

ИССЛЕДОВАНИЕ СТРУКТУРЫ РЕЦЕПТИВНОГО ПОЛЯ В ВИЗУАЛЬНОМ МЕТОДЕ РЕШЕНИЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ*

Н.А. Ольховский

Южно-Уральский государственный университет
(национальный исследовательский университет)

В статье описан метод определения проекции произвольного вектора на гиперплоскость в многомерном пространстве путем взаимодействия суперкомпьютера и искусственной нейронной сети. Суперкомпьютер строит визуальное представление гиперплоскости в окрестности некоторой точки. Визуальный образ обрабатывается нейронной сетью прямого распространения. В качестве ответа нейронная сеть возвращает направление проекции заданного вектора на гиперплоскость. В статье описано построение обучающего множества, проектирование, оптимизация и обучение нейронной сети. Представлены результаты вычислительных экспериментов с различными конфигурациями нейронных сетей.

Ключевые слова: линейное программирование, нестационарные задачи, режим реального времени, метод поверхностного движения, итерационный метод, теорема сходимости, искусственная нейронная сеть, глубокое обучение, параллельный алгоритм, высокопроизводительные вычисления

1. Введение

Одной из важнейших фундаментальных задач современной прикладной математики является задача линейного программирования (ЛП) с большим числом параметров [1]. Оптимизационные модели, основанные на многопараметрической (многомерной) задаче ЛП встречаются в системах поддержки принятия решений в экономике [2, 3], в системах управления беспилотными летательными аппаратами [4], в управлении технологическими процессами [5–7], при построении логистических цепочек [8–10], в оперативном управлении и планировании [11, 12].

В работе [13] предложено оригинальное решение многомерной задачи линейного программирования: построить визуальный образ многогранника, ограничивающего область допустимых значений, и при помощи искусственной нейронной сети (ИНС) прямого распространения определить направление максимального увеличения целевой функции. Движение по поверхности многогранника в указанном направлении позволяет быстро приблизиться к решению задачи. Базовой задачей является с помощью ИНС определить направление в отдельно взятой гиперплоскости. В данной работе описывается попытка построить и обучить нейросетевую модель, определяющую для гиперплоскости произвольной размерности направление, соответствующее максимальному увеличению целевой функции.

Структура статьи следующая. В разделе 2 изложены теоретические основы визуального метода. Раздел 3 раскрывает процедуру генерации обучающего множества и построение нейросетевой модели. В разделе 4 приводятся результаты вычислительных экспериментов с использованием нейронной сети. Наконец, в разделе 5 приведены выводы по полученным результатам и предложены идеи относительно дальнейших исследований.

*Исследование выполнено при финансовой поддержке РФ (проект № 23-21-00356).

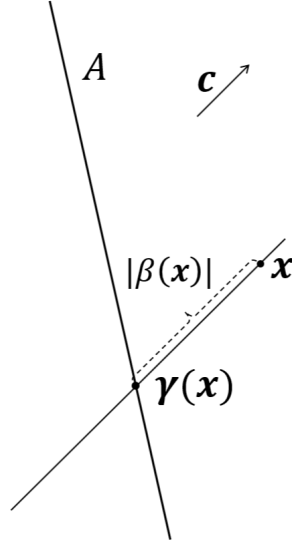


Рис. 1: Целевая проекция точки x на гиперплоскость H_a .

2. Теоретические основы

Без потери общности мы можем ограничиться рассмотрением гиперплоскости, проходящей через некоторую точку $z \in \mathbb{R}^n$. Зафиксируем произвольный ненулевой вектор $a \in \mathbb{R}^n$. Обозначим через H_a – гиперплоскость в пространстве \mathbb{R}^n , для которой вектор a является нормалью:

$$H_a = \{x \in \mathbb{R}^n \mid \langle a, x \rangle = 0\}. \quad (1)$$

Выберем *целевой вектор* $c \in \mathbb{R}^n$. Обозначим $g \in \mathbb{R}^n$ ортогональную проекцию целевого вектора c на гиперплоскость H_a

$$g = c - \frac{\langle a, c \rangle}{\|a\|^2} a. \quad (2)$$

Построим *целевую гиперплоскость* H_c , ортогональную вектору c :

$$H_c = \{x \in \mathbb{R}^n \mid \langle c, x \rangle = 0\}. \quad (3)$$

Целевая проекция $\gamma : H_c \rightarrow H_a$ вычисляется по следующему правилу

$$\gamma(x) = x - \frac{\langle a, x \rangle}{\langle a, c \rangle} c. \quad (4)$$

