УДК 338.24.519.23/.25

НОВЫЕ ФИЗИЧЕСКИЕ ПРИНЦИПЫ ДЕЙСТВИЯ: ЭКОНОФИЗИКА, НЕЙРОННЫЕ СЕТИ, ФРАКТАЛЬНЫЙ АНАЛИЗ – К ВОЗМОЖНОМУ ИХ ПРИМЕНЕНИЮ В МЕДИЦИНЕ КАТАСТРОФ

(ОСНОВЫ НБИКС – формата)

Антипова Татьяна Александровна[1], Ардатова Анастасия Сергеевна[2], Власов Ян Владимирович[3], Гаврилов Владимир Юрьевич[4]

[1]ORCID: 0000-0001-5499-2170 Федеральное государственное бюджетное образовательное учреждение высшего образования «Самарский государственный медицинский университет» Министерства здравоохранения Российской Федерации. Кандидат физико-математических наук. Доцент кафедры медицинской физики, математики и информатики.

[2]ORCID: 0000-0003-3329-9427 Федеральное государственное бюджетное образовательное учреждение высшего образования «Самарский государственный медицинский университет» Министерства здравоохранения Российской Федерации. Ординатор кафедры медицинской реабилитации, спортивной медицины, физиотерапии и курортологии.

[3]ORCID: 0000-0002-9471-9088 Федеральное государственное бюджетное образовательное учреждение высшего образования «Самарский государственный медицинский университет» Министерства здравоохранения Российской Федерации. Доктор медицинских наук. Профессор кафедры неврологии и нейрохирургии. Президент "Общероссийской общественной организации инвалидов-больных рассеянным склерозом" (ОООИ-БРС).

[4]ORCID: 0000-0001-6964-6086 Самарская региональная общественная организация инвалидов - больных рассеянным

склерозом (COPC). Советник по научным вопросам. Член – корреспондент Академии медико-технических наук Российской Федерации.

АННОТАЦИЯ

Помимо составления прогнозов математика находит скрытые закономерности в уже произошедших событиях. Последний финансовый кризис, подкосивший экономики всех без исключения стран, многие сравнивали с цунами. Математик из Нью-Йорка Реджинальд Смит (Redginald Smith) считает, что его развитие, скорее, напоминает эпидемию инфекционного заболевания. В своей статье, опубликованной в журнале *Physical Society of Korea*, ученый выявил очаг заболевания и проследил динамику его распространения по миру.

Математики очень давно изучают развитие эпидемий и обнаружили огромное количество законов, которые управляют заражением в популяциях людей и животных.

Отмечается, что выявленные закономерности вполне применимы для описания распространения скрытых инфекций. Математика очень давно изучает развитие эпидемий, пандемий и обнаружила огромное количество законов, которые управляют заражением в популяциях людей и животных.

Ключевые слова: эконофизика, эволюция нелинейных систем, нейронные сети, медицина катастроф, пандемии, эпидемии, вероятностные и статистические методы, фрактальный анализ, адаптивные системы, энтропия, негэнтропия, запутанные квантовые состояния, декогеренция, квантовая экономика, квантовая биология, НБИКС (нано-, био-, инфо-, когно-, социо-) – формат.

ВВЕДЕНИЕ (ИНФОРМАЦИЯ С РЕСУРСОВ: СМ. В СПИСКЕ ЦИТИРУЕМЫХ ИСТОЧНИКОВ)

«Эконофизика появилась в середине 1990-х в результате попытки заняться сложными проблемами, изложенными экономикой, с точки зрения физических методов. Неудовлетворенность традиционными объяснениями экономистов была обусловлена несоответствием финансовых наборов данных существовавшим теоретическим моделям.

Сам термин эконофизика был введен американским физиком Гарри Юджином Стэнли (en:H. Eugene Stanley) для объединения множества исследований, в которых типично физические методы и приемы использовались при решении экономических задач[1]. Торжественное заседание, посвященное открытию эконофизики было организованным 1998 году в Будапеште Дженосом Кертесзом и Имре Кондором.

В настоящее время, почти регулярные серии встреч на тему эконофизика включают: семинар Никкеи *Econophysics Research*, и симпозиумы APFA, ESHIA, коллоквиум по эконофизике» [1].

«Основными инструментами эконофизики являются вероятностные и статистические методы, часто взятые из статистической физики. Физические модели, которые применялись в экономике, включают кинетическую теорию газа (так называемые модели кинетического обмена на рынках), модели перколяции, хаотические модели, разработанные для изучения остановки сердца, и модели с самоорганизующейся критичностью, а также другие модели, разработанные для предсказания землетрясений. Более того, были попытки использовать математическую теорию сложности и теорию информации, разработанную многими учеными, среди которых Мюррей Гелл-Манн и Клод Э. Шеннон, соответственно. Для потенциальных игр было показано, что порождающее возникновение равновесие, основанное на информации через информационную энтропию

Шеннона, дает такую же равновесную меру (мера Гиббса из статистической механики), что и стохастическое динамическое уравнение, которое представляет зашумленные решения, оба из которых основаны на ограниченных модели рациональности, используемые экономистами. Теорема флуктуации-диссипации соединяет эти два понятия, чтобы установить конкретное соответствие «температуры», «энтропии», «свободного потенциала / энергии» и других физических понятий с экономическими и биологическими системами. Модель статистической механики не строится априори – это результат ограниченно рационального допущения и моделирования существующих неоклассических моделей. Он был использован для доказательства «неизбежности сговора» Хью Диксона в случае, для которого неоклассическая версия модели не предсказывает сговор. Здесь спрос растет, как и в случае с товарами Веблена, покупатели акций с ошибкой «горячей руки», предпочитающие покупать более успешные акции и продавать менее успешные, или среди коротких трейдеров во время короткого сжатия, как это произошло при сговоре группы WallStreetBets с целью поднять курс акций GameStop в 2021 году. Количественные показатели, полученные из теории информации, использовались в нескольких работах эконофизика Аурелио Ф. Баривьера и соавторов для оценки степени информационной эффективности фондовых рынков. Зунино и др. использовать инновационный статистический инструмент в финансовой литературе: плоскость причинности сложности-энтропии. Это декартово представление устанавливает рейтинг эффективности различных рынков и выделяет разную динамику рынка облигаций. Было обнаружено, что в более развитых странах фондовые рынки имеют более высокую энтропию и меньшую сложность, в то время как рынки развивающихся стран имеют более низкую энтропию и более высокую сложность. Более того, авторы приходят к выводу, что классификация, полученная из плоскости причинно-следственной связи сложности и энтропии, согласуется с квалификацией, присвоенной крупными рейтинговыми

компаниями суверенным инструментам. Аналогичное исследование, разработанное Bariviera et al. исследовать взаимосвязь между кредитными рейтингами и информационной эффективностью выборки корпоративных облигаций нефтяных и энергетических компаний США, используя также плоскость причинно-следственной связи сложности-энтропии. Они считают, что эта классификация соответствует кредитным рейтингам, присвоенным Moody's. Еще один хороший пример – теория случайных матриц, которую можно использовать для выявления шума в матрицах финансовой корреляции. В одной статье утверждается, что этот метод может улучшить производительность портфелей, например, при его оптимизации . Однако существуют различные другие инструменты из физики, которые использовались до сих пор, такие как гидродинамика , классическая механика и *квантовая механика* (включая так называемую классическую экономику, *квантовую экономику, квантовую биологию и квантовые финансы)*, а также формулировка интеграла по траекториям статистической механики. . Концепция индекса экономической сложности, представленная физиком Сезаром А. Идальго и экономистом из Гарварда Рикардо Хаусманном и представленная в Обсерватории экономической сложности Массачусетского технологического института, была разработана в качестве инструмента прогнозирования экономического роста в «Атласе экономики» Гарвардской лаборатории роста. По оценкам Хаусманна и Идальго, ECI намного точнее предсказывает рост ВВП, чем традиционные меры управления Всемирного банка . Есть также аналогии между теорией финансов и теорией диффузии . Например, уравнение Блэка-Шоулза для опциона ценообразования является диффузия - адвекция уравнение (см , однако для критики методологии Блэка-Шоулза). Теорию Блэка – Шоулза можно расширить, чтобы дать аналитическую теорию основных факторов экономической деятельности» [2].

«Начиная с середины 1990-х годов в лексикон участников научных конференций вошло странное слово-гибрид - эконофизика. Этот термин был придуман американским физиком Гарри Стэнли (Harry Stanley) для объединения множества исследований, в которых типично физические методы и приемы использовались при решении экономических задач.

Физики и математики пришли на помощь экономистам, так как те не могли справиться с растущим потоком данных, используя применимые в экономике методы анализа. Оказалось, что многие экономические явления, например, развитие фондовых рынков или инфляция, хорошо описываются при помощи математического аппарата теории хаоса или законов, которым подчиняется поведение динамических систем.

Свежий взгляд математиков на экономику позволил выявить несколько нетривиальных закономерностей, которые управляют движениями денежных потоков и ценных бумаг. В 2006 году в авторитетном физическом журнале *Physical Review Letters* появилась статья японских эконофизиков, которые сравнили динамику фондовых рынков с фазовыми переходами в системе конденсированных сред.

Фазовым переходом называют переход вещества из одного термодинамического состояния в другое. Характерным примером фазового перехода является замерзание воды при опускании температуры ниже нуля градусов Цельсия (при нормальном атмосферном давлении). Кристаллы льда образуются по всей емкости с водой практически мгновенно после того, как будет преодолена критическая точка. Авторы работы показали, что обвалы на фондовых рынках подчиняются тем же законам - до определенного момента ситуация стабильна, но после "перевала" индексы начинают необратимо падать.

Оперируя теми же законами, что и японские ученые, российский математик Виктор Маслов, по его словам, предсказал

экономический кризис 2008-2009 годов за шесть месяцев до его начала. О способности предвидеть будущее фондовых рынков в 2009 году заявила группа эконофизиков из Швейцарии и Китая. Проанализировав динамику китайского фондового индекса Shanghai Composite, исследователи заключили, что он представляет собой надувающийся пузырь и предсказали дату, когда пузырь должен лопнуть.

В назначенный срок значение индекса не изменилось, но спустя несколько дней он резко упал. Пока ученые не могут однозначно сказать, было ли это то самое предсказанное эконофизиками падение, или же совпадение по времени было случайным. Коллеги "ясновидцев" также не исключают, что именно их работа и спровоцировала падение индекса - фондовые рынки очень чувствительны к прогнозам, как позитивным, так и негативным (можно ожидать, что в скором времени эта их особенность также будет формализована).

Математика очень давно изучает развитие эпидемий и обнаружила огромное количество законов, которые управляют заражением в популяциях людей и животных. В последние годы список инфекционных агентов, чья деятельность была описана языком формул, пополнили компьютерные вирусы. Летом 2009 года канадские математики рассчитали модель последствий появления на земле «зомби-вируса» [3].

Авторы отмечают, что выявленные закономерности вполне применимы для описания распространения скрытых инфекций.

Американские ученые установили, что эволюция террористических организаций во многом напоминает эволюцию производственных компаний, например ферм (либо консортивных отношений в концентрах экосистем – прим. авт.). Ученые надеются, что новые знания помогут в борьбе с этими организациями. Статья ученых пока еще нигде не опубликована,

однако ее препринт доступен на сайте arXiv.org: https://arxiv.org/abs/0906.3287» [4].

ГЛАВА 1. МЕТОДЫ И МОДЕЛИ ДЛЯ ПРОГНОЗИРОВАНИЯ ПОВЕДЕНИЯ НЕЛИНЕЙНЫХ СИСТЕМ

1.1. Общий обзор моделей и методов прогнозирования

По оценкам зарубежных и отечественных систематиков прогностики, уже насчитывается свыше 100 методов прогнозирования. Число базовых методов прогностики, которые в тех или иных вариациях повторяются в других методах, гораздо меньше. Многие из этих «методов» относятся скорее к отдельным приемам или процедурам прогнозирования, другие представляют набор отдельных приемов, отличающихся от базовых или друг от друга количеством частных приемов и последовательностью их применения.

В литературе имеется большое количество классификационных схем методов прогнозирования. Однако, большинство из них или неприемлемы, или обладают недостаточной познавательной ценностью. Основной погрешностью существующих классификационных схем является нарушение принципов классификации. К числу основных таких принципов, на наш взгляд, относятся: достаточная полнота охвата прогностических методов, единство классификационного признака на каждом уровне членения, открытость классификации. Предлагаемая автором схема классификации методов прогнозирования показана на рис. 1, где МНК – метод наименьших квадратов, МГУА – метод группового учета аргумента, представляющий собой дальнейшее развитие метода регрессивного анализа, основанного на принципах теории обучения и самоорганизации (принцип «селекции», направленный отбор и т. д.) [1].

Кроме того, существуют частные классификационные схемы, предназначенные для определенной цели или задачи. Каждый уровень детализации определяется своим классификационным

признаком: степенью формализации, общим принципом действия, способом получения прогнозной информации.

По степени формализации все методы прогнозирования делятся на интуитивные и формализованные. Интуитивное прогнозирование применяется тогда, когда объект прогнозирования либо слишком прост, либо настолько сложен, что аналитически учесть влияние многих факторов практически невозможно. В этих случаях прибегают к опросу экспертов. Полученные индивидуальные и коллективные экспертные оценки используют как конечные прогнозы или в качестве исходных данных в комплексных системах прогнозирования.

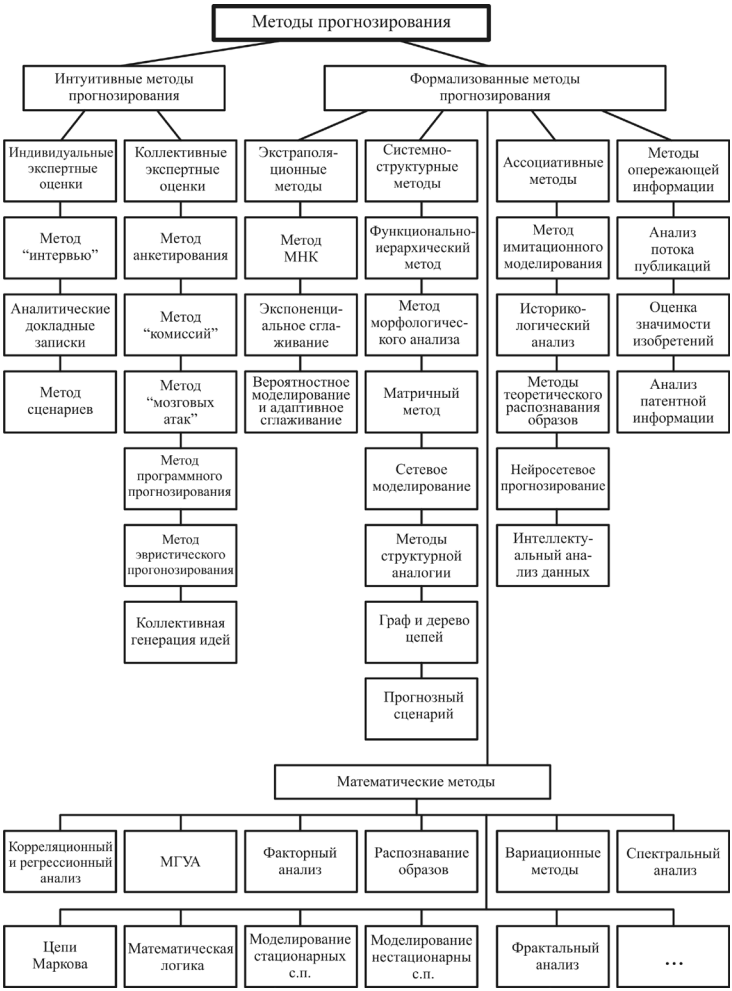


Рис. 1 – Классификационная схема методов прогнозирования [1]

В выборе методов прогнозирования важным показателем является глубина упреждения прогноза. При этом необходимо не только знать абсолютную величину этого показателя, но и отнести его к длительности эволюционного цикла развития объекта прогнозирования. Для этого можно использовать безразмерный показатель глубины (дальности) прогнозирования $^{(\tau)}$ [1]:

$\tau = \Delta t / t_x$,

где Δt – абсолютное время упреждения, а t_x – величина эволюционного цикла объекта прогнозирования.

Формализованные методы прогнозирования являются действенными, если величина глубины упреждения укладывается в рамки эволюционного цикла ($\tau << 1$). При возникновении в рамках прогнозного периода «скачка» в развитии объекта прогнозирования ($\tau \approx 1$) необходимо использовать интуитивные методы, как для определения силы «скачка», так и для оценки времени его осуществления, либо теорию катастроф [2]. В этом случае формализованные методы применяются для оценки эволюционных участков развития до и после скачка. Если же в прогнозном периоде укладывается несколько эволюционных циклов развития объекта прогнозирования ($\tau >> 1$), то при комплексировании систем прогнозирования большее значение имеют интуитивные методы.

В зависимости от общих принципов действия, интуитивные методы прогнозирования, например, можно разделить на две группы: индивидуальные экспертные оценки и коллективные экспертные оценки.

Методы коллективных экспертных оценок уже можно отнести к комплексным системам прогнозирования (обычно неполным), поскольку в последних сочетаются методы индивидуальных экспертных оценок и статистические методы обработки этих оценок. Но так как статистические методы применяются во вспомогательных процедурах выработки прогнозной информации, на наш взгляд, коллективные экспертные оценки целесообразнее отнести к сингулярным методам прогнозирования.

В группу индивидуальных экспертных оценок можно включить (принцип классификации – способ получения прогнозной информации) следующие методы: метод «интервью»,

аналитические докладные записки, написание сценария. В группу коллективных экспертных оценок входят анкетирование, методы «комиссий», «мозговых атак» (коллективной генерации идей).

Класс формализованных методов в зависимости от общих принципов действия можно разделить на группы экстраполяционных, системно-структурных, ассоциативных методов и методов опережающей информации.

В группу методов прогнозной экстраполяции можно включить методы наименьших квадратов, экспоненциального сглаживания, вероятностного моделирования и адаптивного сглаживания. К группе системно-структурных методов можно отнести методы функционально-иерархического моделирования, морфологического анализа, матричный, сетевого моделирования, структурной аналогии. Ассоциативные методы можно разделить на методы имитационного моделирования и историко-логического анализа. В группу методов опережающей информации можно включить методы анализа потоков публикаций, оценки значимости изобретений и анализа патентной информации.

Представленный перечень методов и их групп не является исчерпывающим. Нижние уровни классификации открыты для внесения новых элементов, которые могут появиться в процессе дальнейшего развития инструментария прогностики.

Некоторые не названные здесь методы являются или разновидностью включенных в схему методов, или дальнейшей их конкретизацией.

Ниже приведены основные недостатки методов прогнозирования, не связанных с теорией хаоса:

- Недостатки метода наименьших квадратов (МНК). Использование процедуры оценки, основанной на МНК, предполагает обязательное удовлетворение целого ряда

предпосылок, невыполнение которых может привести к значительным ошибкам:

1. Случайные ошибки имеют нулевую среднюю, конечные дисперсии и ковариации;

2. Каждое измерение случайной ошибки характеризуется нулевым средним, не зависящим от значений наблюдаемых переменных;

3. Дисперсии каждой случайной ошибки одинаковы, их величины независимы от значений наблюдаемых переменных (гомоскедастичность);

4. Отсутствие автокорреляции ошибок, т.е. значения ошибок различных наблюдений независимы друг от друга;

5. Нормальность. Случайные ошибки имеют нормальное распределение;

6. Значения эндогенной переменной x свободны от ошибок измерения и имеют конечные средние значения и дисперсии.

- Проблемы выбора адекватной модели. Выбор модели в каждом конкретном случае осуществляется по целому ряду статистических критериев, например, по дисперсии, корреляционному отношению и др.

- Дисконтирование. Классический метод наименьших квадратов предполагает равноценность исходной информации в модели. В реальной же практике будущее поведение процесса значительно в большей степени определяется поздними наблюдениями, чем ранними. Это обстоятельство породило так называемое дисконтирование, т.е. уменьшение ценности более ранней информации.

- Проблема оценки достоверности прогнозов. Важным моментом получения прогноза с помощью МНК является оценка

достоверности полученного результата. Для этой цели используется целый ряд статистических характеристик:

1. Оценка стандартной ошибки;

2. Средняя относительная ошибка оценки;

3. Среднее линейное отклонение;

4. Корреляционное отношение для оценки надежности модели;

5. Оценка достоверности выбранной модели через значимость индекса корреляции по Z-критерию Фишера;

6. Оценка достоверности модели по F-критерию Фишера;

7. Наличие автокорреляций (критерий Дарбина-Уотсона).

- Недостатки, обусловленные жесткой фиксацией тренда. Жесткие статистические предложения о свойствах временных рядов ограничивают возможности методов математической статистики, теории распознавания образов, теории случайных процессов и т.п., так как многие реальные процессы не могут адекватно быть описаны с помощью традиционных статистических моделей, поскольку, по сути, являются существенно нелинейными и имеют либо хаотическую, либо квазипериодическую, либо смешанную основу.

- Проблемы и недостатки метода экспоненциального сглаживания. Для метода экспоненциального сглаживания основным и наиболее трудным моментом является выбор параметра сглаживания, начальных условий и степени прогнозирующего полинома. Кроме того, для определения начальных параметров модели остаются актуальными перечисленные недостатки МНК и проблема автокорреляций.

- Проблемы и недостатки метода вероятностного моделирования. Недостатком модели является требование большого количества

наблюдений и незнание начального распределения, что может привести к неправильным оценкам.

- Проблемы и недостатки метода адаптивного сглаживания. При наличии достаточной информации можно получить надежный прогноз на интервал больший, чем при обычном экспоненциальном сглаживании. Но это лишь при очень длинных рядах. К сожалению, для данного метода нет строгой процедуры оценки необходимой или достаточной длины исходной информации, для конечных рядов нет конкретных условий оценки точности прогноза. Поэтому для конечных рядов существует риск получить весьма приблизительный прогноз, тем более что в большинстве случаев в реальной практике встречаются ряды, содержащие не более 20-30 точек.

- Проблемы и недостатки метода Бокса-Дженкинса (модели авторегрессии – скользящего среднего). Проблемы связанны, прежде всего, с неоднородностью временных рядов и практической реализации метода из-за своей сложности.

- Проблемы и недостатки методов, реализованных на базе нейронных сетей (НС). Проблемы неопределенности в выборе числа слоев и количества нейронных элементов в слое, медленная сходимость градиентного метода с постоянным шагом обучения, сложность выбора оптимальной скорости обучения, влияние случайной инициализации весовых коэффициентов НС на поиск минимума функции среднеквадратической ошибки. Одна из наиболее серьезных трудностей при обучении – это явление переобучения. Кроме того, использование немасштабированных данных может привести к «параличу» сети.

- Проблемы и недостатки методов, реализованных на базе генетических алгоритмов. Это проблема кодировки информации, содержащейся в модели нейронной сети, а также сложность понимания и программной реализации.

При построении реальных прогнозов всегда необходимо учитывать не только специфические особенности применяемых методов, о и их ограничения и недостатки.

1.2. Бифуркации и катастрофы в нелинейных динамических системах

В настоящее время из систем, рассмотренных в книге (из области техники, медицины и экономики), наиболее хорошо изучены нелинейные динамические системы (НДС). Поэтому, приведем основные понятия НДС, связанные со сложным колебаниями, приводящими к катастрофам.

При математическом моделировании большинства практических задач в нелинейной динамике наиболее часто используются дифференциальные уравнения и отображения, зависящие от ряда параметров. Изменение параметров системы может привести к качественному преобразованию фазового портрета системы, называемому бифуркацией [3]. Под качественным изменением фазового портрета подразумевается его структурная перестройка, нарушающая *топологическую эквивалентность*. Значение параметра, при котором происходит бифуркация, называется бифуркационным значением или точкой бифуркации. Помимо фазового пространства, ДС характеризуется также пространством параметров. Определенный набор значений параметров $\alpha_1, \alpha_2, ..., \alpha_M$ образует радиус-вектор $\vec{\alpha}$ в этом пространстве. В многомерном пространстве параметров системы бифуркационным значениям могут соответствовать определенные множества, представляющие собой точки, линии, поверхности и т. д. Бифуркации характеризуются некоторым количеством условий, налагаемых на параметры системы. Число таких условий называется коразмерностью бифуркации. Например, коразмерность 1 означает, что имеется только одно бифуркационное условие.

Различают локальные и нелокальные бифуркации ДС. Локальные бифуркации связаны с локальной окрестностью траектории на

предельном множестве. Они отражают изменение устойчивости как отдельных траекторий, так и всего предельного множества целиком, и могут свидетельствовать об исчезновении исследуемого предельного множества в результате его слияния с другим предельным множеством. Все перечисленные выше явления могут быть обнаружены в рамках линейного анализа устойчивости. Например, смена знака одного из ляпуновских показателей траектории на предельном множестве свидетельствует о локальной бифуркации предельного множества. *Нелокальные бифуркации* связаны с поведением многообразий предельных седловых множеств, в частности, с образованием сепаратрисных петель, гомоклинических и гетероклинических кривых, а также с возникновением касания аттрактора и сепаратрисной кривой или поверхности. Перечисленные эффекты не могут быть обнаружены в рамках линейного приближения. В такой ситуации необходимо учитывать нелинейные свойства изучаемой системы.

Бифуркации могут происходить с любыми предельными множествами, но наибольший интерес представляют бифуркации аттракторов, поскольку они приводят к изменениям экспериментально наблюдаемых режимов. Бифуркации аттракторов обычно подразделяются на внутренние (мягкие) бифуркации и кризисы (жесткие бифуркации). Внутренние бифуркации связаны с топологическими изменениями самих притягивающих предельных множеств, но не затрагивают их бассейнов притяжения. Кризисы аттракторов сопровождаются качественной перестройкой границ бассейнов притяжения.

Концепция *грубости* (*структурной устойчивости*) ДС тесно связана с бифуркациями. ДС называется *грубой* или *структурно-устойчивой*, если малые гладкие возмущения оператора эволюции приводят к топологически эквивалентным решениям [3]. Бифуркацию ДС можно представлять как переход системы от одного структурно-устойчивого состояния к другому через структурно-неустойчивое состояние в точке бифуркации.

Анализ бифуркаций ДС при изменении параметров системы позволяет построить *бифуркационную диаграмму* системы. *Бифуркационная диаграмма* представляет собой набор точек, линий и поверхностей в пространстве параметров, которые соответствуют различным бифуркациям предельных множеств системы. Если при некоторых значениях параметров существуют несколько предельных множеств, то бифуркационная диаграмма является «многолистной». Сосуществование большого (даже бесконечного) числа предельных множеств типично для систем со сложной динамикой. В этом случае точки бифуркаций могут быть всюду плотны в пространстве параметров. При таких условиях построение полной бифуркационной диаграммы системы становится невозможным, поэтому имеет смысл рассматривать только отдельные ее листы и фрагменты. Помимо бифуркационных диаграмм в пространстве параметров, для наглядного представления часто используются так называемые *фазо-параметрические диаграммы*. При построении такой диаграммы значения управляющего параметра откладываются по оси абсцисс, а ось ординат соответствует одной из динамических переменных, или некоторой величине, связанной с состоянием ДС.

Обычно сложные изменения НДС описывают с помощью оператора эволюции [3]. Резкие изменения состояния системы, вызванные гладкими возмущениями оператора эволюции, в частности малыми вариациями параметров, называет *катастрофами*.

Если описывать катастрофу с помощью параметра бифуркации, то катастрофа описывается с помощью бифуркации «кризис». В [3] делается очень осторожное утверждение, что кризис и катастрофа тождественные понятия. Существенный вклад в развитие теории катастроф был внесен В.И. Арнольдом [4].

1.3. Нейронные сети и генетические алгоритмы

Многие реальные процессы не могут адекватно быть описаны с помощью традиционных статистических моделей, поскольку, по сути, являются существенно нелинейными, и имеют либо хаотическую, либо квазипериодическую, либо смешанную (стохастика+хаос-динамика+детерминизм) основу.

В данной ситуации адекватным аппаратом для решения задач диагностики и прогнозирования могут служить специальные искусственные сети [5-8] реализующие идеи предсказания и классификации при наличии обучающих последовательностей, причем, как весьма перспективные, следует отметить радиально-базисные структуры, отличающиеся высокой скоростью обучения и универсальными аппроксимирующими возможностями.

В основе нейроинтеллекта лежит нейронная организация искусственных систем, которая имеет биологические предпосылки. Способность биологических систем к обучению, самоорганизации и адаптации обладает большим преимуществом по сравнению с современными вычислительными системами. Первые шаги в области искусственных нейронных сетей сделали в 1943 г. В. Мак-Калох и В. Питс. Они показали, что при помощи пороговых нейронных элементов можно реализовать исчисление любых логических функций [6]. В 1949 г. Хебб предложил правило обучения, которое стало математической основой для обучения ряда нейронных сетей. В 1957-1962 гг. Ф. Розенблатт предложил и исследовал модель нейронной сети, которую он назвал персептроном [7]. В 1959 г. В. Видроу и М. Хофф предложили процедуру обучения для линейного адаптивного элемента – ADALINE. Процедура обучения получила название "дельта правило" [6]. В 1969 г. М. Минский и С. Пайперт опубликовали монографию "Персептроны", в которой был дан математический анализ персептрона, и показаны ограничения, присущие ему. В 80-е годы значительно расширяются исследования в области нейронных сетей. Д. Хопфилд в 1982 г. дал анализ устойчивости

нейронных сетей с обратными связями и предложил использовать их для решений задач оптимизации. Т. Кохонен разработал и исследовал самоорганизующиеся НС. Ряд авторов предложил алгоритм обратного распространение ошибки, который стал мощным средством для обучения многослойных НС [5-7]. В настоящее время разработано большое число нейросистем, применяемых в различных областях: прогнозировании, управлении, диагностике в медицине и технике, распознавании образов и т.д. [8-10].

Нейронная сеть – совокупность нейронных элементов и связей между ними. Основной элемент нейронной сети – это *формальный нейрон*, осуществляющий операцию нелинейного преобразования суммы произведений входных сигналов на весовые коэффициенты

$$y = F\left(\sum_{i=1}^{n} w_i x_i\right) F(WX)$$

,

где $X = (x_1, x_2, ..., x_n)^T$ – вектор входного сигнала, $W = (w_1, w_2, ..., w_n)$ – весовой вектор, F – оператор нелинейного преобразования.

Для обучения сети используются различные алгоритмы обучения и их модификации [11-16]. Очень трудно определить, какой обучающий алгоритм будет самым быстрым при решении той или иной задачи. Наибольший интерес для нас представляет алгоритм обратного распространения ошибки, так как является эффективным средством для обучения многослойных нейронных сетей прямого распространения [17,18]. Алгоритм минимизирует среднеквадратичную ошибку нейронной сети. Для этого с целью настройки синаптических связей используется метод градиентного спуска в пространстве весовых коэффициентов и порогов нейронной сети. Следует отметить, что для настройки синаптических связей сети используется не только метод градиентного спуска, но и методы сопряженных градиентов Ньютона, квазиньютоновский метод [19]. Для ускорения

процедуры обучения вместо постоянного шага обучения предложено использовать адаптивный шаг обучения $\alpha(t)$. Алгоритм с адаптивным шагом обучения работает в 4 раза быстрее. На каждом этапе обучения сети он выбирается таким, чтобы минимизировать среднеквадратическую ошибку сети [5,6].

Для прогнозирующих систем на базе НС наилучшие качества показывает гетерогенная сеть, состоящая из скрытых слоев с нелинейной функцией активации нейронных элементов и выходного линейного нейрона. Недостатком большинства рассмотренных нелинейных функций активации является то, что область выходных значений их ограничена отрезком [0,1] или [-1,1]. Это приводит к необходимости масштабирования данных, если они не принадлежат указанным выше диапазонам значений. В работе предложено использовать логарифмическую функцию активации для решения задач прогнозирования, которая позволяет получить прогноз значительно точнее, чем при использовании сигмоидной функции.

Анализ различных типов НС показал, что НС может решать задачи сложения, вычитания десятичных чисел, задачи линейного авто регрессионного анализа и прогнозирования временных рядов с использованием метода «скользящего окна» [14].

Проведенный анализ многослойных нейронных сетей и алгоритмов их обучения позволил выявить ряд недостатков и возникающих проблем:

1. Неопределенность в выборе числа слоев и количества нейронных элементов в слое;

2. Медленная сходимость градиентного метода с постоянным шагом обучения;

3. Сложность выбора подходящей скорости обучения. Так как маленькая скорость обучения приводит к скатыванию НС в локальный минимум, а большая скорость обучения может

привести к пропуску глобального минимума и сделать процесс обучения расходящимся;

4. Невозможность определения точек локального и глобального минимума, так как градиентный метод их не различает;

5. Влияние случайной инициализации весовых коэффициентов НС на поиск минимума функции среднеквадратической ошибки.

Большую роль для эффективности обучения сети играет архитектура НС. При помощи трехслойной НС можно аппроксимировать любую функцию со сколь угодно заданной точностью. Точность определяется числом нейронов в скрытом слое, но при слишком большой размерности скрытого слоя может наступить явление, называемое перетренировкой сети. Для устранения этого недостатка необходимо, чтобы число нейронов в промежуточном слое было значительно меньше, чем число тренировочных образов. С другой стороны, при слишком маленькой размерности скрытого слоя можно попасть в нежелательный локальный минимум. Для нейтрализации этого недостатка можно применять ряд методов описанных в [18,20].

Впервые идея использования генетических алгоритмов для обучения (machine learning) для прогнозирования поведения систем была предложена в 1970-е годы [21-24]. Во второй половине 1980-х к этой идее вернулись в связи с обучением нейронных сетей. Они позволяют решать задачи прогнозирования (в последнее время наиболее широко генетические алгоритмы обучения используются для банковских прогнозов), классификации, поиска оптимальных вариантов, и совершенно незаменимы в тех случаях, когда в обычных условиях решение задачи основано на интуиции или опыте, а не на строгом (в математическом смысле) ее описании. Использование механизмов генетической эволюции для обучения нейронных сетей кажется естественным, поскольку модели нейронных сетей разрабатываются по аналогии с мозгом и реализуют некоторые его

особенности, появившиеся в результате биологической эволюции [9,25,26].

Основные компоненты генетических алгоритмов – это стратегии репродукций, мутаций и отбор "индивидуальных" нейронных сетей (по аналогии с отбором индивидуальных особей) [9].

Важным недостатком генетических алгоритмов является сложность для понимания и программной реализации. Однако преимуществом является эффективность в поиске глобальных минимумов адаптивных рельефов, так как в них исследуются большие области допустимых значений параметров нейронных сетей. Другая причина того, что генетические алгоритмы не застревают в локальных минимумах – случайные мутации, которые аналогичны температурным флуктуациям метода имитации отжига.

В [9,26] указания на достаточно высокую скорость обучения при использовании генетических алгоритмов. Хотя скорость сходимости градиентных алгоритмов в среднем выше, чем генетических алгоритмов.

Генетические алгоритмы дают возможность оперировать дискретными значениями параметров НС. Это упрощает разработку цифровых аппаратных реализаций НС. При обучении на компьютере нейронных сетей, не ориентированных на аппаратную реализацию, возможность использования дискретных значений параметров в некоторых случаях может приводить к сокращению общего времени обучения.

В рамках «генетического» подхода в последнее время разработаны многочисленные алгоритмы обучения НС, различающиеся способами представления данных нейронной сети в "хромосомах", стратегиями репродукции, мутаций, отбора [21,23,24].

1.4. Эконофизика и финансовые временные ряды

Наука о финансовых временных рядах долгое время развивалась в виде двух несвязанных направлений и лишь в последнее время наметилась некоторая тенденция к их сближению [27].

Первое направление, которое можно назвать статистическим, берет свое начало от работы Луиса Башелье 1900 г. [28], где автор, еще за пять лет до Эйнштейна, предложил первую модель броуновского движения (модель случайного блуждания) и применил ее для описания колебания цен акций на фондовой бирже. Строго математически эта модель была обоснована в начале 20-х годов прошлого века Норбертом Винером [29] и применительно к рынку долгое время развивалась исключительно в академической среде. Последнее было связано с тем, что в соответствии с этой моделью принципиально невозможно получить прибыль больше средней по рынку. Однако это противоречило опыту реального трейдинга. Поэтому, среди инвесторов броуновскую модель стали серьезно принимать в расчет только в начале 60-х гг., когда окончательно оформилась концепция «эффективного рынка» (рынка, на котором цены в полной мере отражают всю доступную информацию). Согласно Фамэ [30] для существования такого рынка достаточно предположить, что на нем действует большое число полностью информированных агентов, которые мгновенно реагируют на внешние события, действуя при этом рационально и независимо. Необходимость подобной концепции диктовалась тем, что в ее рамках обретала смысл теория формирования оптимального портфеля Уильяма Шарпа (Ноб. премия по экономике 1990 г.) [31], которая имела большое практическое значение. Основной же моделью поведения цены на таком рынке стала модель броуновского движения. Высшим достижением подхода, основанного на этой модели, стали работы Блэка, Шоулза и Мертона (Ноб. премия по экономике 1997 г.) [32,33], которые позволяли точно рассчитывать «справедливые» цены опционов.

Напомним, что классическая модель броуновского движения основана на двух постулатах. Во-первых, приращения процесса на любом интервале времени имеют нормальное (гауссово) распределение с нулевым средним, которое следует из центральной предельной теоремы и получается как результат суммирования достаточно большого числа независимых случайных факторов. Во-вторых, приращения на непересекающихся интервалах являются статистически независимыми. Однако наблюдения финансовых временных рядов выявили ряд особенностей, которые не согласуются с этими постулатами. Наиболее важные из них связаны с тем, что сильные изменения временного ряда происходят значительно чаще, чем следовало бы ожидать исходя из гауссова распределения (проблема «толстых хвостов»), причем такие изменения обычно разделены колебаниями относительно малой интенсивности (эффект кластеризации волатильности). По этой причине параллельно с броуновской моделью стали развиваться всевозможные ее обобщения, которые были связаны с отказом либо от условия нормальности распределения приращений (первый постулат), либо от условия независимости последних (второй постулат). В первом случае мы приходим к движению Леви (Levi Motion) и, в частности, к устойчивым распределениям Парето [34,35]. Во втором – к процессам с памятью и обобщенному броуновскому движению (Fractional Brownian Motion) [36]. Наконец, если отказаться от обоих постулатов броуновской модели, то мы приходим к идеям авторегрессионой условной гетероскедастичности (зависимости значений временного ряда от его предыдущих значений при изменении дисперсии во времени), или ARCH-моделям Энгла (Ноб. премия по экономике 2003 г.), и различным ее обобщениям (см. обзор [37]). Наиболее полно и последовательно статистический подход, связанный с гипотезой эффективного рынка (Effective Market Hypothesis), изложен у Ширяева [38]. Второе направление, которое естественно назвать *динамическим,* родилось и стало развиваться в

среде практикующих трейдеров в начале 30-х гг. прошлого века. Этот подход получил название «технический анализ» (в отличие от фундаментального анализа, основанного на расчете «справедливой» цены акции в виде дисконтированной стоимости будущих доходов). Идеологической основой такого подхода стало известное положение Чарльза Доу (главного автора известного индекса), выдвинутое им еще в конце XIX в. Доу утверждал, что естественное состояние цены – это тренд (направленное движение вверх или вниз), который является результатом совместного действия толпы и отражает действующую на рынке социальную тенденцию. Поэтому тренд будет продолжаться до тех пор, пока на рынке не произойдет смена этой тенденции. Цель технического анализа – вскрыть внутренние закономерности временного ряда, на основе которых можно прогнозировать переходы из тренда во флэт (относительно стабильное состояние рынка) и обратно. На этом пути были открыты многочисленные формы относительно устойчивого поведения временного ряда (фигуры технического анализа). Первые работы в этом направлении принадлежат Уильяму Ганну и Ральфу Эллиоту [39]. Уже в 50-е гг. почти все классические фигуры технического анализа («треугольник», «трапеция», «голова и плечи» и т.д.) были известны. Однако лишь в 80-е гг. в известных монографиях Джона Мерфи [40] Роберта Прехтера [41] происходит их систематизация. Последняя совпала по времени с неожиданной поддержкой, которую в эти годы технический анализ получил со стороны теории динамического хаоса. Из общей теории следовало, что временной ряд, который внешне выглядел как реализация случайного процесса, вполне мог порождаться нелинейной динамической системой малой размерности. Это означает, что его можно представить в виде одномерной проекции траектории такой системы в расширенном фазовом пространстве, которая описывается с помощью небольшого числа обыкновенных дифференциальных уравнений. Согласно же концепции Доу поведение цен акций определяется стадным инстинктом, который (как убедительно свидетельствуют

данные социальной психологии) подчиняется крайне примитивному механизму. Поэтому вполне обоснованной кажется гипотеза о том, что такой механизм представляет собой динамическую систему. В этом случае, используя теорему Такенса [42], можно восстановить текущее значение временного ряда, исходя из достаточно большого числа исторических данных. Причем для такого восстановления совсем не обязательно знать конкретный вид и число уравнений системы. По существу эта процедура сводит задачу экстраполяции одномерного ряда к задаче интерполяции некоторой многомерной функции. Последняя же является типовой задачей для нейронных сетей [43]. Поэтому теорию динамического хаоса можно считать идеологической основой того мощного внедрения нейротехнологий в бизнес, которое мы повсеместно наблюдали в 90-х гг.