Смещение $\beta : H_c \rightarrow \mathbb{R}$ вычисляется по следующему правилу

$$\beta(x) = -\frac{\langle a, x \rangle}{\langle a, c \rangle} \|c\|. \quad (5)$$

Построим в гиперплоскости H_c набор базисных векторов следующим образом:

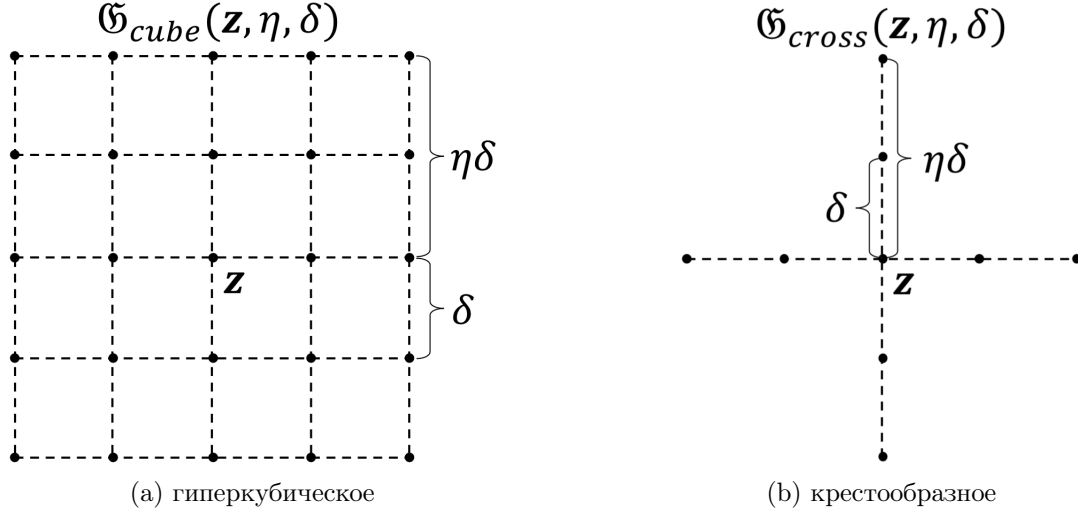


Рис. 2: Две структуры рецептивного поля в пространстве \mathbb{R}^3 .

точки вычисляют по ее номеру в любой момент времени при помощи специального алгоритма 2.

Образом рецептивного поля $\mathcal{I}(\mathfrak{G})$ назовем набор смещений, вычисленных для каждой точки рецептивного поля \mathfrak{G} :

$$\mathcal{I} = \{\beta(\mathbf{x}) \mid \mathbf{x} \in \mathfrak{G}\}. \quad (10)$$

Таким образом, набор данных, характеризующий одну случайную гиперплоскость и соответствующий ей образ, содержит следующие элементы:

- образ рецептивного поля $\mathcal{I}(\mathfrak{G})$, вычисленный для гиперплоскости A ;
- вектор \mathbf{g} , соответствующий целевому вектору \mathbf{c} .