Однако с точки зрения стационарного подхода точный прогноз финансового временного ряда в принципе невозможен. Чтобы разрешить противоречие двух подходов, следует обратиться к результатам экспериментального исследования рынка. Как показывают подобные исследования, на реальном рынке вся совокупность агентов распадается на кластеры (рефферентные группы), в каждом из которых агенты подражают друг другу. Кластеры могут образовывать довольно сложные иерархические связи, могут сливаться в более крупные и распадаться на более мелкие. При этом ясно, что концепция эффективного рынка и концепция Доу описывают два предельных случая. В первом случае на рынке присутствует большое число примерно одинаковых кластеров. Рынок находится в наиболее стабильном состоянии и его эволюция в основном определяется внешней информацией, которая имеет случайный характер. Во втором же случае на рынке присутствует один большой кластер, значительно превосходящий все остальные. Рынок наиболее близок к обвалу и его динамика подчиняется исключительно внутренним факторам. В связи с вышесказанным возникают естественные вопросы: «По каким законам эволюционирует кластерная структура и какова

причина образования больших кластеров?». Ответить на подобные вопросы и означает осуществить синтез двух указанных концепций. В последнее десятилетие серьезные результаты в этом направлении были получены в рамках эконофизики.

Как отдельное направление эконофизика стала оформляться с середины 90-х годов прошлого века, на стыке экономики и физики. При этом само слово «эконофизика» вошло в общее употребление лишь после того, как в 1997 г. Имре Кондор и Янош Кертис организовали в Будапеште «симпозиум по эконофизике» (Workshop on Econophysics). Становление новой дисциплины во многом было связано с приходом в экономику крупных физиков, таких как Филипп Андерсон (Нобелевская премия по физике 1977 г.), Пер Бак, Юджин Стенли и целый ряд других. К тому времени в экономике и, в первую очередь, в финансах накопились задачи, которые не могли быть решены в рамках этих наук. Для решения таких задач предполагалось использовать аппарат и методологию теоретической физики. Заметим, что подобные попытки сблизить экономику и физику многократно предпринимались и раньше. Однако никогда еще этот проект не вызывал такого общественного резонанса. Продолжает неуклонно расти число научных статей, монографий и конференций по эконофизике. Престижные университеты включают соответствующие курсы в свои учебные программы. Все больший интерес проявляют к этой науке и финансовые структуры. Помимо этого, эконофизику уже начинают рассматривать в качестве единой теории, описывающей как функционирование глобальной системы мирового капитала, так и поведение на рынке отдельных экономических субъектов. Следует сказать, что в концептуальном плане эконофизика опирается на позицию, которая не является традиционной в теоретической физике. Такая позиция была представлена Филиппом Андерсом в [44] еще в 1972 г. В этой работе автор утверждал, что «физика элементарных частиц и, в частности, редукционистские подходы обладают лишь ограниченной возможностью объяснить устройство мироздания. Реальность имеет иерархическую

структуру, каждый уровень которой в определенной степени независим от уровней, находящихся выше и ниже. На каждой стадии необходимы совершенно новые законы, концепции и обобщения, требующие не меньшего вдохновения и творчества, чем на предыдущих». «Психология – это не прикладная биология, так же как и биология – это не прикладная химия» – отмечал Андерсон. Таким образом, если раньше «первые принципы, которые не могут быть объяснены в терминах более глубоких принципов» по определению содержала только физика элементарных частиц, теперь оказалось, что такие принципы может содержать любая дисциплина. Концепция Андерсона стала объединяющим лозунгом обширных мульти-дисциплинарных исследований, для проведения которых в середине 80-х гг. в Нью-Мексико был создан Институт Санта-Фе. Предполагалось, что эти исследования внесут «серьезный вклад в решение таких острых долгосрочных проблем, как дефицит торгового баланса, СПИД, генетические дефекты, умственное здоровье, компьютерные вирусы». Именно в этом институте впервые стали появляться работы по экономике с использованием самого современного аппарата теоретической физики. В настоящее время Институт Санта-Фе является одним из главных центров эконофизики, где эта наука развивается в рамках общей теории сложных адаптивных систем. Примерами подобных систем служат центральные нервные системы и нейросети, экосистемы и колонии муравьев, социальные структуры и политические системы, и, конечно, различные структуры, возникающие в экономике. Все эти системы состоят из множества взаимодействующих элементов, которые способны накапливать опыт в процессе взаимодействия с другими элементами, а затем изменяться таким образом, чтобы приспособиться к окружающей среде. Характерным этапом эволюции всех адаптивных систем является процесс самоорганизации, при котором в результате самоусиления отдельных взаимодействий в системе спонтанно возникает порядок. При этом сама система как целое приобретает новое

качество, которое может отсутствовать у отдельных элементов. Базовым примером самоорганизации в экономике служит процесс, управляемый «невидимой рукой» Адама Смита, где множество индивидуумов, стремясь удовлетворить исключительно свои личные материальные потребности, рождают поведенческое целое с принципиально иным качеством. Как показали исследования, проведенные большой группой экономистов, «во многих экспериментах по моделированию рыночных систем плохо информированные, склонные к ошибкам и непонятливые субъекты, контактируют между собой на основе установленных правил и создают социальные алгоритмы по максимизации общих материальных ценностей, явно приближающиеся к оптимальным результатам, которые, как традиционно считалось, можно было бы получить лишь на основе полной информации и с помощью когнитивно-рациональных личностей». Кроме того, эти работы также показывают, что с течением времени рынок может приближаться к эффективному. Это означает, что он способен агрегировать в себе с помощью цен всю значимую информацию. При этом на промежуточных временах из-за того, что субъекты постоянно совершают как ошибки восприятия, так и ошибки под действием эмоций, а также из-за того, что при недостатке информации они обычно начинают подражать друг другу, в системе могут возникать самоорганизующиеся информационные «миражи», которые в скором времени распадаются. Таким образом, можно сказать, что «люди эволюционируют к рациональности, учась на ошибках». Эти экспериментальные исследования стали побудительным мотивом для появления целого раздела эконофизики, посвященного «игре в меньшинство» (Minority Game) [27]. Цель этой «игры» – показать на простой модели, каким образом экономические агенты с ограниченной рациональностью при неполной информации могут создавать эффективный рынок.

1.5. «Самоорганизованная критичность» в экономических системах

Для описания катастроф в экономических системах в [27] вводят управляющий параметр $\alpha(t) = I(t)/N$, где $I(t)$ – число различных состояний фундаментальной информации, N – число возможных состояний. Например, при описании динамики финансового рынка через N обозначают число агентов, каждый из которых может выбирать одно из двух состояний: buy (+1) или sell (–1). Эти состояния описываются функциями $a_i(t)$ [$i = \overline{1,N}$; $t = \overline{0,\infty}$] [3,44]. Выражение $A(t)$ определяется по формуле [27]:

$$A(t) = \sum_{i=1}^{N} a_i(t)$$

.

В некоторой точке α_C в нелинейной системе наблюдается фазовый переход. При этом, если $\alpha < \alpha_C$ система находится в симметричной фазе, где не существует никакой дополнительной информации, которую можно было бы использовать для более точного предсказания, и рынок оказывается эффективным. Напротив, если $\alpha > \alpha_C$ система находится в асиметричной фазе, где наиболее информированные агенты имеют действительное преимущество над остальными и рынок оказывается не эффективным. Кроме того, модель позволяет детально проследить, как на рынке образуется равновесие и как оно нарушается по причине того, что агенты, которые ведут себя независимо, начинают вести себя единым образом. На эту модель можно отчасти ориентироваться и при решении более важных проблем. Одна из проблем подобного рода, которая возникает при исследовании любой адаптивной системы, связана с очень быстрым и очень резким изменением ее состояния. В результате такого изменения, называемого катастрофой, система приходит в соответствие с окружающей средой. На практике катастрофы приносят огромные разрушения и неисчислимые бедствия. Поэтому очень важно понять их причину и научиться их вовремя предсказывать. Пер Бак [45] разработал целостную теорию таких явлений и назвал ее теорией

«самоорганизованной критичности» (Self-Organized Criticality). Он предположил, что катастрофы в сложных системах происходят не только вследствие внешних причин, но и вследствие того, что мелкие события, складываясь вместе, могут приводить к цепной реакции. Для иллюстрации подобного явления Бак обычно использует метафору кучи песка, которая медленно насыпается сверху. При этом очевидно, что время от времени будет возникать ситуация, когда достаточно всего лишь одной песчинки, чтобы вызвать лавину. После этого куча оседает, и процесс повторяется снова. Метафора кучи песка позволяет понять многие природные и социальные системы, у которых мы видим одну и ту же динамику: эволюционируя, эти системы самоорганизуются до критического предела, после чего стремительно разрушаются, чтобы затем опять спонтанно организоваться. В работе [46], которая является одной из базовых в эконофизике, авторы применили теорию самоорганизованной критичности к фондовому рынку. Они построили модель, в которой все действующие на этом рынке агенты были разделены на рациональных инвесторов (агентов, которые покупают и продают акции исходя из разницы между котировкой акции и «справедливой» ценой) и шумовых трейдеров (агентов, которые следуют тренду, чтобы извлечь прибыль благодаря краткосрочным изменениям на рынке). Большую часть времени число первых и вторых сбалансировано. Однако, когда цены начинают расти, увеличивается число рациональных инвесторов, желающих продать акции и уйти с рынка. На их место приходит все большее число шумовых трейдеров, которых привлекают растущие цены. Таким образом, возникает фазовый переход, в результате которого резко возрастает относительное число шумовых трейдеров. Это приводит к сильному росту цен, образованию «пузырей» и последующим обвалам. Следует заметить, что теория самоорганизованной критичности дает возможность лишь качественно понять возникновение катастроф. Однако она не позволяет проследить возникновение и развитие каждой отдельной катастрофы. Эту проблему применительно к

фондовым рынкам отчасти решил Дидье Сорнетте [47]. Он показал, что нелинейное взаимодействие рациональных инвесторов и шумовых трейдеров может приводить к появлению критической точки t_C на временной оси, в которой вероятность резкого обвала является максимальной. В окрестности этой точки ценовой ряд $p(t)$ имеет вид [47]:

$$p(t) = a_0 + a_1(t_C - t)^{\gamma}[1 + b\cos(\omega \ln(t_C - t) + C)], \qquad (1)$$

где $a_0, a_1, \gamma, b, \omega, C$ – константы.

Из формулы (1) видно, что по мере приближения к моменту t_C функция $p(t)$ совершает все более быстрые колебания, период которых стремится к нулю. Таким образом, появление у функции $p(t)$ подобных осцилляции можно рассматривать в качестве предвестника катастрофы. При этом, поскольку отношение двух последовательных периодов таких осцилляции остается постоянным, то исходя из положения трех последовательных локальных максимумов t_n, t_{n+1}, t_{n+2} можно оценить значение t_C по формуле:

$$t_C = \frac{t_{n+1}^2 - t_{n+2}t_n}{2t_{n+1} - t_n - t_{n+2}}. \qquad (2)$$

Используя эту методику, Сорнет исследовал все основные крахи, известные из истории финансовых рынков. В результате выяснилось, что поведение цен перед крахом во всех случаях достаточно хорошо можно приблизить формулой (2). Однако при попытке использовать (2) для предсказания обвалов в реальном режиме времени, выяснилось, что она работает лишь в большинстве, но не во всех случаях. Как оказалось, окончательное решение проблемы предсказания катастроф является задачей более сложной и, возможно, до конца принципиально не разрешимой.

Особое место в эконофизике занимает работа [48] где неожиданно была обнаружена аналогия, которая существует между энергетическим каскадом в гидродинамической турбулентности и информационным каскадом на финансовом рынке. Как известно, в трехмерном турбулентном потоке кинетическая энергия передается от больших пространственных масштабов к малым, начиная с некоторого характерного масштаба, на котором происходит подкачка энергии со стороны внешней силы [27]. При этом существует некоторый интервал масштабов (инерционный интервал), где все наблюдаемые характеристики $g_i(x)$ (в данном случае статистические моменты поля скорости) уже не зависят от характерного масштаба и являются пространственно однородными. Это означает, что при изменении масштаба $x \to \lambda x$ функция $g_i(x)$ просто умножается на некоторое число $\mu = \mu(\lambda)$. Другими словами,

$g_i(\lambda x) = \mu(\lambda) g_i(x)$.

Решением этого уравнения является степенная функция

$g_i(x) = c\, x^{\alpha_i}$,

где c, α_i – некоторые константы. Сравните с фрактальной ролью степенных законов по Шредеру.

При этом, показатель α_i, называется размерностью функции $g_i(x)$. Степенные зависимости являются отличительным признаком масштабной инвариантности, поскольку коэффициент

$$\lambda^{\alpha_i} = \frac{g_i(\lambda x)}{g_i(x)}$$

не зависит от x. Это означает, что относительное значение наблюдаемой величины зависит только от отношения масштабов. При этом, распределение вероятности изменения Δx на временном

масштабе Δt является однородным при вариациях Δt от нескольких минут до нескольких дней. Кроме того, это семейство распределений $P_{\Delta t}(\Delta x)$ удивительно напоминает при соответствующей нормировке семейство распределений $P_{\Delta t}(\Delta u)$ для разности скоростей Δu в точках, находящихся на расстоянии Δl в турбулентном потоке. Это вызвано тем, что на рынке существует механизм распространения возмущений, напоминающий турбулентный каскад [27]. При этом, роль пространственных масштабов играют временные масштабы, а роль энергии – информация. Более того, как показано в [48], даже хорошо известная перемежаемость между турбулентным и ламинарным движением имеет соответствие на рынке в виде перемежаемости кластеров высокой и низкой волатильности временного ряда. Надо сказать, что эта работа оказалось правильной лишь отчасти. Однако во многом именно она инициировала целый поток работ, посвященный устойчивым степенным законам. Такие законы были установлены, в частности, для распределения доходности акций, объема продаж и числа сделок [27].

К самому известному результату эконофизики, можно отнести работу Мантенья и Стенли, где они экспериментально показали [27], что плотности вероятности $P(Z_{\Delta t})$ изменения индекса S&P500 на интервале Δt от 1 мин. до 1 мес. описываются «усеченным» распределением Леви. Такое распределение в своей центральной части (где изменения не превышают шести стандартных отклонений) совпадает с обычным распределением Леви, а на краях оно «опускается» ниже последнего, оставаясь при этом значительно выше гауссовых распределений с тем же σ. Распределение Леви нормированной на а величины $Z_{\Delta t}$ в данном случае для всех интервалов Δt имеет вид [27]:

$$P_l(\beta) = \frac{2}{\pi}\int_0^{\infty} \cos(\beta x)\, e^{-\gamma \Delta x^{\alpha}}\, dx \quad , \qquad (3)$$

где $\alpha = 1.4$, $\gamma = 0.00375$, $\beta = Z_{\Delta t} / \sigma$, $Z_{\Delta t} = Y(t) - Y(t - \Delta t)$, $Y(t)$ – значение индекса S&P500 в момент времени t. Заметим, что если графики с различными Δt привести к нормированному масштабу

$Z_S = Z_{\Delta t} / (\Delta t^{1/\alpha})$,

то все они совпадут. Это означает, что усеченное распределение Леви для интервалов от 1 мин. до 1 мес. является однородным по Δt, так же как и чистое распределение Леви (3). Дальнейшие исследования показали, что этот закон остается верным и для других индексов, а так же для отдельных акций. При этом в каждом случае появляется свой показатель α. Кроме того, оказалось, что начиная от $\Delta t = 1$ мес. и больше, это универсальное распределение начинает сходиться к гауссовому распределению.

1.6. Фрактальные структуры

Один из основополагающих принципов науки – существование объективных законов, которым подчиняется любой, спонтанно возникающий порядок, а также вера в то, что порядок может быть описан на языке математики [49].

Одним из примеров такого порядка является фрактальные структуры, которые возникает равно и в теории фазовых переходов, и в турбулентности, и в теории самоорганизованной критичности.

Согласно одному из определений Бенуа Мандельброта [49] «фракталом называется множество, хаусдорфова размерность которого строго больше его топологической размерности». Чтобы раскрыть это определение, напомним о двух принципиально различных подходах к понятию размерности. С точки зрения первого, размерность геометрической фигуры - это минимальное число координат, необходимых для ее описания как множества точек с сохранением структуры естественной близости: например, для описания линии достаточно одной координаты, для описания

поверхности – двух, для описания тела – трех координат. Поскольку эта размерность является топологическим инвариантом (т.е. сохраняется при взаимно-однозначном и непрерывном в обе стороны отображении), то ее называют топологической размерностью и обозначают D_T. Этот подход отражен в известных работах Пуанкаре, Брауэра, Менгера, Урысона, Лебега и др. С точки зрения второго подхода, размерность – это число D выражающее связь естественной меры геометрической фигуры (например, длины, площади или объема) с величиной (в данном случае длиной), положенной в основу исходной метрической системы. Если метрический эталон такой величины, принятый за единицу, увеличить (уменьшить) в b раз, то указанная мера уменьшится (увеличится) в b^D раз. Эту размерность называют метрической. Источником такого определения является следующее выражение для обычной меры M (длины, площади или объема) произвольной геометрической кривой, поверхности или тела:

$$M = \lim_{\delta \to 0}[N(\delta)\delta^D],\ D = 1,2,3,$$

где $N(\delta)$ – число симплексов (отрезков, клеток или кубов) с геометрическим фактором (линейным размером) δ, определяющих аппроксимацию исходного множества. На основе именно этого выражения, Хаусдорф в 1919 г. предложил свое знаменитое определение размерности для случая компактного множества в произвольном метрическом пространстве [50]:

$$D = \lim_{\delta \to 0}[\ln N(\delta) / \ln(1/\delta)], \qquad (4)$$

где $N(\delta)$ – минимальное количество шаров радиуса δ, покрывающих это множество. Заметим, что для обычных регулярных геометрических фигур $D = D_T$. Однако, для более экзотических множеств (а именно фракталов) $D > D_T$. Следует отметить, что если исходное множество погружено в евклидово

пространство, то в определении (4) вместо покрытий этого множества шарами можно брать любые другие его аппроксимации простыми фигурами (например, клетками) с геометрическим фактором δ. При этом наряду с исходной сферической размерностью D появляются новые фрактальные размерности (клеточная, внутренняя и т.д.), которые, как предельные значения при $\delta \to 0$, обычно совпадают. Однако скорость сходимости к этому пределу для таких размерностей может быть весьма различной. Поскольку на практике размерность вычисляется на основе конечного числа покрытий или других аппроксимаций, то правильный выбор последних может иметь принципиальное значение.

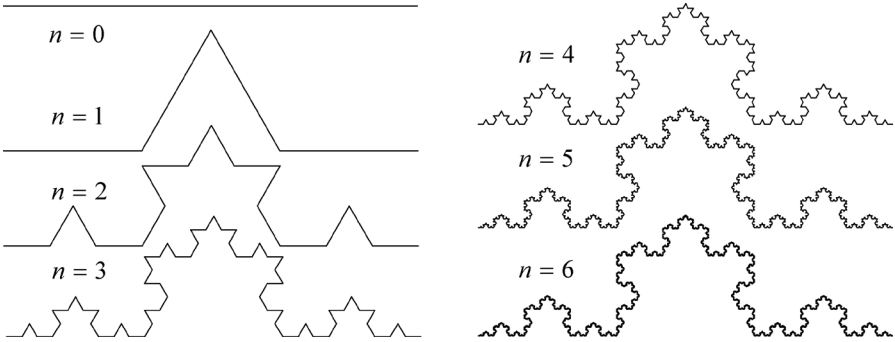


Рис. 2 – Кривая Коха на первых шести шагах итерации

Первоначальное определение (4) рассматривалось лишь как удобное средство для систематизации различных «патологических» множеств типа функции Веерштрасса, кривых Пеано и их многочисленных разновидностей, которые возникают естественным образом при обсуждении оснований математического анализа. Поскольку все такие множества, несмотря на свою крайнюю нерегулярность, обычно инвариантны относительно масштабных преобразований, то все они имеют хаусдорфову размерность. Причем, как правило, эта размерность оказывается дробной. Далее мы рассмотрим простейшие примеры

таких множеств, чтобы на этих примерах пояснить идею, которая станет основной при исследовании временных рядов.

Первый пример – это кривая Коха. Строится она с помощью итеративной (повторяющейся) процедуры. На нулевом шаге берется единичный отрезок (рис. 2). На первом шаге этот отрезок делится на три равные части. Затем на средней части строится равносторонний треугольник и его основание выбрасывается. Далее на каждом следующем шаге повторяется та же процедура со всеми оставшимися отрезками. Множество, которое получается в результате такой итеративной процедуры, называется кривой Коха. Надо сказать, что все модельные фракталы строятся как предел последовательности некоторых комплексов. Такие комплексы Мандельброт назвал *предфракталами*. Для вычисления размерности кривой Коха в качестве последовательности аппроксимаций удобно воспользоваться ее представлением на n-м шаге итерации (предфракталом n-го поколения). В этом случае оно будет аппроксимировано 4^n отрезками, уменьшенными в 3^n раз. И хотя такие аппроксимации не являются покрытиями, однако именно они для данного объекта являются наиболее естественными. Если теперь взять $\delta = (1/3)^n$, то $N(\delta) = 4^n$. Переходя к пределу при $n \to \infty$ по формуле получаем $D = \ln 4 / \ln 3 \approx 1.26...$. Такая размерность называется внутренней. Нетрудно показать, что в данном случае она совпадает с хаусдорфовой размерностью.

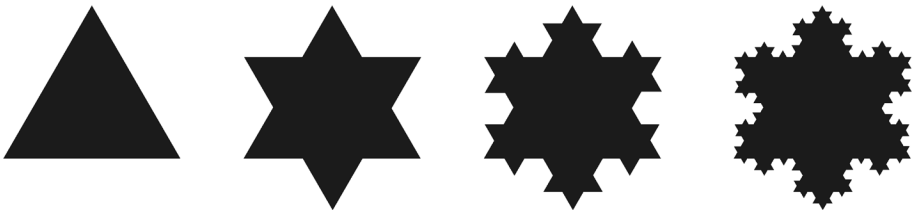
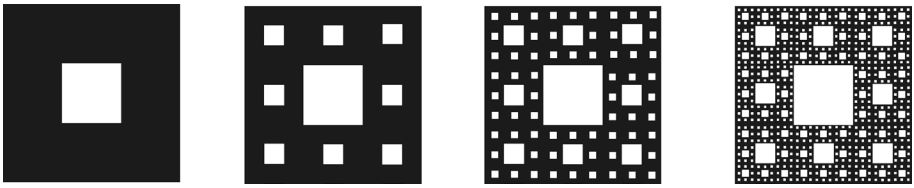
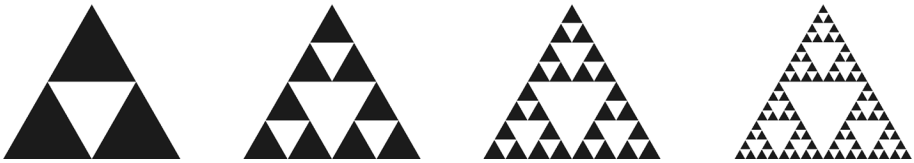


Рис. 3 – Построение снежинки Коха



а)



б)

Рис. 4 – Построение ковра (*а*) и салфетки (*б*) Серпинского

Попутно заметим, что если в нулевом приближении взять не отрезок, а треугольник с единичной стороной и проделать с каждой из его сторон описанную выше процедуру, то в результате получится множество, известное как снежинка Коха (рис. 3).

Второй пример – это ковер Серпинского, который строится так. На нулевом шаге берется единичный квадрат. На первом – этот квадрат делится на девять равных квадратов и выбрасывается средний. Далее на каждом следующем шаге эта процедура повторяется со всеми оставшимися квадратами. Множество, которое получается в пределе такой итеративной процедуры, называется ковром Серпинского (рис. 4,*а*). Для вычисления фрактальной размерности этого множества в качестве n-й аппроксимации опять используем предфрактал n-го поколения. В этом случае оно будет покрыто 8^n квадратами, уменьшенными в 3^n раз. Это означает, что при $\delta = (1/3)^n$, $N(\delta) = 8^n$. Переходя к пределу при $n \to \infty$, получаем $D = \ln 8 / \ln 3 \approx 1.89...$. Аналогичным образом строится салфетка Серпинского, представленная на рис. 4,*б*.

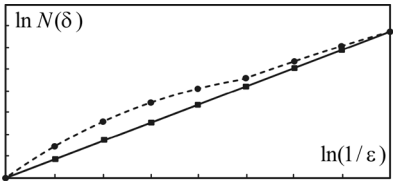


Рис. 5 – Зависимость в двойном логарифмическом масштабе числа симплексов $N(\delta)$ составляющих аппроксимацию, от характерного размера симплекса δ. Если такие аппроксимации являются предфракталами, то эта зависимость является линейной. В противном случае эта зависимость будет лишь приближаться к линейной в асимптотике при $\delta \to 0$ [27]

Заметим, что аппроксимации, которые мы использовали в этих случаях, в некотором смысле оптимальным образом приближают соответствующий фрактал на каждом шаге итерации. Именно этот факт позволяет нам получить значение D уже на первом шаге, что невозможно при использовании других аппроксимаций. Для того чтобы пояснить это более подробно, для какого-либо из этих двух примеров построим график, где по горизонтальной оси отложены значения $\ln(1/\delta)$, а по вертикальной – значения $\ln N(\delta)$ (рис. 5). Поскольку в этом масштабе все степенные функции являются линейными, то степень D определяется как тангенс угла наклона соответствующей прямой. Очевидно, что при $\delta = (1/3)^n$ все данные для указанного примера будут находиться на одной прямой. Если же для вычисления D в этом примере использовать другие аппроксимации, то получится график, который лишь асимптотически приближается к прямой. При этом, если аппроксимации выбраны неудачно, то такое приближение может быть весьма медленным. Далее, при анализе временных рядов, это соображение будет основным для осознания необходимости перехода к оптимальным аппроксимациям.

Когда рассматривались подобные примеры, то никто не надеялся, что множества с нетривиальной хаусдорфовой размерностью могут иметь какое-либо отношение к природе. Теперь, во многом благодаря усилиям Мандельброта мы знаем, что фракталы окружают нас повсюду. Некоторые из фракталов непрерывно меняются, подобно движущимся облакам или мерцающему пламени, в то время как другие, подобно деревьям или нашим сосудистым системам, сохраняют структуру, приобретенную в процессе эволюции. При этом реальный диапазон масштабов, где может наблюдаться фрактальная структура, простирается от расстояний между молекулами в полимерах до расстояния между скоплениями галактик во вселенной. Можно сказать, что все сильные нерегулярности в природе стремятся обрести самоподобную (фрактальную) структуру как наиболее энергетически выгодную.

Общую специфику природных фракталов мы поясним на примере береговой линии, тем более, что этот пример представляет исторический интерес. Именно здесь впервые была обнаружена закономерность, впоследствии осмысленная как фрактальность. В работе известного английского метеоролога и картографа Ричардсона (вышедшей уже после его смерти в 1961 г.) с помощью последовательности все более точных карт измерялся периметр береговой линии Великобритании. Данные наносились на график, где по горизонтальной и вертикальной осям откладывались соответственно логарифмы масштабного фактора карты m и периметра $P(m)$. Результат оказался поразительным. Данные почти точно легли на прямую. Это означает, что благодаря «довескам», которые появляются по мере уменьшения масштабного фактора карты, периметр «расходится» (т.е. $P(m) \to \infty$ при $m \to 0$), и причем по степенному закону. Отсюда следует, что береговая линия имеет фрактальную размерность. Действительно, поскольку масштабный фактор карты m прямо пропорционален минимальному различимому размеру δ («разрешению» карты), то измерение

периметра с помощью последовательности все более точных карт можно представлять как измерение с помощью последовательности все более точных аппроксимаций береговой линии ломаными с размером звена δ. Тогда выполнение степенного закона при переходе к более точным картам просто означает что выполняется степенной закон [27]:

$$P(\delta) \sim \delta^{-\alpha},$$

где $P(\delta)$ – периметр, соответствующий разрешению δ, α – константа, а символ $\sim$ означает, что при $\delta \to 0$ выражения справа и слева (которые обычно стремятся к нулю или бесконечности) различаются не более чем на константу. Если теперь учесть, что $P(\delta) = N(\delta)\delta$, где $N(\delta)$ – число звеньев ломаной, аппроксимирующей периметр, то для $N(\delta)$ получим выражение:

$$N(\delta) \sim \delta^{-(\alpha+1)},$$

откуда сразу следует, что береговая линия – фрактал с размерностью $D = \alpha + 1$ (см. формулу (4)).

На основе этого примера отметим основные особенности всех природных фракталов в их отличии от модельных [27]:

- во-первых, свойство самоподобия для них выполняется как правило лишь в среднем;

- во-вторых, при вычислении фрактальной размерности степенной закон проявляет себя как «промежуточная асимптотика» (т.е. при $\delta \to 0$ берется масштаб, малый по сравнению с некоторым характерным, но много больше некоторого минимального).

Последнее означает, что при построении соответствующей зависимости в двойном логарифмическом масштабе следует исключить масштабы порядка минимального, причем особую роль

здесь играет «искусство» правильного выбора системы аппроксимаций.

1.7. Определение фрактальной размерности временных рядов

Ниже приведем простой способ вычисления фрактальной размерности D через клеточную размерность D_C. Для определения размерности D_C разобьем плоскость, на которой определен график временного ряда на клетки размером δ и определим число клеток $N(\delta)$, где находится хотя бы одна точка этого графика [27]. Затем, при различных δ в двойном логарифмическом масштабе построим функцию $N(\delta)$, которая аппроксимируется прямой, например, с помощью метода наименьших квадратов (МНК). Тогда D_C определяется по углу наклона этой прямой. Главным недостатком такого метода является то, что выход на степенную асимптотику функции $N(\delta)$ при $\delta \to 0$ обычно происходит крайне медленно. Более популярным является метод определения D через показатель Херста H, который для гауссовых процессов связан с D соотношением $H = 2 - D$. Однако для надежного вычисления H требуется слишком большой репрезентативный масштаб, содержащий несколько тысяч данных [50]. Внутри этого масштаба временной ряд, как правило, меняет характер своего поведения много раз.

Чтобы связать локальную динамику соответствующего процесса с фрактальной размерностью временного ряда, необходимо определить размерность D локально. Для этого попытаемся найти последовательность аппроксимаций, которая при фиксированном δ была бы в некотором смысле оптимальной. Умножим обе части (4) на $1/\delta$ и введем D под знак логарифма. В результате получим:

$N(\delta) \sim \delta^{-D}$ при $\delta \to 0$. (5)

Если теперь умножить обе части (5) на δ^2, то определение размерности можно переписать в виде степенного закона для площади аппроксимаций $S(\delta)$:

$S(\delta) \sim \delta^{2-D}$ при $\delta \to 0$. (6)

Заметим, что такая форма в отличие от (5) не требует, чтобы симплексы, из которых состоит аппроксимация, были одинаковыми. Достаточно того, чтобы они имели один и тот же геометрический фактор δ. Это позволяет нам использовать аппроксимации, которые при данном δ в некотором смысле наилучшим образом приближают исходную функцию.

1.8. Фрактальное моделирование одномерных временных рядов

Действительно, пусть на отрезке $[a,b]$ задана функция $y = f(t)$, имеющая не более конечного числа точек разрыва первого рода: именно такие функции естественно рассматривать в качестве модельных, например, для финансовых временных рядов [27]. Введем равномерное разбиение отрезка

$\omega_m = [a = t_0 < t_1 < ... < t_m = b]$,

где $t_i - t_{i-1} = \delta = (b-a)/m \ \ (i = 1,2,...,m)$. Покроем график этой функции прямоугольниками таким образом, чтобы это покрытие было минимальным (по площади) в классе покрытий прямоугольниками с основанием δ (рис. 6). Тогда высота прямоугольника на отрезке $[t_{i-1}, t_i]$ будет равна амплитуде $A_i(\delta)$, которая является разностью между максимальным и минимальным значением функции $f(t)$ на этом отрезке.

Введем величину:

$$V_f(\delta) \equiv \sum_{i=1}^{m} A_i(\delta) \quad . \qquad (7)$$

Тогда полную площадь минимального покрытия $S_\mu(\delta)$ можно записать в виде:

$$S_\mu(\delta) = V_f(\delta)\delta \quad .$$

Поэтому из (6) следует, что

$$V_f(\delta) \sim \delta^{-\mu} \text{ при } \delta \to 0 \, , \qquad (8)$$

где

$$\mu = D_\mu - 1 \quad . \qquad (9)$$

Размерность D_μ в [27] названа ***размерностью минимального покрытия.*** Чтобы соотнести D_μ с другими размерностями и, в частности, с клеточной размерностью D_C, построим клеточное разбиение плоскости графика функции $f(t)$, как показано на рис. 6. Пусть $N_i(\delta)$ – число клеток, покрывающих график $f(t)$ внутри отрезка $[t_{i-1}, t_i]$. Тогда:

$$0 < N_i(\delta)\delta^2 - A_i(\delta)\delta < 2\delta^2 \quad . \qquad (10)$$

Разделим это соотношение на δ и просуммируем по i с учетом (7). В результате получим

$$0 < N(\delta)\delta - V_f(\delta) < 2(b-a) \, ,$$

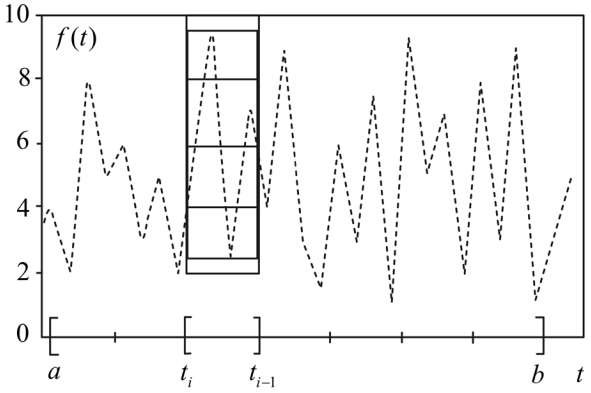


Рис. 6 – Минимальное (черный прямоугольник) и клеточное (серый прямоугольник) покрытия функции $f(t)$ на интервале $[t_{i-1},t_i]$, длиной δ [27]

где $N(\delta)=\sum N_i(\delta)$ – есть полное число клеток размера δ, покрывающих график функции $f(t)$ на отрезке $[a,b]$. Переходя к пределу при $\delta\to 0$, с учетом (8) и (9), получим:

$$N(\delta)\delta \cdot V_f(\delta) \cdot \delta^{\mu} = \delta^{1-D_{\mu}},$$

С другой стороны, согласно (6)

$$N(\delta)\delta = S_C(\delta)\delta^{-1} \cdot \delta^{1-D_C}.$$

Следовательно, $D_C = D_{\mu}$.

Заметим, однако, что, несмотря на это равенство, для реальных фрактальных функций минимальные и клеточные покрытия могут давать различные приближения величины $S(\delta)$ к асимптотическому режиму (6), причем величина этого различия может быть весьма значительной.

Возвращаясь к формуле (9) заметим, что поскольку $D_{\mu} = D$ и для одномерной функции $D_T = 1$, то $\mu = D - D_T$. В [27] индекс μ назван

индексом фрактальности. При анализе финансовых временных рядов его рассматривают как основной фрактальный показатель. На рис. 7 представлено поэтапное фрактальное моделирование временного ряда на основе его тренда.

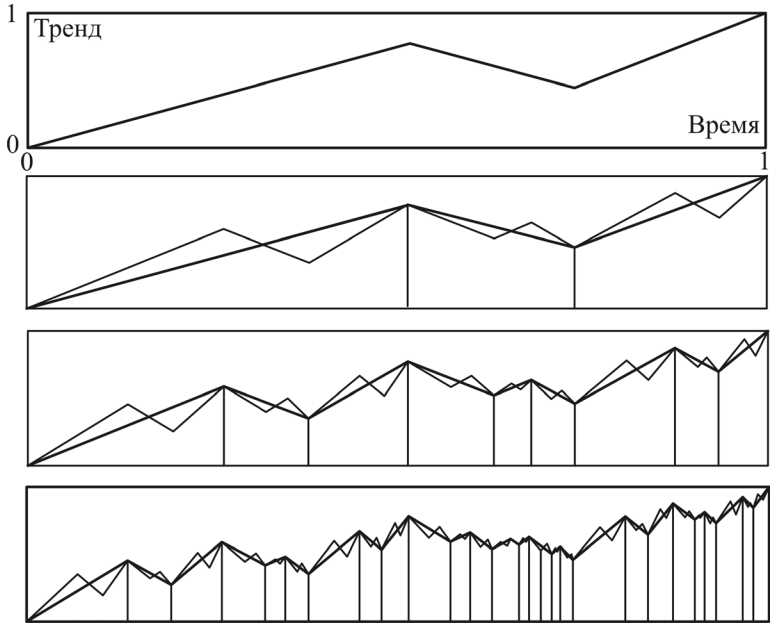


Рис. 7 – Поэтапное фрактальное моделирование временного ряда на основе его тренда

1.9. Финансовые временные ряды

Особое значение фрактального анализа временных рядов в том, что он учитывает поведение системы не только в период измерений, но и его предысторию [27].

Для фрактальных временных рядов на интервале $t_0 < t < T$ размах параметра R определяется как

$$R(\tau) = \max_{1 \le t \le \tau} B(t) - \min_{1 \le t \le \tau} B(t)$$,

и зависит от времени t степенным образом:

$$R(t) = R(t_0)\left(\frac{t}{t_0}\right)^{2-D}$$,

где D – фрактальная размерность временного ряда. Исходя из данного выражения, можно предсказать возможное значение размаха интересующего параметра в будущем.

Фрактальная размерность, является показателем сложности кривой. Анализируя чередование участков с различной фрактальной размерностью и тем, как на систему воздействуют внешние и внутренние факторы, можно научиться предсказывать поведение системы.

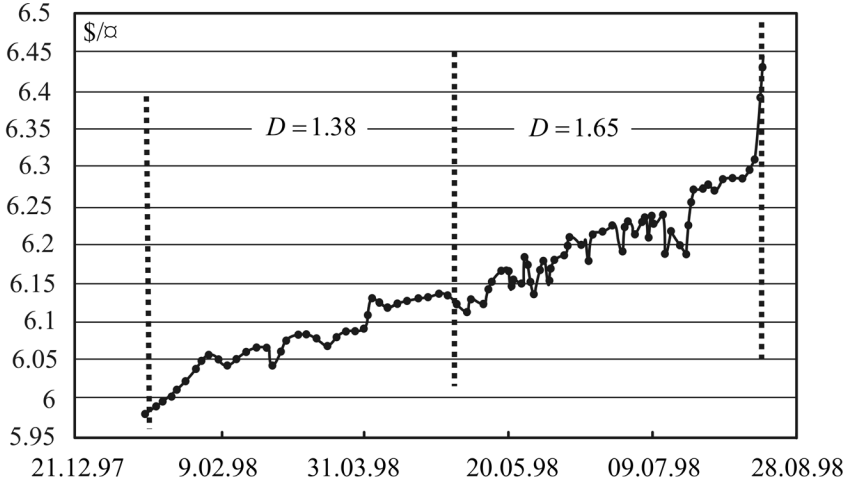


Рис. 8 – Зависимость курса доллара по отношению к российскому рублю во время кризиса 1998 года

И что самое главное, можно научиться предсказывать диагностировать и предсказывать нестабильные состояния.

Существенным моментом этого подхода является наличие критического значения фрактальной размерности временной кривой, при приближении к которому система теряет устойчивость и переходит в нестабильное состояние и параметры быстро либо возрастает, либо убывает, в зависимости от тенденции, имеющей место в данное время.

Это хорошо видно, если анализировать динамику курса доллара по отношению к российскому рублю во время кризиса 1998 года. Она изображена на рис. 8. Проследив изменение динамики фрактальной размерности временного ряда, можно заметить резкий подъем фрактальной размерности непосредственно перед скачком курса доллара.

То есть фрактальная размерность определенной величины может использоваться как индикатор кризиса или "флаг" катастрофы.

Анализ экспериментальных данных показывает, что линия тренда для временного ряда хорошо описывается уравнением [27]

$$\overline{y}(t) = \overline{y}(t_0) + \frac{K_f(t_0)(t - t_0)}{(D - D_0)^{\beta}},$$

где $\overline{y}(t_0)$ – среднее значение величины за период, предшествующий прогно-зируемому, K_f, β – коэффициенты, t_0 – период времени, предшествующий прогнозируемому, t – время на которой делается прогноз, D_0 – фрактальная размерность на периоде предыдущем прогнозируемому.

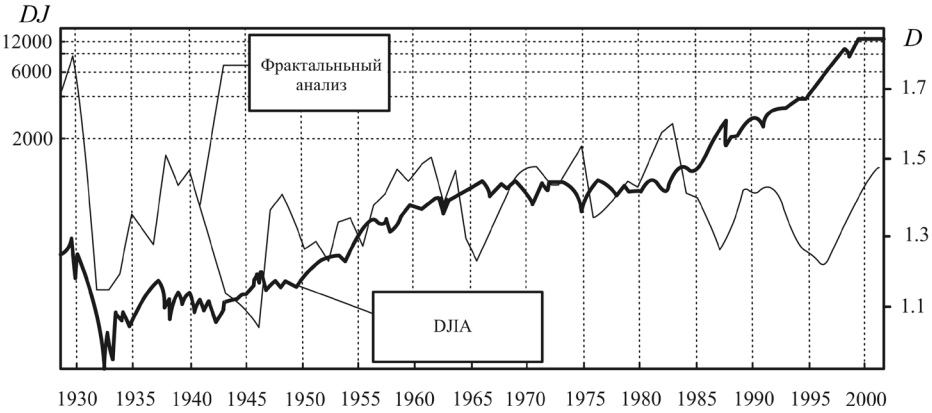


Рис. 9 – Динамика биржевого индекса Доу-Джонса [27]

Также величина фрактальной размерности может служить индикатором количества факторов, влияющих на систему. При фрактальной размерности менее 1.4, на систему влияет одна или несколько сил, двигающих систему в одном направлении. Если размерность около 1.5, то силы, действующие на систему, разнонаправлены, но более или менее компенсируют друг друга. Поведение системы в этом случае является стохастическим и хорошо описывается классическими статистическими методами. Если же фрактальная размерность значительно более 1.6, система становится неустойчивой и готова перейти в новое состояние.

В этом плане можно рассмотреть динамику биржевого индекса Доу-Джонса. Она показана на рис. 9. Там же представлена среднегодовая фрактальная размерность временного ряда этого биржевого индекса. Во время достаточно стабильных периодов и медленных подъемов фрактальная размерность временного ряда оставалась достаточно невысокой, в то время как в периоды кризисов суммарная фрактальная размерность возрастала.

Естественно, это общие закономерности, и для каждой системы надо устанавливать конкретные закономерности факторов влияния.

Классические финансовые модели предсказывают, что критические события должны происходить крайне редко. Как правило, они базируются на вероятности, вычисляемой по Гауссу или Пуассону. При этом существенно занижается вероятность критических событий. Краеугольный камень финансов современная портфельная теория (portfolio theory), которая пытается максимизировать отдачу для данного уровня риска. Математика, лежащая в основе портфельной теории, обращается с чрезвычайными ситуациями с некоторым пренебрежением: она считает большие рыночные изменения слишком маловероятными. На наш взгляд она недооценивает формирующее влияние кризисов на систему.

Фрактал – геометрическая форма, которая может быть разделена на части, каждая из которых – уменьшенная версия целого. В финансах эта концепция – не беспочвенная абстракция, а теоретическая переформулировка практичной рыночной поговорки – а именно, что движения акции или валюты внешне похожи, независимо от масштаба времени и цены. Наблюдатель не может сказать по внешнему виду графика, относятся ли данные к недельным, дневным или же часовым изменениям. Известно правило первого месяца, согласно которому, как утверждают некоторые биржевые аналитики, как дела на бирже идут в первый месяц, примерно так же они будут идти в течение всего года. Это качество определяет диаграммы как фрактальные кривые и делает доступными многие мощные инструменты из математического и компьютерного анализа.

Наиболее популярными представителями фрактальных временных функций являются финансовые временные ряды (ряды цен акций и курсов валют). Существует надежное численное подтверждение фрактальной структуры таких рядов [27]. Теоретически же фрактальность обычно связывают с тем, что для устойчивости рынка на нем должны присутствовать инвесторы с разными инвестиционными горизонтами (от нескольких часов до

нескольких лет). Это и приводит к масштабной инвариантности (отсутствию выделенного масштаба) ценовых рядов на соответствующем временном интервале. С помощью индекса фрактальности μ были исследованы ценовые ряды акций тридцати компаний, входящих в индекс Доу-Джонса (Dow Jones Industrial Index) с 1970 по 2002 г. Каждая запись соответствует одному торговому дню и содержит четыре значения: информацию о минимальной и максимальной цене, а также цены открытия и закрытия. В литературе финансовые ряды обычно изображают с использованием т.н. «японских свечей». Фрагмент такого ряда для акций компании Coca-Cola представлен на рис. 10.

Для простоты анализа ограничимся последними $2^{12} = 4096$ записями для каждой компании. При вычислении индекса μ использовалась последовательность m вложенных разбиений ω_m, где $m = 2^n$, $n = 0,1,2,...,12$. Каждое разбиение состояло из 2^n интервалов, содержащих 2^{12-n} торговых дней. Для каждого разбиения ω_m вычислялось значение $V_f(\delta)$ (7). Здесь $A_i(\delta)$ равна разности между максимальной и минимальной ценой на интервале $[t_{i-1}, t_i]$ (в частности, если $\delta = \delta_0$, то $A_i(\delta)$ равна разности между максимальной и минимальной ценой за день). Типичный пример поведения $V_f(\delta)$ в двойном логарифмическом масштабе представлен на рис. 11 для компании Microsoft. Из рисунка видно, что данные почти точно ложатся на прямую линию, кроме двух последних точек, где линейный режим имеет излом.

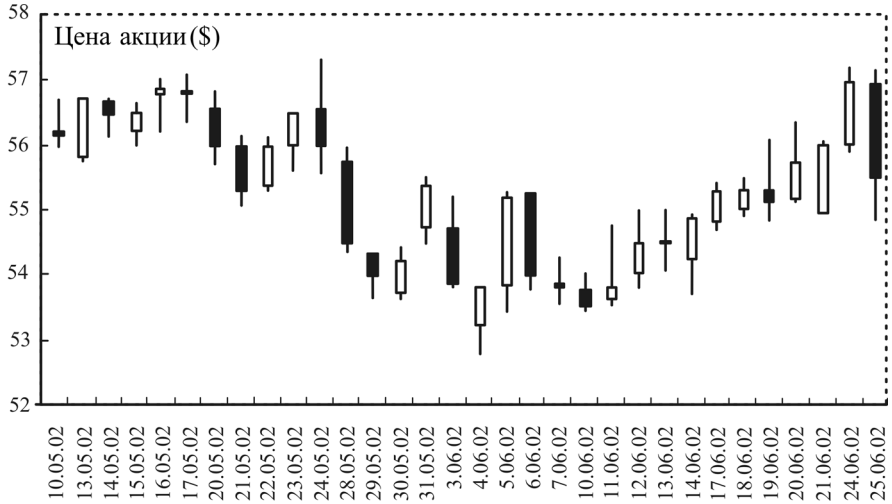


Рис. 10 – Типичное поведение цен на интервале 32 дня (использован дневной график цен акций компании Coca-Cola). В финансах графики цен принято изображать не одномерными линиями, а интервалами (так называемые баровые графики или графики в виде японских свечей). Один прямоугольник (называемый телом свечи) с двумя штрихами сверху и снизу (называемыми тенями свечи) изображает колебания цен в течение дня. Верхняя точка верхней тени показывает максимум цены, нижняя точка нижней тени - минимум цены за день. Верхняя и нижняя границы тела свечи показывают цену открытия и закрытия торгов. При этом, если тело белого (черного) цвета, то закрытие выше (ниже) открытия

Для определения значения μ по этим данным следует исключить две последние точки и найти линию регрессии $y = ax + b$ с помощью метода наименьших квадратов (МНК). Тогда $\mu = -a$.

При уровне надежности $a = 0.95$ в приведенном примере $\mu = 0.472 \pm 0.008$, $R^2 = 0.999$. Здесь R^2 – коэффициент детерминации для линии регрессии. Результаты для остальных компаний следующие:

$\mu_{\min} = 0.469 \pm 0.019$, $R^2 = 0.999$ (Intel Corporation);

$\mu_{\max} = 0.532 \pm 0.007$, $R^2 = 0.997$ (International Paper Company).

В [27] говорится, что для каждой из 30 известных компаний, график $V_f(\delta)$ почти точно ложится на прямую также и на всех меньших репрезентативных интервалах вплоть до 32-*x*, а иногда даже до 16-*mu* дней. Типичный пример $V_f(\delta)$ на интервале 32 дня представлен на рис. 12. При этом на интервалах меньших, чем 500 дней, излом линейной части графика, как правило, исчезает.

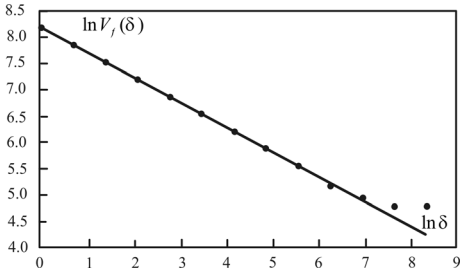


Рис. 11 – Результат вычисления индекса вариации для временного ряда цен акций компании Microsoft на интервале 4096 дней. Прямая $y = ax + b$ определялась методом МНК по всем точкам, исключая две последние

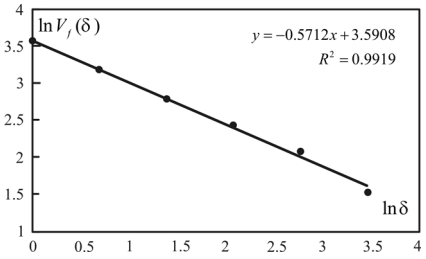


Рис. 12 – Результат вычисления $V_f(\delta)$ в двойном логарифмическом масштабе для временного ряда, представленного на рис. 10. Прямая $y = ax + b$ построена методом МНК. Для определения индекса μ следует отождествить $\mu = -a$

Чтобы оценить преимущества построенного алгоритма [27] сравним его с методом вычисления фрактальной размерности с помощью показателя Херста H. Для фрактальных временных рядов этот метод традиционно считается наиболее эффективным.

Как известно, показатель Херста H определяется на основе предположения, что

$$< \left| f(t+\delta) - f(t) \right| > \propto \delta^H \text{ при } \delta \to 0, \qquad (11)$$

где угловые скобки означают усреднение по временному интервалу. Чтобы сравнить индекс μ с показателем H, введем следующее естественное определение средней амплитуды $< A_i(\delta) >$ на разбиении ω_m:

$$< A(\delta) > \equiv m^{-1} \sum_{i=1}^{m} A_i(\delta). \qquad (12)$$

Умножим (7) на $m^{-1} \propto \delta$ и подставим в (10). Получим:

$$< A(\delta) > \propto \delta^{H_\mu} \text{ при } \delta \to 0, \qquad (13)$$

где

$$H_\mu \equiv 1 - \mu. \qquad (14)$$

Если $f(t)$ – реализация гауссова случайного процесса, то показатель H связан с размерностью D а следовательно, и с индексом μ, соотношением:

$$H = 2 - D_\mu = 1 - \mu.$$

Следовательно, в этом случае $H = H_\mu$. Однако, реальные финансовые ряды, как правило, не являются гауссовыми (см., например, [51]) и поэтому значения H_μ и H могут сильно различаться.

Действительно, в формуле (13) мы имеем степенной закон для средней амплитуды функции $f(t)$ на интервале длиной δ, в то время как в формуле (11) мы имеем степенной закон для средней

разности между начальным и конечным значением $f(t)$ на том же интервале. Как оказывается, индекс μ вычисляется на порядок более точно, чем показатель Херста H в подавляющем большинстве случаев. Например, это справедливо для ценового ряда компании Alcoa Inc., которая стоит первой по алфавиту в списке индекса Доу-Джонса. Рассмотрим последовательность 32-х дневных интервалов исходного ценового ряда (состоящего из 8145 торговых дней, каждому из которых соответствует 4 значения), смещенных друг относительно друга на один день. Общее число таких интервалов $N = 8145 - 32 = 8113$. При вычислении H для этих интервалов будем использовать $32 + 1 = 33$ цены закрытия $C(t)$ и усреднять по непересекающимся подинтервалам длиной $\delta = 2^n$, где $n = \overline{0,5}$. Это означает, что:

$$\left\langle \left| C(t+\delta) - C(t) \right| \right\rangle = (\delta / 32) \sum_{i=0}^{32/\delta} \left| C(t_{i+1}) - C(t_i) \right|,$$

где $t_{i+1} = t_i + \delta$.

В соответствии с предположением (11) вычислим для каждого δ

$$x = \ln \delta, \quad y = \left\langle \left| C(t+\delta) - C(t) \right| \right\rangle$$

и аппроксимируем результаты вычисления прямой

$$y = ax + b \tag{15}$$

с помощью МНК. После чего положим $H = a$.

Для определения индекса μ используем (7) в виде

$$V_f(\delta) = \sum_{i=1}^{32/\delta} A_i(\delta),$$

где $A_f(\delta)$ – амплитуда $f(t)$ на интервале $[t_i, t_i + \delta]$.

В соответствии с предположением (8) вычислим для каждого δ

$$x = \ln \delta, \quad y = \ln V_f(\delta)$$

и аппроксимируем полученные результаты прямой (15) с помощью МНК. После чего положим $\mu = -a$.

В качестве критериев точности расчетов выберем ширину доверительного интервала Δ, в который значение величины μ (или H) попадает с вероятностью 0.90 и $K = 1 - R^2$, где R^2 – коэффициент детерминации (если точки ложатся точно на прямую, то $R^2 = 1$ и $K = 0$.

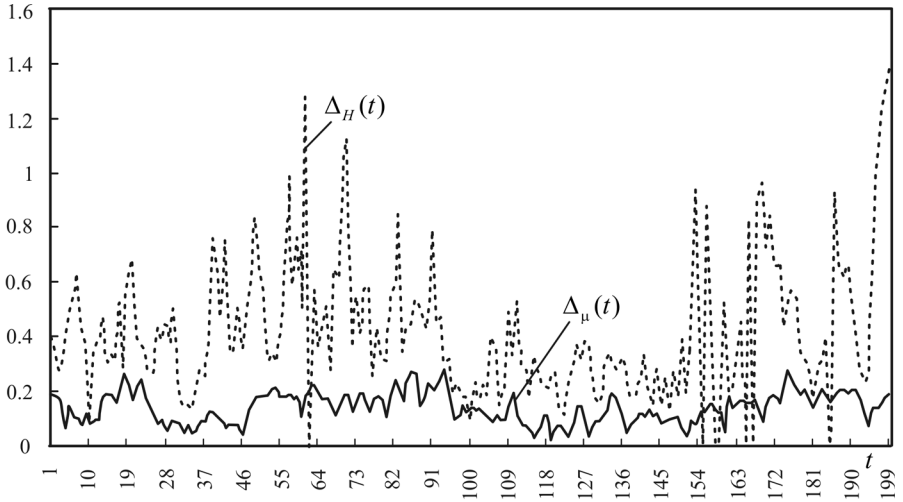


Рис. 13 – Типичный фрагмент временного ряда ширины доверительных интервалов $\Delta_H(t)$ и $\Delta_\mu(t)$, построенных по ряду цен закрытия $C(t)$ для компании Alcoa Inc.

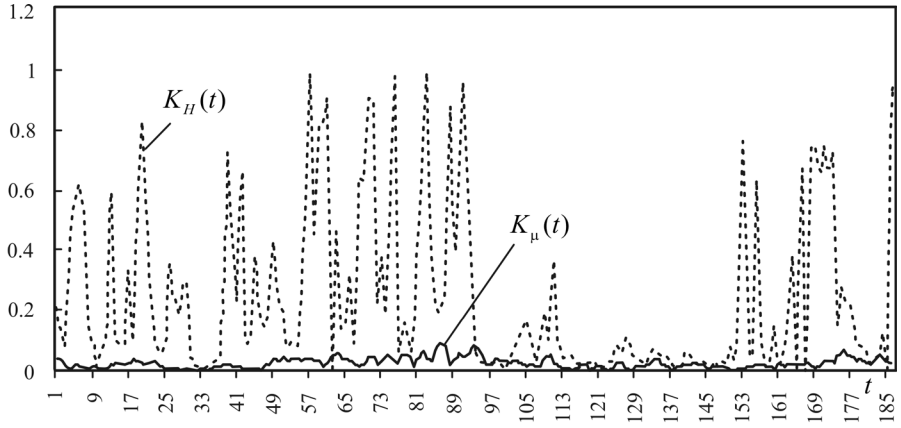


Рис. 14 – Соответствующий фрагмент ряда для значений $K_H(t)$ и $K_\mu(t)$, построенных по тому же самому ряду $C(t)$

Типичные фрагменты графиков функций $\Delta_\mu(t)$, $\Delta_H(t)$ и $K_\mu(t)$, $K_H(t)$, построенных по интервалам, правый конец которых совпадает с моментом t, представлены на рис. 13 и рис. 14. Как видно из этих рисунков, в подавляющем большинстве случаев индекс μ определен на порядок точнее, чем H.

Результаты проведенных вычислений следующие [27]:

$< \Delta_\mu >= 0.107$, $< \Delta_H >= 0.452$;

$< K_\mu >= 0.0147$, $< K_H >= 0.245$.

Кроме этого, оказалось, что $\Delta_\mu < \Delta_H$ для 99 %, и $K_\mu < K_H$ для 91 % исследованных интервалов. Аналогичные результаты были получены и для других ценовых рядов. В качестве типичного примера на рис. 13 и рис. 14 мы представляем результаты вычисления μ и H на одном и том же 32-дневном интервале, показанном на рис. 10. Для $a = 0.90$ результаты оказались следующими:

$\mu = 0.571 \pm 0.071$, $H_\mu = 0.429 \pm 0.071$, $R^2_\mu = 0.992$,

$H = 0.229 \pm 0.405$, $R_H^2 = 0.382$.

Как и следовало ожидать, индекс μ определен намного точнее, чем показатель H.

Таким образом, главным преимуществом индекса μ по сравнению с другими фрактальными показателями (в частности с показателем Херста H) является то, что соответствующая ему величина $V_f(\delta)$ имеет быстрый выход на степенной асимптотический режим. Это дает возможность использовать его в качестве локальной характеристики, определяющей динамику исходного процесса, поскольку репрезентативный масштаб, необходимый для надежного определения индекса μ можно считать имеющим тот же порядок, что и характерный масштаб основных состояний процесса. К таким состояниям относятся флэты (периоды относительного спокойствия) и тренды (периоды относительно длительного движения вверх или вниз). Чтобы соотнести значение μ с поведением временного ряда, естественно ввести функцию $\mu(t)$ как такое значение μ, которое еще может быть вычислено с приемлемой точностью на минимальном, предшествующем t интервале τ_μ. В случае непрерывного аргумента t в качестве такого интервала можно было бы взять произвольно малый интервал. Однако поскольку на практике временной ряд всегда имеет минимальный масштаб (в нашем случае он равен одному дню), то τ_μ имеет конечную длину ($\tau_\mu = 32$ дня).

Использование функции $\mu(t)$ позволяет существенно продвинуться и в решении двух основных задач анализа временных рядов – задачи идентификации и задачи прогноза.

1.10. Задача идентификации системы

Задача идентификации обычно заключается в корректном определении макросостояния системы на основе наблюдаемой реализации временного ряда [27]. Для решения этой задачи применительно к финансовым рядам была рассчитана функция $\mu(t)$ для каждой из компаний, входящих в индекс Доу-Джонса. На рис. 15 представлен типичный фрагмент ценового ряда одной из таких компаний вместе с вычисленной для этого фрагмента функцией $\mu(t)$. Достаточно беглого взгляда на рис. 15, чтобы понять, что индекс μ имеет отношение к поведению временного ряда. Действительно, на интервале между 1-м и 39-м днем, где цены ведут себя относительно стабильно (флэт), $\mu(t) > 0.5$. Далее, одновременно с развитием тренда на графике цен, $\mu(t)$ резко падает ниже значения $\mu = 0.5$ и, наконец, после 56-го дня, где цены находятся в промежуточном состоянии между трендом и флэтом, $\mu(t)$ возвращается к значению $\mu \approx 0.5$. Таким образом, исходный ряд оказывается тем стабильнее, чем больше значение μ. При этом, если $\mu > 0.5$, то наблюдается флэт, если $\mu < 0.5$, наблюдается тренд. Наконец, если $\mu \approx 0.5$, то процесс находится в промежуточном состоянии между трендом и флэтом. Подобная корреляция между значением μ и характером поведения исходного временного ряда наблюдается для подавляющего большинства его фрагментов. Чтобы представить агрегированные результаты, рассмотрим снова ценовой ряд компании Alcoa Inc. за период 1970-2002 гг. и 8113 интервалов длиной 32 дня, смещенных друг относительно друга на один день. Для каждого из таких интервалов выберем пять различных показателей стабильности:

$F_1 = \log(C_i / C_{i-32})$ – логарифмическое увеличение цены закрытия C_i за 32-дневный интервал;

$F_2 = A_i / A_{i-32}$, где A_i – амплитуда колебаний цен за последние 32 дня;

$F_3 = \sigma[\log(C)]$ – стандартное отклонение цен закрытия за последние 32 дня;

F_4 – значение коэффициента наклона линии линейной регрессии, проведенной по ряду цен;

$F_5 = \left(C_i - C_{i-32} / \sum_{j=1-31}^{i} \left| C_j - C_{j-1} \right| \right)$ – разность между текущей ценой и ценой 32 дня назад, нормированная на сумму модулей ежедневных приращений цен за 32 дня.

Рис. 15 – Ежедневные цены акций компании Exxon Mobil Corporation (правая шкала, японские свечи) и график функции $\mu(t)$ (левая шкала, сплошная линия)

Для всех выбранных показателей стабильности справедливо следующее утверждение: чем более стабильно поведение исходного ряда (колебания происходят возле одного уровня), тем

ближе значение F_m к нулю и наоборот, чем ярче выражен тренд, тем больше по модулю значение F_m.

В [27] показана явная корреляция между значением индекса μ и стабильностью ценового ряда. При этом, чем больше значение μ, тем стабильнее поведение ряда, и чем меньше μ, тем сильнее в исходном ряде выражен тренд.

Корреляцию можно объяснить путем рассмотрения винеровского случайного процесса, который является классической моделью броуновского движения. Напомним, что из постулатов этой модели (см. введение) следует, что:

$$< (X(t) - X(t_0))^2 >= \sigma^2 |t - t_0|, \qquad (16)$$

где угловые скобки означают статистическое усреднение по временному интервалу; $X(t)$ и $X(t_0)$ – значения процесса соответственно в моменты времени t и t_0, а σ^2 – дисперсия за единицу времени (в финансах параметр а известен как волатильность). Из (16) можно получить, что этот процесс преобразуется сам в себя при изменении масштаба времени в b раз и одновременном изменении пространственного масштаба в $b^{1/2}$ раз. Поэтому фрактальная размерность графика реализации такого процесса $D = 1.5$ ($\mu = 0.5$).

Далее, поскольку винеровский процесс не имеет памяти, то этот график в любой своей части занимает промежуточное положение между трендом и флэтом. С другой стороны, поскольку при вычислении индекса μ соответствующий репрезентативный масштаб $\tau_\mu \square 1$, то исходя из формул (12)-(13) процесс тем стабильнее, чем больше значение μ. Поэтому, случай $\mu < 0.5$ естественно интерпретировать как тренд, а случай $\mu > 0.5$ – как

флэт. Таким образом, индекс μ действительно является показателем стабильности исходного временного ряда.

Напомним, что в соответствии с гипотезой эффективного рынка поведение цен должно быть близким к случайному блужданию. Проверка этого предположения обычно сводится к исследованию распределения ценовых приращений на нормальность и к изучению их автокорреляционной функции на предмет наличия зависимости [27]. Выводы, которые можно сделать из подобных исследований, не позволяют оценить степень отклонения реальных финансовых рядов от случайного блуждания. Использование индекса вариации μ позволяет провести более подробный анализ.

Оценим долю общего времени, которую ряд проводит в различных состояниях. Для этого вернемся к рассмотренным выше 32-дневным интервалам и вычислим для каждого из них значения μ, μ_- и μ_+, где μ_- и μ_+ – соответственно нижняя и верхняя границы доверительного интервала, в который истинное значение μ попадает с вероятностью 0.9. Будем считать, что при $\mu_+ < 0.5$ ($\mu_- > 0.5$) ряд находится в состоянии тренда (флэта), а при $\mu_- \le 0.5 \le \mu_+$ – в состоянии броуновского движения. Рассчитаем общее количество отрезков для каждого типа поведения и вычислим долю времени, проводимую исходным временным рядом в каждом из состояний. Результаты расчетов для ценовых рядов некоторых компаний представлены в табл. 3.1 [27].

Таблица 1.1

Состояние временного	Броуновское	Тренд	Флэт
Alcoa Inc	23%	43%	34 %
Boeing Corp	24%	37%	39 %
IBM	25%	39%	36%
Microsoft Corp	26%	36%	38%
Exxon Mobile Corp	15%	50%	35%

Аналогичные результаты получены для всего списка компаний, входящих в индекс Доу-Джонса. Как видно из табл. 1.1. времённые ряды проводят в состоянии близком к случайному блужданию менее 30 % общего времени.

Реальные временные ряды демонстрируют сложное непериодическое поведение, при котором тренды и флэты хаотическим образом сменяют броуновское движение. При этом, зная значение функции $\mu(t)$, можно сказать, какой тип поведения преобладает в каждой точке ряда.

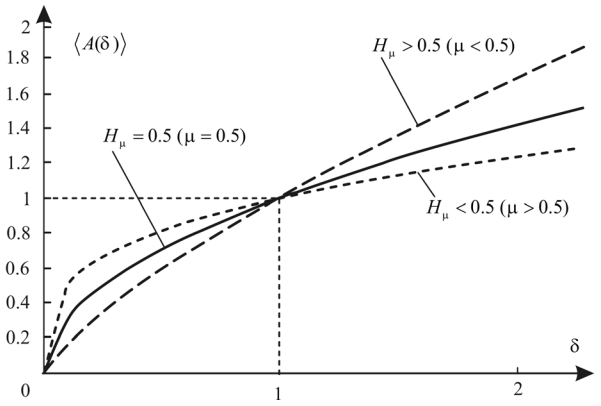


Рис. 16 – Зависимость $< A(\delta) >$ для временного ряда при различных значениях μ (H_μ): сплошная линия соответствует изменению амплитуды для броуновского движения, пунктирная и штрихпунктирная – для трендов и флэтов, соответственно

1.11. Фрактальное прогнозирование поведения системы

Однако наиболее интересным представляется использование построенного фрактального анализа для решения задач прогноза [27]. Оказывается, что наличие степенной зависимости для

функции $V_f(\delta)$ в достаточно широком диапазоне масштабов позволяет предложить новый подход к прогнозированию фазовых переходов хаотических системах. Действительно, обратимся к формулам (13), (14) и рис. 16. Предположим, что временной ряд находится в состоянии случайного блуждания. В этом случае $D = 1.5$ ($\mu = 0.5$) и зависимость $< A(\delta) >$ описывается сплошной линией на рис. 16. Предположим, что в системе произошел фазовый переход, в результате которого временной ряд переходит в состояние тренда. Это означает, что через некоторое время (т.е. для больших δ) амплитуда колебаний увеличится (стрелка (1) на рис. 16). Однако как видно из графиков, переход временного ряда в новое состояние вызовет одновременное уменьшение амплитуды колебаний на малых масштабах (стрелка (2) на рис. 16). Таким образом, увеличение крупномасштабных флуктуации ведет к подавлению мелкомасштабных флуктуации и наоборот. Этот эффект, который служит ключом к прогнозу сильных флуктуации на больших масштабах и следует из наличия степенного закона, был действительно обнаружен и подтвержден обработкой большого количества эмпирических данных.

Таким образом, для одномерной фрактальной функции $f(t)$ на основе величины $V_f(\delta)$ (7) введены новые фрактальные показатели [27]: индекс фрактальности μ и размерность минимального покрытия D_μ (9) тесно связанную с индексом μ. Как предельное значение при $\delta \to 0$, D_μ совпадает с обычной фрактальной размерностью D. Однако по сравнению с другими известными фрактальными показателями алгоритм вычисления размерности D_μ (и соответственно индекса μ) имеет быстрый выход на степенной асимптотический режим для D. Численные расчеты, выполненные для ценовых рядов акций компаний, входящих в индекс Доу-Джонса, показали, что репрезентативный масштаб, необходимый для определения индекса μ с приемлемой

точностью, на два порядка меньше, чем, например, соответствующий масштаб для определения показателя Херста H. Это позволяет рассматривать индекс μ в качестве локального фрактального показателя. Поэтому для каждого момента t временного ряда в [27] введена функция $\mu(t)$ как значение μ, вычисленное на минимальном, предшествующем t интервале τ_μ. Использование этой функции позволило серьезно продвинуться как в плане идентификации, так и в плане прогноза финансовых временных рядов. В плане идентификации в [27] теоретически обосновали и подтвердили численно на основе достаточно большого количества эмпирических данных тот факт, что индекс μ является показателем стабильности временного ряда. Чем больше значение μ, тем стабильнее ряд. При этом случай $\mu < 0.5$ может быть интерпретирован как тренд, а случай $\mu > 0.5$ – как флэт. Случай же $\mu \approx 0.5$ соответствует броуновскому движению. Использование функции $\mu(t)$ позволило протестировать исходные ценовые ряды с тем, чтобы выделить в них броуновскую компоненту. Как оказалось, ее доля составляет менее 30 %. Полученный результат дает оценку степени обоснованности гипотезы эффективного рынка. В плане же прогноза обоснован теоретически и подтвержден расчетами эффект увеличения крупномасштабных флуктуации при подавлении мелкомасштабных. Этот эффект, очевидно, может быть полезным для предсказания сильных колебаний на финансовом рынке [27].

Как оказывается, фрактальный анализ имеет гораздо более широкую область применения. В частности, использование функции $\mu(t)$ позволяет решить задачу о распределении трендов, имеющую большое практическое значение. При этом оказывается, что вероятность продолжения тренда, который длится, скажем, 7 дней значимо выше, чем соответствующая вероятность для тренда, который длится только 5 дней.

Построенный локальный фрактальный анализ можно использовать также для прогноза землетрясений, ишемических заболеваний и т.д. [27].

ГЛАВА 2. ОБЩИЕ СВЕДЕНИЯ О НЕЙРОНАХ

2.1. Введение

Нейронные сети (НС), или, точнее, искусственные НС (ИНС), представляют собой технологию, уходящую корнями во множество дисциплин: нейрофизиологию, математику, статистику, физику, компьютерные науки и технику [52]. Они находят свое применение в таких разнородных областях, как моделирование, анализ временных рядов, распознавание образов, обработка сигналов и управление благодаря одному важному свойству – способности обучаться на основе данных при участии учителя или без его вмешательства.

Под нейронными сетями подразумеваются вычислительные структуры, которые моделируют простые биологические процессы, обычно ассоциируемые с процессами человеческого мозга [53]. Они представляют собой распределенные и параллельные системы, способные к адаптивному обучению путем анализа положительных и отрицательных воздействий. Элементарным преобразователем в данных сетях является искусственный нейрон или просто нейрон, названный так по аналогии с биологическим прототипом. Искусственные НС строятся по принципам организации и функционирования их биологических аналогов. Они способны решать широкий круг задач распознавания образов, идентификации, прогнозирования, оптимизации, управления сложными объектами. Дальнейшее повышение производительности компьютеров все в большой мере связывают с ИНС, в частности, с нейрокомпьютерами (НК), основу

которых составляет искусственная НС. Основные проблемы, решаемые при помощи ИНС, с точки зрения прогнозирования можно свести к двум, связанным между собой задачам [53]: *аппроксимация функций, предсказание/прогноз.*

2.2. Биологический нейрон

Нервная система и мозг человека состоят из нейронов, соединенных между собой нервными волокнами [53]. Нервные волокна способны передавать электрические импульсы между нейронами. Все процессы передачи раздражений от кожи, ушей и глаз к мозгу, процессы мышления и управления действиями - все это реализовано в живом организме как передача электрических импульсов между нейронами.

Нейрон (нервная клетка) является особой биологической клеткой, которая обрабатывает информацию (рис. 17). Он состоит из тела, или сомы, и отростков нервных волокон двух типов - *дендритов*, по которым принимаются импульсы, и единственного *аксона*, по которому нейрон может передавать импульс. Тело нейрона включает *ядро*, которое содержит информацию о наследственных свойствах, и *плазму,* обладающую молекулярными средствами для производства необходимых нейрону материалов.

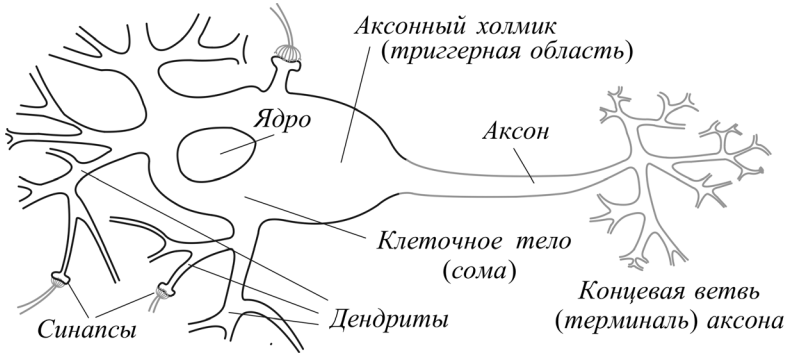


Рис. 17 – Упрощенное изображение биологического нейрона [84]

Нейрон получает сигналы (импульсы) от аксонов других нейронов через дендриты (приемники) и передает сигналы, сгенерированные телом клетки, вдоль своего аксона (передатчика), который в конце разветвляется на волокна. На окончаниях этих волокон находятся специальные образования - *синапсы*, которые влияют на величину импульсов.

Синапс является элементарной структурой и функциональным узлом между двумя нейронами (волокно аксона одного нейрона и дендрит другого). Когда импульс достигает синаптического окончания, высвобождаются химические вещества, называемые нейротрансмиттерами. Нейротрансмиттеры диффундируют через синаптическую щель, возбуждая или затормаживая, в зависимости от типа синапса, способность нейрона-приемника генерировать электрические импульсы. Результативность передачи импульса синапсом может настраиваться проходящими через него сигналами так, что синапсы могут обучаться в зависимости от активности процессов, в которых они участвуют. Эта зависимость от предыстории действует как память, которая, возможно, ответственна за память человека. Важно отметить, что веса синапсов могут изменяться со временем, а значит, меняется и поведение соответствующих нейронов.

Кора головного мозга человека содержит около 10^{11} нейронов и представляет собой протяженную поверхность толщиной от 2 до 3 мм с площадью около $2200\,см^2$. Каждый нейрон связан с $10^3 - 10^4$ другими нейронами. В целом мозг человека содержит приблизительно от 10^{14} до 10^{15} взаимосвязей.

Нейроны взаимодействуют короткими сериями импульсов продолжительностью, как правило, несколько миллисекунд. Сообщение передается посредством частотно-импульсной модуляции. Частота может изменяться от нескольких единиц до сотен герц, что в миллион раз медленнее, чем быстродействующие переключательные электронные схемы Тем не менее сложные

задачи распознавания человек решает за несколько сотен миллисекунд. Эти решения контролируются сетью нейронов, которые имеют скорость выполнения операций всего несколько миллисекунд. Это означает, что вычисления требуют не более 100 последовательных стадий. Другими словами, для таких сложных задач мозг «запускает» параллельные программы, содержащие около 100 шагов. Рассуждая аналогичным образом, можно обнаружить, что количество информации, посылаемое от одного нейрона другому, должно быть очень малым (несколько бит). Отсюда следует, что основная информация не передается непосредственно, а захватывается и распределяется в связях между нейронами.

2.3. Структура и свойства искусственного нейрона

Нейрон представляет собой единицу обработки информации в нейронной сети [52]. На блок-схеме рис. 18 показана *модель* нейрона, лежащего в основе ИНС. В этой модели можно выделить три основных элемента.

1. Набор *синапсов* или *связей*, каждый из которых характеризуется своим *весом* или *силой*. В частности, сигнал x_j на входе синапса j, связанного с нейроном k, умножается на вес w_{kj}. Важно обратить внимание на то, в каком порядке указаны индексы синаптического веса w_{kj}. Первый индекс относится к рассматриваемому нейрону, а второй – ко входному окончанию синапса, с которым связан данный вес. В отличие от синапсов мозга синаптический вес искусственного нейрона может иметь как положительные, так и отрицательные значения.

2. *Сумматор* складывает входные сигналы, взвешенные относительно соответствующих синапсов нейрона. Эту операцию можно описать как *линейную комбинацию.*

3. *Функция активации* ограничивает амплитуду выходного сигнала нейрона. Эта функция также называется функцией *сжатия*. Обычно нормализованный диапазон амплитуд выхода нейрона лежит в интервале [0,1] или [-1,1].

В модель нейрона, показанную на рис. 18, включен *пороговый элемент*, который обозначен символом b_k. Эта величина отражает увеличение или уменьшение входного сигнала, подаваемого на функцию активации.

В математическом представлении функционирование нейрона *к* можно описать следующей парой уравнений:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \,, \quad y_k = \varphi(u_k + b_k) \,, \qquad (17)$$

где $x_1, x_2, ..., x_m$ – входные сигналы, $w_{k1}, w_{k2}, ..., w_{km}$ – синаптические веса k-го нейрона, u_k – линейная комбинация входных воздействий, b_k – порог, $\varphi(\cdot)$ – *функция активации, y_k* – выходной сигнал нейрона. Использование порога b_k обеспечивает эффект *афинного преобразования* выхода линейного сумматора u_k.

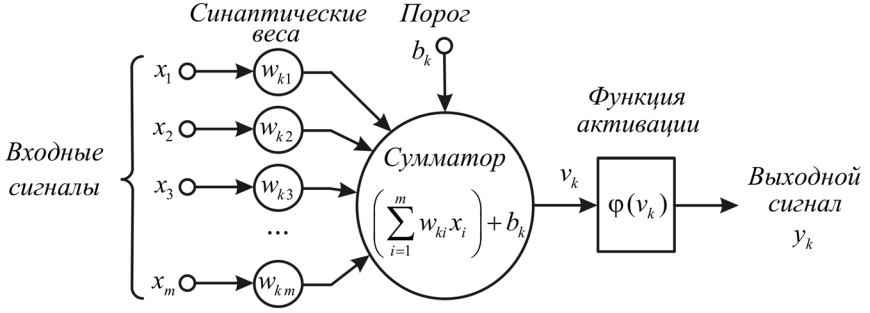
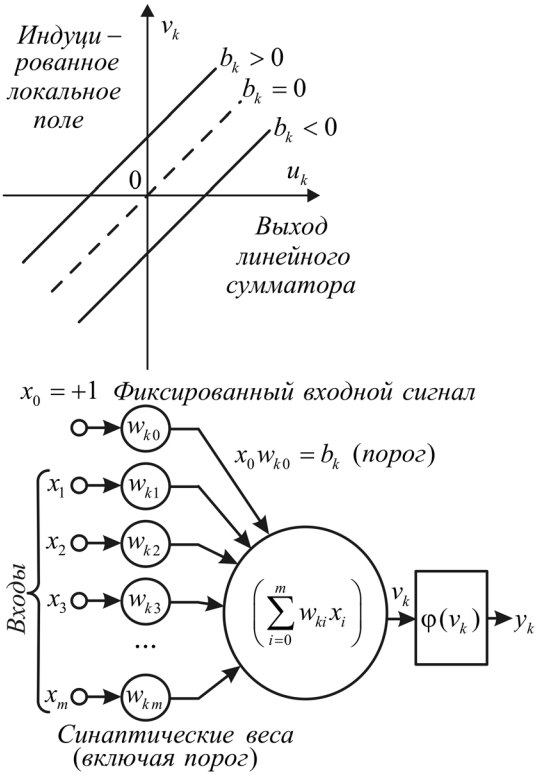


Рис. 18 – Нелинейная модель нейрона [52]



а) *б*)

Рис. 19 – Афинное преобразование, вызванное наличием порога (*а*), и соответствующая ему беспороговая модель нейрона (*б*) [52]

В модели, показанной на рис. 18, постсинаптический потенциал v_k вычисляется следующим образом:

$$v_k = u_k + b_k . \qquad (18)$$

В частности, в зависимости от того, какое значение принимает порог b_k, положительное или отрицательное, *индуцированное локальное поле* или *потенциал активации* v_k нейрона k

изменяется так, как показано на рис. 19,*а*. Теперь, график v_k уже не проходит через начало координат, как график u_k.

Порог b_k является внешним параметром искусственного нейрона k. Его присутствие мы видим в выражении (17). Принимая во внимание выражение (18), формулы (17) можно преобразовать к следующему виду:

$$u_k = \sum_{j=0}^{m} w_{kj} x_j, \quad y_k = \varphi(u_k), \qquad (19)$$

В первом выражении (19) добавился новый синапс. Его входной сигнал имеет значение $x_0 = +1$, а его вес, соответственно $w_{k0} = b_k$.

Это позволило трансформировать модель нейрона к виду, показанному на рис. 19,*б*. На этом рисунке видно, что в результате введения порога добавляется новый входной сигнал фиксированной величины +1, а также появляется новый синаптический вес, равный пороговому значению b_k. Хотя модели, показанные на рис. 18 и 19,*б*, внешне не очень схожи, математически они эквивалентны.

2.4. Типы функций активации

Функции активации, представленные в формулах как $\varphi(v)$, определяют выходной сигнал нейрона в зависимости от индуцированного локального поля v. Можно выделить три основных типа функций активации.