Коэффициентом угла наклона вектора \mathbf{g} к базисному вектору $\mathbf{e}^{(i)}$ рецептивного поля назовем косинус угла наклона, вычисляемый следующим образом.

$$\cos \alpha_i = \frac{\langle \mathbf{e}^{(i)}, \mathbf{g} \rangle}{\|\mathbf{g}\|}. \quad (11)$$

Пример коэффициентов углов наклона для рецептивного поля в \mathbb{R}^3 показан на Рисунке 3. Взятые вместе, коэффициенты углов наклона образуют *вектор правильных ответов* $\mathbf{y}(\mathbf{g})$

$$\mathbf{y}(\mathbf{g}) = \left\{ \frac{\langle \mathbf{e}^{(i)}, \mathbf{g} \rangle}{\|\mathbf{g}\|} \mid i \in 1, \dots, n-1 \right\}. \quad (12)$$

В этом случае один *прецедент обучающего множества* будет включать:

- образ рецептивного поля $\mathcal{I}(\mathfrak{G})$;
- вектор правильных ответов $\mathbf{y}(\mathbf{g})$.

3. Обучение нейронных сетей

Создание обучающего множества и обучение искусственной нейронной сети производилось по схеме, представленной на Рис. 4. Сначала программа генерации случайных гиперплоскостей, приняв в качестве входного параметра число N , генерирует N пар $\{\mathbf{a}, \mathbf{c}\}$, где каждая пара состоит из случайного вектора \mathbf{a} , определяющего гиперплоскость (3) и

Algorithm 1 Построение крестообразного рецептивного поля $\mathfrak{G}_{\text{cross}}(z, \eta, \delta)$

Require: $z \in \mathbb{R}^n$, $\eta \in \mathbb{N}$, $\delta \in \mathbb{R}_{>0}$

```

1:  $\mathfrak{G} := \emptyset$ 
2: for  $t = 1 \dots n - 1$  do
3:   for  $i = 1 \dots 2\eta$  do
4:      $s := \mathbf{0}$ 
5:     for  $j = 1 \dots n - 1$  do
6:       if  $j \neq t$  then
7:          $s_j = 0$ 
8:       else
9:         if  $j \leq \eta$  then
10:           $s_j := (i - \eta - 1)\delta$ 
11:        else
12:           $s_j := (i - \eta)\delta$ 
13:        end if
14:      end if
15:       $s := s + s_j e^{(j)}$ 
16:    end for
17:     $\mathfrak{G} := \mathfrak{G} \cup \{s + z\}$ 
18:  end for
19: end for
20:  $\mathfrak{G} := \mathfrak{G} \cup \mathbf{0}$ 
21: return  $\mathfrak{G}$ 

```

случайного вектора целевой функции \mathbf{c} . Массив сгенерированных данных помещается в файл, передаваемый программе визуализации. Визуализатор принимает входные параметры, определяющие положение точек рецептивного поля:

- ранг рецептивного поля η ;
- плотность рецептивного поля δ ;
- форма рецептивного поля **cube** (гиперкубическое) или **cross** (крестообразное).

Визуализатор вычисляет координаты точек рецептивного поля и для каждой точки определяет смещение. Множество смещений нормализуется по формуле:

$$x = \frac{x - \min(\mathcal{I}(\mathfrak{G}))}{\max(\mathcal{I}(\mathfrak{G})) - \min(\mathcal{I}(\mathfrak{G}))} \cdot 511 - 256, \text{ где } x \in \mathcal{I}(\mathfrak{G}). \quad (13)$$

Также визуализатор вычисляет вектор правильных ответов. В результате работы визуализатора формируется файл прецедентов, содержащий N строк. В каждой строке записаны K нормализованных по формуле (13) коэффициентов, формирующих образ рецептивного поля $\mathcal{I}(\mathfrak{G})$. И $n - 1$ коэффициентов угла наклона, формирующих вектор правильного ответа $\mathbf{y}(g)$.

При обучении ИНС множество прецедентов делилось по правилу 80/15/5: 80% прецедентов использовались для обучения, 15% – для валидации в конце каждой эпохи, 5% – для тестирования нейронной сети после окончания обучения.