1. *Функция единичного скачка,* или пороговая функция – это известная функция Хевисайда. Внешний вид этой функции показан на рис. 20*а*, а сама она описывается следующим образом:

$$\varphi(v) = \begin{cases} 1, \text{ при } v \geq 0; \\ 0, \text{при } v < 0; \end{cases} \qquad (20)$$

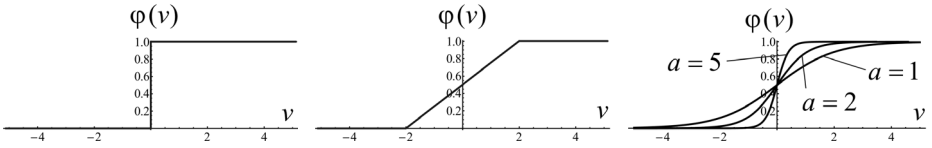
Модель нейрона, использующую данную функцию в качестве функции активации в литературе называют *моделью Мак-Каллока-Питца*, отдавая дань пионерской работе [54]. В этой модели выходной сигнал нейрона принимает значение 1, если индуцированное локальное поле этого нейрона не отрицательно, и 0 в противном случае. Это выражение описывает свойство *"все или ничего"* модели Мак-Каллока-Питца [52].

2. *Кусочно-линейная функция*. Кусочно-линейная функция, показанная на рис. 20,*б*, описывается следующим выражением:

$$\varphi(v) = \begin{cases} 1, & v \leq -0.5; \\ |v|, & +0.5 > v > -0.5; \\ 0, & v \leq -0.5, \end{cases} \qquad (21)$$

где коэффициент усиления в линейной области оператора предполагается равным единице. Эту функцию активации можно рассматривать как *аппроксимацию* нелинейного усилителя.

3. *Сигмоидалъная функция*. Сигмоидальная функция, график которой напоминает букву S, является, пожалуй, самой распространенной функцией, используемой для создания ИНС. Это быстро возрастающая функция, которая поддерживает баланс между линейным и нелинейным поведением. Примером сигмоидальной функции может служить *логистическая функция*, задаваемая следующим выражением:



а) *б*) *в*)

Рис. 20 – Виды функций активации: функция единичного скачка (*а*); кусочно-линейная функция (*б*) и сигмоидальная функция для различных значений параметра a (*в*) [53]

$$\varphi(v) = \frac{1}{1 + e^{-av}}, \qquad (22)$$

где a – *параметр наклона* сигмоидальной функции. Изменяя этот параметр, можно получить различную крутизну функции $\varphi(v)$, как это показано на рис. 20*в*. Первый график соответствует величине параметра, равной $a = 1/4$. В пределе, когда параметр наклона достигает бесконечности, сигмоидальная функция вырождается в пороговую. Если пороговая функция может принимать только значения 0 и 1, то сигмоидальная функция принимает бесконечное множество значений в диапазоне от 0 до 1. При этом следует заметить, что сигмоидальная функция является дифференцируемой, в то время как пороговая таковой не является. Как будет показано дальше, дифференцируемость активационной функции играет важную роль в теории НС.

Область значений функций активации, определенных формулами (20) – (22), представляет собой отрезок от 0 до +1. Однако иногда требуется функция активации, имеющая область значений от -1 до +1. В этом случае функция активации должна быть симметричной относительно начала координат. В частности, пороговую функцию в данном случае можно определить следующим образом:

$$\varphi(v) = \begin{cases} 1, & v > 0; \\ 0, & v = 0; \\ -1, & v < 0. \end{cases}$$

Иногда сигмоидальная функция аппроксимируется *гиперболическим тангенсом:*

$$\varphi(v) = \tanh(v) = \frac{e^{av} - e^{-av}}{e^{av} + e^{-av}}.$$

Следует отметить, что сигмоидальные функции дифференцируемы на всей оси абсцисс, что используется в некоторых алгоритмах обучения. Кроме того, они обладают свойством усиливать слабые сигналы лучше, чем большие, и предотвращает насыщение от больших сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон.

ГЛАВА 3. ОБЩИЕ СВЕДЕНИЯ О НЕЙРОННЫХ СЕТЯХ

3.1. Классификация НС и их свойства

Нейронная сеть представляет собой совокупность нейроподобных элементов, определенным образом соединенных друг с другом и с внешней средой с помощью связей, определяемых весовыми коэффициентами [53]. В зависимости от функций, выполняемых нейронами в сети, можно выделить три их типа:

1. *Входные нейроны,* на которые подается вектор, кодирующий входное воздействие или образ внешней среды; в них обычно не осуществляется вычислительных процедур, а информация передается с входа на выход путем изменения их активации.
2. *Выходные нейроны,* выходные значения которых представляют выходы нейронной сети; преобразования в них осуществляются по выражениям (17) и (18).
3. *Промежуточные нейроны,* составляющие основу нейронных сетей, преобразования в которых выполняются также по выражениям (17) и (18).

В большинстве нейронных моделей тип нейрона связан с его расположением в сети. Если нейрон имеет только выходные связи, то это входной нейрон, если наоборот – выходной нейрон. Однако возможен случай, когда выход топологически внутреннего нейрона рассматривается как часть выхода сети. В процессе функционирования сети осуществляется преобразование входного вектора в выходной, некоторая переработка информации. Конкретный вид выполняемого сетью преобразования данных обусловливается не только характеристиками нейроподобных элементов, но и особенностями ее архитектуры, а именно топологией межнейронных связей, выбором определенных подмножеств нейроподобных элементов для ввода и вывода

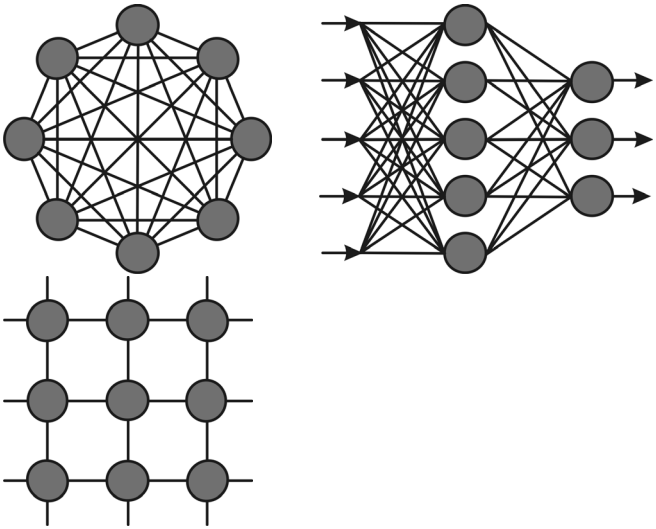
информации, способами обучения сети, наличием или отсутствием конкуренции между нейронами, направлением и способами управления и синхронизации передачи информации между нейронами.

С точки зрения топологии можно выделить три основных типа НС:

1. *Полносвязные*, топология которых показана на рис. 21,*а*).
2. *Многослойные* или *слоистые* (рис. 21,*б*).
3. *Слабосвязные* (с локальными связями) (рис. 21,*в*).

В *полносвязных нейронных сетях* каждый нейрон передает свой выходной сигнал остальным нейронам, в том числе и самому себе. Все входные сигналы подаются всем нейронам. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети.

В *многослойных* НС нейроны объединяются в слои. Слой содержит совокупность нейронов с едиными входными сигналами. Число нейронов в слое может быть любым и не зависит от количества нейронов в других слоях.



а) *б*) *в*)

Рис. 21 – Архитектуры нейронных сетей: *а* - полносвязная сеть, *б* - многослойная сеть с последовательными связями, *в* - слабосвязная сеть [53]

В общем случае сеть состоит из Q слоев, пронумерованных слева направо. Внешние входные сигналы подаются на входы нейронов входного слоя (его часто нумеруют как нулевой), а выходами сети являются выходные сигналы последнего слоя. Кроме входного и выходного слоев в многослойной нейронной сети есть один или несколько скрытых слоев. Связи от выходов нейронов некоторого слоя q к входам нейронов следующего слоя $(q+1)$ называются последовательными.

В свою очередь, среди многослойных НС выделяют следующие типы:

1) Монотонные НС. Это частный случай слоистых сетей с дополнительными условиями на связи и нейроны. Каждый слой кроме последнего (выходного) разбит на два блока: возбуждающий и тормозящий. Связи между

блоками тоже разделяются на тормозящие и возбуждающие. Для нейронов монотонных сетей необходима монотонная зависимость выходного сигнала нейрона от параметров входных сигналов.

2) *Сети без обратных связей.* В таких сетях нейроны входного слоя получают входные сигналы, преобразуют их и передают нейронам первого скрытого слоя, и так далее вплоть до выходного, который выдает сигналы для интерпретатора и пользователя. Если не оговорено противное, то каждый выходной сигнал q-го слоя подастся на вход всех нейронов $(q+1)$-го слоя, однако возможен вариант соединения q-го слоя с произвольным $(q+1)$-м слоем. Среди многослойных сетей без обратных связей различают полносвязанные (выход каждого нейрона q-го слоя связан с входом каждого нейрона $(q+1)$-го слоя) и частично полносвязанные.

3) *Сети* с *обратными связями.* В сетях с обратными связями информация с последующих слоев передается на предыдущие. Среди них, в свою очередь, выделяют следующие: *слоисто-циклические, слоисто-полносвязанные, полносвязанно-слоистые*.

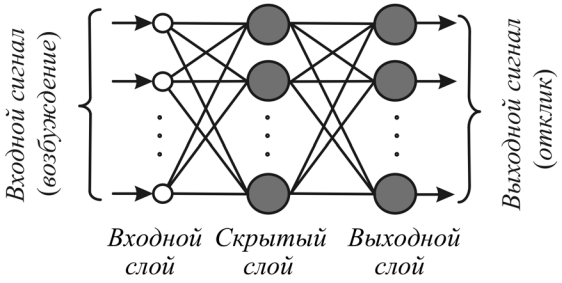
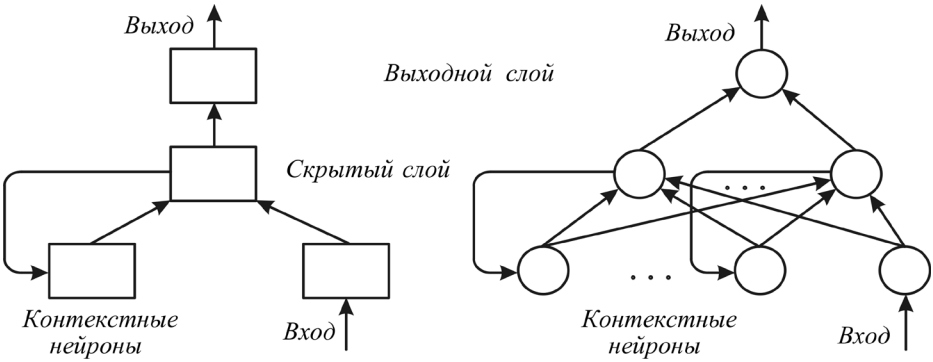
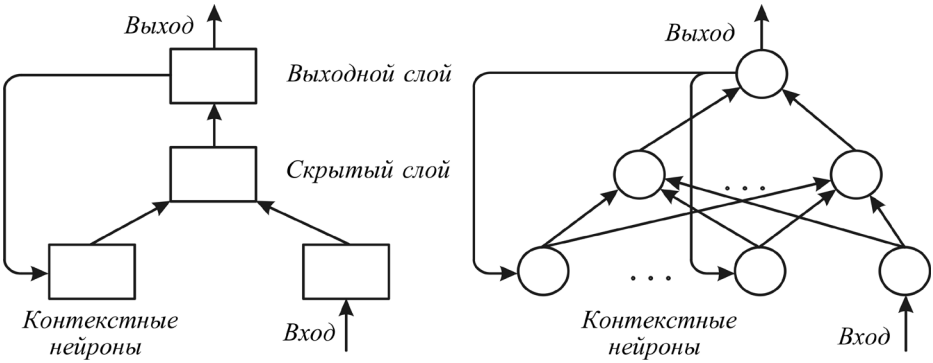


Рис. 22 – Многослойная (двухслойная) сеть прямого распространения [53]



а)



б)

Рис. 23 – Частично-рекуррентные сети: *а* - Элмана, *б* – Жордана [53]

В качестве примера сетей с обратными связями на рис. 23 представлены частично-рекуррентные сети Элмана и Жордана.

В *слабосвязных нейронных сетях* нейроны располагаются в узлах прямоугольной или гексагональной решетки Каждый нейрон связан с четырьмя (окрестность фон Неймана), шестью (окрестность Голея) или восемью (окрестность Мура) своими ближайшими соседями.

Известные НС можно разделить по типам структур нейронов на гомогенные (однородные) и гетерогенные. Гомогенные сети состоят из нейронов одного типа с единой функцией активации, а в гетерогенную сеть входят нейроны с различными функциями активации.

Существуют бинарные и аналоговые сети. Первые из них оперируют только двоичными сигналами, и выход каждого нейрона может принимать значение либо логического ноля (заторможенное состояние) либо логической единицы (возбужденное состояние).

Еще одна классификация делит НС на синхронные и асинхронные. В первом случае в каждый момент времени лишь один нейрон меняет свое состояние, во втором – состояние меняется сразу у целой группы нейронов, как правило, у всего слоя. Алгоритмически ход времени в НС задается итерационным выполнением однотипных действий над нейронами. Далее будут рассматриваться только синхронные сети.

Сети можно классифицировать также по числу слоев. Теоретически число слоев и число нейронов в каждом слое может быть произвольным, однако фактически оно ограничено ресурсами компьютера или специализированных микросхем, на которых обычно реализуется нейронная сеть. Чем сложнее сеть, тем более сложные задачи она может решать.

Выбор структуры НС осуществляется в соответствии с особенностями и сложностью задачи. Для решения отдельных типов задач уже существуют оптимальные конфигурации. Если же задача не может быть сведена ни к одному из известных типов, приходится решать сложную проблему синтеза новой конфигурации. При этом необходимо руководствоваться следующими основными правилами:

- возможности сети возрастают с увеличением числа нейронов сети, плотности связей между ними и числом слоев;
- введение обратных связей наряду с увеличением возможностей сети поднимает вопрос о динамической устойчивости сети;
- сложность алгоритмов функционирования сети, введение нескольких типов синапсов способствует усилению мощности НС.

В задачах прогнозирования в качестве входных сигналов используются временные ряды, представляющие значения контролируемых переменных на некотором интервале времени. Выходной сигнал – множество переменных, которое является подмножеством переменных входного сигнала.

В результате отображение $X \to Y$ должно обеспечить формирование правильных выходных сигналов в соответствии:

- со всеми примерами обучающей выборки;
- со всеми возможными входными сигналами, которые не вошли в обучающую выборку.

3.2. Динамические НС

В задачах прогнозирования особое внимание следует уделить динамическим или рекуррентным НС [55]. Они построены из динамических нейронов, чье поведение описывается дифференциальными или разностными уравнениями, как правило, первого порядка. Сеть организована так, что каждый нейрон получает входную информацию от других нейронов (возможно, и от себя самого) и из окружающей среды. Этот тип сетей имеет важное значение, так как с их помощью можно моделировать

нелинейные динамические системы. При работе с временными рядами наиболее часто используются НС с временной задержкой и сети Хопфилда.

3.3. Нейронные сети с временной задержкой

Перед тем, как описать собственно динамические сети, рассмотрим, как сеть с прямой связью используется для обработки временных рядов [55]. Метод состоит в том, чтобы разбить временной ряд на несколько отрезков, и, получить, таким образом, статический образец для подачи на вход многослойной сети с прямой связью. Это осуществляется с помощью, так называемой, разветвленной линии задержки, структурная схема которой показана на рис. 24.

Архитектура такой нейронной сети с временной задержкой позволяет моделировать любую конечную временную зависимость вида:

$$y(t) = F[x(t), x(t-1), ..., x(t-k)],$$

Поскольку рекуррентные связи отсутствуют, такая сеть может быть обучена при помощи стандартного алгоритма обратного распространения ошибки или какого-то из его многочисленных вариантов. Сети такой конструкции успешно применялись в задачах распознавания речи, предсказания нелинейных временных рядов и нахождения закономерностей в хаосе [55].

3.4. Сети Хопфилда

С помощью рекуррентных сетей Хопфилда можно обрабатывать неупорядоченные (рукописные буквы), упорядоченные но времени (временные ряды) или пространстве (графики, грамматики) образцы [55]. Рекуррентная НС простейшего вида была введена

Хопфилдом; она построена из N нейронов, связанных каждый с каждым, причем все нейроны являются выходными (рис. 25).

Сети такой конструкции используются, главным образом, в качестве ассоциативной памяти, а также *в* задачах нелинейной фильтрации данных и грамматического вывода. Кроме этого, относительно недавно они были применены дли предсказания [56, 57] и для распознавания закономерностей в поведении цен акций [58].

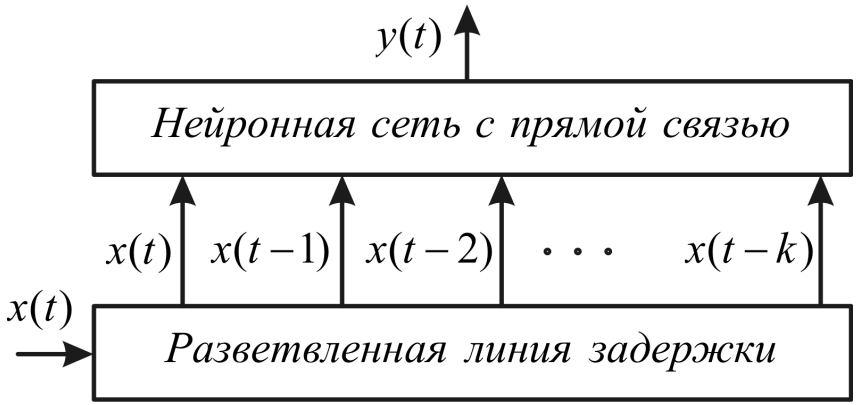


Рис. 24 – НС с временной задержкой [55]

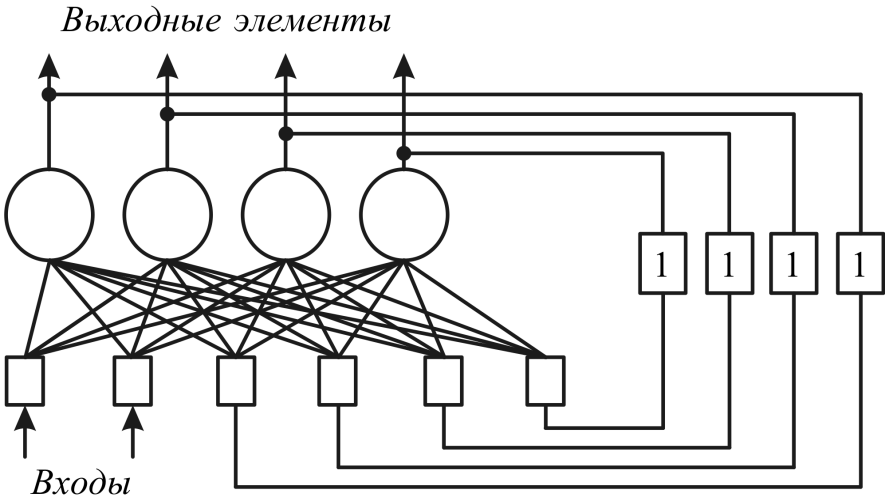


Рис. 25 – Сеть Хопфилда [55]

ГЛАВА 4. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРОЕКТИРОВАНИЯ НЕЙРОННЫХ СЕТЕЙ

4.1. Теорема Колмогорова-Арнольда

Построить многомерное отображение $X \to Y$ – это значит представить его с помощью математических операций над не более, чем двумя переменными [53].

Проблема представления функций многих переменных в виде суперпозиции функций меньшего числа переменных восходит к 13-й проблеме Гильберта. В результате многолетней научной полемики между А.Н. Колмогоровым и В.И. Арнольдом был получен ряд важных теоретических результатов, опровергающих тезис непредставимости функции многих переменных функциями меньшего числа переменных.

4.2. Теорема Хехт Нильсена

Теорема о представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения в 1987 году была переложена Хехт-Нильсеном для нейронных сетей [53].

Теорема Хехт-Нильсена доказывает представимость функции многих переменных достаточно общего вида с помощью двухслойной нейронной сети с прямыми полными связями с n нейронами входного слоя, $(n+1)$ нейронами скрытого слоя с заранее известными ограниченными функциями активации (например, сигмоидальными) и m нейронами выходного слоя с неизвестными функциями активации.

Теорема, таким образом, в неконструктивной форме теорема доказывает решаемость задачи представления функции

произвольного вида на НС и указывает для каждой задачи минимальные числа нейронов сети, необходимых для ее решения.

4.3. Следствия из теоремы Колмогорова-Арнольда-Хехт-Нильсена

Для нейросетевого проектирования важными являются следующие следствия и утверждение, из теоремы Колмогорова-Арнольда и работы Хехт-Нильсена [53]:

Следствие 1. Из теоремы Хехт-Нильсена следует представимость любой многомерной функции нескольких переменных с помощью НС фиксированной размерности. Неизвестными остаются следующие характеристики функций активации нейронов:

- ограничения области значений (координаты асимптот) сигмоидальных функций активации нейронов скрытого слоя;
- наклон сигмоидальных функций активации;
- вид функций активации нейронов выходного слоя.

Про функции активации нейронов выходного слоя из теоремы Хехт-Нильсена известно только то, что они представляют собой нелинейные функции общего вида. В одной из работ, продолжающих развитие теории, связанной с рассматриваемой теоремой, доказывается, что функции активации нейронов выходного слоя должны быть монотонно возрастающими. Это утверждение в некоторой степени сужает класс функций, которые могут использоваться при реализации отображения с помощью двухслойной нейронной сети.

На практике, требования теоремы Хехт-Нильсена к функциям активации удовлетворяются следующим образом. В нейронных

сетях, как для первого (скрытого), так и для второго (выходного) слоя, используют сигмоидальные передаточные функции с настраиваемыми параметрами. То есть, в процессе обучения, индивидуально для каждого нейрона, задается максимальное и минимальное значение, а также наклон сигмоидальной функции.

Следствие 2. Для любого множества пар (X^k, Y^k) (где Y^k -скаляр) существует двухслойная однородная (с одинаковыми функциями активации) нейронная сеть первого порядка с последовательными связями и с конечным числом нейронов, которая выполняет отображение $X \to Y$, выдавая на каждый входной сигнал X^k правильный выходной сигнал Y^k. Нейроны в такой двухслойной нейронной сети должны иметь сигмоидальные передаточные функции.

К сожалению, эта теорема не конструктивна. В ней не заложена методика определения числа нейронов в сети для некоторой конкретной обучающей выборки. Для многих задач, единичной размерности выходного сигнала недостаточно. Необходимо иметь возможность строить с помощью НС функции $X \to Y$, где Y^k имеет произвольную размерность. Следующее утверждение является теоретической основой для построения таких функций на базе однородных нейронных сетей.

Утверждение. Для любого множества пар входных-выходных векторов произвольной размерности $\{(X^k, Y^k), k = 1...N\}$ существует однородная двухслойная нейронная сеть с последовательными связями, с сигмоидальными передаточными функциями и с конечным числом нейронов, которая для каждого входного вектора X^k. Сформирует соответствующий ему выходной вектор Y^k.

Таким образом, для представления многомерных функций многих переменных может быть использована однородная двухслойная НС с сигмоидальными передаточными функциями.

Для оценки числа нейронов с скрытых слоях однородных нейронных сетей можно воспользоваться формулой для оценки необходимого числа синаптических весов L_w в многослойной сети с сигмоидальными передаточными функциями:

$$\frac{mN}{1+\log_2 N} \le L_w \le m\left(\frac{N}{m}+1\right)(n+m+1)+m,$$

где n - размерность входного сигнала, m - размерность выходного сигнала, N - число элементов обучающей выборки.

Оценив необходимое число весов, можно рассчитать число нейронов в скрытых слоях. Например, для двухслойной сети это число составит:

$$L = \frac{L_w}{n+m}.$$

Известны и другие формулы для оценки, например:

$$2(n+L+m) \le N \le 10(n+L+m)$$

$$\frac{N}{10} - n - m \le L \le \frac{N}{2} - n - m$$

Точно так же можно рассчитать число нейронов в сетях с большим числом слоев. Иногда, целесообразно использовать сети с большим числом слоев. Такие многослойные нейронные сети могут иметь меньшие размерности матриц синаптических весов нейронов одного слоя, чем двухслойные сети, реализующие то же самое отображение. Однако строгой методики построения таких сетей пока нет. Аналогичная ситуация складывается и с многослойными нейронными сетями, в которых помимо последовательных связей используются и прямые (связи от слоя с номером q к слою с номером $(q+p)$, где $p>1$). Нет строгой теории, которая показывала бы возможность и целесообразность построения таких сетей.

Наибольшие проблемы возникают при использовании сетей циклического функционирования. К этой группе относятся многослойные сети с обратными связями (от слоя с номером q к слою с номером $(q+p)$, где $p<0$, а также полносвязные сети. Для успешного функционирования таких сетей необходимо соблюдение условий динамической устойчивости, иначе сеть может не сойтись к правильному решению, либо, достигнув на некоторой итерации правильного значения выходного сигнала, после нескольких итераций уйти от этого значения. Проблема динамической устойчивости подробно исследована, пожалуй, лишь для одной модели из рассматриваемой группы - нейронной сети Хопфилда.

Отсутствие строгой теории для перечисленных моделей нейронных сетей не препятствует исследованию возможностей их применения.

В отечественной литературе, приведенные результаты известны в более фрагментарной форме - в виде так называемой *теоремы о полноте.*

Теорема о полноте. Любая непрерывная функция на замкнутом ограниченном множестве может быть равномерно приближена функциями, вычисляемыми НС, если функция активации нейрона дважды непрерывно дифференцируема и непрерывна.

Таким образом, НС являются универсальными структурами, позволяющими реализовать любой вычислительный алгоритм.

ГЛАВА 5. ПОСТАНОВКА И ВОЗМОЖНЫЕ ПУТИ РЕШЕНИЯ ЗАДАЧИ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

5.1. Теоретические основы обучения

Очевидно, что процесс функционирования НС, сущность действий, которые она способна выполнять, зависит от величин синаптических связей [83]. Поэтому, задавшись определенной структурой сети, соответствующей какой-либо задаче, необходимо найти оптимальные значения всех переменных весовых коэффициентов (некоторые синаптические связи могут быть постоянными).

Этот этап называется обучением НС, и от того, насколько качественно он будет выполнен, зависит способность сети решать поставленные перед ней проблемы во время функционирования.

В процессе функционирования нейронная сеть формирует выходной сигнал Y в соответствии с входным сигналом X, реализуя некоторую функцию $g: Y = g(X)$. Если архитектура сети задана, то вид функции g определяется значениями синаптических весов и смещений сети. Обозначим через G множество всех возможных функций g, соответствующих заданной архитектуре сети.

Пусть решение некоторой задачи есть функция $r: Y = r(X)$, заданная парами входных-выходных данных $(X^1, Y^1),...,(X^k, Y^k)$, для которых $Y^k = r(X^k)$, $k = 1...N$, E -- функция ошибки (функционал качества), показывающая для каждой из функций g степень близости к r.

Решить поставленную задачу с помощью НС заданной архитектуры - это значит построить (синтезировать) функцию $g \in G$, подобрав параметры нейронов (синаптические веса и

смещения) таким образом, чтобы функционал качества обращался в оптимум для всех пар (X^k, Y^k).

Таким образом, задача обучения нейронной сети определяется совокупностью пяти компонентов:

$\langle X, Y, r, G, E \rangle$.

Задача обучения состоит в поиске (синтезе) функции g, оптимальной по E. Она требует длительных вычислений и представляет собой итерационную процедуру. Число итераций может составлять от 10^3 до 10^8. На каждой итерации происходит уменьшение функции ошибки.

Если выбраны множество обучающих примеров и способ вычисления функции ошибки, обучение НС превращается в задачу многомерной оптимизации, для решения которой могут быть использованы следующие методы:

- локальной оптимизации с вычислением частных производных первого порядка;
- локальной оптимизации с вычислением частных производных первого и второго порядка;
- стохастической оптимизации;
- глобальной оптимизации.

К первой группе относятся: градиентный метод (наискорейшего спуска); методы с одномерной и двумерной оптимизацией целевой функции в направлении антиградиента; метод сопряженных градиентов; методы, учитывающие направление антиградиента на нескольких шагах алгоритма.

Ко второй группе относятся: метод Ньютона, методы оптимизации с разреженными матрицами Гессе, квазиньютоновские методы, метод Гаусса-Ньютона, метод Левенберга-Маркардта.

Стохастическими методами являются: поиск в случайном направлении, имитация отжига, метод Монте-Карло (численный метод статистических испытаний). Задачи глобальной оптимизации решаются с помощью перебора значений переменных, от которых зависит целевая функция.

Для сравнения методов обучения НС необходимо использовать два критерия: количество шагов алгоритма для получения решения и количество дополнительных переменных для организации вычислительного процесса.

Для иллюстрации термина «дополнительные переменные» приведем следующий пример. Пусть p_1 и p_2 - некоторые параметры НС с заданными значениями. В процессе обучения сети на каждом шаге, по меньшей мере, два раза потребуется выполнить умножение $p_1 p_2$. Дополнительная переменная требуется для сохранения значения произведения после первого умножения.

Предпочтение следует отдавать тем методам, которые позволяют обучить нейронную сеть за небольшое число шагов и требуют малого числа дополнительных переменных. Это связано с ограничением ресурсов вычислительных средств. Как правило, для обучения используются персональные компьютеры.

Пусть НС содержит P изменяемых параметров (синаптических весов и смещений). Существует лишь две группы алгоритмов обучения, которые требуют менее $2P$ дополнительных параметров, и при этом дают возможность обучать нейронные сети за приемлемое число шагов. Это алгоритмы с вычислением частных производных первого порядка и, возможно, одномерной оптимизации.

Хотя указанные алгоритмы дают возможность находить только локальные экстремумы, они могут быть использованы на практике для обучения НС с многоэкстремальными целевыми функциями

(функциями ошибки), так как экстремумов у целевой функции, как правило, не очень много. Достаточно лишь раз или два вывести сеть из локального минимума с большим значением целевой функции для того, чтобы в результате итераций в соответствии с алгоритмом локальной оптимизации сеть оказалась в локальном минимуме со значением целевой функции, близким к нулю. Если после нескольких попыток вывести сеть из локального минимума нужного эффекта добиться не удается, необходимо увеличить число нейронов во всех слоях с первого по предпоследний и присвоить случайным образом их синаптическим весам и смещениям значения из заданного диапазона.

Эксперименты по обучению нейронных сетей показали, что совместное использование алгоритма локальной оптимизации процедуры вывода сети из локального минимума и процедуры увеличения числа нейронов приводит к успешному обучению нейронных сетей.

Кратко опишем недостатки других методов. Стохастические алгоритмы требуют очень большого числа шагов обучения. Это делает невозможным их практическое использование для обучения НС больших размерностей. Экспоненциальный рост сложности перебора с ростом размерности задачи в алгоритмах глобальной оптимизации при отсутствии априорной информации о характере целевой функции также делает невозможным их использование для обучения НС больших размерностей. Метод сопряженных градиентов очень чувствителен к точности вычислений, особенно при решении задач оптимизации большой размерности. Для методов, учитывающих направление антиградиента на нескольких шагах алгоритма, и методов, включающих вычисление матрицы Гессе, необходимо более чем $2P$ дополнительных переменных. В зависимости от способа разрежения, вычисление матрицы Гессе требует от $2P$ до P^2 дополнительных переменных.

Рассмотрим один из самых распространенных алгоритмов обучения – алгоритм обратного распространения ошибки.

5.2. Обучение с учителем

В многослойных НС оптимальные выходные значения нейронов всех слоев, кроме последнего, как правило, неизвестны [53]. Трех- или более слойный персептрон уже невозможно обучить, руководствуясь только величинами ошибок на выходах сети.

Один из вариантов решения этой проблемы – разработка наборов выходных сигналов, соответствующих входным, для каждого слоя НС, что, конечно, является очень трудоемкой операцией и не всегда осуществимо. Второй вариант - динамическая подстройка весовых коэффициентов синапсов, в ходе которой выбираются, как правило, наиболее слабые связи и изменяются на малую величину в ту или иную сторону, а сохраняются только те изменения, которые повлекли уменьшение ошибки на выходе всей сети. Очевидно, что данный метод, несмотря на кажущуюся простоту, требует громоздких рутинных вычислений. И, наконец, третий, более приемлемый вариант - распространение сигналов ошибки от выходов нейронной сети к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Этот алгоритм обучения получил название процедуры *обратного распространения ошибки.* Именно он рассматривается ниже.

5.3. Алгоритм обратного распространения ошибки

На рис. 26 показан архитектурный граф многослойного персептрона с двумя скрытыми слоями и одним выходным слоем [52]. Показанная на рисунке сеть является *полносвязной*, что характерно для многослойного персептрона общего вида. Это значит, что каждый нейрон в любом слое сети связан со всеми

нейронами (узлами) предыдущего слоя. Сигнал передается по сети исключительно в прямом направлении, слева направо, от слоя к слою.

На рис. 27 показан фрагмент многослойного персептрона. Для этого типа сети выделяют два типа сигналов [59].

1. *Функциональный сигнал*. Это входной сигнал (стимул), поступающий в сеть и передаваемый вперед от нейрона к нейрону по всей сети. Такой сигнал достигает конца сети в виде выходного сигнала. Данный сигнал называется *функциональным* по двум причинам [52]. Во-первых, он предназначен для выполнения некоторой функции на выходе сети. Во-вторых, в каждом нейроне, через который передается этот сигнал, вычисляется некоторая *функция* с учетом весовых коэффициентов.

2. *Сигнал ошибки*. Сигнал ошибки берет свое начало на выходе сети и распространяется в обратном направлении (от слоя к слою). Он получил свое название благодаря тому, что вычисляется каждым нейроном сети на основе функции ошибки, представленной в той или иной форме.

Выходные нейроны (вычислительные узлы) составляют выходной слой сети. Остальные нейроны (вычислительные узлы) относятся к скрытым слоям. Таким образом, скрытые узлы не являются частью входа или выхода сети – отсюда они и получили свое название. Первый скрытый слой получает данные из входного слоя, составленного из сенсорных элементов (входных узлов). Результирующий сигнал первого скрытого слоя, в свою очередь, поступает на следующий скрытый слой, и т.д., до самого конца сети.

Любой скрытый или выходной нейрон многослойного персептрона может выполнять два типа вычислений:

1. Вычисление функционального сигнала на выходе нейрона, реализуемое в виде непрерывной нелинейной функции от входного сигнала и синаптических весов, связанных с данным нейроном.
2. Вычисление оценки вектора градиента (т.е. градиента поверхности ошибки по синаптическим весам, связанным со входами данного нейрона), необходимого для обратного прохода через сеть.

Для того чтобы упростить понимание математических выкладок, сначала приведем полный список используемых обозначений:

n – итерация (такт времени), соответствующий n-му обучающему образу (примеру), поданному на вход сети.

$E(n)$ – текущая сумма квадратов ошибок (или энергия ошибки) на итерации n. Среднее значение $E(n)$ по всем значениям n (т.е. по всему обучающему множеству) называется средней энергией ошибки E_{av}.

$e_j(n)$ – сигнал ошибки на выходе нейрона j на итерации n.

$d_j(n)$ – желаемый отклик нейрона j и используется для вычисления $e_j(n)$.

$y_j(n)$ – функциональный сигнал, генерируемый на выходе нейрона j на итерации n.

$w_{ji}(n)$ – синаптический вес, связывающий выход нейрона i со входом нейрона j на итерации n.

$\Delta w_{ji}(n)$ – коррекция, применяемая к весу $w_{ji}(n)$ на шаге n.

$v_j(n)$ – индуцированное локальное поле (т.е. взвешенная сумма всех синаптических входов плюс порог) нейрона j на итерации n. Это значение передается в функцию активации, связанную с нейроном j.

$\varphi_j(\cdot)$ – функция активации, соответствующая нейрону j и описывающая нелинейную взаимосвязь входного и выходного сигналов этого нейрона.

b_j – порог нейрона j. (Если используется беспороговая модель нейрона, то вместо него используется нулевой синаптический вес $w_{j0} = b_j$, соединенный с сигналом с фиксированным значением +1).

$x_i(n)$ – i-й элемент входного вектора.

$o_k(n)$ – k-й элемент выходного вектора (образа).

η – параметр скорости (интенсивности) обучения.

m_l – размерность (количество узлов) слоя l многослойного персептрона ($l = 1,2,...,L$);

L – "глубина" сети.

m_0 – размерность входного слоя;

m_1 – размерность первого скрытого слоя;

$m_L = M$ – размерность выходного слоя.

Индексы i, j и k относятся к различным нейронам сети. Когда сигнал проходит по сети слева направо, считается, что нейрон j находится на один слой правее нейрона i, а нейрон k – еще на один слой правее нейрона j, если последний принадлежит скрытому слою.

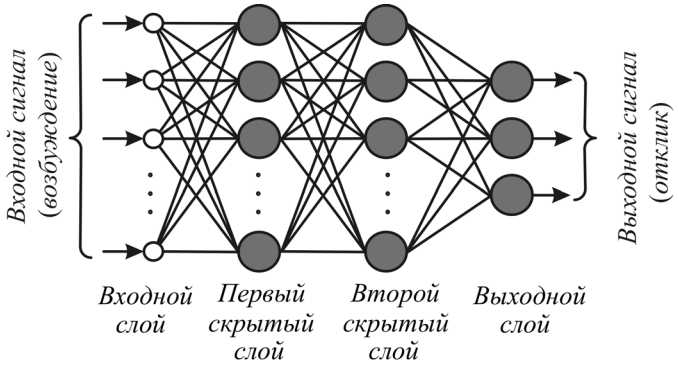


Рис. 26 – Архитектурный граф многослойного персептрона с двумя скрытыми слоями

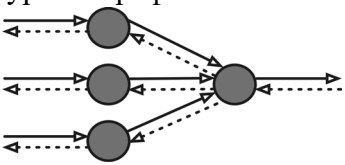


Рис. 27 – Направление двух основных потоков сигнала для многослойного персептрона: прямое распространение функционального сигнала (сплошные линии) и обратное распространение сигнала ошибки (пунктирные линии) [52]

Сигнал ошибки выходного нейрона j на итерации n (соответствующей n-му примеру обучения) определяется соотношением:

$$e_j(n) = d_j(n) - y_j(n). \qquad (23)$$

Текущее значение энергии ошибки нейрона j определим как $\frac{1}{2}e_j^2(n)$. Соответственно, текущее значение $\mathbf{E}(n)$ общей энергии ошибки сети вычисляется путем сложения величин $\frac{1}{2}e_j^2(n)$ по *всем нейронам выходного слоя.* Это "видимые" нейроны, для которых сигнал ошибки может быть вычислен непосредственно. Таким образом, можно записать

$$\mathbf{E}(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n), \qquad (24)$$

где множество C включает все нейроны выходного слоя сети. Пусть N – общее число образов в обучающем множестве (т.е. мощность этого множества). *Энергия среднеквадратической ошибки* в таком случае вычисляется как нормализованная по N сумма всех значений энергии ошибки $\mathbf{E}(n)$:

$$\mathbf{E}_{av}(n) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{E}(n). \qquad (25)$$

Текущая энергия ошибки $\mathbf{E}(n)$, а значит и средняя энергия ошибки $\mathbf{E}_{av}$, являются функциями всех свободных параметров (т.е. синаптических весов и значений порога) сети. Для данного обучающего множества энергия $\mathbf{E}_{av}$ представляет собой *функцию стоимости* – меру эффективности обучения. Целью процесса обучения является настройка свободных параметров сети с целью минимизации величины $\mathbf{E}_{av}$. В процессе минимизации будем использовать аппроксимацию. В частности, рассмотрим простой метод обучения, в котором веса обновляются *для каждого обучающего примера* в пределах одной *эпохи*, т.е. всего обучающего множества. Настройка весов выполняется в соответствии с ошибками, вычисленными для *каждого* образа, представленного в сети.

Арифметическое среднее отдельных изменений для весов сети по обучающему множеству может служить *оценкой* реальных изменений, произошедших в процессе минимизации функции стоимости $\mathbf{E}_{av}$ на множестве обучения.

На рис. 27 изображен нейрон j, на который поступает поток сигналов от нейронов, расположенных в предыдущем слое. Индуцированное локальное поле $v_j(n)$, полученное на входе функции активации, связанной с данным нейроном, равно

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_j(n), \qquad (26)$$

где m – общее число входов (за исключением порога) нейрона j. Синаптический вес w_{j0} (соответствующий фиксированному входу $y_0 = +1$) равен порогу b_j, применяемому к нейрону j. Функциональный сигнал $y_j(n)$ на выходе нейрона j на итерации n равен

$$y_j(n) = \varphi_j(v_j(n)). \qquad (27)$$

Алгоритм обратного распространения состоит в применении к синаптическому весу $w_{ji}(n)$ коррекции $\Delta w_{ji}(n)$, пропорциональной частной производной $\partial \mathbf{E}(n) / \partial w_{ji}(n)$. В соответствии с *правилом цепочки* градиент можно представить следующим образом:

$$\frac{\partial \mathbf{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathbf{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_j(n)}. \qquad (28)$$

Частная производная $\partial \mathbf{E}(n) / \partial w_{ji}(n)$ представляет собой *фактор чувствительности*, определяющий направление поиска в пространстве весов для синаптического веса $w_{ji}(n)$.

Определяя, производные в правой части (29) из уравнений (23)-(28), окончательно получим:

$$\frac{\partial \mathbf{E}(n)}{\partial w_{ji}(n)} = -e_j(n)\varphi_j'(v_j(n))y_j(n) \quad . \qquad (29)$$

Коррекция $\Delta w_{ji}(n)$, применяемая к $w_{ji}(n)$, определяется согласно *дельта-правилу*:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathbf{E}(n)}{\partial w_{ji}(n)} , \qquad (30)$$

где η *– параметр скорости обучения* алгоритма обратного распространения. Использование знака "минус" в (30) связано с реализацией *градиентного спуска* в пространстве весов (т.е. поиском направления изменения весов, уменьшающего значение энергии ошибки $\mathbf{E}(n)$. Следовательно, подставляя (29) в (30), получим:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_j(n) , \qquad (31)$$

где *локальный градиент* $\delta_j(n)$ определяется выражением

$$\delta_j(n) = -\frac{\partial \mathbf{E}(n)}{\partial v_j(n)} = -\frac{\partial \mathbf{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n)\varphi_j'(v_j(n)) \quad . \qquad (32)$$

Локальный градиент указывает на требуемое изменение синаптического веса. В соответствии с (32) локальный градиент $b_j(n)$ выходного нейрона j равен произведению соответствующего сигнала ошибки $e_j(n)$ этого нейрона и производной $\varphi_j'(v_j(n))$ соответствующей функции активации.

Из выражений (31) и (32) видно, что ключевым фактором в вычислении величины коррекции $\Delta w_{ji}(n)$ весовых коэффициентов является сигнал ошибки $e_j(n)$ нейрона j. В этом контексте можно

выделить два различных случая, определяемых положением нейрона j в сети. В первом случае нейрон j является выходным узлом. Это довольно простой случай, так как для каждого выходного узла сети известен соответствующий желаемый отклик. Следовательно, вычислить сигнал ошибки можно с помощью простой арифметической операции. Во втором случае нейрон j является скрытым узлом. Однако даже если скрытый нейрон непосредственно недоступен, он несет ответственность за ошибку, получаемую на выходе сети. Вопрос состоит в том, как персонифицировать вклад (положительный или отрицательный) отдельных скрытых нейронов в общую ошибку. Эта проблема называется *задачей назначения коэффициентов доверия*. Она очень элегантно решается с помощью метода обратного распространения сигнала ошибки по сети.

Случай 1. Нейрон j – выходной узел. Если нейрон j расположен в выходном слое сети, для него известен соответствующий желаемый отклик. Значит, с помощью выражения (23) можно определить сигнал ошибки $e_j(n)$ (см. рис. 27) и без проблем вычислить градиент $\delta_j(n)$ по формуле (32).

Случай 2. Нейрон j – скрытый узел. Если нейрон *j* расположен в скрытом слое сети, желаемый отклик для него неизвестен. Следовательно, сигнал ошибки скрытого нейрона должен рекурсивно вычисляться на основе сигналов ошибки всех нейронов, с которым он непосредственно связан. Именно здесь алгоритм обратного распространения сталкивается с наибольшей сложностью. Рассмотрим ситуацию, изображенную на рис. 28, где изображен скрытый нейрон j. Согласно (32), локальный градиент $\delta_j(n)$ скрытого нейрона j можно переопределить следующим образом

$$\delta_j(n) = -\frac{\partial \mathbf{E}(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = -\frac{\partial \mathbf{E}(n)}{\partial y_j(n)} \varphi_j'(v_j(n))$$
,

где j – скрытый нейрон. Для вычисления частной производной $\partial \mathbf{E}(n)/\partial y_j(n)$ выполним некоторые преобразования. На рис. 28 видно, что

$$\mathbf{E}(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n) \quad , \qquad (33)$$

где k – выходной нейрон. Соотношение (33) — это выражение (24), в котором индекс j заменен индексом k. Это сделано для того, чтобы избежать путаницы с индексом j, использованным ранее для скрытого нейрона. Дифференцируя (33) по функциональному сигналу $y_j(n)$, получим:

$$\frac{\partial \mathbf{E}(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial y_j(n)} .$$

Окончательно запишем *формулу обратного распространения* для локального градиента $\delta_j(n)$ скрытого нейрона j [53]:

$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) . \qquad (34)$$

Графическое представление формулы (34) представлено на рис. 29. Формула записано в предположении, что выходной слой состоит из m_L нейронов.

Множитель $\varphi_j'(v_j(n))$, использованный в формуле (34) для вычисления локального градиента $\delta_j(n)$, зависит исключительно от функции активации, связанной со скрытым нейроном j. Второй множитель формулы (сумма по k) зависит от двух множеств параметров.

Первое из них – $\delta_k(n)$ – требует знания сигналов ошибки $e_k(n)$ для всех нейронов слоя, находящегося правее скрытого нейрона j, которые непосредственно связаны с нейроном j (см. рис. 28).

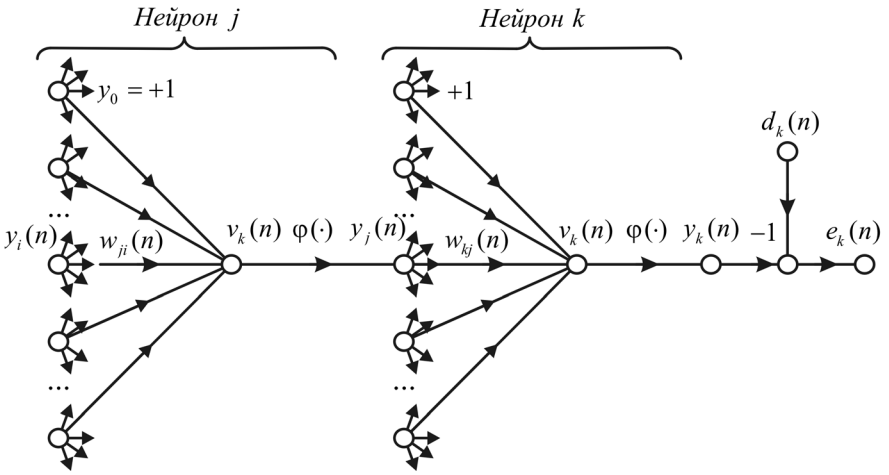


Рис. 28 – Граф передачи сигнала, детально отражающий связь выходного нейрона k со скрытым нейроном j [52]

Второе множество – $w_{kj}(n)$ – состоит из синаптических весов этих связей.

Теперь можно свести воедино все соотношения, которые были введены для алгоритма обратного распространения. Во-первых, коррекция веса $\Delta w_{ji}(n)$, применяемая к j синаптическому весу, соединяющему нейроны i и j, определяется следующим дельта-правилом:

$$\Delta w_{ji}(n) = \eta \cdot \delta_j(n) \cdot y_i(n), \qquad (35)$$

где η – параметр скорости обучения, $\delta_j(n)$ – локальный градиент, $y_i(n)$ – входной сигнал нейрона j.

Во-вторых, значение локального градиента $\delta_j(n)$ зависит от положения нейрона в сети.

1. Если нейрон j – выходной, то градиент $\delta_j(n)$ равен произведению производной $\varphi_j'(v_j(n))$ на сигнал ошибки $e_j(n)$ для нейрона j (см. выражение (32)).

2. Если нейрон j – скрытый, то градиент $\delta_j(n)$ равен произведению производной $\varphi_j'(v_j(n))$ на взвешенную сумму градиентов, вычисленных для нейронов следующего скрытого или выходного слоя, которые непосредственно связаны с данным нейроном j (см. выражение (34)).

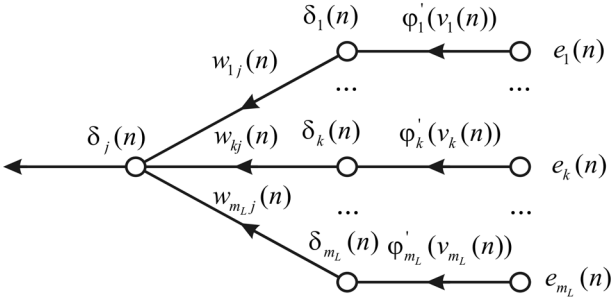


Рис. 29 – Граф передачи сигнала для фрагмента системы, задействованного в обратном распространении сигналов ошибки [52]

5.4. Два прохода вычислений

При использовании алгоритма обратного распространения различают два прохода, выполняемых в процессе вычислений [52]. Первый проход называется прямым, второй – обратным.

При *прямом проходе* синаптические веса остаются неизменными во всей сети, а функциональные сигналы вычисляются последовательно, от нейрона к нейрону. Функциональный сигнал на выходе нейрона j вычисляется по формуле

$$y_j(n) = \varphi(v_j(n)),$$

где $v_j(n)$ – индуцированное локальное поле нейрона j; определяемое выражением

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n),$$

где m – общее число входов (за исключением порога) нейрона j, $w_{ji}(n)$ – синаптический вес, соединяющий нейроны i и j, $y_i(n)$ – входной сигнал нейрона j или выходной сигнал нейрона i. Если нейрон j расположен в первом скрытом слое сети, то $m = m_0$, а индекс i относится к i-му входному терминалу сети, для которого можно записать

$$y_j(n) = x_i(n),$$

где $x_i(n)$ – i-й элемент входного вектора (образа). С другой стороны, если нейрон j расположен в выходном слое сети, то $m = m_L$, а индекс j означает j-й выходной терминал сети, для которого можно записать

$$y_i(n) = o_j(n),$$

где $o_j(n)$ – j-й элемент выходного вектора. Выходной сигнал сравнивается с желаемым откликом $d_j(n)$, в результате чего вычисляется сигнал ошибки $e_j(n)$ для j-го выходного нейрона. Таким образом, прямая фаза вычислений начинается с представления входного вектора первому скрытому слою сети и завершается в последнем слое вычислением сигнала ошибки для каждого нейрона этого слоя.

Обратный проход начинается с выходного слоя предъявлением ему сигнала ошибки, который передается справа налево от слоя к слою с параллельным вычислением локального градиента для каждого нейрона. Этот рекурсивный процесс предполагает изменение синаптических весов в соответствии с дельта-правилом (35). Для нейрона, расположенного в выходном слое, локальный градиент равен соответствующему сигналу ошибки, умноженному на первую производную нелинейной функции активации. Затем соотношение (35) используется для вычисления изменений весов, связанных с выходным слоем нейронов. Зная локальные градиенты для всех нейронов выходного слоя, с помощью (34) можно вычислить локальные градиенты всех нейронов предыдущего слоя, а значит, и величину коррекции весов связей с этим слоем. Такие вычисления проводятся для всех слоев в обратном направлении.

Заметим, что после предъявления очередного обучающего примера он "фиксируется" на все время прямого и обратного проходов.

Вычисление локального градиента δ для каждого нейрона многослойного персептрона требует знания производной функции активации $\varphi(\cdot)$, связанной с этим нейроном. Для существования такой производной функция активации должна быть непрерывной. Другими словами, *дифференцируемость* является единственным требованием, которому должна удовлетворять функция активации. Примером непрерывно дифференцируемой нелинейной функции активации, которая часто используется в многослойных

персептронах, является *сигмоидалъная нелинейная функция*, две формы которой описываются ниже.

5.5. Дифференцирование логистической функции активации

Эта форма сигмоидальной нелинейности в общем виде определяется следующим образом [52]:

$$\varphi_j(v_j(n)) = \frac{1}{1+e^{-av_j(n)}},\ a>0,\ -\infty < v_j(n) < +\infty, \qquad (36)$$

где $v_j(n)$ – индуцированное локальное поле нейрона j. Из (36) видно, что амплитуда выходного сигнала нейрона с такой активационной функцией лежит в диапазоне $0 < y_j < 1$.

Дифференцируя (36) по $v_j(n)$, получим:

$$\varphi_j'(v_j(n)) = \frac{a e^{-av_j(n)}}{\left(1+e^{-av_j(n)}\right)^2}. \qquad (37)$$

Так как $y_j(n) = \varphi_j(v_j(n))$, то можно избавиться от экспоненты $e^{-av_j(n)}$ в выражении (37) и представить производную функции активации в следующем виде:

$$\varphi_j'(v_j(n)) = a y_j(n)\left[1 - y_j(n)\right]. \qquad (38)$$

Для нейрона j, расположенного в выходном слое, $y_j(n) = o_j(n)$. Отсюда локальный градиент нейрона j можно выразить следующим образом:

$$\delta_j(n) = e_j(n)\varphi_j'(v_j(n)) = a\left[d_j(n) - o_j(n)\right]o_j(n)\left[1 - o_j(n)\right], \qquad (39)$$

где $o_j(n)$ – функциональный сигнал на выходе нейрона j, $d_j(n)$ – его желаемый сигнал. С другой стороны, для произвольного скрытого нейрона j локальный градиент можно выразить так:

$$\delta_j(n) = \varphi_j'(v_j(n))\sum_k \delta_k(n) w_{kj}(n) = a y_j(n)\left[1 - y_j(n)\right]\sum_k \delta_k(n) w_{kj}(n) \quad . \quad (40)$$

Согласно выражению (38), производная $\varphi_j'(v_j(n))$ достигает своего максимального значения при $y_j(n) = 0.5$, а минимального (нуля) – при $y_j(n) = 0$ и $y_j(n) = 1,0$. Так как величина коррекции синаптических весов в сети пропорциональна производной $\varphi_j'(v_j(n))$, то для максимального изменения синаптических весов нейрона его функциональные сигналы (вычисленные в соответствии с сигмоидальной функцией) должны находиться в середине диапазона. Согласно [60], именно это свойство алгоритма обратного распространения вносит наибольший вклад в его устойчивость как алгоритма обучения.

5.6. Дифференцирование функции гиперболического тангенса

Еще одной часто используемой формой сигмоидальной нелинейности является функция гиперболического тангенса, которая в общем виде описывается выражением [52]:

$$\varphi_j'(v_j(n)) = a\tanh(b v_j(n)), \quad (a,b) > 0,$$

где a и b – константы. В действительности функция гиперболического тангенса является не более чем логистической функцией, только масштабированной и смещенной. Ниже приведем формулы для локального градиента для этой формы сигмоидальной нелинейности.

Для нейрона j, расположенного в выходном слое, локальный градиент будет иметь вид

$$\delta_j(n) = e_j(n)\varphi_j'(v_j(n)) = \frac{b}{a}\left[d_j(n) - o_j(n)\right]\left[a - o_j(n)\right]\left[a + o_j(n)\right]. \quad (41)$$

Для нейрона j, расположенного в скрытом слое, локальный градиент приобретает вид

$$\delta_j(n) = \varphi_j'(v_j(n))\sum_k \delta_k(n) w_{kj}(n) =$$

$$= \frac{b}{a}\left[d_j(n) - y_j(n)\right]\left[a + y_j(n)\right]\sum_k \delta_k(n) w_{kj}(n). \quad (42)$$

Используя формулы (39) и (40) для логистической функции и формулы (41) и (42) для гиперболического тангенса, можно вычислить локальный градиент даже без явного знания активационной функции.

5.7. Скорость обучения

Алгоритм обратного распространения обеспечивает построение в пространстве весов "аппроксимации" для траектории, вычисляемой методом наискорейшего спуска. Чем меньше параметр скорости обучения η, тем меньше корректировка синаптических весов, осуществляемая на каждой итерации, и тем более гладкой является траектория в пространстве весов [52]. Однако это улучшение происходит за счет замедления процесса обучения. С другой стороны, если увеличить параметр η для повышения скорости обучения, то результирующие большие изменения синаптических весов могут привести систему в неустойчивое состояние. Простейшим способом повышения скорости обучения без потери устойчивости является изменение дельта-правила (31) за счет добавления к нему момента инерции [60]:

$$\Delta w_{ji}(n) = a\Delta w_{ji}(n-1) + \eta\delta_j(n)y_i(n), \qquad (43)$$

где a – как правило, положительное значение, называемое постоянной момента. Для частного случая было показано [61], что использование постоянной момента a уменьшает диапазон устойчивости параметра скорости обучения η и может привести к неустойчивости, если последний выбран неверно. Более того, с увеличением значения a рассогласование возрастает.

Как показано на рис. 30, значение a является управляющим воздействием в контуре обратной связи для $\Delta w_{ji}(n)$, где z^{-1} – оператор единичной задержки. Уравнение (43) называется обобщенным дельта-правилом, а вывод алгоритма обратного распространения с учетом постоянной момента содержится в [62]. При $a = 0$ оно вырождается в обычное дельта-правило.

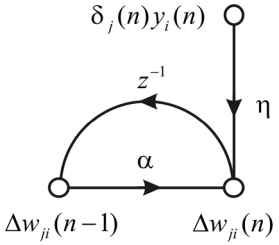


Рис. 30 – Граф прохождения сигнала, иллюстрирующий постоянную момента a [52]

Для того чтобы оценить влияние последовательности представления обучающих примеров на синаптические веса в зависимости от константы a, перепишем (43) в виде временного ряда с индексом t. Значение индекса t будет изменяться от нуля до текущего значения n. При этом выражение (43) можно рассматривать как разностное уравнение первого порядка для

коррекции весов $\Delta w_{ji}(n)$. Решая это уравнение относительно $\Delta w_{ji}(n)$, получим временной ряд длины $n+1$:

$$\Delta w_{ji}(n) = \eta \sum_{t=0}^{n} a^{n-1} \delta_j(t) y_i(t) . \qquad (44)$$

Следовательно, соотношение (44) можно переписать в эквивалентной форме:

$$\Delta w_{ji}(n) = -\eta \sum_{t=0}^{n} a^{n-t} \frac{\partial \mathbf{E}(t)}{\partial w_{ji}(t)} .$$

Основываясь на этом выражении, можно сделать следующие интуитивные наблюдения [63, 64]:

1. Текущее значение коррекции весов $\Delta w_{ji}(n)$ представляет собой сумму экспоненциально взвешенного временного ряда. Для того чтобы этот ряд сходился, постоянная момента должна находиться в диапазоне $0 < |a| < 1$. Если константа a равна нулю, алгоритм обратного распространения работает без момента. Следует также заметить, что константа a может быть и отрицательной, хотя эти значения не рекомендуется использовать на практике.

2. Если частная производная $\partial \mathbf{E}(n) / \partial w_{ji}(n)$ имеет один и тот же алгебраический знак на нескольких последовательных итерациях, то экспоненциально взвешенная сумма $\Delta w_{ji}(n)$ возрастает по абсолютному значению, поэтому веса $w_{ji}(n)$ могут изменяться на очень большую величину. Включение момента в алгоритм обратного распространения ведет к *ускорению спуска* в некотором постоянном направлении.

3. Если частная производная $\partial \mathbf{E}(n)/\partial w_{ji}(n)$ на нескольких последовательных итерациях меняет знак, экспоненциально взвешенная сумма $\Delta w_{ji}(n)$ уменьшается по абсолютной величине, поэтому веса $w_{ji}(n)$ изменяются на небольшую величину. Таким образом, добавление момента в алгоритм обратного распространения ведет к *стабилизирующему эффекту* для направлений, изменяющих знак.

Включение момента в алгоритм обратного распространения обеспечивает незначительную модификацию метода корректировки весов, оказывая положительное влияние на работу алгоритма обучения. Кроме того, слагаемое момента может предотвратить нежелательную остановку алгоритма в точке какого-либо локального минимума на поверхности ошибок.

При выводе алгоритма обратного распространения предполагалось, что параметр интенсивности обучения представлен константой η. Однако на практике он может задаваться как η_{ji}. Это значит, что параметр скорости обучения является локальным и определяется для каждой конкретной связи. В самом деле, применяя различные параметры скорости обучения в разных областях сети, можно добиться интересных результатов. Более подробно на этом вопросе мы остановимся в последующих разделах.

Следует отметить, что при реализации алгоритма обратного распространения можно изменять как все синаптические веса сети, так и только часть из них, оставляя остальные на время адаптации фиксированными. В последнем случае сигнал ошибки распространяется по сети в обычном порядке, однако фиксированные синаптические веса будут оставаться неизменными. Этого можно добиться, установив для

соответствующих синаптических весов w_{ji} параметр интенсивности обучения η_{ji} равным нулю.

ГЛАВА 6. ПОТОКОВЫЕ ГРАФЫ И ГРАДИЕНТНЫЕ МЕТОДЫ ОБУЧЕНИЯ

6.1. Потоковые графы и их применение для генерации градиента

Знакомство с формулами для расчета градиента показывает, что они довольно сложны и неудобны для практического применения, особенно если сеть содержит более одного скрытого слоя [54]. Поэтому представляется интересным, что на основе метода потоковых графов удается построить очень простые правила формирования компонентов градиента, которые имеют постоянную структуру, не зависящую от сложности сети. При этом базу таких правил составляют соотношения, полученные в результате анализа чувствительности сети методом сопряженных элементов. В теории систем [65] под полной чувствительностью объекта понимается производная любого циркулирующего в нем сигнала относительно значений весов, которая может быть рассчитана на основании знаний о сигналах, распространяющихся по обычному графу (обозначаемому G) и сопряженному с ним графу (обозначаемому $\bar{G}$). Граф $\bar{G}$ определяется как исходный граф G, в котором направленность всех дуг изменена на противоположную. Линейная дуга графа G и соответствующая ей дуга сопряженного графа $\bar{G}$ имеют идентичные описания. В случае нелинейной связи $f(x,k)$ где x – входной сигнал, а k – параметр, соответствующая ей дуга графа $\bar{G}$ линеаризуется с коэффициентом $\beta = \frac{\partial f(x,k)}{\partial x}$, рассчитанным для фактического входного сигнала x графа G.

Как показано в работах [65-67], метод расчета чувствительности НС с использованием потоковых графов основан на анализе

исходного графа G и сопряженного с ним графа $\hat{G}$ при возбуждении последнего единичным сигналом, подаваемым на вход $\hat{G}$ (соответствующий выходу G). Чувствительность графа G относительно параметров дуг этого графа к произвольному входному сигналу v_0 можно выразить следующим образом:

а) для линейной дуги w_{ij} графа G

$$\frac{dv_0}{dw_{ij}} = v_j \hat{v}_i ,$$

где w_{ij} – это коэффициент усиления линейной дуги, направленной от j-го узла к i-му, v_j – сигнал j-го узла графа G, а $\hat{v}_i$ - сигнал i-го узла сопряженного графа $\hat{G}$, для которого в качестве входного сигнала задается значение $\hat{v}_0 = 1$;

б) для нелинейной дуги графа G, объединяющей l-й и k-й узлы и описываемой функцией $v_k = f_{kl}(v_l, K)$, чувствительность относительно параметра K определяется выражением

$$\frac{dv_0}{dK} = \hat{v}_k \frac{\partial f_{kl}(v_l, K)}{\partial K} ,$$

где $\frac{\partial f_{kl}(v_l, K)}{\partial K}$ рассчитывается для сигнала v_l l-го узла графа G.

Обозначим через w вектор оптимизированных параметров (весов w_i) системы, представленной графом G, $w = [w_1, w_2, ..., w_n]^T$, а через $E(w)$ – целевую функцию. Тогда градиент $\nabla E(w)$, сокращенно обозначаемый как $g(w) = \nabla E(w)$, можно определить в виде

$$g(w) = \left[\frac{\partial E(w)}{\partial w_1}, \frac{\partial E(w)}{\partial w_2}, ..., \frac{\partial E(w)}{\partial w_n} \right]^T$$

(в этом выражении для обозначения элементов вектора w использован один индекс $i = 1,2,...,n$). Если представить целевую функцию в форме, учитывающей только одну обучающую выборку

$$E(w) = \frac{1}{2}\sum_{i=1}^{M}(y_i - d_i)^2,$$

где d_i обозначено ожидаемое значение i-го выходного нейрона, $i = 1,2,...,M$, то градиент целевой функции принимает вид:

$$g(w) = \left[g_1(w), g_2(w), ..., g_n(w)\right]^T,$$

в которой

$$g_k(w) = \sum_{i=1}^{M}(y_i - d_i)\frac{\partial y_i}{\partial w_k}.$$

Для задания вектора градиента также необходимы производные выходных сигналов y_i графа относительно весов w_k (показатели их чувствительности), умноженные на величину погрешности $(y_i - d_i)$. Благодаря использованию методов теории графов все эти операции (в том числе и суммирование) можно выполнить за один шаг с помощью исходного графа G и сопряженного с ним графа $\hat{G}$ при соблюдении соответствующих условий возбуждения графа $\hat{G}$. Как показано в [67], вследствие линейности сопряженного графа все эти операции могут быть реализованы автоматически в случае, когда в сопряженном графе вместо единичных возбуждений генерируются сигналы в виде разностей между фактическими y_i, и ожидаемыми d_i значениями выходных сигналов. Способ формирования сопряженного графа $\hat{G}$ и методика его возбуждения для автоматического расчета вектора градиента на основе анализа только двух графов G и $\hat{G}$ представлены на рис. 31. При замене

всех единичных возбуждений в $\hat{G}$ на $(y_i - d_i)$ любой компонент вектора градиента $g_k(w)$ может быть рассчитан по соответствующим сигналам исходного графа G и сопряженного с ним графа $\hat{G}$ точно так же, как и при определении обычной чувствительности. Для линейной дуги графа G, описываемой весом w_{ij}, формула имеет вид:

$$\frac{\partial E(w)}{\partial w_{ij}} = v_j \hat{v}_i .$$

Для нелинейной дуги графа G, описываемой функцией $w_{kl}(v_l, K)$, получаем:

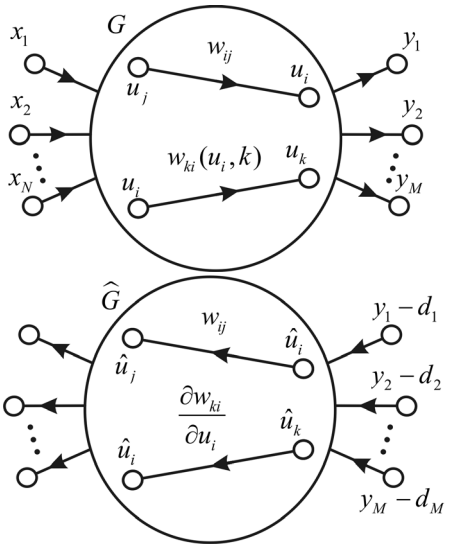
$$\frac{\partial E(w)}{\partial K} = \hat{v}_k \frac{\partial w_{kl}(v_l, K)}{\partial K} .$$

Представленные выражения применимы для любых систем (линейных, нелинейных, рекуррентных и т.п.). Они практически применяются для анализа однонаправленных многослойных НС, описываемых потоковым графом прохождения сигналов.

Рассмотрим изображенную на рис. 32,*а* типовую многослойную сеть (состоящую из m слоев) с произвольной непрерывной функцией активации нейронов. Количество нейронов в каждом слое будем обозначать $K_j (j = 1, 2, ..., m)$ причем последний слой является выходным слоем сети, содержащим $K_m = M$ нейронов. Выходные сигналы нейронов в конкретных слоях обозначим $v_j^{(k)}$ причем *для* последнего слоя $v_j^{(m)} = y_j$.

Для определения компонентов градиента относительно весов конкретных слоев сети будем применять формулировки, относящиеся к сопряженному графу. На рис 32,*б* представлен сопряженный граф сети, изображенной на рис. 32,*а*, причем для унификации описания все сигналы в сопряженном графе

обозначены символами с "шапочкой" ($\hat{u}$). Сопряженный граф возбуждается разностями между фактическими y_i и ожидаемыми d_i, значениями выходных сигналов. Нелинейные дуги графа G заменяются в сопряженном графе $\hat{G}$ производными $\frac{df(x)}{dx}$, значения которых рассчитываются раздельно для каждого слоя в точках $x = u_i$.



а) *б*)

Рис. 31 – Иллюстрация применения способа формирования и возбуждения сопряженного графа: *а*) исходный граф G, *б*) сопряженный граф $\hat{G}$ [54]

Если, например, функция активации нейронов имеет сигмоидальную униполярную форму $\frac{df(x)}{dx} = f(x)(1 - f(x))$ рассчитывается непосредственно на основе известного значения

сигмоидальной функции в точке x и не требует никаких дополнительных вычислений.

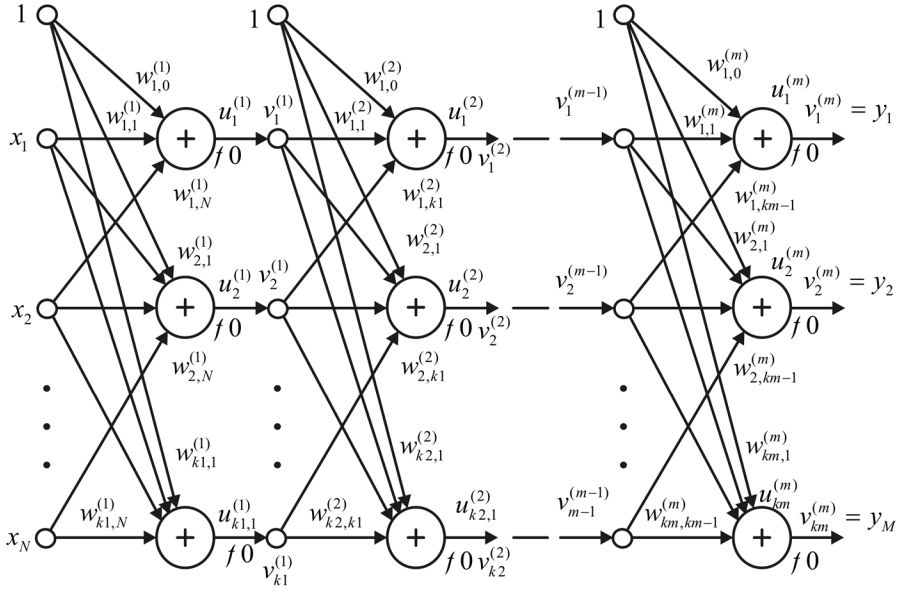
Опираясь на предложенный алгоритм определения градиента методами теории графов, можно рассчитать конкретные компоненты вектора градиента $g(w)$ для любого слоя нейронов [54].

Еще одним важным достоинством графического метода, помимо значительного упрощения вычислительных процедур, считается возможность учета равенства значений различных весов сети [65]. Если, например, вес со значением w относится к дуге w_{ij}, соединяющей i-й и j-й узлы (в направлении от j-го к i-му), и к дуге w_{kl}, соединяющей k-й и l-й узлы (в направлении от l-го к i-му), то легко заметить, что вес w будет присутствовать в двух различных позициях выражения, определяющего целевую функцию. Согласно правилу дифференцирования составной функции ее производная представляется в виде суммы производных относительно w_{ij} и w_{kl}. Следовательно,

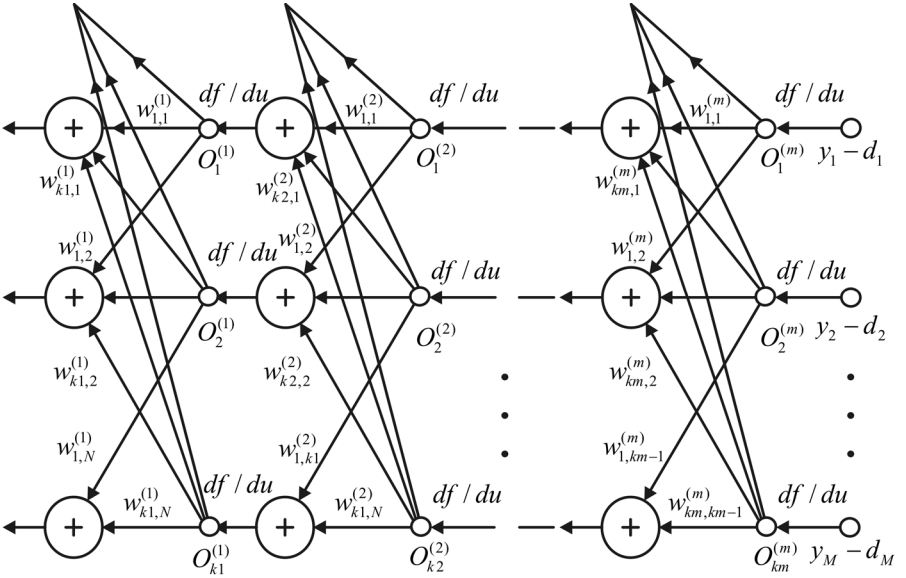
$$\frac{\partial E(w)}{\partial w} = \frac{\partial E}{\partial w_{ij}} + \frac{\partial E}{\partial w_{kl}}.$$

С учетом рассуждений относительно сопряженного графа при вводе унифицированных обозначений сигналов для любых узлов в виде $v(\widehat{v})$ с соответствующими индексами получаем окончательное выражение

$$\frac{\partial E(w)}{\partial w} = v_j \widehat{u}_i + v_l \widehat{u}_k. \qquad (45)$$


1
1
1
x_1
x_2
x_N
$w_{1,0}^{(1)}$
$w_{1,1}^{(1)}$
$w_{1,N}^{(1)}$
$w_{2,1}^{(1)}$
$w_{2,N}^{(1)}$
$w_{k1,1}^{(1)}$
$w_{k1,N}^{(1)}$
$u_1^{(1)}$
$u_2^{(1)}$
$u_{k1,1}^{(1)}$
$v_1^{(1)}$
$v_2^{(1)}$
$v_{k1}^{(1)}$
$f\,0$
$w_{1,0}^{(2)}$
$w_{1,1}^{(2)}$
$w_{1,k1}^{(2)}$
$w_{2,1}^{(2)}$
$w_{2,k1}^{(2)}$
$w_{k2,1}^{(2)}$
$w_{k2,k1}^{(2)}$
$u_1^{(2)}$
$u_2^{(2)}$
$u_{k2,1}^{(2)}$
$v_1^{(2)}$
$v_2^{(2)}$
$v_{k2}^{(2)}$
$v_1^{(m-1)}$
$v_2^{(m-1)}$
$v_{m-1}^{(m-1)}$
$w_{1,0}^{(m)}$
$w_{1,1}^{(m)}$
$w_{1,km-1}^{(m)}$
$w_{2,1}^{(m)}$
$w_{2,km-1}^{(m)}$
$w_{km,1}^{(m)}$
$w_{km,km-1}^{(m)}$
$u_1^{(m)}$
$u_2^{(m)}$
$u_{km}^{(m)}$
$v_1^{(m)} = y_1$
$v_2^{(m)} = y_2$
$v_{km}^{(m)} = y_M$

а)


df / du
$w_{1,1}^{(1)}$
$w_{k1,1}^{(1)}$
$w_{1,2}^{(1)}$
$w_{k1,2}^{(1)}$
$w_{1,N}^{(1)}$
$w_{k1,N}^{(1)}$
$O_1^{(1)}$
$O_2^{(1)}$
$O_{k1}^{(1)}$
$w_{1,1}^{(2)}$
$w_{k2,1}^{(2)}$
$w_{1,2}^{(2)}$
$w_{k2,2}^{(2)}$
$w_{1,k1}^{(2)}$
$w_{k1,N}^{(2)}$
$O_1^{(2)}$
$O_2^{(2)}$
$O_{k2}^{(2)}$
$w_{1,1}^{(m)}$
$w_{km,1}^{(m)}$
$w_{1,2}^{(m)}$
$w_{km,2}^{(m)}$
$w_{1,km-1}^{(m)}$
$w_{km,km-1}^{(m)}$
$O_1^{(m)}$ $y_1 - d_1$
$O_2^{(m)}$ $y_2 - d_2$
$O_{km}^{(m)}$ $y_M - d_M$

б)

Рис. 32 – Иллюстрация метода сопряженных графов для генерации вектора градиента однонаправленной многослойной сети: *а*) выходной граф сети, *б*) сопряженный граф [54]

Как следует из формулы (45), учет равенства отдельных весов $w_{ij} = w_{lk}$ не только не усложняет общую расчетную формулу, но, напротив, упрощает ее за счет уменьшения количества переменных. Необходимо отметить, что совпадающие веса могут лежать как в одном и том же, так и в совершенно разных слоях. Сущность формулы (45) при этом совершенно не меняется. В этом заключается важнейшее отличие метода генерации градиента, основанного на потоковых графах распространения сигналов, от классического подхода, широко представленного в мировой литературе [68, 69].

6.2. Основные положения градиентных алгоритмов обучения сети

Задачу обучения нейронной сети будем рассматривать на данном этапе как требование минимизировать априори определенную целевую функцию $E(w)$ [54]. При таком подходе можно применять для обучения алгоритмы, которые в теории оптимизации считаются наиболее эффективными. К ним, без сомнения, относятся градиентные методы, чью основу составляет выявление градиента целевой функции. Они связаны с разложением целевой функции $E(w)$ в ряд Тейлора в ближайшей окрестности точки имеющегося решения w. В случае целевой функции от многих переменных ($w = [w_1, w_2, ..., w_n]^T$) такое представление связывается с окрестностью ранее определенной точки (в частности, при старте алгоритма это исходная точка w_0) в направлении p. Подобное разложение описывается универсальной формулой вида [70, 71]

$$E(w+p) = E(w) + \left[g(w)\right]^T p + \frac{1}{2} p^T H(w) p + ..., \quad (46)$$

где $g(w) = \nabla \mathbf{E} = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, ..., \frac{\partial E}{\partial w_n}\right]^T$ – вектор градиента, а симметричная квадратная матрица

$$H(w) = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1 \partial w_1} & ... & \frac{\partial^2 E}{\partial w_1 \partial w_n} \\ ... & ... & ... \\ \frac{\partial^2 E}{\partial w_n \partial w_1} & ... & \frac{\partial^2 E}{\partial w_n \partial w_n} \end{bmatrix}$$

является матрицей производных второго порядка, называемой *гессианом.*

В выражении (46) p играет роль направляющего вектора, зависящего от фактических значений вектора w. На практике чаще всего рассчитываются три первых члена ряда (46), а последующие просто игнорируются. При этом, зависимость (46) может считаться квадратичным приближением целевой функции $E(w)$ в ближайшей окрестности найденной точки w с точностью, равной локальной погрешности отсеченной части $O(h^3)$, где $h = \|p\|$. Для упрощения описания значения переменных, полученные в k-м цикле, будем записывать с нижним индексом k. Точкой решения $w = w_k$ будем считать точку, в которой достигается минимум целевой функции $E(w)$ и $g(w_k) = 0$, а гессиан $H(w)$ является положительно определенным [71, 72]. При выполнении этих условий функция в любой точке, лежащей в окрестности w_k, имеет большее значение, чем в точке w_k, поэтому точка w_k является решением, соответствующим критерию минимизации целевой функции.

В процессе поиска минимального значения целевой функции направление поиска p и шаг h подбираются таким образом, чтобы для каждой очередной точки $w_k + 1 = w_k + \eta_k p_k$ выполнялось условие $E(w_{k+1}) < E(w_k)$. Поиск минимума продолжается, пока норма градиента не упадет ниже априори заданного значения допустимой погрешности либо пока не будет превышено максимальное время вычислений (количество итераций).

Универсальный оптимизационный алгоритм обучения нейронной сети можно представить в следующем виде [54] (будем считать, что начальное значение оптимизируемого вектора известно и составляет $w_k = w_0$):

1. Проверка сходимости и оптимальности текущего решения w_k. Если точка w_k отвечает градиентным условиям остановки процесса – завершение вычислений. В противном случае перейти к п.2.

2. Определение вектора направления оптимизации p_k *для* точки w_k.

3. Выбор величины шага η_k в направлении p_k при котором выполняется условие $E(w_k + \eta_k p_k) < E(w_k)$.

4. Определение нового решения $w_{k+1} = w_k + \eta_k p_k$, а также соответствующих ему значений $E(w_k)$ и $g(w_k)$, а если требуется – то и $H(w_k)$, и возврат к п.1.

6.2. Алгоритм наискорейшего спуска

Если при разложении целевой функции $E(w)$ в ряд Тейлора ограничиться ее линейным приближением, то мы получим

алгоритм наискорейшего спуска [54]. Для выполнения соотношения $E(w_{k+1}) = E(w_k)$ достаточно подобрать $g(w_k)^T p < 0$. Условию уменьшения значения целевой функции отвечает выбор вектора направления

$$p_k = -g(w_k). \qquad (47)$$

Именно выражением (47) определяется вектор направления p в методе наискорейшего спуска.

Ограничение слагаемым первого порядка при разложении функции в ряд Тейлора не позволяет использовать информацию о ее кривизне. Это обусловливает медленную сходимость метода (она остается линейной). Указанный недостаток, а также резкое замедление минимизации в ближайшей окрестности точки оптимального решения, когда градиент принимает очень малые значения, делают алгоритм наискорейшего спуска низкоэффективным. Тем не менее, с учетом его простоты, невысоких требований к объему памяти и относительно небольшой вычислительной сложности, именно этот метод в течение многих лет был и остается в настоящее время основным способом обучения многослойных сетей. Повысить его эффективность удается путем модификации (как правило, эвристической) выражения, определяющего направление. Хорошие результаты приносит применение метода обучения с так называемым моментом. При этом подходе уточнение весов сети ($w_{k+1} = w_k + \Delta w_k$) производится с учетом модифицированной формулы определения значения Δw_k

$$\Delta w_k = \eta_k p_k + \alpha(w_k - w_{k-1}), \qquad (48)$$

где α – это коэффициент момента, принимающий значения в интервале $[0,1]$. Первое слагаемое этого выражения соответствует обычному обучению по методу наискорейшего спуска, тогда как второе учитывает последнее изменение весов и не зависит от

фактического значения градиента. Чем больше значение коэффициента α, тем большее значение оказывает показатель момента на подбор весов. Это влияние существенно возрастает на плоских участках целевой функции, а также вблизи локального минимума, где значение градиента близко к нулю.