Algorithm 2 Функция G вычисляет точку рецептивного поля по ее номеру k

Require: $z \in \mathbb{R}^n$, $\eta \in \mathbb{N}$, $\delta \in \mathbb{R}_{>0}$

```
1: function  $G(k, n, z, \eta, \delta)$ 
2:   if  $k = 2\eta + 1$  then
3:      $g := \mathbf{0}$ 
4:     return  $g$ 
5:   end if
6:    $l := \lfloor (k - 1)/2\eta \rfloor + 1$ 
7:    $k := (k - 1) \bmod 2\eta + 1$ 
8:    $g := z$ 
9:   if  $k \leq \eta$  then
10:     $g := g + (i - \eta - 1)\delta e^{(l)}$ 
11:  else
12:     $g := g + (i - \eta)\delta e^{(l)}$ 
13:  end if
14:  return  $g$ 
15: end function
```

Обучение производилось при помощи библиотек **keras** и **TensorFlow**. Для оценки достигаемой точности ответов нейронной сети была взята косинусовая мера (Cosine Similarity):

$$CS = -\frac{\sum_{k=1}^n (\alpha_k \cdot \mathbf{y}_k)}{\sqrt{\sum_{k=1}^n \alpha_k^2} \cdot \sqrt{\sum_{k=1}^n \mathbf{y}_k^2}}. \quad (14)$$

Здесь α_k – это значения коэффициентов наклона, предсказанные нейронной сетью, а \mathbf{y}_k – коэффициенты наклона, рассчитанные визуализатором.

4. Вычислительные эксперименты

Были проведены три серии вычислительных экспериментов. Вычисления проводились на комплексе «Нейрокомпьютер» Южно-Уральского государственного университета [14].

В первой серии экспериментов был осуществлен подбор гиперпараметров для обучения нейронной сети для крестообразного рецептивного поля со следующей конфигурацией: $n = 4$, $\eta = 5$, $\delta = 1$. Для осуществления поиска в пространстве гиперпараметров использовался Байесовский подход. Варианты гиперпараметров ИНС, среди которых осуществлялся поиск:

- число скрытых слоев равно размерности $n = 3$;
- число нейронов в скрытом слое $\{1024, 2048, 4096\}$;
- коэффициент прореживания (dropout) $\{0., 0.15, 0.30, 0.45\}$;
- функция активации $\{\text{ReLU}, \text{sigmoid}, \text{tanh}\}$;
- размер выборки (batch size) $\{128, 256, 512, 1024\}$;
- скорость обучения $\{0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00005, 0.00001\}$.

Для каждого количества скрытых слоев выполнялось 30 попыток подбора гиперпараметров. Наилучшую точность по косинусовой мере продемонстрировала ИНС с архитектурой, представленной на Рис. 5

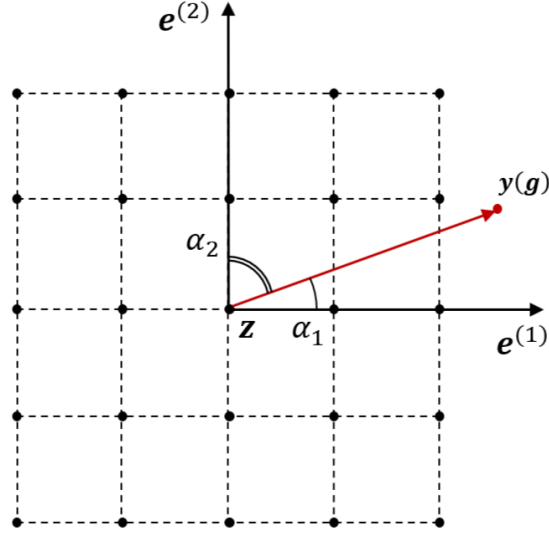


Рис. 3: Коэффициенты углов наклона в рецептивном поле в \mathbb{R}^3 .

Во второй серии экспериментов была исследована зависимость точности, достигаемой при обучении нейронных сетей, от конфигурации и плотности рецептивного поля. Для этого в пространстве \mathbb{R}^3 и \mathbb{R}^4 были взяты рецептивные поля с рангом от 1 до 5 и плотностью, подобранной так, чтобы количество точек рецептивного поля по каждому измерению равнялась 10. Параметры рецептивных полей и соответствующее им количество точек для конфигураций **cube** и **cross** представлены в Таблице 1.

Таблица 1: Параметры рецептивных полей в \mathbb{R}^3 и \mathbb{R}^4 .

Параметры	K	
	<i>cube</i>	<i>cross</i>
$n = 3, \eta = 1, \delta = 5$	9	5
$n = 3, \eta = 2, \delta = 2.5$	25	9
$n = 3, \eta = 3, \delta = 1.66667$	49	13
$n = 3, \eta = 4, \delta = 1.25$	81	17
$n = 3, \eta = 5, \delta = 1$	121	21
$n = 4, \eta = 1, \delta = 5$	27	7
$n = 4, \eta = 2, \delta = 2.5$	125	13
$n = 4, \eta = 3, \delta = 1.66667$	343	19
$n = 4, \eta = 4, \delta = 1.25$	729	25
$n = 4, \eta = 5, \delta = 1$	1331	31

Для каждой конфигурации нейронная сеть имеет входной слой, состоящий из K нейронов, и выходной слой из числа нейронов, равного размерности пространства минус единица $n - 1$. Каждому коэффициенту $\cos \alpha_i$ соответствовал один нейрон выходного слоя.

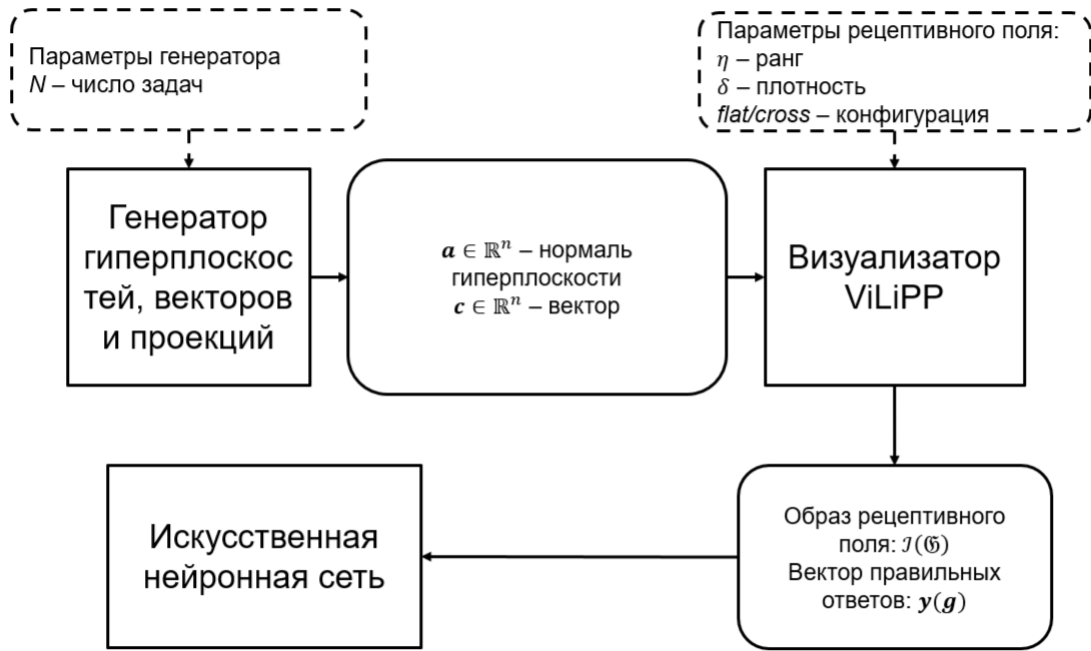


Рис. 4: Архитектура программного комплекса, генерирующего прецеденты для обучения ИНС.