На плоских участках целевой функции приращение весов (при постоянном значении коэффициента обучения $\eta_k = \eta$), остается приблизительно одним и тем же. Это означает, что $\Delta w_k = \eta p_k + \alpha \Delta w_k$, поэтому эффективное приращение значений весов можно описать отношением

$$\Delta w_k = \frac{\eta}{1-\alpha} p_k .$$

При значении $\alpha = 0.9$ это соответствует 10-кратному увеличению эффективного значения коэффициента обучения и, следовательно, также 10-кратному ускорению процесса обучения.

Вблизи локального минимума показатель момента, не связанный с градиентом, может вызвать слишком большое изменение весов, приводящее к увеличению значения целевой функции и к выходу из "зоны притяжения" этого минимума. При малых значениях градиента показатель момента начинает доминировать в выражении (48), что приводит к такому приращению весов Δw_k, которое соответствует увеличению значения целевой функции, позволяющему выйти из зоны локального минимума. Однако показатель момента не должен полностью доминировать на протяжении всего процесса обучения, поскольку это привело бы к нестабильности алгоритма. Для предотвращения такого избыточного доминирования значение целевой функции E контролируется так, чтобы допускать его увеличение только в ограниченных пределах, например не более 4%. При таком подходе, если на очередных (k-м и $(k+1)$-м) шагах итерации

выполняется условие $E(k+1) < 1.04E(k)$, то изменения игнорируются и считается, что $(w_k - w_k) = 0$. При этом показатель градиента начинает доминировать над показателем момента и процесс развивается в направлении минимизации, заданном вектором градиента. Следует подчеркнуть, что подбор величины коэффициента момента является непростым делом и требует проведения большого количества экспериментов, имеющих целью выбрать такое значение, которое наилучшим образом отражало бы специфику решаемой проблемы

Алгоритм переменной метрики. В методе переменной метрики используется квадратичное приближение функции $E(w)$ в окрестности полученного решения w_k. Если в формуле (46) ограничиться тремя первыми слагаемыми, то получим:

$$E(w_k + p_k) \cong E(w_k) + g(w_k)^T p_k + \frac{1}{2} p_k^T H(w_k) p_k + O(h^3). \qquad (49)$$

Для достижения минимума функции (49) требуется, чтобы $\frac{dE(w_k + p_k)}{dp_k} = 0$. При выполнении соответствующего дифференцирования можно получить условие оптимальности в виде

$$g(w_k) + H(w_k) p_k = 0.$$

Элементарное преобразование этого выражения дает очевидное решение

$$p_k = -\left[\mathbf{H}(w_k)\right]^{-1} g(w_k). \qquad (50)$$

Формула (50) однозначно указывает направление p_k, которое гарантирует достижение минимального для данного шага значения целевой функции. Из него следует, что для определения этого направления необходимо в каждом цикле вычислять значение

градиента g и гессиана $\mathbf{H}$ в точке известного (последнего) решения w_k.

Формула (50), представляющая собой основу ньютоновского алгоритма оптимизации, является чисто теоретическим выражением, поскольку ее применение требует положительной определенности гессиана на каждом шаге, что в общем случае практически неосуществимо. По этой причине в имеющихся реализациях алгоритма, как правило, вместо точно определенного гессиана $\mathbf{H}(w_k)$ используется его приближение $\mathbf{G}(w_k)$. Одним из наиболее популярных считается метод переменной метрики [70, 71]. В соответствии с этим методом на каждом шаге гессиан или обратная ему величина, полученная на предыдущем шаге, модифицируется на величину некоторой поправки. Если прирост вектора w_k и градиента g на двух последовательных шагах итерации обозначить соответственно s_k и r_k, т.е. $s_k = w_k - w_{k-1}$ и $r_k = g(w_k) - g(w_{k-1})$, а матрицу, обратную приближению гессиана $\mathbf{V}_k = \left[\mathbf{G}(w_k)\right]^{-1}$, $\mathbf{V}_{k-1} = \left[\mathbf{G}(w_{k-1})\right]^{-1}$, обозначить $\mathbf{V}$, то в соответствии с очень эффективной формулой Бройдена-Флетчера-Гольдфарба-Шенно (БФГШ) процесс уточнения значения матрицы $\mathbf{V}$ можно описать рекуррентной зависимостью [70]:

$$\mathbf{V}_k = \mathbf{V}_{k-1} + \left[1 + \frac{r_k^T \mathbf{V}_{k-1} r_k}{s_k^T r_k}\right] \frac{s_k s_k^T}{s_k^T r_k} - \frac{s_k r_k^T \mathbf{V}_{k-1} r_k s_k^T}{s_k^T r_k}.$$

В другом известном алгоритме Девидона-Флетчера-Пауэлла (ДФП) значение гессиана уточняется согласно выражению [70]

$$\mathbf{V}_k = \mathbf{V}_{k-1} + \frac{s_k s_k^T}{s_k^T r_k} - \frac{\mathbf{V}_{k-1} r_k r_k^T \mathbf{V}_{k-1}}{r_k^T \mathbf{V}_{k-1} r_k}.$$

В качестве начального значения обычно принимается $\mathbf{V}_0 = 1$, а первая итерация проводится в соответствии с алгоритмом

наискорейшего спуска. Как показано в [70, 71], при начальном значении $\mathbf{V}_0 = 1$ и при использовании направленной минимизации на каждом шаге оптимизации можно обеспечить положительную определенность аппроксимированной матрицы гессиана. Направленная минимизация необходима при реализации как стратегии БФГШ, так и ДФП, причем в соответствии с проведенными тестами метод БФГШ менее чувствителен к различным погрешностям вычислительного процесса. По этой причине, несмотря на несколько большую вычислительную сложность, метод БФГШ применяется чаще, чем ДФП.

Метод переменной метрики характеризуется более быстрой сходимостью, чем метод наискорейшего спуска. Кроме того, факт положительной определенности гессиана на каждом шаге итерации придает уверенность в том, что выполнение условия $g(w_k) = 0$ действительно гарантирует решение проблемы оптимизации. Именно этот метод считается в настоящее время одним из наиболее эффективных способов оптимизации функции нескольких переменных. Его недостаток состоит в относительно большой вычислительной сложности (связанной с необходимостью расчета в каждом цикле n^2 элементов гессиана), а также в использовании значительных объемов памяти для хранения элементов гессиана, что в случае оптимизации функции с большим количеством переменных может стать серьезной проблемой. По этой причине метод переменной метрики применяется для не очень больших сетей. В частности, с использованием персонального компьютера была доказана его эффективность для сети, содержащей не более тысячи взвешенных связей.

6.4. Алгоритм Левенберга-Марквардта

Другим приложением ньютоновской стратегии оптимизации является алгоритм Левенберга-Марквардта [70,54]. При его использовании точное значение гессиана $\mathbf{H}(w)$ в формуле (50) заменяется аппроксимированным значением $\mathbf{G}(w)$ которое рассчитывается на основе содержащейся в градиенте информации с учетом некоторого регуляризационного фактора.

Для описания этого метода представим целевую функцию в виде, отвечающем существованию единственной обучающей выборки,

$$E(w) = \frac{1}{2}\sum_{i=1}^{M}\left[e_i(w)\right]^2, \qquad (51)$$

где $e_i = [y_i(w) - d_i]$. При использовании обозначений

$$e(w) = \begin{bmatrix} e_1(w) \\ e_2(w) \\ \dots \\ e_M(w) \end{bmatrix}, \quad \mathbf{J}(w) = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \dots & \frac{\partial e_1}{\partial w_n} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \dots & \frac{\partial e_2}{\partial w_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_M}{\partial w_1} & \frac{\partial e_M}{\partial w_2} & \dots & \frac{\partial e_M}{\partial w_n} \end{bmatrix},$$

вектор градиента и аппроксимированная матрица гессиана, соответствующие целевой функции (51), определяются в виде

$$g(w) = \left[\mathbf{J}(w)\right]^T e(w),$$

$$\mathbf{G}(w) = \left[\mathbf{J}(w)\right]^T \mathbf{J}(w) + \mathbf{R}(w), \qquad (52)$$

где через $\mathbf{R}(w)$ обозначены компоненты гессиана $\mathbf{H}(w)$, содержащие высшие производные относительно w. Сущность подхода Левенберга-Марквардта состоит в аппроксимации $\mathbf{R}(w)$ с

помощью регуляризационного фактора $v1$, в котором переменная v, называемая параметром Левенберга-Марквардта, является скалярной величиной, изменяющейся в процессе оптимизации. Таким образом, аппроксимированная матрица гессиана на k-*м* шаге алгоритма приобретает вид:

$$\mathbf{G}(w_k) = \left[\mathbf{J}(w_k)\right]^T \mathbf{J}(w_k) + v_k 1 . \qquad (53)$$

В начале процесса обучения, когда фактическое значение w_k еще далеко от искомого решения (велико значение вектора погрешности e) используется значение параметра v_k, намного превышающее собственное значение матрицы $\left[\mathbf{J}(w_k)\right]^T \mathbf{J}(w_k)$. В таком случае гессиан фактически подменяется регуляризационным фактором:

$$\mathbf{G}(w_k) \equiv v_k 1 .$$

а направление минимизации выбирается по методу наискорейшего спуска:

$$p_k = -\frac{g(w_k)}{v_k} .$$

По мере уменьшения погрешности и приближения к искомому решению величина параметра v_k понижается и первое слагаемое в формуле (52) начинает играть все более важную роль.

На эффективность алгоритма влияет грамотный подбор величины v_k. Слишком большое начальное значение v_k по мере прогресса оптимизации должно уменьшаться вплоть до нуля при достижении фактического решения, близкого к искомому. Известны различные способы подбора этого значения, но мы ограничимся описанием только одной оригинальной методики, предложенной Д. Марквардтом [73]. Пусть значения целевой функции на k-*м* и

$(k-1)$-*м* шагах итерации обозначаются соответственно E_k и E_{k-1}, а значения параметра v на этих же шагах – v_k и v_{k-1}. Коэффициент уменьшения значения v обозначим через r, причем $r>1$. В соответствии с классическим алгоритмом Левенберга-Марквардта значение v изменяется по следующей схеме:

- если $E\left(\frac{v_{k-1}}{r}\right) \le E_k$, то принять $v_k = \frac{v_k - 1}{r}$;
- если $E\left(\frac{v_{k-1}}{r}\right) > E_k$ и $E(v_{k-1}) < E_k$, то принять $v_k = v_{k-1}$;
- если $E\left(\frac{v_{k-1}}{r}\right) > E_k$ и $E(v_{k-1}) > E_k$, то увеличить последовательно m раз значение v до достижения $E(v_{k-1}r^m) \le E_k$, одновременно принимая $v_k = v_{k-1}r^m$.

Такая процедура изменения значения v выполняется до момента, в котором так называемый коэффициент верности отображения q, рассчитываемый по формуле

$$q = \frac{E_k - E_{k-1}}{[\Delta w_k]^T g_k + 0.5[\Delta w_k]^T \mathbf{G}_k \Delta w_k},$$

достигнет значения, близкого к единице. При этом квадратичная аппроксимация целевой функции имеет высокую степень совпадения с истинными значениями, что свидетельствует о близости оптимального решения. В такой ситуации регуляризационный фактор $v_k 1$ в формуле (53) может быть опущен ($v_k = 0$), процесс определения гессиана сводится к непосредственной аппроксимации первого порядка, а алгоритм Левенберга-Марквардта превращается в алгоритм Гаусса-Ньютона, характеризующийся квадратичной сходимостью к оптимальному решению.

6.5. Алгоритм сопряженных градиентов

В этом методе при выборе направления минимизации не используется информация о гессиане [54]. Направление поиска до выбирается таким образом, чтобы оно было ортогональным и сопряженным ко всем предыдущим направлениям $p_0, p_1, ..., p_{k-1}$. Множество векторов p_i, $i = 0,1,..,k$, будет взаимно сопряженным относительно матрицы $\mathbf{G}$, если

$p_i^T \mathbf{G} p_j = 0$, $i \neq j$.

Как показано в [70, 74], вектор p_k, удовлетворяющий заданным выше условиям, имеет вид:

$$p_k = -g_k + \beta_{k-1} p_{k-1}, \qquad (54)$$

где $g_k = g(w_k)$ обозначает фактическое значение вектора градиента.

Из формулы (54) следует, что новое направление минимизации зависит только от значения градиента в точке решения w_k и от предыдущего направления поиска p_{k-1} умноженного на коэффициент сопряжения β_{k-1}. Этот коэффициент играет очень важную роль, аккумулируя в себе информацию о предыдущих направлениях поиска. Существуют различные правила расчета его значения. Наиболее известны среди них [70, 71]

$$\beta_{k-1} = \frac{g_k^T (g_k - g_{k-1})}{g_{k-1}^T g_{k-1}}$$

и

$$\beta_{k-1} = \frac{g_k^T (g_k - g_{k-1})}{-p_{k-1} g_{k-1}}.$$

Ввиду накопления погрешностей округления в последовательных циклах вычислений практическое применение метода сопряженных градиентов связано с постепенной утратой свойства ортогональности между векторами направлений минимизации. По этой причине, после выполнения n итераций (значение n рассчитывается как функция от количества переменных, подлежащих оптимизации) производится рестарт процедуры, на первом шаге которой направление минимизации из точки полученного решения выбирается по алгоритму наискорейшего спуска. Метод сопряженных градиентов имеет сходимость, близкую к линейной, и он менее эффективен, чем метод переменной метрики, однако заметно быстрее метода наискорейшего спуска. Он широко применяется как единственно эффективный алгоритм оптимизации при весьма значительном количестве переменных, которое может достигать нескольких десятков тысяч. Благодаря невысоким требованиям к памяти и относительно низкой вычислительной сложности метод сопряженных градиентов позволяет успешно решать очень серьезные оптимизационные задачи.

ГЛАВА 7. ОБОБЩАЮЩИЕ СВОЙСТВА НЕЙРОННЫХ СЕТЕЙ

7.1. Способность к обобщению

Одно из важнейших свойств нейронной сети – это способность к обобщению полученных знаний [54]. Сеть, натренированная на некотором множестве обучающих выборок, генерирует ожидаемые результаты при подаче на ее вход данных, относящихся к тому же множеству, но не участвовавших непосредственно в процессе обучения. Разделение данных на обучающее и тестовое подмножества представлено на рис. 33. Множество данных, на котором считается истинным некоторое правило R, разбито на подмножества Z, и G.

При этом в составе L, в свою очередь, можно выделить определенное подмножество контрольных данных V, используемых для верификации степени обучения сети. Обучение проводится на данных, составляющих подмножество L. Способность отображения сетью элементов L может считаться показателем степени накопления обучающих данных, тогда как способность распознавания данных, входящих во множество G, и не использованных для обучения, характеризует ее возможности обобщения (генерализации) знаний. Данные, входящие и в L, и в G, должны быть типичными элементами множества R. В обучающем подмножестве не должно быть уникальных данных, свойства которых отличаются от ожидаемых типичных значений.

Феномен обобщения возникает вследствие большого количества комбинаций входных данных, которые могут кодироваться в сети с N входами. Если в качестве простого примера рассмотреть однослойную сеть с одним выходным нейроном, то для нее может быть составлено 2^N входных выборок. Каждой выборке может соответствовать единичное или нулевое состояние выходного

нейрона. Таким образом, общее количество различаемых сигналов составит 2^N. Если для обучения сети используются p из общего числа 2^N входных выборок, то оставшиеся иезадействованными $(2^N - p)$ допустимых комбинаций характеризуют потенциально возможный уровень обобщения знаний.

Подбор весов сети в процессе обучения имеет целью найти такую комбинацию их значений, которая наилучшим образом воспроизводила бы последовательность ожидаемых обучающих пар (x_i, d_i). При этом наблюдается тесная связь между количеством весов сети (числом степеней свободы) и количеством обучающих выборок. Если бы целью обучения было только запоминание обучающих выборок, их количество могло быть равным числу весов. В таком случае каждый вес соответствовал бы единственной обучающей паре. К сожалению, такая сеть не будет обладать свойством обобщения и сможет только восстанавливать данные.

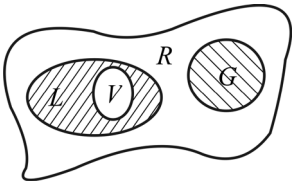


Рис. 33 – Иллюстрация разделения данных, подчиняющихся правилу R, на обучающее подмножество L, тестовое подмножество G и контрольное подмножество V [54]

Для обретения способности обобщать информацию сеть должна тренироваться на избыточном множестве данных, поскольку тогда веса будут адаптироваться не к уникальным выборкам, а к их статистически усредненным совокупностям. Следовательно, для усиления способности к обобщению необходимо не только оптимизировать структуру сети в направлении ее минимизации, но и оперировать достаточно большим объемом обучающих данных.

Обратим внимание на определенную непоследовательность процесса обучения сети. Собственно обучение ведется путем минимизации целевой функции $E(w)$ определяемой только на обучающем подмножестве L, при этом $E(w) = \sum_{k=1}^{p} E(y_k(w), d_k)$, где p обозначено количество обучающих пар (x_k, d_k), а y_k – вектор реакции сети на возбуждение x_k. Минимизация этой функции обеспечивает достаточное соответствие выходных сигналов сети ожидаемым значениям из обучающих выборок.

Истинная цель обучения состоит в таком подборе архитектуры и параметров сети, которые обеспечат минимальную погрешность распознавания тестового подмножества данных, не участвовавших в обучении. Эту погрешность будем называть погрешностью обобщения $E_G(w)$. Со статистической точки зрения погрешность обобщения зависит от уровня погрешности обучения $E_L(w)$ и от доверительного интервала ε. Она характеризуется отношением [68]:

$$E_G(w) \le E_L(w) + \varepsilon(\frac{p}{h}, E_L).$$

В работе [68] показано, что значение ε функционально зависит от уровня погрешности обучения $E_L(w)$ и от отношения количества обучающих выборок p к фактическому значению η параметра, называемого мерой Вапника-Червоненкиса и обозначаемого $VC_{\dim}$. Мера $VC_{\dim}$ отражает уровень сложности нейронной сети и тесно связана с количеством содержащихся в ней весов. Значение ε уменьшается по мере возрастания отношения количества обучающих выборок к уровню сложности сети.

По этой причине обязательным условием выработки хороших способностей к обобщению считается грамотное определение

меры Вапника-Червоненкиса для сети заданной структуры. Метод точного определения этой меры не известен, о нем можно лишь сказать, что ее значение функционально зависит от количества сииаптических весов, связывающих нейроны между собой. Чем больше количество различных весов, тем больше сложность сети и соответственно значение меры $VC_{\dim}$. В [76] предложено определять верхнюю и нижнюю границы этой меры в виде

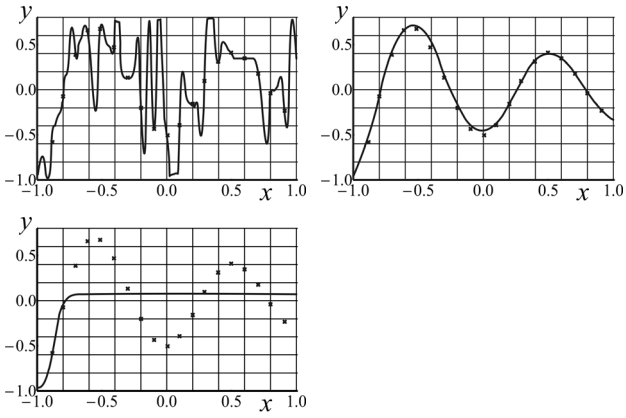
$$2\left[\frac{K}{2}\right]N \le VC_{\dim} \le 2N_w(1+\lg N_n), \qquad (55)$$

где через оператор $[\cdot]$, обозначена целая часть числа, N – размерность входного вектора, K – количество нейронов скрытого слоя, N_w – общее количество весов сети, а N_n – общее количество нейронов сети.

Из выражения (55) следует, что нижняя граница диапазона приблизительно равна количеству весов, связывающих входной и скрытый слои, тогда как верхняя граница превышает двукратное суммарное количество всех весов сети. В связи с невозможностью точного определения меры $VC_{\dim}$ в качестве ее приближенного значения используется общее количество весов нейронной сети.

Таким образом, на погрешность обобщения оказывает влияние отношение количества обучающих выборок к количеству весов сети. Небольшой объем обучающего подмножества при фиксированном количестве весов вызывает хорошую адаптацию сети к его элементам, однако не усиливает способности к обобщению, так как в процессе обучения наблюдается относительное превышение числа подбираемых параметров (весов) над количеством пар фактических и ожидаемых выходных сигналов сети. Эти параметры адаптируются с чрезмерной (а вследствие превышения числа параметров над объемом обучающего множества – и неконтролируемой) точностью к значениям конкретных выборок, а не к диапазонам, которые эти

выборки должны представлять. Фактически задача аппроксимации подменяется в этом случае задачей приближенной интерполяции. В результате всякого рода нерегулярности обучающих данных и измерительные шумы могут восприниматься как существенные свойства процесса. Функция, воспроизводимая в точках обучения, будет хорошо восстанавливаться только при соответствующих этим точкам значениях. Даже минимальное отклонение от этих точек вызовет значительное увеличение погрешности, что будет восприниматься как ошибочное обобщение. По результатам разнообразных численных экспериментов установлено, что высокие показатели обобщения достигаются в случае, когда количество обучающих выборок в несколько раз превышает меру $VC_{\dim}$ [75].



а) *б*) *в*)

Рис. 34 – Графическая иллюстрация способности сети к обобщению на примере аппроксимации одномерной функции: *а*) слишком большое количество скрытых нейронов; *б*) правильно подобранное количество нейронов; *в*) слишком малое количество нейронов [54]

На рис. 34,*а* представлена графическая иллюстрация эффекта гиперразмерности сети (слишком большого количества нейронов и весов). Аппроксимирующая сеть, скрытый слой которой состоит из 80 нейронов, на основе интерполяции в 21-й точке адаптировала свои выходные сигналы с нулевой погрешностью обучения. Минимизация этой погрешности на слишком малом (относительно количества весов) количестве обучающих выборок, спровоцировала случайный характер значений многих весов, что при переходе от обучающих выборок к тестовым, стало причиной значительных отклонений фактических значений y от ожидаемых значений d. Уменьшение количества скрытых нейронов до 5 при неизменном объеме обучающего множества позволило обеспечить и малую погрешность обучения, и высокий уровень обобщения (рис. 34,*б*). Дальнейшее уменьшение количества скрытых нейронов может привести к потере сетью способности восстанавливать обучающие данные (т.е. к слишком большой погрешности обучения $E_L(w)$). Подобная ситуация иллюстрируется на рис. 34,*в*, где задействован только один скрытый нейрон.

Сеть оказалась не в состоянии корректно воспроизвести обучающие данные, поскольку количество ее степеней свободы слишком мало по сравнению с необходимым для такого воспроизведения. Очевидно, что в этом случае невозможно достичь требуемого уровня обобщения, поскольку он явно зависит от погрешности обучения $E_L(w)$. На практике, подбор количества скрытых нейронов (и связанный с ним подбор количества весов) может, в частности, выполняться путем тренинга нескольких сетей с последующим выбором той из них, которая содержит наименьшее количество скрытых нейронов при допустимой погрешности обучения.

7.2. Выбор схемы сети для наилучшего обобщения

Решение по выбору окончательной схемы сети может быть принято только после полноценного обучения (с уменьшением погрешности до уровня, признаваемого удовлетворительным) различных вариантов ее структуры [54]. Однако нет никакой уверенности в том, что этот выбор будет оптимальным, поскольку тренируемые сети могут отличаться различной чувствительностью к подбору начальных значений весов и параметров обучения. По этой причине базу для редукции сети составляют алгоритмы отсечения взвешенных связей либо исключения нейронов в процессе обучения или после его завершения.

Как правило, методы непосредственного отсечения связей, основанные на временном присвоении им нулевых значений, с принятием решения о возобновлении их обучения по результатам наблюдаемых изменений величины целевой функции (если это изменение слишком велико, следует восстановить отсеченную связь), оказываются неприменимыми из-за слишком высокой вычислительной сложности. Большинство применяемых в настоящее время алгоритмов редукции сети можно разбить на две категории. Методы первой группы исследуют чувствительность целевой функции к удалению веса или нейрона.

С их помощью устраняются веса с наименее заметным влиянием, оказывающие минимальное воздействие на величину целевой функции, и процесс обучения продолжается уже на редуцированной сети.

Методы второй группы связаны с модификацией целевой функции, в которую вводятся компоненты, штрафующие за неэффективную структуру сети. Чаще всего это бывают элементы, усиливающие малые значения амплитуды весов. Такой способ менее эффективен по сравнению с методами первой группы, поскольку малые значения весов не обязательно ослабляют их влияние на функционирование сети.

Принципиально иной подход состоит в начале обучения при минимальном (обычно нулевом) количестве скрытых нейронов и последовательном их добавлении, вплоть до достижения требуемого уровня натренированности сети на исходном множестве обучающих выборок. Добавление нейронов, как правило, производится по результатам оценивания способности сети к обобщению, после определенного количества циклов обучения. В частности, именно такой прием реализован в алгоритме каскадной корреляции Фальмана.

При обсуждении способности сети к обобщению невозможно обойти вниманием влияние на ее уровень длительности обучения. Численные эксперименты показали, что погрешность обучения при увеличении количества итераций монотонно уменьшается, тогда как погрешность обобщения снижается только до определенного момента, после чего начинает расти. Типичная динамика этих показателей представлена на рис. 35, где погрешность обучения E_L означена сплошной, а погрешность обобщения E_G – пунктирной линией.

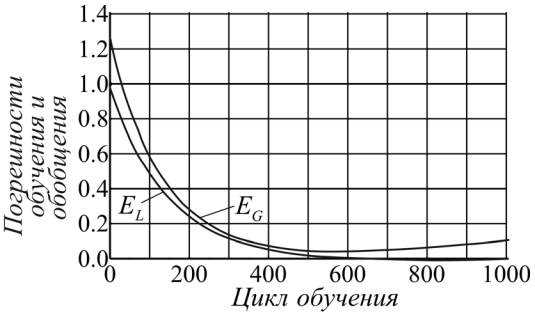


Рис. 35 – Иллюстрация влияния длительности обучения на погрешность обучения E_L и на погрешность тестирования (обобщения) E_G [54]

Приведенный график однозначно свидетельствует, что слишком долгое обучение может привести к "переобучению" сети, которое выражается в слишком детальной адаптации весов к несущественным флуктуациям обучающих данных.

Такая ситуация имеет место при использовании сети с чрезмерным (по сравнению с необходимым) количеством весов, и она тем более заметна, чем больше "лишних" весов содержит сеть. Излишние веса адаптируются к любым нерегулярностям обучающих данных, воспринимая их в качестве важных характеристик. Как следствие, на этапе тестирования они становятся причиной возникновения значительных погрешностей воспроизведения.

Для предупреждения переобучения в обучающем множестве выделяется область контрольных данных (подмножество V на рис. 33), которые в процессе обучения применяются для оперативной проверки фактически набранного уровня обобщения.

Обучение прекращается, когда погрешность обобщения на этом подмножестве достигнет минимального значения (или начнет возрастать).

7.3. Достаточный объем примеров обучения для корректного обобщения

Нейронная сеть, спроектированная с учетом хорошего обобщения, будет осуществлять корректное отображение входа на выход даже тогда, когда входной сигнал слегка отличается от примеров, использованных для обучения сети [52]. Однако, если сеть обучается на слишком большом количестве примеров, все может закончиться только запоминанием данных обучения. Это может произойти за счет нахождения таких признаков (например, благодаря шуму), которые присутствуют в примерах обучения, но не свойственны самой моделируемой функции отображения. Такое

явление называют *избыточным обучением,* или *переобучением*. Если сеть "переучена", она теряет свою способность к обобщению на аналогичных входных сигналах.

Способность к обобщению определяется тремя факторами: размером обучающего множества и его представительностью, архитектурой нейронной сети и физической сложностью рассматриваемой задачи. Естественно, последний фактор выходит за пределы нашего влияния. В контексте остальных факторов вопрос обобщения можно рассматривать с двух различных точек зрения [75]:

1. Архитектура сети фиксирована (получена в соответствии с физической сложностью рассматриваемой задачи), и вопрос сводится к определению размера обучающего множества, *необходимого* для хорошего обобщения.

2. Размер обучающего множества фиксирован, и вопрос сводится к определению наилучшей архитектуры сети, позволяющей достичь хорошего обобщения.

Обе точки зрения по-своему правильны. Упоминавшаяся ранее величина $VC_{\dim}$ обеспечивает теоретический базис адекватности размера обучающей выборки. К сожалению, часто оказывается, что между действительно достаточным размером множества обучения и этими оценками может существовать большой разрыв. Из-за этого расхождения возникает *задача сложности выборки*, открывающая новую область исследований.

На практике оказывается, что для хорошего обобщения достаточно, чтобы размер обучающего множества N удовлетворял следующему соотношению:

$$N = O(W / \varepsilon), \qquad (56)$$

где W – общее количество свободных параметров (т.е. синаптических весов и порогов) сети, ε – допустимая точность

ошибки классификации, $O(\cdot)$ – порядок заключенной в скобки величины. Например, для ошибки в 10% количество примеров обучения должно в 10 раз превосходить количество свободных параметров сети.

Выражение (56) получено из *эмпирического правила Видроу*, утверждающего, что время стабилизации процесса линейной адаптивной временной фильтрации примерно равно *объему памяти* линейного адаптивного фильтра в задаче фильтра на *линии задержки с отводами*, деленному на величину *рассогласования* [75, 76]. Рассогласование выступает в роли ошибки ε из выражения (56).

ГЛАВА8. ОПТИМИЗАЦИЯ ПРОГНОЗИРУЮЩИХ НС И ПРАКТИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИХ ПОСТРОЕНИЮ

8.1. Алгоритмы сокращения числа нейронов в скрытых слоях НС в процессе обучения

Для того, чтобы многослойная нейронная сеть реализовывала заданное обучающей выборкой отображение, она должна иметь достаточное число нейронов в скрытых слоях [53]. В настоящее время, нет формул для точного определения необходимого числа нейронов в сети, по заданной обучающей выборке. Однако, предложены способы настройки числа нейронов, в процессе обучения, которые обеспечивают построение нейронной сети для решения задачи и дают возможность избежать избыточности. Эти способы настройки можно разделить на две группы *алгоритмы сокращения* и *конструктивные алгоритмы*

В начале работы алгоритма обучения с сокращением, число нейронов в скрытых слоях сети заведомо избыточно. Затем из НС постепенно удаляются синапсы и нейроны. Существуют два подхода к реализации алгоритмов сокращения:

1) *Метод штрафных функций.* В целевую функцию алгоритма обучения вводится штрафы за то, что значения синаптических весов отличны от нуля, например:

$$C = \sum_{i=1}^{L} \sum_{j=1}^{n} w_{ij}^2 ,$$

где w_{ij} – синаптический вес, i – номер нейрона, j – номер входа, L – число нейронов скрытого слоя, n – размерность входного сигнала.

2) *Метод проекций.* Синаптический вес обнуляется, если его значение попало в заданный диапазон

$$w_{ij} = \begin{cases} 0, & |w_{ij}| \le \varepsilon, \\ w_{ij}, & |w_{ij}| > \varepsilon, \end{cases}$$

где ε – некоторая константа.

Алгоритмы сокращения имеют, по крайней мере, два недостатка:

• нет методики определения числа нейронов скрытых слоев, которое является избыточным, поэтому перед началом работы алгоритма нужно угадать это число;

• в процессе работы алгоритма сеть содержит избыточное число нейронов, поэтому обучение идет медленно.

8.2. Конструктивные алгоритмы увеличения числа нейронов НС и их расщепление в процессе обучения

Предшественником конструктивных алгоритмов можно считать методику обучения многослойных НС, включающую в себя следующие шаги [53]:

1. Выбор начального числа нейронов в скрытых слоях.
2. Инициализация сети, заключающаяся в присваивании синаптическим весам и смещениям сети случайных значений из заданного диапазона.
3. Обучение сети по заданной выборке.
4. Завершение в случае успешного обучения. Если сеть обучить не удалось, то число нейронов увеличивается и повторяются шаги со второго по четвертый.

В конструктивных алгоритмах число нейронов в скрытых слоях также изначально мало и постепенно увеличивается В отличие от этой методики, в конструктивных алгоритмах сохраняются навыки, приобретенные сетью до увеличения числа нейронов.

Конструктивные алгоритмы различаются правилами задания значений параметров в новых, добавленных в сеть, нейронах:

• значения параметров являются случайными числами из заданного диапазона;

• значения синаптических весов нового нейрона определяются путем *расщепления* одного из старых нейронов.

Первое правило не требует значительных вычислений, однако его использование приводит к некоторому увеличению значения функции ошибки после каждого добавления нового нейрона. В результате случайного задания значений параметров новых нейронов, может появиться избыточность в числе нейронов скрытого слоя. Расщепление нейронов лишено обоих указанных недостатков. Суть алгоритма расщепления проиллюстрирована на рис. 36.

На этом рисунке показан вектор весов нейрона скрытого слоя на некотором шаге обучения и векторы изменения весов, соответствующие отдельным обучающим примерам. Векторы изменений имеют два преимущественных направления и образуют в пространстве область, существенно отличающуюся от сферической.

Суть алгоритма заключается в выявлении и расщеплении таких нейронов. В результате расщепления, вместо одного, исходного в сети, оказывается два нейрона. Первый из этих нейронов имеет вектор весов, представляющий из себя сумму вектора весов исходного нейрона и векторов изменений весов одного из преимущественных направлений. В результате суммирования векторов, изменений весов другого преимущественного

направления и вектора весов исходного нейрона получают синаптические веса второго нового нейрона.

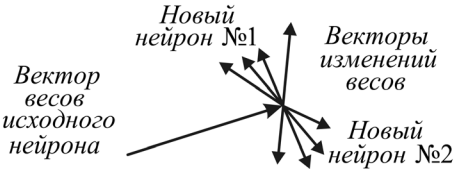


Рис. 36 – Вектор весов нейрона скрытого слоя и изменения, соответствующие отдельным обучающим примерам [53]

Расщеплять нейроны, векторы изменений которых имеют два преимущественных направления, необходимо потому, что наличие таких нейронов приводит к осцилляциям при обучении методом обратного распространения. При обучении методом с интегральной функцией ошибки наличие таких нейронов приводит к попаданию в локальный минимум с большим значением ошибки.

Алгоритм расщепления включает в себя:

• построение ковариационной матрицы векторов изменений синаптических весов;

• вычисление собственных векторов и собственных значений полученной матрицы с помощью итерационного алгоритма Ойа (Oja), в соответствии с которым выполняется стохастический градиентный подъем и ортогонализация Грамма-Шмидта.

Недостатком алгоритма является экспоненциальный рост времени вычислений при увеличении размерности сети.

Для преодоления указанного недостатка предложен *упрощенный алгоритм расщепления,* который не требует значительных вычислений. Общее число нейронов в сети, построенной с помощью этого алгоритма по заданной обучающей выборке,

может быть несколько больше, чем у сети, построенной с помощью исходного алгоритма.

В упрощенном алгоритме для расщепления выбирается нейрон с наибольшим значением функционала:

$$F_i = \frac{\sum_{k=1}^{N} \left| \delta w_i^k \right|}{\left| \sum_{k=1}^{N} \delta w_i^k \right|},$$

где δw – вектор изменений синаптических весов нейрона, i – номер нейрона, $i = \overline{1, L}$, L – число нейронов, k – номер обучающего примера; N – число примеров в обучающей выборке.

Таким образом, в качестве критерия выбора нейрона для расщепления используется отношение суммы длин векторов изменений синаптических весов нейрона, соответствующих различным обучающим примерам, к длине суммы этих векторов.

В результате расщепления вместо исходного нейрона в сеть вводятся два новых нейрона. Значение каждого синаптического веса нового нейрона есть значение соответствующего веса старого нейрона плюс некоторый небольшой шум. Величины весов связей выходов новых нейронов и нейронов следующего слоя равны половине величин весов связей исходного нейрона с соответствующими весами следующего слоя. Упрощенный алгоритм, как и исходный, гарантирует, что функция ошибки после расщепления увеличиваться не будет.

Кроме описанных способов выбора нейронов для расщепления, может быть использован анализ чувствительности, в процессе которого строятся матрицы Гессе для вторых производных функции ошибки по параметрам сети. По величине модуля второй производной судят о важности значения данного параметра для решения задачи. Параметры с малыми значениями вторых

производных обнуляют. Анализ чувствительности требует больших вычислительных ресурсов.

8.3. Выбор коэффициентов функции активации

Многослойный персептрон, обучаемый по алгоритму обратного распространения, может в принципе обучаться быстрее (в терминах требуемого для обучения количества итераций), если сигмоидальная функция активации нейронов сети является антисимметричной, а не симметричной [52]. Функция активации $\varphi(v)$ называется *антисимметричной* (т.е. четной функцией своего аргумента), если

$$\varphi(-v) = -\varphi(v)$$.

Стандартная логистическая функция не удовлетворяет этому условию. Известным примером антисимметричной функции активации является сигмоидальная нелинейная функция *гиперболического тангенса*

$$\varphi(v) = a\tanh(bv)$$,

где a и b – константы. Удобными значениями для данных констант являются следующие [78, 79]:

$a = 1,7159$, $b = 2/3$.

Определенная таким образом функция гиперболического тангенса имеет ряд полезных свойств.

- $\varphi(1) = 1$ и $\varphi(-1) = -1$.
- В начале координат тангенс угла наклона (т.е. эффективный угол) функции активации близок к единице:

$\varphi(0) = ab = 1.1424$.

- Вторая производная $\varphi(v)$ достигает своего максимального значения при $v = 1$.

8.4. Емкость нейронной сети

Рассмотрим вопрос о емкости нейронной сети, т. е. числа образов, предъявляемых на ее входы, которые она способна научиться распознавать [53]. Для сетей с числом слоев больше двух, этот вопрос остается открытым. Для сетей с двумя слоями, детерминистская емкость сети C_d оценивается следующим образом:

$$\frac{L_w}{m} < C_d < \frac{L_w}{m} \log\left(\frac{L_w}{m}\right),$$

где L_w – число подстраиваемых весов, m – число нейронов в выходном слое.

Данное выражение получено с учетом некоторых ограничений. Во-первых, число входов n и нейронов в скрытом слое L должно удовлетворять неравенству $(n + L) > m$. Во-вторых, $L_w / m > 1000$. Однако приведенная оценка выполнена для сетей с пороговыми активационными функциями нейронов, а емкость сетей с гладкими активационными функциями, обычно больше. Кроме того, термин детерминистский означает, что полученная оценка емкости подходит для всех входных образов, которые могут быть представлены n входами. В действительности распределение входных образов, как правило, обладает некоторой регулярностью, что позволяет нейронной сети проводить обобщение и, таким образом, увеличивать реальную емкость. Так как распределение образов, в общем случае, заранее не известно, можно говорить о реальной емкости только предположительно, но обычно она раза в два превышает детерминистскую емкость.

Вопрос о емкости нейронной сети тесно связан с вопросом о требуемой мощности выходного слоя сети, выполняющего окончательную классификацию образов. Например, для разделения множества входных образов по двум классам достаточно одного выходного нейрона. При этом каждый логический уровень («1» и «0») будет обозначать отдельный класс. На двух выходных нейронах с пороговой функцией активации можно закодировать уже четыре класса. Для повышения достоверности классификации желательно ввести избыточность путем выделения каждому классу одного нейрона в выходном слое или, что еще лучше, нескольких, каждый из которых обучается определять принадлежность образа к классу со своей степенью достоверности, например: высокой, средней и низкой. Такие нейронные сети позволяют проводить классификацию входных образов, объединенных в нечеткие (размытые или пересекающиеся) множества. Это свойство приближает подобные сети к реальным условиям функционирования биологических нейронных сетей.

Рассматриваемая НС имеет несколько «узких мест». Во-первых, в процессе большие положительные или отрицательные значения весов могут сместить рабочую точку на сигмоидах нейронов в область насыщения. Малые величины производной от логистической функции приведут к остановке обучения, что парализует сеть. Во-вторых, применение метода градиентного спуска не гарантирует нахождения глобального минимума целевой функции. Это тесно связано вопросом выбора скорости обучения. Приращения весов и, следовательно, скорость обучения для нахождения экстремума должны быть бесконечно малыми, однако в этом случае обучение будет происходить неприемлемо медленно. С другой стороны, слишком большие коррекции весов могут привести к постоянной неустойчивости процесса обучения. Поэтому в качестве коэффициента скорости обучения η обычно выбирается число меньше 1 (например, 0.1), которое постепенно

уменьшается в процессе обучения. Кроме того, для исключения случайных попаданий сети в локальные минимумы иногда, после стабилизации значений весовых коэффициентов, η кратковременно значительно увеличивают, чтобы начать градиентный спуск из новой точки. Если повторение этой процедуры несколько раз приведет сеть в одно и то же состояние, можно предположить, что найден глобальный минимум.

Существует другой метод исключения локальных минимумов и паралича сети, заключающийся в применении стохастических нейронных сетей.

Дадим изложенному геометрическую интерпретацию.

В алгоритме обратного распространения вычисляется вектор градиента поверхности ошибок. Этот вектор указывает направление кратчайшего спуска по поверхности из текущей точки, движение по которому приводит к уменьшению ошибки. Последовательность уменьшающихся шагов приведет к минимуму того или иного типа. Трудность здесь представляет вопрос подбора длины шагов.