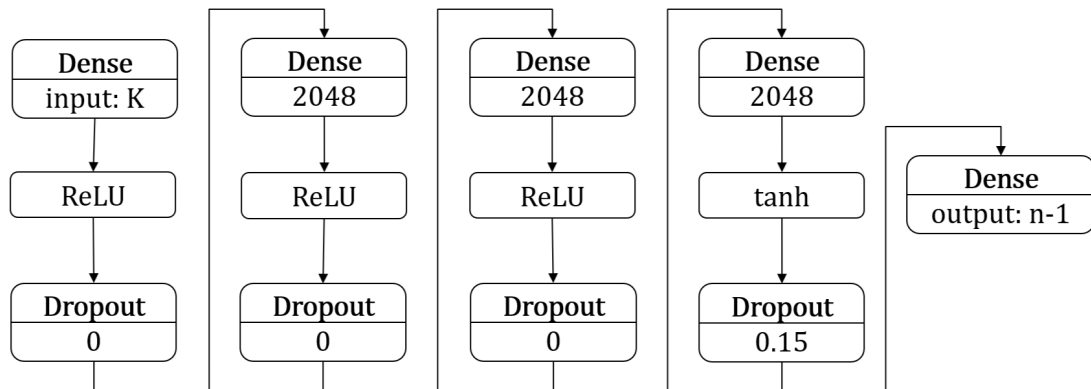
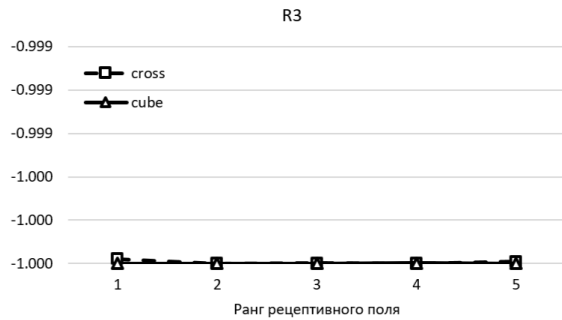


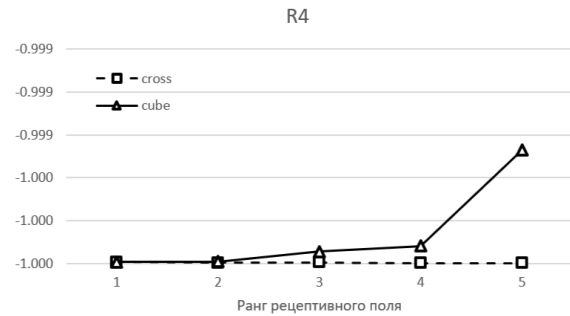
Рис. 5: Архитектура нейронной сети.

Результаты работы ИНС в \mathbb{R}^3 представлена на Рис. 6а. Видно, что точность работы сети близка к -1 независимо от ранга рецептивного поля и его конфигурации. У крестообразного рецептивного поля при ранге $\eta = 1$ можно заметить падение точности на 0.000015 . Вопрос о том, является ли это снижение значимым или представляет собой погрешность вычислений, требует дополнительной проверки. Результаты экспериментов в \mathbb{R}^4 показали, что, начиная с ранга 3, точность гиперкубического рецептивного поля демонстрирует существенное снижение. Результаты для ранга $\eta = 5$ на 0.0005 хуже, чем для ранга $\eta = 1$. Крестообразное рецептивное поле демонстрирует стабильно высокие результаты, близкие к -1 , независимо от ранга. Графики представлены на Рис. 6б.

В третьей серии экспериментов было исследовано изменение точности с повышением размерности пространства, в которой работает ИНС. Было использовано крестообразное рецептивное поле ранга $\eta = 5$ с плотностью $\delta = 1$. Число скрытых слоев равнялось размерности пространства. Архитектура нейронной сети строилась следующим образом. За основу бралась архитектура, представленная на Рис. 5. При повышении размерности на единицу после входного слоя (после первого Dropout) добавлялся новый скрытый слой, состоящий из 2048 нейронов с функцией активации ReLU и Dropout 0. Таким образом, для каждой



(a) Cosine Similarity в \mathbb{R}^3 .



(b) Cosine Similarity в \mathbb{R}^4 .

Рис. 6: Точность работы ИНС.