При большой величине шага сходимость будет более быстрой, но имеется опасность перепрыгнуть через решение или в случае сложной формы поверхности ошибок уйти в неправильном направлении, например, продвигаясь по узкому оврагу с крутыми склонами, прыгая с одной его стороны на другую. Напротив, при небольшом шаге и верном направлении потребуется очень много итераций. На практике величина шага берется пропорциональной крутизне склона, так что алгоритм замедляет ход вблизи минимума. Правильный выбор скорости обучения зависит от конкретной задачи и обычно делается опытным путем. Эта константа может также зависеть от времени, уменьшаясь по мере продвижения алгоритма.

Обычно этот алгоритм видоизменяется таким образом, чтобы включать слагаемое импульса (или инерции). Это способствует продвижению в фиксированном направлении, поэтому, если было сделано несколько шагов в одном и том же направлении, то алгоритм увеличивает скорость, что иногда позволяет избежать локального минимума, а также быстрее проходить плоские участки.

На каждом шаге алгоритма на вход сети поочередно подаются все обучающие примеры, реальные выходные значения сети сравниваются с требуемыми значениями, и вычисляется ошибка. Значение ошибки, а также градиента поверхности ошибок используется для корректировки весов, после чего все действия повторяются. Процесс обучения прекращается либо когда пройдено определенное количество эпох, либо когда ошибка достигнет некоторого определенного малого уровня, либо когда ошибка перестанет уменьшаться.

Рассмотрим проблемы обобщения и переобучения нейронной сети более подробно. Обобщение – это способность нейронной сети делать точный прогноз на данных, не принадлежащих исходному обучающему множеству. Переобучение же представляет собой чрезмерно точную подгонку, которая имеет место, если алгоритм обучения работает слишком долго, а сеть слишком сложна для такой задачи или для имеющегося объема данных.

Продемонстрируем проблемы обобщения и переобучения на примере аппроксимации некоторой зависимости не нейронной сетью, а посредством полиномов, при этом суть явления будет абсолютно та же.

Графики полиномов могут иметь различную форму, причем, чем выше степень и число членов, тем более сложной может быть эта форма. Для исходных данных можно подобрать полиномиальную кривую (модель) и получить, таким образом, объяснение имеющейся зависимости. Данные могут быть зашумлены, поэтому

нельзя считать, что лучшая модель в точности проходит через все имеющиеся точки. Полином низкого порядка может лучше объяснять имеющуюся зависимость, однако, быть недостаточно гибким средством для аппроксимации данных, в то время как полином высокого порядка может оказаться чересчур гибким, но будет точно следовать данным, принимая при этом замысловатую форму, не имеющую никакого отношения к настоящей зависимости.

НС сталкиваются с такими же трудностями. Сети с большим числом весов моделируют более сложные функции и, следовательно, склонны к переобучению. Сети же с небольшим числом весов могут оказаться недостаточно гибкими, чтобы смоделировать имеющиеся зависимости. Например, сеть без скрытых слоев моделирует лишь обычную линейную функцию.

8.5. Выбор правильной степени сложности сети

Почти всегда более сложная сеть дает меньшую ошибку, но это может свидетельствовать не о хорошем качестве модели, а о переобучении сети [53].

Выход состоит в использовании контрольной *кросс-проверки.* Для этого резервируется часть обучающей выборки, которая используется не для обучения сети по алгоритму обратного распространения ошибки, а для независимого контроля результата в ходе алгоритма. В начале работы ошибка сети на обучающем и контрольном множествах будет одинаковой. По мере обучения сети ошибка обучения убывает, как и ошибка на контрольном множестве. Если же контрольная ошибка перестала убывать или даже стала расти, это указывает на то, что сеть начала слишком близко аппроксимировать данные (переобучилась) и обучение следует остановить. Если это случилось, то следует уменьшить число скрытых элементов и/или слоев, ибо сеть является слишком

мощной для данной задачи. Если же обе ошибки (обучения и кросс-проверки) не достигнут достаточного малого уровня, то переобучения, естественно не произошло, а сеть, напротив, является недостаточно мощной для моделирования имеющейся зависимости.

Описанные проблемы приводят к тому, что при практической работе с нейронными сетями приходится экспериментировать с большим числом различных сетей, порой обучая каждую из них по несколько раз и сравнивая полученные результаты. Главным показателем качества результата является здесь контрольная ошибка. При этом, в соответствии с общесистемным принципом, из двух сетей с приблизительно равными ошибками контроля имеет смысл выбрать ту, которая проще.

Необходимость многократных экспериментов приводит к тому, что контрольное множество начинает играть ключевую роль в выборе модели и становится частью процесса обучения. Тем самым ослабляется его роль как независимого критерия качества модели. При большом числе экспериментов есть большая вероятность выбрать удачную сеть, дающую хороший результат на контрольном множестве. Однако для того чтобы придать окончательной модели должную надежность, часто (когда объем обучающих примеров это позволяет) поступают следующим образом: резервируют *тестовое* множество примеров. Итоговая модель тестируется на данных из этого множества, чтобы убедиться, что результаты, достигнутые на обучающем и контрольном множествах примеров, реальны, а не являются артефактами процесса обучения. Разумеется, для того чтобы хорошо играть свою роль, тестовое множество должно быть использовано *только один раз:* если его использовать повторно для корректировки процесса обучения, то оно фактически превратится в контрольное множество.

С целью ускорения процесса обучения сети предложены многочисленные модификации алгоритма обратного распространения ошибки, связанные с использованием различных функций ошибки, процедур определения направления и величин шага.

1) Функции ошибки:

• интегральные функции ошибки по всей совокупности обучающих примеров;

• функции ошибки целых и дробных степеней

2) Процедуры определения величины шага на каждой итерации

• дихотомия;

• инерционные соотношения;

• отжиг.

3) Процедуры определения направления шага:

• с использованием матрицы производных второго порядка (метод Ньютона);

• с использованием направлений на нескольких шагах (партан метод).

8.6. Инициализация нейронной сети

Хороший выбор начальных значений синаптических весов и пороговых значений сети может оказать неоценимую помощь в проектировании [52]. Естественно, возникает вопрос: "А что такое хорошо?" Если синаптические веса принимают большие начальные значения, то нейроны, скорее всего, достигнут режима насыщения. Если такое случится, то локальные градиенты алгоритма обратного распространения будут принимать малые

значения, что, в свою очередь, вызовет торможение процесса обучения. Если же синаптическим весам присвоить малые начальные значения, алгоритм будет очень вяло работать в окрестности начала координат поверхности ошибок. В частности, это верно для случая антисимметричной функции активации, такой как гиперболический тангенс. К сожалению, начало координат является *седловой точкой*, т.е. стационарной точкой, где образующие поверхности ошибок вдоль одной оси имеют положительный градиент, а вдоль другой – отрицательный. По этим причинам нет смысла использовать как слишком большие, так и слишком маленькие начальные значения синаптических весов. Как всегда, золотая середина находится между этими крайностями.

8.7. Методы инициализации весов

Обучение НС, даже при использовании самых эффективных алгоритмов, представляет собой трудоемкий процесс, далеко не всегда дающий ожидаемые результаты [54]. Проблемы возникают из-за нелинейных функций активации, образующих многочисленные локальные минимумы, к которым может сводиться процесс обучения. Конечно, применение продуманной стратегии поведения (например, имитации отжига, метода мультистара, генетических алгоритмов) уменьшает вероятность остановки процесса в точке локального минимума, однако платой за это становится резкое увеличение трудоемкости и длительности обучения. Кроме того, для применения названных методов необходим большой опыт в области решения сложных проблем глобальной оптимизации, особенно для правильного подбора управляющих параметров.

На результаты обучения огромное влияние оказывает подбор начальных значений весов сети. Идеальными считаются начальные значения, достаточно близкие к оптимальным. При этом удается не

только устранить задержки в точках локальных минимумов, но и значительно ускорить процесс обучения. К сожалению, не существует универсального метода подбора весов, который бы гарантировал нахождение наилучшей начальной точки для любой решаемой задачи. По этой причине в большинстве практических реализаций чаще всего применяется случайный подбор весов с равномерным распределением значений в заданном интервале.

Неправильный выбор диапазона случайных значений весов может вызвать слишком раннее насыщение нейронов, в результате которого, несмотря на продолжающееся обучение, среднеквадратичная погрешность будет оставаться практически постоянной. Явление этого типа не означает попадания в точку локального минимума, а свидетельствует о достижении седловой зоны целевой функции вследствие слишком больших начальных значений весов. При определенных обучающих сигналах в узлах суммирующих нейронов генерируются сигналы $u_t = \sum_j w_{ij} x_j$ со значениями, соответствующими глубокому насыщению сигмоидальной функции активации. При этом поляризация насыщения обратна ожидаемой (выходной сигнал нейрона равен +1 при ожидаемой величине -1 и обратно). Значение возвратного сигнала, генерируемое в методе обратного распространения, пропорционально величине производной от функции активации $\frac{\partial \varphi}{\partial x}$, в точке насыщения близко нулю. Поэтому изменения значений весов, выводящие нейрон из состояния насыщения, происходят очень медленно. Процесс обучения надолго застревает в седловой зоне. Следует обратить внимание, что в состоянии насыщения может находиться одна часть нейронов, тогда как другая часть остается в линейном диапазоне, и для них обратный обучающий сигнал принимает нормальный вид. Это означает, что связанные с такими нейронами веса уточняются нормальным образом, и процесс их обучения ведет к быстрому уменьшению погрешности. Как следствие, нейрон, остающийся в состоянии насыщения, не

участвует в отображении данных, сокращая таким образом эффективное количество нейронов в сети. В итоге процесс обучения чрезвычайно замедляется, поэтому состояние насыщения отдельных нейронов может длиться практически непрерывно вплоть до исчерпания лимита итераций.

Случайная инициализация, считающаяся единственным универсальным способом приписывания начальных значений весам сети, должна обеспечить такую стартовую точку активации нейронов, которая лежала бы достаточно далеко от зоны насыщения. Это достигается путем ограничения диапазона допустимых разыгрываемых значений. Оценки нижней и верхней границ такого диапазона, предлагаемые различными исследователями на основании многочисленных компьютерных экспериментов, отличаются в деталях, однако практически все лежат в пределах $(0,1)$.

В работе [17] предложено равномерное распределение весов, нормализованное для каждого нейрона по амплитуде $2/\sqrt{n_{in}}$, где n_{in} означает количество входов нейрона. Значения весов поляризации для нейронов скрытых слоев должны принимать случайные значения из интервала $\left[-\sqrt{n_{in}}/2, \sqrt{n_{in}}/2\right]$, а для выводных нейронов – нулевые значения.

В своих рассуждениях на тему оптимальных значений начальных весов Д. Нгуен и Б. Видроу используют кусочно-линейную аппроксимацию сигмоидальной функции активации. На этой основе они определили оптимальную длину случайного вектора весов нейронов скрытых слоев равной $\sqrt[n_{in}]{N_h}$, где N_h означает количество нейронов в скрытом слое, а n_{in} – количество входов данного нейрона. Оптимальный диапазон весов поляризации для нейронов скрытого слоя определен в пределах $\left[-\sqrt[n_{in}]{N_h}, \sqrt[n_{in}]{N_h}\right]$. Начальные веса выходных нейронов, по мнению упомянутых

авторов, не должны зависеть от топологии области допустимых значений и могут выбираться случайным образом из интервала $[-0.5, 0.5]$.

Решение представленных проблем случайной инициализации весов сети опирается либо на интуицию исследователя, либо на результаты большого количества численных экспериментов. Более детальный анализ событий, происходящих в процессе обучения, позволит точнее выявить причины замедления обучения персептронной сети, задержек в седловых зонах, а также слишком раннего завершения обучения в точках локальных минимумов, далеких от оптимального решения. Результатом такого анализа должны стать меры предупреждения этих нежелательных явлений за счет применения соответствующих процедур предварительной обработки обучающих данных для необходимой инициации как структуры сети, так и значений весов. Эти процедуры базируются либо на анализе данных с использованием конкуренции [81], подобно тому, как это происходит в сетях, самоорганизующихся на основе конкуренции, либо на использовании информации о корреляционных зависимостях обучающих данных [82].

8.8. Скорость обучения нейронной сети

Все нейроны многослойного персептрона в идеале должны обучаться с одинаковой скоростью [62]. Однако последние слои обычно имеют более высокие значения локальных градиентов, чем начальные слои сети. Исходя из этого, параметру скорости обучения η, следует назначать меньшие значения для последних слоев сети, и большие – для первых. Чтобы время обучения для всех нейронов сети было примерно одинаковым, нейроны с большим числом входов должны иметь меньшее значение параметра обучения, чем нейроны с малым количеством входов. В [78] предлагается назначать параметр скорости обучения для каждого нейрона обратно пропорционально квадратному корню из

суммы его синаптических связей. Более подробно о параметре скорости обучения речь пойдет ниже.

8.9. Подбор коэффициента скорости обучения

Алгоритмы, представленные в предыдущем подразделе, позволяют определить только направление, в котором уменьшается целевая функция, но не говорят ничего о величине шага, при котором эта функция может получить минимальное значение [54]. После выбора правильного направления p_k следует определить на нем новую точку решения w_{k+1}, в которой будет выполняться условие $E(w_k) < E(W_{k+1})$. Необходимо подобрать такое значение η_k, чтобы новое решение $w_{k+1} = w_k + \eta_k p_k$ лежало как можно ближе к минимуму функции $E(w)$ в направлении p_k. Грамотный подбор коэффициента η_k оказывает огромное влияние на сходимость алгоритма оптимизации к минимуму целевой функции. Чем сильнее величина η_k отличается от значения, при котором $E(w)$ достигает минимума в выбранном направлении p_k, тем большее количество итераций потребуется для поиска оптимального решения. Слишком малое значение η не позволяет минимизировать целевую функцию за один шаг и вызывает необходимость повторно двигаться в том же направлении. Слишком большой шаг приводит к "перепрыгиванию" через минимум функции и фактически заставляет возвращаться к нему.

Существуют различные способы подбора значения η, называемого в теории НС *коэффициентом обучения.* Простейший из них (относительно редко применяемый в настоящее время, главным образом для обучения в режиме "онлайн") основан на фиксации постоянного значения η на весь период оптимизации. Этот способ практически используется только совместно с методом

наискорейшего спуска. Он имеет низкую эффективность, поскольку значение коэффициента обучения никак не зависит от вектора фактического градиента и, следовательно, от направления p на данной итерации. Величина η подбирается, как правило, раздельно для каждого слоя сети с использованием различных эмпирических зависимостей. Один из подходов состоит в определении минимального значения коэффициента η для каждого слоя по формуле [83]

$$\eta \le \min(1/n_i),$$

где n_i – количество входов i-го нейрона в слое.

Другой более эффективный метод основан на адаптивном подборе коэффициента η учетом фактической динамики величины целевой функции в результате обучения. В соответствии с этим методом стратегия изменения значения η определяется путем сравнения суммарной погрешности ε на i-й итерации с ее предыдущим значением, причем ε рассчитывается по формуле

$$\varepsilon = \sqrt{\sum_{j=1}^{M}(y_j - d_j)^2}.$$

Для ускорения процесса обучения следует стремиться к непрерывному увеличению η при одновременном контроле прироста погрешности ε по сравнению с ее значением на предыдущем шаге. Незначительный рост этой погрешности считается допустимым.

Если погрешности на $(i-1)$ и i-й итерациях обозначить соответственно ε_{i-1} и ε_i а коэффициенты обучения на этих же итерациях η_{i-1} и η_i, то в случае $\varepsilon_i > k_w \varepsilon_{i-1}$ (k_w – коэффициент допустимого прироста погрешности) значение η должно уменьшаться в соответствии с формулой

$\eta_{i+1} = \eta_i p_d$,

где p_d – коэффициент уменьшения η. В противном случае, когда $\varepsilon_i \le k_w \varepsilon_{i-1}$, принимается

$\eta_{i+1} = \eta_i p_i$,

где p_i – коэффициент увеличения η. Несмотря на некоторое возрастание объема вычислений (необходимых для дополнительного расчета значений ε), возможно существенное ускорение процесса обучения. Например, реализация представленной стратегии в программе *MATLAB* [84] со значениями $k_w = 1.41$, $p_d = 0.7$, $p_i = 1.05$ позволила в несколько раз ускорить обучение при решении проблемы аппроксимации нелинейных функций.

Интересно проследить характер изменения коэффициента η в процессе обучения. Как правило, на начальных этапах доминирует тенденция к его увеличению, однако при достижении некоторого квазистационарного состояния величина η постепенно уменьшается, но не монотонно, а циклически возрастая и понижаясь в следующих друг за другом циклах.

Однако необходимо подчеркнуть, что адаптивный метод подбора η сильно зависит от вида целевой функции и значений коэффициентов k_w, p_d и p_i. Значения, оптимальные для функции одного вида, могут замедлять процесс обучения при использовании другой функции. Поэтому при практической реализации этого метода следует обращать внимание на механизмы контроля и управления значениями коэффициентов, подбирая их в соответствии со спецификой решаемой задачи.

Наиболее эффективный, хотя и наиболее сложный, метод подбора коэффициента обучения связан с направленной минимизацией

целевой функции в выбранном заранее направлении p_k. Необходимо так подобрать скалярное значение η_k, чтобы новое решение $w_{k+1} = w_k + \eta_k p_k$, соответствовало минимуму целевой функции в данном направлении p_k. В действительности получаемое решение w_{k+1} только с определенным приближением может считаться настоящим минимумом. Это результат компромисса между объемом вычислений и влиянием величины η_k на сходимость алгоритма.

Среди наиболее популярных способов направленной минимизации можно выделить *безградиентные* и *градиентные методы.* В *безградиентных методах* используется только информация о значениях целевой функции, а ее минимум достигается в процессе последовательного уменьшения диапазона значений вектора w. Примерами могут служить методы деления пополам, золотого сечения либо метод Фибоначчи [70, 71], различающиеся способом декомпозиции получаемых поддиапазонов.

Заслуживает внимания метод аппроксимации целевой функции $E(w)$ в предварительно выбранном направлении p_k с последующим расчетом минимума, получаемого таким образом, функции одной переменной η. Выберем для аппроксимации многочлен второго порядка вида

$$E(w) \to P_2(\eta) = a_2\eta^2 + a_1\eta + a_0, \qquad (57)$$

где a_2, a_1 и a_0 обозначены коэффициенты, определяемые в каждом цикле оптимизации. Выражение (57) – это многочлен P_2 одной скалярной переменной η. Если для расчета входящих в P_2 коэффициентов используются три произвольные точки w_1, w_2 и w_3, лежащие в направлении p_k, т.е. $w_1 = w + \eta_1 p_k$, $w_2 = w + \eta_2 p_k$ и $w_3 = w + \eta_3 p_k$ (в этом выражении w обозначено предыдущее

решение), а соответствующие этим точкам значения целевой функции $E(w)$ обозначены $E_1 = E(w_1)$, $E_2 = E(w_2)$, $E_3 = E(w_3)$, то

$$P_2(\eta_1) = E_1,\ P_2(\eta_2) = E_2,\ P_2(\eta_3) = E_3. \qquad (58)$$

Коэффициенты a_2, a_1 и a_0 многочлена P_2 рассчитываются в соответствии с системой линейных уравнений, описываемых в (58). Для определения минимума этого многочлена его производная $\frac{dP_2}{d\eta} = 2a_2\eta + a_1$ приравнивается к нулю, что позволяет получить значение η *в* виде $\eta_{\min} = \frac{-a_1}{2a_2}$. После подстановки выражений для E_1, E_2 и E_3 в формулу расчета $\eta_{\min}$ получаем:

$$\eta_{\min} = \eta_2 - \frac{1}{2}\frac{(\eta_2 - \eta_1)^2(E_2 - E_3) - (\eta_2 - \eta_3)^2(E_2 - E_1)}{(\eta_2 - \eta_1)(E_2 - E_3) - (\eta_2 - \eta_3)(E_2 - E_1)}.$$

Однако лучшим решением считается применение градиентных методов, в которых, кроме значения функции, учитывается также и ее производная вдоль направляющего вектора p_k. Они позволяют значительно ускорить достижение минимума, поскольку используют информацию о направлении уменьшения величины целевой функции. В такой ситуации применяется, как правило, аппроксимирующий многочлен третьего порядка

$$P_3(\eta) = a_3\eta^3 + a_2\eta^2 + a_1\eta + a_0.$$

Значения четырех коэффициентов a_i этого многочлена можно получить исходя из информации о величине функции и ее производной всего лишь в двух точках. Если приравнять к нулю производную многочлена относительно η то можно получить формулу для расчета $\eta_{\min}$ в виде

$$\eta_{\min} = \frac{-a_2 + \sqrt{a_2^2 - 3a_2 a_1}}{3a_3}.$$

Более подробное описание этого подхода можно найти в первоисточниках, посвященных теории оптимизации [70, 71].

ГЛАВА 9. ПОСТРОЕНИЕ НС ДЛЯ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ СТАТИСТИЧЕСКИМИ МЕТОДАМИ

9.2. Статистический анализ временных рядов

Подробное описание методов статистического анализа временных рядов выходит за рамки этой книги [55]. Начиная с пионерской работы Юла [85], центральное место в статистическом анализе временных рядов заняли линейные модели ARMA. Со временем эта область оформилась в законченную теорию с набором методов – теорию Бокса-Дженкинса (см. [86]). В этом подходе модель задается двумя компонентами, характеризующими *авторегрессию* и *скользящее среднее.* Общая формула для процесса с авторегрессией и скользящим средним порядка (p,q) имеет вид:

$$x_t = a_0 + \sum_{j=1}^{p} a_j x_{t-j} + \sum_{j=1}^{q} b_j \varepsilon_{t-j},$$

где p – порядок авторегрессии (положительное целое число), q – порядок скользящего среднего, ε_t – шум (некоррелированный временной ряд, подчиненный гауссову распределению с нулевым средним и дисперсией σ_ε^2). Коэффициенты a_j и b_j – являются параметрами модели. Если $q = 0$, то получается авторегрессионная модель $AR(p)$, а если $p = 0$, – модель скользящего среднего $MA(q)$.

Присутствие в модели ARMA авторегрессионного члена выражает то обстоятельство, что текущие значения переменной зависят от ее прошлых значений. Такие модели называются одномерными. Часто, однако, значения исследуемой целевой переменной связаны с несколькими разными временными рядами. Так будет, например, если целевая переменная – курс обмена валют, а другие участвующие переменные – процентные ставки (в каждой из двух

валют). Соответствующие методы называются *многомерными.* Общий вид уравнения многомерной модели такой:

$$x_t = a_0 + \sum_{k=1}^{N} \sum_{j=1}^{p(k)} a_j^{(k)} x_{t-\tau_j} + \varepsilon_t ,$$

где k – номер временного ряда (всего их – N). Математическая структура линейных моделей довольно проста, и расчеты по ним могут быть без особых трудностей выполнены с помощью стандартных пакетов численных методов. Следующим шагом в анализе временных рядов стала разработка моделей, способных учитывать нелинейности, присутствующие, как правило, в реальных процессах и системах. Одна из первых таких моделей была предложена Тонгом [87] и называется пороговой авторегрессионной моделью (TAR). В ней, при достижении определенных (установленных заранее) пороговых значений, происходит переключение с одной линейной AR-модели на другую. Тем самым в системе выделяется несколько режимов работы. Через θ_t, обозначим номер режима в момент $t(\theta_t = 1,2,...,r)$. Тогда одномерная AR-модель с соответствующим номером дает:

$$x_t = a_0^{(\theta_t)} + \sum_{j=1}^{p} a_j^{(\theta_t)} x_{t-j} .$$

Затем были предложены STAR-, или «гладкие» TAR-модели. Такая модель представляет собой линейную комбинацию нескольких моделей, взятых с коэффициентами, которые являются непрерывными функциями времени. Примером может служить следующее уравнение модели, в котором θ – гладкая функция, принимающая значения от 0 до 1:

$$x_t = \left(a_0 + \sum_{j=1}^{p} a_j x_{t-j} \right) \left[1 - \theta(x_{t-1}) \right] + \left(b_0 + \sum_{j=1}^{p} b_j x_{t-j} \right) \theta(x_{t-1}) + \varepsilon_t .$$

Были предложены также многочисленные другие нелинейные модели анализа временных рядов, см. [88, 89].

9.1. Задача статистического анализа временных рядов

Временной ряд – это упорядоченная последовательность вещественных чисел x_t, $t = 1, 2, ..., T$, представляющих собой результаты наблюдений некоторой величины [55]. Эти значения обычно получают как результаты измерений в некоторой физической системе. Если нас интересуют зависимости между текущими и прошлыми значениями, то нужно рассматривать вектор задержки $(x_{t-1}, x_{t-2}, ..., x_{t-n})$ в n-мерном пространстве сдвинутых во времени значений, или пространстве задержки.

Цель анализа временных рядов состоит в том, чтобы извлечь из данного ряда полезную информацию. Для этого необходимо построить математическую модель явления. Такая модель должна объяснять существо процесса, порождающего данные, в частности – описывать характер данных (случайные, имеющие тренд, периодические, стационарные и т.п.). После этого можно применять различные методы фильтрации данных (сглаживание, удаление выбросов и др.) с конечной целью – предсказать будущие значения.

Таким образом, этот подход основан на предположении, что временной ряд имеет некоторую математическую структуру (которая, например, может быть следствием физической сути явления). Эта структура существует в так называемом *фазовом пространстве,* координаты которого – это независимые переменные, описывающие состояние динамической системы. Поэтому первая задача, с которой придется столкнуться при моделировании – это подходящим образом определить фазовое

пространство. Для этого нужно выбрать некоторые характеристики системы в качестве фазовых переменных. После этого уже можно ставить вопрос о предсказании или экстраполяции. Как правило, во временных рядах, полученных в результате измерений, в разной пропорции присутствуют случайные флуктуации и шум. Поэтому качество модели во многом определяется ее способностью аппроксимировать предполагаемую структуру данных, отделяя ее от шума.

Что могут дать в этом отношении нейронные сети? В этой главе будет показано, что нейронные сети можно рассматривать как обобщение традиционных подходов к анализу временных рядов. Нейронные сети дают дополнительные возможности в моделировании нелинейных явлений и распознавании хаотического поведения. Благодаря своей большой гибкости (на одной топологии можно реализовать много различных отображений), сети могут ухватывать самые разные структуры в фазовом пространстве.

9.3. Модели, основанные на НС с прямой связью

Любопытно заметить, что все описанные в предыдущем пункте модели могут быть реализованы посредством нейронных сетей [55]. Любая зависимость вида $x_t = f\left(x_{t-1}, x_{t-2}, ..., x_{t-p}\right) + \varepsilon_t$ с непрерывной нелинейной функцией f может быть воспроизведена на многослойной сети. Пример приведен на рис. 37. Вместо того, чтобы отображать поверхность во входном (фазовом) пространстве, образованную данными, посредством одной гиперплоскости (AR), нескольких гиперплоскостей (TAR), или нескольких гиперплоскостей, гладко соединенных друг с другом (STAR), нейронная сеть может осуществить произвольное нелинейное отображение. Мы говорим это не для того, чтобы представить нейронные сети как универсальную модель в анализе

временных рядов, а просто, чтобы показать все многообразие структур, которые таким способом можно моделировать. Недавние исследования показали, что нейронные сети имеют, по сравнению с классическими моделями, более высокие потенциальные возможности при анализе сложной динамической структуры, но при этом дают лучшие результаты и на таких известных типах временных рядов, как стационарные, периодические, трендовые и некоторые другие (см. [90]). Мы согласны с мнением Куама [91], что перед окончательным формированием нейронной сети необходимо проделать моделирование на основе модульного подхода с выделением тренда и сезонных колебаний.

При этом, однако, исследования в области моделирования временных рядов при помощи сетей продолжаются и в настоящее время, и никаких стандартных методов здесь пока не выработано. В нейронной сети многочисленные факторы взаимодействуют весьма сложным образом, и успех здесь пока приносит только эвристический («кустарный») подход.

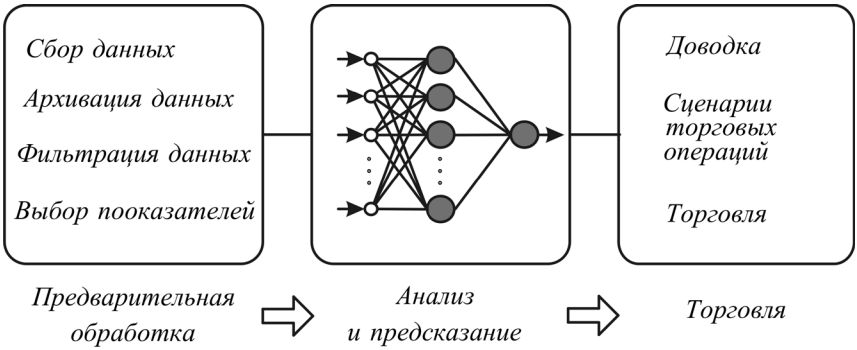


Рис. 37 – Реализация ARMA $^{(p,q)}$ модели на простейшей нейронной сети [55]

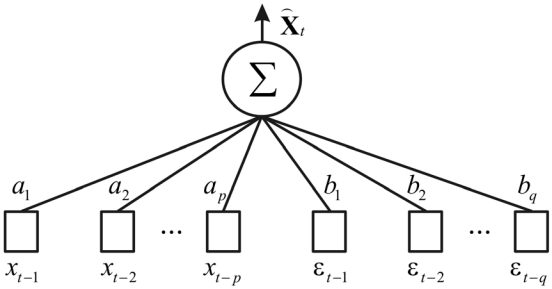


Рис. 38 – Блок-схема финансового прогнозирования при помощи нейронных сетей [55]

Типичная последовательность действий при решении задачи прогнозирования финансовых показателей с помощью нейронных сетей показана на рис. 38.

Действия на первом этапе – этапе *предварительной обработки данных,* очевидно, сильно зависят от специфики задачи. Нужно правильно выбрать число и вид показателей, характеризующих процесс, в том числе, — структуру задержек. После этого надо выбрать топологию сети. Если применяются сети с прямой связью, нужно определить число скрытых элементов. Далее, для нахождения параметров модели нужно выбрать критерий ошибки и оптимизирующий (обучающий) алгоритм. Затем, используя средства диагностики, следует проверить различные свойства модели. Наконец, нужно проинтерпретировать выходную информацию сети и, может быть, подать ее на вход какой-то другой системы поддержки принятия решений. Далее мы рассмотрим вопросы, которые приходится решать на этапах предварительной обработки, оптимизации и анализа *(доводки)* сети.

ГЛАВА 10. ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ДАННЫХ

10.1. Предварительная обработка данных

При моделировании реальных процессов «чистые» данные — это редкая роскошь [55]. В силу самой своей природы, реальные данные содержат шумы и бывают неравномерно распределены. Очень часто практик просто собирает данные и подает их на вход модели, надеясь, что все получится. Однако, при сетевом подходе (и это верно здесь даже в большей степени, чем для классического статистического анализа) тщательная *предварительная* обработка данных может сэкономить массу времени и уберечь от многих разочарований. Рассмотрим следующие относящиеся сюда вопросы: сбор данных; их анализ и очистку; их преобразование с целью сделать входную информацию более содержательной и удобной для сети.

10.2. Необходимые этапы нейросетевого анализа

На практике, именно предобработка данных может стать наиболее трудоемким элементом нейросетевого анализа [45]. Причем, знание основных принципов и приемов предобработки данных не менее, а может быть даже более важно, чем знание собственно нейросетевых алгоритмов. Последние, как правило, уже входят в состав различных нейроэмуляторов, доступных на современном программном рынке. Сам же процесс решения прикладных задач, в том числе и подготовка данных, прерогатива того, кто решает конкретную задачу. Далее кратко описаны *технологии* нейросетевого анализа.

В общем виде, технологическая цепочка, т.е. необходимые этапы нейросетевого анализа, выглядят следующим образом:

- **Кодирование входов-выходов:** нейросети могут работать только с числами.
- **Нормировка данных:** результаты нейроанализа не должны зависеть от выбора единиц измерения.
- **Предобработка данных:** удаление очевидных регулярностей из данных облегчает нейросети выявление нетривиальных закономерностей.
- **Обучение нескольких нейросетей** с различной архитектурой: результат обучения зависит как от размеров сети, так и от ее начальной конфигурации.
- **Отбор оптимальных сетей:** тех, которые дадут наименьшую ошибку предсказания на неизвестных пока данных.
- **Оценка значимости предсказаний:** оценка ошибки предсказаний не менее важна, чем само предсказанное значение.

Хотя перобработка не связана непосредственно с нейросетями, она является одним из ключевых элементов этой информационной технологии. Успех обучения нейросети может решающим образом зависеть оттого, в каком виде представлена информация для ее обучения.

10.3. Сбор данных

Самое важное решение, которое должен принять аналитик, – это выбор совокупности переменных для описания моделируемого процесса [55]. Чтобы представить себе возможные связи между разными переменными, нужно хорошо понимать существо задачи. В этой связи очень полезно будет побеседовать с опытным специалистом в данной предметной области. Относительно

выбранных переменных нужно понимать, значимы ли они сами по себе, или же в них всего лишь отражаются другие, действительно, существенные переменные. Проверка на значимость включает в себя кросс-корреляционный анализ. С его помощью можно, например, выявить временную связь типа запаздывания (лаг) между двумя рядами. То, насколько явление может быть описано линейной моделью, проверяется с помощью регрессии по методу наименьших квадратов (OLS). Полученная после оптимизации невязка R^2 может принимать значения от 0 (полное несоответствие) до 1 (точное соответствие). Часто бывает так, что для линейных систем OLS-метод дает такие результаты, которые уже нельзя сколько-нибудь значительно улучшить применением нейронных сетей.

В целом, можно сказать, что предварительная обработка через формирование совокупности переменных и проверку их значимости существенно улучшает качество модели. Если никаких теоретических методов проверки в распоряжении нет, переменные можно выбирать методом проб и ошибок, или с помощью формальных методов типа генетических алгоритмов [92, 93].

10.4. Кодирование входов-выходов

В отличие от обычных компьютеров, способных обрабатывать любую символьную информацию, нейросетевые алгоритмы работают только с числами, ибо их работа базируется на арифметических операциях умножения и сложения [43]. Именно таким образом набор синаптических весов определяет ход обработки данных.

Между тем, не всякая входная или выходная переменная в исходном виде может иметь численное выражение. Соответственно, все такие переменные следует закодировать – перевести в численную форму, прежде чем начать собственно

нейросетевую обработку. Рассмотрим, прежде всего, основной руководящий принцип, общий для всех этапов предобработки данных – максимизацию энтропии.

10.5. Максимизация энтропии как цель предобработки

Допустим, что в результате перевода всех данных в числовую форму и последующей нормировки все входные и выходные переменные отображаются в единичном кубе [43]. Задача нейросетевого моделирования - найти статистически достоверные зависимости между входными и выходными переменными. Единственным источником информации для статистического моделирования являются примеры из обучающей выборки. Чем больше бит информации принесет каждый пример – тем лучше используются имеющиеся данные.

Рассмотрим произвольную компоненту нормированных (предобработанных) данных: $\tilde{x}_i$. Среднее количество информации, приносимой каждым примером $\tilde{x}_i^\alpha$, равно энтропии распределения значений этой компоненты $H(\tilde{x}_i)$. Если эти значения сосредоточены в относительно небольшой области единичного интервала, информационное содержание такой компоненты мало. В пределе нулевой энтропии, когда все значения переменной совпадают, эта переменная не несет никакой информации.

Напротив, если значения переменной $\tilde{x}_i^\alpha$ равномерно распределены в единичном интервале, информация такой переменной максимальна.

Общий принцип предобработки данных для обучения, таким образом, состоит в максимизации энтропии входов и выходов. Этим принципом следует руководствоваться и на этапе кодирования нечисловых переменных.

10.6. Типы нечисловых переменных

Можно выделить два основных типа нечисловых переменных: *упорядоченные* (называемые также *ординальными* - от англ. *order*-порядок) и *категориальные [43].* В обоих случаях переменная относится к одному из дискретного набора классов $\{c_1, c_2, ..., c_n\}$. Но в первом случае эти классы упорядочены – их можно *ранжировать*: $c_1 > c_2 > ... > c_n$, тогда как во втором, такая упорядоченность отсутствует. В качестве примера упорядоченных переменных можно привести сравнительные категории: плохо – хорошо – отлично, или медленно – быстро. Категориальные переменные просто обозначают один из классов, являются *именами* категорий. Например, это могут быть имена людей или названия цветов: белый, синий, красный.

10.7. Кодирование ординальных переменных

Ординальные переменные более близки к числовой форме, т.к. числовой ряд также упорядочен [43]. Соответственно, для кодирования таких переменных остается лишь поставить в соответствие номерам категорий такие числовые значения, которые сохраняли бы существующую упорядоченность. Естественно, при этом имеется большая свобода выбора – любая монотонная функция от номера класса порождает свой способ кодирования. Какая же из бесконечного многообразия монотонных функций – наилучшая?

В соответствии с изложенным выше общим принципом, необходимо стремиться к тому, чтобы максимизировать энтропию закодированных данных. При использовании сигмоидальных функций активации все выходные значения лежат в конечном интервале – обычно [0,1] или [-1,1]. Из всех статистических функций распределения, определенных на конечном интервале, максимальной энтропией обладает равномерное распределение.

Применительно к данному случаю это подразумевает, что кодирование переменных числовыми значениями должно приводить, по возможности, к равномерному заполнению единичного интервала закодированными примерами. (Захватывая заодно и этап нормировки.) При таком способе "оцифровки" все примеры будут нести примерно одинаковую информационную нагрузку.

Исходя из этих соображений, можно предложить следующий практический рецепт кодирования ординальных переменных.

Единичный отрезок разбивается на n отрезков – по числу классов – с длинами, пропорциональными числу примеров каждого класса в обучающей выборке:

$$\Delta x_k = P_k / P,$$

где P_k – число примеров класса k, а P – общее число примеров.

Центр каждого такого отрезка будет являться численным значением для соответствующего ординального класса (см. рис. 39).

10.8. Кодирование категориальных переменных

В принципе, категориальные переменные также можно закодировать описанным выше способом, пронумеровав их произвольным образом [43]. Однако, такое навязывание несуществующей упорядоченности только затруднит решение задачи. Оптимальное кодирование не должно искажать структуры соотношений между классами. Если классы не упорядочены, такова же должна быть и схема кодирования.

Наиболее естественной выглядит и чаще всего используется на практике двоичное кодирование типа $"n \to n"$, когда имена n категорий кодируются значениями n бинарных нейронов, причем

первая категория кодируется как $(1,0,0,...,0)$, вторая, соответственно _ $(0,1,0,...,0)$ и т.д., вплоть до n-ной: $(0,0,0,...,1)$. Можно использовать биполярную кодировку, в которой нули заменяются на "-1". Легко убедиться, что в такой симметричной кодировке расстояния между всеми векторами-категориями равны.

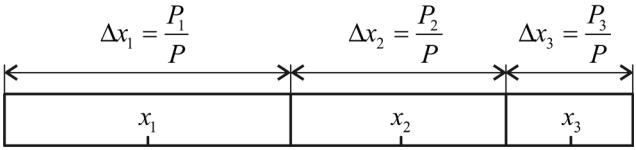


Рис. 39 – Иллюстрация способа кодирования кардинальных переменных с учетом количества примеров каждой категории [43]

Такое кодирование, однако, неоптимально в случае, когда классы представлены существенно различающимся числом примеров. В этом случае, функция распределения значений переменной крайне неоднородна, что существенно снижает информативность этой переменной. Тогда имеет смысл использовать более компактный, но симметричный код $n \to m$, когда имена n классов кодируются m-битным двоичным кодом. Причем, в новой кодировке активность кодирующих нейронов должна быть равномерна: иметь приблизительно одинаковое среднее по примерам значение активации. Это гарантирует одинаковую значимость весов, соответствующих различным нейронам.

В качестве примера рассмотрим ситуацию, когда один из четырех классов (например, класс c_x) некой категориальной переменной представлен гораздо большим числом примеров, чем остальные: $P_x \gg P_2 \sim P_3 \sim P_4$. Простое кодирование $n \to n$ привело бы к тому, что первый нейрон активировался бы гораздо чаще остальных. Соответственно, веса оставшихся нейронов имели бы меньше возможностей для обучения. Этой ситуации можно избежать,

закодировав четыре класса двумя бинарными нейронами следующим образом:

$c_1 = (0,0)$, $c_2 = (1,0)$, $c_3 = (0,1)$, $c_4 = (1,1)$,

обеспечивающим равномерную "загрузку" кодирующих нейронов.

10.9. Отличие между входными и выходными переменными

Отметим одно существенное отличие способов кодирования входных и выходных переменных, вытекающее из определения градиента ошибки [43]:

$$\frac{\partial E}{\partial w_{ij}^{[n]}} = \delta_i^{[n]} x_j^{[n]}$$

.

А именно, входы участвуют в обучении непосредственно, тогда как выходы лишь *опосредованно* – через ошибку верхнего слоя. Поэтому при кодировании категорий в качестве выходных нейронов можно использовать как логистическую функцию активации

$$\varphi(x) = \frac{1}{1+e^{-ax}},$$

определенную на отрезке [0,1], так и ее антисимметричный аналог для отрезка $[-1,1]$, например гипертангенсальную:

$$\varphi(x) = \frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}}.$$

При этом кодировка выходных переменных из обучающей выборки будет либо $\{0,1\}$, либо $\{-1,1\}$. Выбор того или иного варианта никак не скажется на обучении [43].