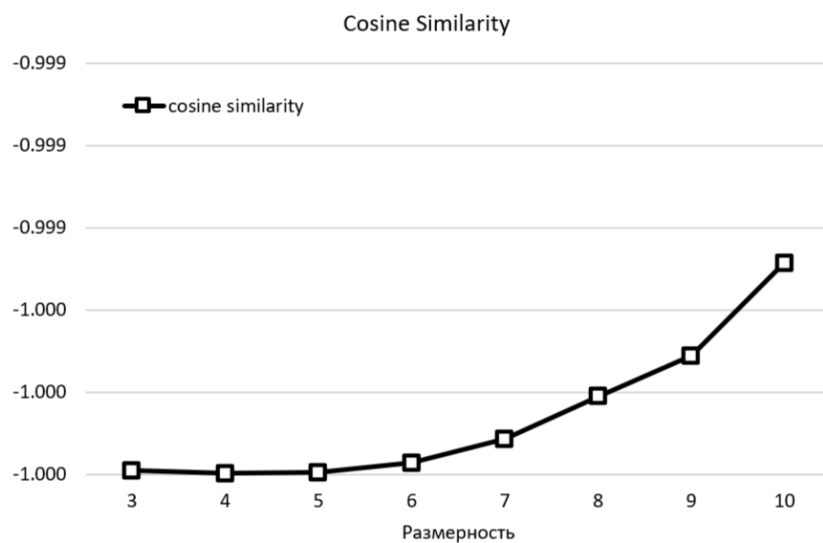


Рис. 7: Точность работы ИНС в пространстве размерности от 3 до 10.

размерности все скрытые слои, кроме последнего, были одинаковыми: 2048 нейронов, функция активации ReLU, Dropout 0. Последний слой: 2048 нейронов, функция активации ReLU, Dropout 0.15.

Из результатов, представленных на графике Рис. 7, можно отметить, что с повышением размерности достигаемая при обучении точность существенно снижается. Для размерности $n = 10$ падение достигает 0.0005. При этом важно отметить, что при обучении ИНС использовалось фиксированное число эпох 50. И на момент окончания обучения достигнутая точность продолжала расти! То есть высока вероятность, что при высоких размерностях сниженная точность обусловлена не объективной неспособностью ИНС продолжать обучение, а эффектом замедления обучения, связанным с ростом числа слоев. Требуется больше времени на обучение, чтобы установить реальную достижимую точность ИНС при работе в высоких размерностях.

5. Заключение

В настоящей работе была исследована зависимость точности нейронной сети, от конфигурации рецептивного поля. Установлено, что при низкой размерности нейронные сети на основе обеих конфигураций рецептивного поля (гиперкубической и крестообразной) де-

монстрируют прекрасные результаты, достигая точности, близкой к -1.0.

Изменение ранга рецептивного поля в пространстве \mathbb{R}^3 не оказывает влияния на результаты работы ИНС. Однако в \mathbb{R}^4 с увеличением ранга существенно падает точность ИНС на основе гиперкубического рецептивного поля, что может быть связано с экспоненциальным ростом числа точек рецептивного поля и соответствующим усложнением вычислений. Эта версия косвенно подтверждается тем фактом, что точность работы ИНС на основе крестообразного поля, в котором с увеличением ранга число точек растет линейно, остается высокой и близкой к наилучшей.

Результаты экспериментов с повышением размерности до 10 включительно свидетельствуют, что точность, достигаемая при фиксированном числе эпох (времени обучения), снижается экспоненциально. Таким образом проявляется замедление обучения при увеличении числа скрытых слоев. Необходимо дополнительно проверить, какая точность может быть достигнута при существенном увеличении времени обучения.

При проведении дальнейших исследований необходимо ответить на следующие вопросы. Какова достижимая точности за разумное время обучения для высоких размерностей: 100, 1000 и т.д.? Как изменяется точность при изменении числа скрытых слоев для высоких размерностей? Возможно ли для высокой размерности добиться увеличения точности за счет более удачного подбора гиперпараметров?