В случае со входными переменными дело обстоит по-другому: обучение весов нижнего слоя сети определяется *непосредственно*

значениями входов: на них умножаются невязки, зависящие от выходов. Между тем, если с точки зрения операции умножения значения ±1 равноправны, между 0 и 1 имеется существенная асимметрия: нулевые значения не дают никакого вклада в градиент ошибки. Таким образом, выбор схемы кодирования входов влияет на процесс обучения [43]. В силу логической равноправности обоих значений входов, более предпочтительной выглядит симметричная кодировка: {-1,1}, сохраняющая это равноправие в процессе обучения.

10.10. Очистка и преобразование базы данных

Стоит начать с того, чтобы изобразить распределение переменной с помощью гистограммы или же рассчитать для него характеристики асимметрии (симметричность распределения) и эксцесса (весомости «хвостов» распределения) [55]. В результате будет получена информация о том, насколько распределение данных близко к нормальному. Многие методы моделирования, в том числе, – НС, дают лучшие результаты на нормализованных данных. Далее, с помощью специальных статистических тестов, например, на расстояние Махаланобиса, можно выявить многомерные выбросы, с которыми затем нужно разобраться на предмет достоверности соответствующих данных. Эти выбросы могут порождаться ошибочными данными или крайними значениями, вследствие чего структура связей между переменными может (а может и не) нарушаться (см. [94]). В некоторых приложениях выбросы могут нести положительную информацию, и их не следует автоматически отбрасывать.

Предварительное, до подачи на вход сети, преобразование данных с помощью стандартных статистических приемов может существенно улучшить как параметры обучения (длительность, сложность), так и работу системы. Например, если входной ряд имеет отчетливый экспоненциальный вид, то после его

логарифмирования получится более простой ряд, и если в нем имеются сложные зависимости высоких порядков, обнаружить их теперь будет гораздо легче. Очень часто ненормально распределенные данные предварительно подвергают нелинейному преобразованию: исходный ряд значений переменной преобразуется некоторой функцией, и ряд, полученный на выходе, принимается за новую входную переменную. Типичные способы преобразования – возведение в степень, извлечение корня, взятие обратных величин, экспонент или логарифмов (см. [95]). Нужно проявить осторожность в отношении функций, которые определены не всюду (например, логарифм отрицательных чисел не определен). После этого могут быть применены дополнительные преобразования для изменения формы кривой регрессии. Часто это на порядок уменьшает требования к обучению [96, 97].

Для того чтобы улучшить информационную структуру данных, могут оказаться полезными определенные комбинации переменных – произведения, частные и т.д. Например, когда вы пытаетесь предсказать изменения цен акций по данным о позициях на рынке опционов, отношение числа *опционов пут* (т.е. опционов на продажу) к числу опционов колл (т.е. опционов на покупку) более информативно, чем оба этих показателя в отдельности. К тому же, с помощью таких промежуточных комбинаций часто можно получить более простую модель, что особенно важно, когда число степеней свободы ограниченно.

Наконец, для некоторых функций преобразования, реализованных в выходном узле, возникают проблемы с масштабированием. Сигмоид определен на отрезке [0,1], поэтому выходную переменную нужно масштабировать так, чтобы она принимала значения в этом интервале. Известно несколько способов масштабирования: сдвиг на константу, пропорциональное изменение значений с новым минимумом и максимумом, центрирование путем вычитания среднего значения, приведение

стандартного отклонения к единице, стандартизация (два последних действия вместе). Имеет смысл сделать так, чтобы значения всех входных и выходных величин в сети всегда лежали, например, в интервале [0,1] (или [-1,1]), – тогда можно будет без риска использовать любые функции преобразования.

Еще одна важная проблема (которая одновременно является основным преимуществом нейронно-сетевых методов) – способность работать с данными качественного характера. Отношения эквивалентности или порядка нужно суметь записать для входа (или выхода) сети. Это можно сделать, вводя искусственные переменные, принимающие значения 1 или 0.

10.11. Нормировка данных

Как входами, так и выходами нейросети могут быть совершенно разнородные величины [43]. Очевидно, что результаты нейросетевого моделирования не должны зависеть от единиц измерения этих величин. А именно, чтобы сеть трактовала их значения единообразно, все входные и выходные величины должны быть приведены к единому – единичному – масштабу. Кроме того, для повышения скорости и качества обучения полезно провести дополнительную предобработку данных, выравнивающую распределение значений еще до этапа обучения.

Приведение данных к единичному масштабу обеспечивается нормировкой каждой переменной на диапазон разброса ее значений. В простейшем варианте, это линейное преобразование вида:

$$\tilde{x}_i = \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}}$$

в единичный отрезок: $\tilde{x}_i \in [0,1]$. Обобщение для отображения данных в интервал $[-1,1]$, рекомендуемого для входных данных тривиально.

Линейная нормировка оптимальна, когда значения переменной x_i плотно заполняют определенный интервал. Но подобный "прямолинейный" подход применим далеко не всегда. Так, если в данных имеются относительно редкие выбросы, намного превышающие типичный разброс, именно эти выбросы определят согласно предыдущей формуле масштаб нормировки. Это приведет к тому, что основная масса значений нормированной переменной $\tilde{x}_i$ сосредоточится вблизи нуля: $|\tilde{x}_i| \ll 1$.

Гораздо надежнее, поэтому, ориентироваться при нормировке не на экстремальные значения, а на типичные, т.е. статистические характеристики данных, такие как среднее и дисперсия:

$$\tilde{x}_i = \frac{x_i - \overline{x}_i}{\sigma_i}, \quad \overline{x}_i \equiv \frac{1}{P}\sum_{\alpha=1}^{P} x_i^{\alpha}, \quad \sigma_i^2 \equiv \frac{1}{P-1}\sum_{\alpha=1}^{P}\left(x_i^{\alpha} - \overline{x}_i\right)^2.$$

В этом случае основная масса данных будет иметь единичный масштаб, т.е. типичные значения всех переменных будут сравнимы, как показано на рис. 40.

Однако, теперь, нормированные величины не принадлежат гарантированно единичному интервалу. Более того, максимальный разброс значений $\tilde{x}_i$ заранее не известен.

Для входных данных это может быть и не важно, но выходные переменные будут использоваться в качестве эталонов для выходных нейронов. В случае, если выходные нейроны – сигмоидальные, они могут принимать значения лишь в единичном диапазоне. Чтобы установить соответствие между обучающей выборкой и НС, в данном случае необходимо ограничить диапазон изменения переменных.

Линейное преобразование, как было показано [43], неспособно отнормировать основную массу данных и одновременно ограничить диапазон возможных значений этих данных. Естественный выход из этой ситуации - использовать для предобработки данных функцию активации тех же нейронов. Например, нелинейное преобразование

$$\tilde{x}_i = \varphi\left(\frac{x_i - \overline{x}_i}{\sigma_i}\right), \quad \varphi(x) = \frac{1}{1+e^{-ax}}$$

нормирует основную массу данных одновременно гарантируя, что $\tilde{x}_i \in [0,1]$, как показано на рис. 41)

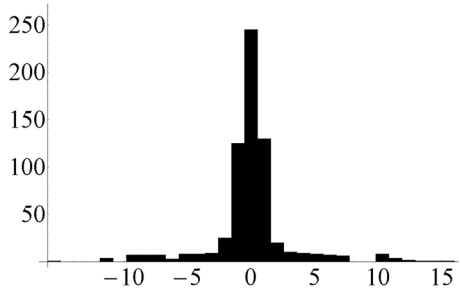


Рис. 40 – Гистограмма значений переменной при наличии редких, но больших по амплитуде, отклонений от среднего [43]

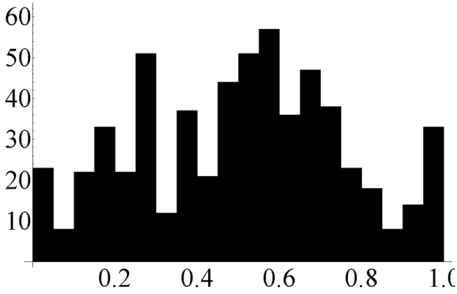


Рис. 41 – Нелинейная нормировка, использующая логистическую функцию активации [43]

Как видно из приведенного выше рисунка, распределение значений после такого нелинейного преобразования гораздо ближе к равномерному.

Однако, максимизируя *совместную* энтропию каждого входа (выхода) можно добиться гораздо большего. Рассмотрим эту технику на примере совместной нормировки входов, подразумевая, что с таким же успехом ее можно применять и для выходов а также для всей совокупности входов-выходов.

10.12. Совместная нормировка и выбеливание входов

Если два входа статистически независимы, то и совместная энтропия меньше суммы индивидуальных энтропии [43]:

$$H(\tilde{x}_i\tilde{x}_j) \le H(\tilde{x}_i) + H(\tilde{x}_j)\,.$$

Поэтому, добившись статистической независимости входов, мы, тем самым, повысим информационную насыщенность входной информации. Это, однако, потребует более сложной процедуры совместной нормировки входов.

Вместо того, чтобы использовать для нормировки индивидуальные дисперсии, будем рассматривать входные данные в совокупности. Мы хотим найти такое линейное преобразование, которое максимизировало бы их совместную энтропию. Для упрощения задачи вместо более сложного условия статистической независимости потребуем, чтобы новые входы после такого преобразования были декоррелированы. Для этого рассчитаем средний вектор и ковариационную матрицу данных по формулам:

$$\overline{\mathbf{x}} \equiv \frac{1}{P}\sum_{\alpha=1}^{P}\mathbf{x}^{\alpha}, \qquad \sum_{ij}^{X} \equiv \frac{1}{P-1}\sum_{\alpha=1}^{P}\left(x_i^{\alpha} - \overline{x}_i\right)\left(x_j^{\alpha} - \overline{x}_j\right).$$

Затем найдем линейное преобразование, диагонализующее ковариационную матрицу.

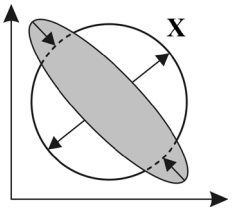


Рис. 42 – Выбеливание входной информации: повышение информативности входов за счет выравнивания функции распределения [43]

Соответствующая матрица составлена из столбцов – собственных векторов ковариационной матрицы:

$$\sum_j \sum_{ij}^{X} U_{jk} = \lambda_k U_{ik}$$

Легко убедиться, что линейное преобразование, называемое *выбеливанием*

$$\tilde{x}_i = (x_k - \overline{x}_k) U_{ki} / \sqrt{\lambda_i} ,$$

превратит все входы в некоррелированные величины с нулевым средним и единичной дисперсией.

Если входные данные представляют собой многомерный эллипсоид, то графически выбеливание выглядит как растяжение этого эллипсоида по его главным осям, как показано на рис. 42.

Очевидно, такое преобразование увеличивает совместную энтропию входов, т.к. оно выравнивает распределение данных в обучающей выборке.

10.13. Построение модели

Значения целевого ряда (это тот ряд, который нужно найти, например, доход по акциям на день вперед) зависят от N факторов, среди которых могут быть комбинации переменных, прошлые значения целевой переменной, закодированные качественные показатели. Эти факторы определяются обычными методами статистики (метод наименьших квадратов, кросс-корреляция и т.д.) [55]. Входные и выходные переменные преобразуются (масштабированием, стандартизацией) так, чтобы они принимали значения от 0 до 1 (или от -1 до 1). В результате мы получаем первоначальную модель, которую можно пытаться оптимизировать с помощью нейронной сети.

Главной нерешенной проблемой в области анализа временных рядов с помощью нейронных сетей остается определение топологии сети (или числа степеней свободы в модели). Нужно или прямо указать размеры сети (число скрытых слоев, скрытых элементов, структуру связей), или настроить модель на имеющиеся данные, применяя конструктивный либо деструктивный подход (например, уменьшение весов).

10.14. Оптимизация обучения

Следующая задача – найти параметры (веса) модели [55]. Это делается с помощью алгоритма оптимизации (обучения). Известно несколько таких алгоритмов, в частности, методы обратного распространения и «замораживания». Этот этап может занять продолжительное время и потребует большой технической работы (установка начальных значений весов, выбор критерия остановки и др.), однако в конце его мы получим некоторую разумную совокупность весов. Нужно следить за тем, чтобы сеть не запоминала шумы, присутствующие во временных рядах (переобучение). Для этого на протяжении всего процесса оптимизации следует проверять, согласуется ли работа модели на обучающем множестве с соответствующими результатами на подтверждающем множестве.

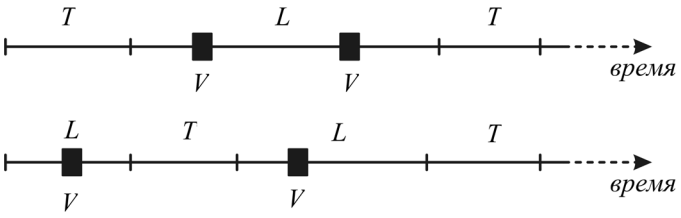
10.15. Статическое и адаптивное обучение

При моделировании финансовых временных рядов вопрос о том, как разбить все имеющиеся данные на обучающее, подтверждающее и тестовое множества, является нетривиальным [55]. Например, если данные, касающиеся биржевого краха, отнести к подтверждающему множеству, это даст искаженные результаты. При статическом подходе обычно поступают так: берут два небольших промежутка времени до и после обучающего

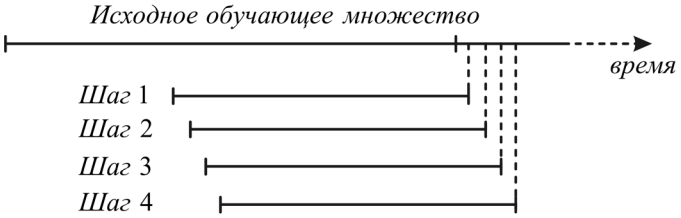
множества и из них случайным образом выбирают образцы в подтверждающее множество. На рис. 43,*а* показан ряд примеров построения подтверждающего множества. Никогда не будет лишним проверить, насколько изменятся результаты, если множество выбрать иначе.

Изменчивый характер финансовых рынков плохо согласуется с долгосрочными моделями устойчивости. Под действием краткосрочных «модных» тенденций или паники на бирже может существенно измениться реакция людей на те или иные показатели рынка. Чтобы справиться с этой трудностью, были предложены так называемые адаптивные нейронные сети, в которых веса модели непрерывно уточняются с помощью процедуры обучения на все новых временных промежутках. Пример такой модели, учитывающей вновь поступающую информацию, показан на рис. 43,*б*. Варфиз и Версино [98, 99] применили эту идею для предсказания изменений ежемесячных индексов промышленных и энергетических компаний.

Результаты совпали с тем, что получается по модели Бокса-Дженкинса. Де Гроот [100] занимался задачей прогноза обменных курсов валют. Модель видоизменялась каждые три месяца, и результаты оказались существенно лучше, чем по методу линейной регрессии.



а)



б)

Рис. 4.27 – Различные статические методы обучения (*а*) и обучение по меняющимся промежуткам времени (*б*) (здесь L -- обучающее множество, T -- тестовое множество, V -- подтверждающее множество) [55]

10.16. Отбор и диагностика модели

Проверка свойств модели временного ряда – необходимое условие для надежного предсказания и для понимания природы имеющихся закономерностей [55]. К сожалению, этот вопрос слабо освещен в литературе.

Разности между истинными и оцененными значениями должны подчиняться гауссовскому распределению с нулевым средним. Если оказалось, что распределение имеет слишком тяжелые хвосты или несимметрично, то нужно пересмотреть модель. Среди значений разностей могут выявиться закономерности или последовательные корреляции, тогда необходимо дополнительное обучение или улучшение модели. Здесь следует отметить, что рассматриваемые в данной монографии системы не обладают гауссовским распределением.

Оценка качества модели обычно основывается на критерии согласия типа средней квадратичной ошибки (MSE) или квадратного корня из нее (RMSE). Эти критерии показывают, насколько предсказанные значения оказались близки к обучающему, подтверждающему или тестовому множествам. Для

рядов с большим разбросом Лапедес [101] предложил критерий средней относительной вариации:

$$\mathrm{arv}(S) = \frac{\sum_{t\in S}(d_t - x_t)^2}{\sum_{t\in S}\left(d_t - \langle x_t\rangle\right)^2} = \frac{\sum_{t\in S} e_t^2}{N\sigma^2},$$

где S – временной ряд, e_t – разность (истинное значение d_t минус x_t) в момент t, $\langle x_t\rangle$ – оценка для среднего значения ряда, N – число данных в ряде. Последующая нормализация с помощью оценки для вариации позволяет проводить более надежные сравнения для различных приложений.

Используя предыдущее значение x_{t-1} целевой переменной, можно оценить способность модели *предсказывать на один шаг вперед.* Подставляя предсказанные значения на место истинных, получим метод *предсказания на* k *шагов вперед.* Если через несколько шагов модель начинает отклоняться от настоящей траектории, это значит, что в ней происходит накопление и рост ошибки (см. [102, 90]).

Самый распространенный метод выбора нейронно-сетевой модели с наилучшим обобщением – это проверка критерия согласия (MSE, ARV и др.) на тестовом множестве, которое не использовалось при обучении. Если же данных мало, разбивать их на обучающее и подтверждающее множество нужно разными способами. Такое перекрестное подтверждение может потребовать много времени, особенно для нейронных сетей с их длительным процессом обучения.

В линейном анализе временных рядов можно получить несмещенную оценку способности к обобщению, исследуя результаты работы на обучающем множестве (MSE), число свободных параметров (W) и объем обучающего множества (N). Оценки такого тина называются *информационными критериями*

(IC) и включают в себя компоненту, соответствующую критерию согласия, и компоненту штрафа, которая учитывает сложность модели. Барроном [103] были предложены следующие информационные критерии: нормализованный IC Акаике (NAIC), нормализованный байесовский IC (NBIC) и итоговая ошибка прогноза (FPE):

$$\mathrm{NAIC}=\ln(\mathrm{MSE})+\frac{2W}{N},$$

$$\mathrm{NBIC}=\ln(\mathrm{MSE})+\frac{2W}{N}\ln N,$$

$$\mathrm{FPE}=\mathrm{MSE}\left(\frac{1+\mathrm{W/N}}{1-\mathrm{W/N}}\right).$$

Было показано [104], что FPE представляет собой несмещенную оценку способности к обобщению для нелинейных моделей, в частности, – для нейронных сетей. К сожалению, при этом предполагается, что в нашем распоряжении имеется бесконечное число наблюдений, – в этом случае оценка надежности модели, вообще, не представляет особых сложностей.

Ясно, что информационные критерии дают информацию об адекватности модели и помогают выбрать модель подходящего уровня сложности. Другие методы диагностики позволяют, если такая задача стоит, избежать подхода к системе как к «черному ящику». Поскольку основное отличие сети от линейной регрессии – это возможность применять нелинейные преобразователи, имеет смысл посмотреть, насколько глубоко модель использует свои нелинейные возможности. Проще всего это сделать с помощью введенного Вигендом [90] отношения:

$$\Re=\frac{\Delta N}{\Delta\delta},$$

где ΔN – вариация сетевых разностей, $\Delta\delta$ – вариация разностей регрессии.

Для более тщательной проверки нелинейных возможностей нужно изобразить распределение выходных значений для скрытых элементов. Слишком большая доля крайних значений (0 или 1) говорит о том, что некоторые элементы попали в режим насыщения. Еще один способ – построить совместное распределение линейного и нелинейного выходов и применить линейную регрессию. Отклонения от наклона с углом 45° говорят о том, что нелинейные возможности задействованы [100]. Между прочим, встречается точка зрения, что появление во время обучения резких изменений разностей говорит об использовании нелинейностей. К сожалению, это не соответствует действительности.

Следует также проверить, скоррелированны ли действия скрытых элементов. В многомерном регрессионном анализе при росте мультиколлинеарности значения коэффициентов регрессии становятся все менее надежными. Так же и здесь предпочтительно, чтобы выходы скрытых элементов одного слоя были некоррелированны. Нужно найти собственные значения корреляционной матрицы для выходов скрытых узлов по данным обработки всех обучающих примеров. При полной некоррелированности все собственные значения будут равны единице, а отличия от единицы говорят об избыточном числе скрытых элементов. Кроме того, для анализа внутреннего представления нейронно-сетевой модели часто применяются методы кластерного анализа (см. [105]).

10.17. Доводка

При построении системы прогноза преследуется цель не только расширить наше понимание процессов, но и получить помощь для

принятия решений в финансовой области [55]. Такого рода руководства можно создать с помощью комбинаций нескольких нейронных сетей обученных на разных множествах данных и разных отрезках времени. Например, сигналы на покупку или продажу будут даваться по пороговым значениям, которые настроены с учетом предыдущих позиций и ошибок. Очень важно также распознать момент, когда эффективность модели начинает падать.

СПИСОК ЦИТИРУЕМЫХ ИСТОЧНИКОВ

ССЫЛКИ НА ЭЛЕКТРОННЫЕ РЕСУРСЫ (К ВВЕДЕНИЮ)

1. Электронный ресурс – URL – https://ru.wikipedia.org/wiki/Эконофизика
2. Электронный ресурс – URL – https://ru.abcdef.wiki/wiki/Econophysics
3. Электронный ресурс – URL – https://science.uottawa.ca/mathstat/en
4. Электронный ресурс – URL – https://lenta.ru/articles/2009/09/08/fusion/
5.

ЛИТЕРАТУРА ПО ОСНОВНОМУ СОДЕРЖАНИЮ

1. Тихонов Э.И. Методы прогнозирования в условиях рынка: учебное пособие. Невинномысск, 2006. 221 с.

2. Каста Дж. Связность, сложность и катастрофы: Пер. с англ. М. : Мир, 1982. 216 с.

3. Анищенко В.С., Астахов В.В., Вадивасова Т.Е., Нейман А.Б., Стрелкова Г.И., Шимановский-Гайер Л. Нелинейные эффекты в хаотических и стохастических системах. М., Ижевск : Институт Компьютерных исследований, 2003. 544 с.

4. Арнольд В.И. Теория катастроф. 3-е изд., доп. М. : Наука. Гл. ред. физ. мат. лит., 1990. 128 с.

5. Галушкин А.И. Теория нейронных сетей. Кн. 1: Учеб. Пособие для вузов. М. : ИПРЖР, 2001. 385 с.

6. Головко В.А. Нейронные сети: обучение, организация и применение. Кн.4:Учеб.пособие для вузов/Общая ред. А.И. Галушкина. М. : ИПРЖР, 2001. 256 с.

7. Розенблат Ф. Принципы нейродинамики: Персептрон и теория механизмов мозга. Пер. с англ. М. : Мир, 1965. 175 с.

8. Сигеру О., и др. Нейроуправление и его приложения / Пер. с англ. под ред. А.М. Ганшина. М. : ИПРЖР, 2001. 321 с.

9. Вороновский Г.К., Махотило К. В. Петрашев С. Н. Сергеев С. А.. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. Харьков : ОСНОВА, 1997. 112 с.

10. Еремин Д.М. Система управления с применением нейронных сетей // Приборы и системы. Управление, контроль, диагностика. 2001. № 9 С. 8–11.

11. Барский А.Б. Обучение нейросети методом трассировки // Труды VIII Всероссийской конференции «Нейрокомпьютеры и их применение»: Сб. докл., 2002. С. 862–898.

12. Белим С.В. Математическое моделирование квантового нейрона // Труды VIII Всероссийской конференции «Нейрокомпьютеры и их применение»: Сб. докл., 2002. С. 899–900.

13. Бутенко А.Х. и др. Обучение нейронной сети при помощи алгоритма фильтра Калмана // Труды VIII Всероссийской конференции «Нейрокомпьютеры и их применение »: Сб. докл., 2002. С. 1120–1125.

14. Лащев А.Я., Глушич Д.В. Синтез алгоритмов обучения нейронных сетей // Труды VIII Всероссийской конференции «Нейрокомпьютеры и их применение»: Сб. докл., 2002. С. 997–999.

15. Гусак А.Н. и др. Подход к послойному обучению нейронной сети прямого распространения // Труды VIII Всероссийской конференции «Нейрокомпьютеры и их применение»: Сб. докл., 2002. С. 931–933.

16. Шибхузов З.М. Конструктивный TOWER алгоритм для обучения нейронных сетей из 1П-нейронов // Труды VIII Всероссийской конференции «Нейрокомпьютеры и их применение»: Сб. докл., 2002. С. 1066–1072.

17. Медведев В.С., Потемкин В.Г. Нейронные сети. MATLAB 6 / Под общ. ред. к.т.н. В.Г. Потемкина. М. : ДИАЛОГ-МИФИ, 2002. 496 с.

18. Уоссермен Ф. Нейрокомпьютерная техника: теория и практика. М. : ЮНИТИ, 1992. 240 с.

19. Новиков А.В., и др. Метод поиска экстремума функционирования оптимизации для нейронной сети с полными последовательными связями //Труды VIII Всероссийской конференции «Нейрокомпьютеры и их применение»: Сб. докл., 2002. С. 1000–1006.

20. Бодянский Е.В., Кучеренко Е.И. Диагностика и прогнозирование временных рядов многослойной радиально-базисной нейронной сети // Труды VIII Всероссийской конференции «Нейрокомпьютеры и их применение»: Сб. докл., 2002. С. 69–72.

21. Генетические алгоритмы и машинное обучение. Режим доступа [http://www.math.tsu.ru/russian/center/ai_group/ai_collection/docs/faqs/ai/part5/faq3.html]

22. Генетические алгоритмы обучения. Режим доступа [http://www.hamovniki.net/~alchemist/NN/DATA/Gonchar/Main.htm]

23. Лекции по нейронным сетям и генетическим алгоритмам. Режим доступа [http://www.infoart.baku.az/inews/30000007.htm]

24. (EHIPS) Генетические алгоритм. Режим доступа [http://www.iki.rssi.ru/ehips/genetics.htm 29.08.2002]

25. Батищев Д.И. Генетические алгоритмы решения экстремальных задач. Воронеж : ВГУ, 1994. 135 с.

26. Пятецкий В.Е., Бурдо А.И. Имитационное моделирование процесса создания обучающихся систем. В сб.: Имитационное моделирование производственных процессов / Под. Ред. Мироносецкого Н.Б. Новосибирск, 1979. 68 с.

27. Дубовиков М.М., Старченко Н.В. Эконофизика и анализ финансовых временных рядов. «Эконофизика. Современная физика в поисках экономической теории» / Под. ред. В.В. Харитонова и А.А. Ежова. М. : МИФИ, 2007. С. 244–293.

28. Bachelier L. Theory of Speculation (Translation of 1900 French edn) / P.H. Cootner (Ed.) // The Random Character of Stock Market Prices, The MIT Press, Cambridge. 1964. P. 17–78.

29. Wiener N. Differential-space // J. Math. Phys. Math. Inst. Technol. 1923. № 2. P. 131–174.

30. Fama E.F. Mandelbrot and the stable Paretian hypothesis // J. Bus. 1963. (36). P. 420–429.

31. Sharpe W.F. Portfolio Theory and Capital Markets. N.Y. : McGraw-Hill, 1970. 341 p.

32. Black F., Scholes M. The pricing of options and corporate liabilities // J. Polit. Econ. 1973. № 3. P. 637–659.

33. Merton R. Theory of rational option pricing // Bell J. Econ. Manage. Sci. 1973. № 4. P. 141–183.

34. Mandelbrot B. Sur certains prix speculatifs: faits empiriques et modele base sur des processus stables additifs de Paul Levy // C. R. Acad. Sci. 1962. (254). P. 3968–3970.

35. Мандельброт Б. Фракталы, случай и финансы. М. ; Ижевск : НИЦ "Регулярная и хаотическая динамика", 2004. 256 с.

36. Mandelbrot B. Une classe de processus stochastiques homotheH tiques a soi; application a la hi climatohgique de H. E. Hurst // C R. Acad. Sci. 1965. (260). P. 3274–3277.

37. Лоскутов А.Ю., Бредихин А. А. ARCH-модели на финансовом рынке России // Обозрение прикладной и промышленной математики. 2004. Т. 11. № 3. С. 468–486.

38. Ширяев А.Н. Основы стохастической финансовой математики. М. : Фазис, 1998. 544 с.

39. Elliott R. N. Elliott's Masterworks: the Definitive Collection, Gainesville. New Classics Library, 1994.

40. Мерфи Дж. Технический анализ фьючерсных рынков. М. : Сокол, 1996. 592 с.

41. Prechter R.R., Frost A.J. ELLIOTT WAVE PRINCIPLE KEY TO MARKET BEHAVIOR. Gainesville, New Classics Library, 1978. 248 p.

42. Takens F. *Dynamical Systems and Turbulence.* Berlin : Springer-Verlag, 1981. 898 p.

43. Ежов А.А., Шумский С.А. Нейрокомпьютинг и его приложения в экономике и бизнесе. М. : МИФИ, 1998. 222 с.

44. Anderson P.V. More is different // Science, 1972. 177. P. 393–396.

45. Bak P. How Nature Works: The Science of Self-Organized Criticality. New York : Copernicus, 1996. 226 p.

46. Per Bak, M. Paczuski, M. Shubik, *Price Variations in a Stock Market with Many Agents,* Working paper 96-05-078, Santa Fe Institute Economics Research Programm, 1996.

47. Сорнетте Дидье. Как предсказывать крахи финансовых рынков. Критические события в комплексных финансовых системах. М. : Интернет-Трейдинг, 2003. 400 с.

48. Kolmogorov A. N. A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high Reynolds number // J. Fluid Mech., 1962, vol. 13, P. 82–85.

49. Mandelbrot B. *The fractal geometry of nature. Benoit B. Mandelbrot. W. H. Freeman and co., San Francisco, 1982. 460 p.*

50. Кузнецов С.П. Динамический хаос. М. : Физматлит., 2001. 296 с.

51. Петерс Э. Хаос и порядок на рынках капитала. Новый аналитический взгляд на циклы, цены и изменчивость рынка / Пер. с англ. М. : Мир, 2000. 333 с.

52. Хайкин, Саймон. Нейронные сети: полный курс, 2-е издание. : Пер. с англ. М. : Издательский дом "Вильямс", 2006. 1104 с.

53. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. 2-е изд., стереотип. М. : Горячая линия-Телеком, 2002. 382 с.

54. McCulloch W.S., Pitts W. A logical calculus of the ideas immanent in nervous activity // Bulletin of Mathematical Biophysics, 1943. vol. 5, P. 115–133.

55. Бэстенс Д.-Э., ван ден Берг В.-М., Вуд Д. Нейронные сети и финансовые рынки: принятие решений в торговых операциях. М. : ТВП, 1997. 236 с.

56. Dorizzi B., Duval J.M., Debar H. Utilisation de reseaux recurrents pour la prevision de consommation electrique // Proceedings of NeuroNimes, Journées internationals. 1992. № 5. P. 141–150.

57. Connor J., Atlas L. Recurrent neural networks and time series prediction // Conference on Neural Networks (IJCNN'91–Seattle), Seattle, Washington, July 1991. P. 301–306.

58. Kamijo K., Tanigawa T. Stock price pattern recognition: A recurrent network approach // Proceedings of the IEEE International Joint Conference on Neural Networks, 1990. P. 1215–1221.

59. Parker D.B. Optimal algorithms for adaptive networks: Second order back propagation, second order direct propagation and second order Hebbian learning // IEEE 1st International Conference on Neural Networks, San Diego, CA., 1987, vol. 2, P. 593–600.

60. Rumelhart D.E., Hinton G.E., Williams R.J. Learning representations of back-propagation errors // Nature (London), 1986. vol. 323. P. 533–536.

61. Roy S., Shynk J.J. Analysis of the momentum LMS algorithm // IEEE Transactions on Acoustics, Speech and Signal Processing, 1990. vol. ASSP–38. P. 2088–2098.

62. Hagiwara M. Theoretical derivation of momentum term in back-propagation // International Joint Conference on Neural Networks, Baltimore, 1992. vol. I, P. 682–686.

63. Jacobs R.A. Increased rates of convergence through learning rate adaptation // Neural Networks, 1988. vol. 1, P. 295–307.

64. Watrous R.L. Learning algorithms for connectionist networks: Aplied gradient methods of nonlinear optimization // First IEEE

Infemational Conference on Neural Networks. San Diego, CA., 1987, vol. 2, P. 619–627.

65. Osowski S. Sieci neuronowe w ujeciu algorytmicznym. Warszawa : WNT, 1996. 350 p.

66. Osowski S. Sieci neuronowe. - Warszawa: Oficyna Wydawnicza PW, 1994.

67. Osowski S.t Stodolski KL, Bojarczak P. Fast second order learning algorithm for feedforward multilayer neural networks and its applications // Neural Networks, 1996. Vol. 9. P. 1583–1596.

68. Haykin S. Neural networks, a comprehensive foundation. N.Y. : Macmillan College Publishing Company, 1994.

69. Hertz J., Krogh A., Palmer R. Wstep do teorii obliczen neuronowych. Wyd. II. Warszawa : WNT, 1995.

70. Gill P., Murray W., Wright M. Practical Optimization. N.Y. : Academic Press, 1981.

71. Widrow B., Stearns S.D. Adaptive Signal Processing, Englewood Cliffs, N.Y. : Prentice-Hall, 1985.

72. Golub G., Van Loan C. Matrix computations. N.Y. : Academic Press, 1991.

73. Marquardt D. An algorithm for least squares estimation of nonlinear parameters, S1AM, 1963. P. 431–442.

74. Widrow B., Hoff M. Adaptive switching circuits // Proc. IRE WESCON Convention Record, 1960. P. 107–115.

75. Lawton J.H. More time means more variation // Nature. 1988. V.334. P. 563.

76. Hush D.R., Home. B.G. Progress in supervised neural networks: What's new since Lipmann? // IEEE Signal Processing Magazine, 1993. vol. 10. P. 8–39.

77. Vapnik V. N., Chervonenkis A. On the uniform convergence of relative frequencies of events to their probabilities // Theory of Probability and its Applications, 1971. Vol. 16. P. 264–280.

78. LeCun Y. Efficient Learning and Second-order Methods, A Tutorial at NIPS 93, Denver, 1993.

79. LeCun Y. Generalization and network design strategies // Technical Report CRG-TR-89-4, Department of Computer Science, University of Toronto, Canada, 1989.

80. Thimm G,, Fiesler E. Neural network initialization // Natural to Artificial Neural Computation / Eds. J. Mira, F. Sandoval. – Malaga : IWANN, 1995. P. 533–542.

81. Denoeux J., Lengalle R. Initializing back propagation networks with prototypes // Neural Networks, 1993. Vol. 6. P. 351–363.

82. Karayiannis N. Accelerating training of feedforward neural network using generalized hebbian rules for initializing the internal representation // IEEE Proc. ICNN, Orlando, 1994. P. 1242–1247.

83. Klimauskas G. Neural Ware–User manual Natick, USA : Neural Ware Inc. 1992.

84. Demuth K., Beale M. Neural Network Toolbox for use with Matlab. – Natick : The MathWorks Inc. 1992.

85. Yule G.U. On a method of investigating periodicities in disturbed series, with special reference to Wolfer's sunspot numbers // Phil Trans. Royal Society London 1927. A226. 267 p.

86. Box G.E.P., Jenkins G.M. Time Series Analysis: Forecasting and Control. San Franisco : Holden-Day. 1970.

87. Tong H. Threshold Models in Non-linear Time Series Analysis. Lectures Notes Statistics, Vol. 21, Springer, 1983.

88. Granger C.W.J., Anderson T.W. Introduction to Bilinear Time Series Models. Gottingen : Vandenhoeck und Ruprecht., 1978.

89. Priestley M.B. State-dependent models: A general approach to non-linear time series analysis // Journal of Time Series Analysis, 1980. Vol. 1. P. 1.

90. Weigend A.S., Huberman B.A., Rumelhart D.E. Predicting the future: A connectionist approach // International Journal of Neural Systems, 1990. Vol. 1. P. 193–209.

91. Kouam A., Badran F. and Thiria S. Approche methodologique pour l'etude de la prevision a l'aide de reseaux de neurons // Proceedings of Neuro-Nimes, 1992.

92. Colin A. Exchange rate forecasting at Citibank London // Proceedings of Neural Computing, London. 1991.

93. Colin A. Machine learning techniques for foreign exchange trading // Proceedings PASE 1991, Zurich, Dec. Parallel problem solving-applications in statistics and economics, 115 p.

94. Baestaens D.-E. Distributional analysis to model atypical behaviour// European Journal of Operational Research, April 1994. Vol.74. № 4. P. 230–242.

95. Stein R. Preprocessing data for neural networks // AI Expert, 1993. P. 32–37.

96. Windsor C.G., Harker A.H. Multi-variate financial index prediction: A neural network study // Euro Conference 88. 1988. P. 357.

97. Stein R. Selecting data for neural networks // AI Expert, 1993. P. 42–47.

98. Varfis A., Versino C. Univariate economic time series forecasting by connestionist methods // Proc. International Neural Network Conference, July, Paris, 1990.

99. Varfis A., Versino C. Murtagh F. Neural networks for economic time series forecasting // Neural networks for statistical and economic data. PASE. 1990. P. 155–159.

100. DeGroot C. Nonlinear time series analysis with connectionist networks // IPS Research Report № 93-03, Diss. ETH № 10038, Zurich. 1993.

101. Lapedes A., Farber R. Nonlinear signal processing using neural networks: prediction and system modeling // TR LA-UR-87-2662, Los Alamos, 1987.

102. Wurtz D., DeGroot C. Forecasting time series with connectionist nets: Applications in statistics, signal processing and economies // Lecture Notes in Artificial Intelligence, 604, Heidelberg : Springer, 1992.

103. Barron A. Predicted squared error: a criterion for automatic model selection // Self-Organizing Methods in Modeling, N.Y. : Marcel Dekker, 1984.

104. Moody J.E. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems // ANIPS 4, 1992. P. 847–854.

105. Gorman R.P., Sejnowski T.J. Analysis of hidden units in a layered network trained to classify sonar targets // Neural Networks, Vol. 1, 1988, P. 75–89.

106. Хакен Г. Синергетика:Иерархия неустойчивостей в самоорганизующихся системах: Пер. с англ. М. : Мир, 1985. 423 с.

107. Малинецкий Г.Г., Потапов А.Б., Подлазов А.В. Нелинейная динамика: Подходы, результаты, надежды. Изд. 2-е. М. : КомКнига, 2009. 280 с.

108. Головко В., Чумерин Ю. Нейросетевые методы определения спектра Ляпунова хаотических процессов // «Нейрокомпьютеры: Разработка и применение», 2004. № 1.

109. Головко В.А., Савицкий Ю.В. Нейросетевые методы определения спектра Ляпунова // Международный журнал «Компьютинг». 2002. № 1. С. 80–86.

110. Головко В.А., Чумерин Н.Ю., Савицкий Ю.В. Нейросетевой метод оценки спектра Ляпунова по наблюдаемым реализациям // Вестник Брестского государственного технического университета. 2002. №4. С. 66–70.

111. Ширяев В.И. Финансовые рынки: Нейронные сети, хаос и нелинейная динамика: Учебное пособие. Изд. 2-е, испр. и доп. М. : Книжный дом «ЛИБРОКОМ», 2009. 232 с.

112. Мусалимов В.М., Резников С.С., Чан Нгок Чау. Специальные разделы высшей математики. СПб : Санкт-Петербургский Государственный Университет Информационных технологий Механики и Оптики (СПбГУ ИТМО), 2006. 80 с.

113. Постнов Д.Э., Павлов А.Н., Астахов С.В. Методы нелинейной динамики: Учеб. пособие для студ. физ. фак. Саратов 2008. 120 с.

114. Безручко Б.П., Смирнов Д.А. Математическое моделирование и хаотические временные ряды. Саратов : Гос УНЦ «Колледж», 2005. 320 с.

115. Кузнецов С.П. Динамический хаос. М. : Физматлит., 2001. 296 с.

116. Антипов О.И., Неганов В.А., Потапов А.А. Детерминированный хаос и фракталы в дискретно-нелинейных системах. М. : Радиотехника, 2009. 235 с.

117. Grassberger P., Procaccia I. Measuring the strangeness of strange attractors // Physica D. № 9, 1983.

118. Rosenstein M.T., Colins J.J., De Luca C.J. Reconstruction expansion as a geometry-based framework for choosing proper delay time // Physica D. – 73, 1994, P. 82–89.

119. Fraser A.M., Swinney H.L. Independent coordinates for strange attractor from mutual information // Physical Review A. – 33, 1986, P. 1134–1140.

120. Abarbanel H.D.I., Brown R., Sidorovich J., Tsimring L. The analysis of observed chaotic data in physical systems // Review of Modern Physics, Vol. 65. № 4. 1993. P. 1331–1392.

121. Головко В.А. Нейросетевые методы обработки хаотических процессов // VII Всероссийская научно-техническая конференция «Нейроинформатика 2005»: Лекции по нейроинформатике. М. : МИФИ, 2005. С. 43–91.

122. Takens F. Detecting Strange Attractors in Turbulence // Dynamical Systems and Turbulence. Lecture Notes in Mathematics. Berlin : Springer – Verlag, 1981. 898 p. P. 366–381.

123. Дмитриев А.С., Ефремова Е.В., Кузьмин Л.В., Атанов Н.В. Генерация потока хаотических импульсов в динамической системе с внешним (периодическим) воздействием // Радиотехника и электроника, 2006. Т.51. № 5. С. 593–604.