Для применения полученных результатов к решению многомерной задачи линейного программирования целесообразно провести исследование достигаемой точности в размерностях 1000, 10 000, 100 000 при ограниченном небольшом числе скрытых слоев.

Также необходимо исследовать возможности по определению проекций векторов в пространства, образуемые пересечением гиперплоскостей высоких размерностей.

Литература

1. Соколинская И.М., Соколинский Л.Б. О решении задачи линейного программирования в эпоху больших данных // Параллельные вычислительные технологии (ПаВТ'2017). Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2017. С. 471–484. URL: <http://omega.sp.susu.ru/pavt2017/short/014.pdf>.
2. Brogaard J., Hendershott T., Riordan R. High-Frequency Trading and Price Discovery // Review of Financial Studies. 2014. Vol. 27, no. 8. P. 2267–2306. DOI: 10.1093/rfs/hhu032.
3. Deng S., Huang X., Wang J., et al. A Decision Support System for Trading in Apple Futures Market Using Predictions Fusion // IEEE Access. 2021. Vol. 9. P. 1271–1285. DOI: 10.1109/ACCESS.2020.3047138.
4. Seregin G. Lecture notes on regularity theory for the Navier-Stokes equations. Singapore: World Scientific Publishing Company, 2014. 268 p. DOI: 10.1142/9314.
5. Demin D.A. Synthesis of optimal control of technological processes based on a multialternative parametric description of the final state. Eastern-European Journal of Enterprise Technologies. 2017. Vol. 3, 4(87). P. 51–63. DOI: 10.15587/1729-4061.2017.105294.
6. Kazarinov L.S., Shnayder D.A., Kolesnikova O.V. Heat load control in steam boilers. 2017 International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM 2017 - Proceedings. IEEE, 2017. DOI: 10.1109/ICIEAM.2017.8076177.
7. Zagoskina E.V., Barbasova T.A., Shnaider D.A. Intelligent Control System of Blast-furnace Melting Efficiency. SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings. IEEE, 2019. P. 710–713. DOI: 10.1109/SIBIRCON48586.2019.8958221.

8. *Fleming J., Yan X., Allison C., et al.* Real-time predictive eco-driving assistance considering road geometry and long-range radar measurements. IET Intelligent Transport Systems. 2021. Vol. 15, no. 4. P. 573–583. DOI: 10.1049/ITR2.12047.
9. *Scholl M., Minnerup K., Reiter C., et al.* Optimization of a thermal management system for battery electric vehicles. 14th International Conference on Ecological Vehicles and Renewable Energies, EVER 2019. IEEE, 2019. DOI: 10.1109/EVER.2019.8813657.
10. *Meisel S.* Dynamic Vehicle Routing. Anticipatory Optimization for Dynamic Decision Making. Operations Research/Computer Science Interfaces Series, vol. 51. New York, NY: Springer, 2011. P. 77–96. DOI: 10.1007/978-1-4614-0505-4_6.
11. *Cheng A.M.K.* Real-Time Scheduling and Schedulability Analysis. Real-Time Systems: Scheduling, Analysis, and Verification. John Wiley, Sons, 2002. P. 41–85. DOI: 10.1002/0471224626.CH3.
12. *Kopetz H.* Real-Time Scheduling. Real-Time Systems. Real-Time Systems Series. Boston, MA: Springer, 2011. P. 239–258. DOI: 10.1007/978-1-4419-8237-7_10.
13. *Ольховский Н.А., Соколинский Л.Б.* Визуальное представление многомерных задач линейного программирования // Вестник ЮУрГУ. Серия: Вычислительная Математика и Информатика. 2022. Т. 11, № 1. С. 31–56. DOI: 10.14529/cmse220103.
14. *Биленко Р. В., Долганина Н. Ю., Иванова Е. В., Рекачинский А. И.* Высокопроизводительные вычислительные ресурсы Южно-Уральского государственного университета // Вестник ЮУрГУ. Серия: Вычислительная Математика и Информатика. 2022. Т. 11, № 1. С. 15–30. DOI: 10.14529/cmse220102.