Вестник КРАУНЦ. Физ.-мат. науки. 2022. Т. 40. №3. С. 153-164. ISSN 2079-6641

ИНФОРМАЦИОННЫЕ И ВЫЧИСЛИТЕЛЬНЫЕ ТЕХНОЛОГИИ

УДК 519.652 Научная статья

Подходы к решению систем линейных алгебраических уравнений с помощью нейронных сетей

В. А. Галкин[1,2], *Т. В. Гавриленко*[1,2], *А. Д. Смородинов*[1,2]

[1] Сургутский филиал ФГУ ФНЦ НИИСИ РАН 628426, Тюменская область, Ханты-Мансийский автономный округ – Югра, г. Сургут, ул. Энергетиков, д. 4.
[2] БУ ВО «Сургутский государственный университет», 628412, Тюменская область, Ханты-Мансийский автономный округ – Югра, г. Сургут, пр. Ленина, д. 1
E-mail: Sachenka_1998@mail.ru

СЛАУ является основной решения существенного класса задач математического моделирования. Исследование возможности решения СЛАУ с использованием нейронных сетей позволит создать новые подходы в решении задач математического моделирования. Представляется новый способ решения систем линейных уравнений с помощью нейронных сетей. Используются сети прямого распространения и алгоритм стохастического градиентного спуска. Описываются этапы проектирования нейронной сети, а также процесс выбора оптимальной структуры НС, основанный на проведенных вычислительных экспериментах. Приводятся результаты использования нейронных сетей для решения систем линейных уравнений. Обосновывается целесообразность применения НС для задач данного типа.

Ключевые слова: системы линейных алгебраических уравнений, нейронные сети, градиентный спуск

DOI: 10.26117/2079-6641-2022-40-3-153-164

Поступила в редакцию: 03.10.2022 В окончательном варианте: 25.11.2022

Для цитирования. Галкин В. А., Гавриленко Т. В., Смородинов А. Д. Подходы к решению систем линейных алгебраических уравнений с помощью нейронных сетей // *Вестник КРАУНЦ. Физ.-мат. науки.* 2022. Т. 40. № 3. С. 153-164. DOI: 10.26117/2079-6641-2022-40-3-153-164

Финансирование. Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (Выполнение фундаментальных научных исследований ГП 47) по теме № 0580-2021-0007 «Развитие методов математического моделирования распределенных систем и соответствующих методов вычисления».

Введение

Искусственный интеллект занимает ведущие позиции в современном мире. Алгоритмы, основанные на нейронных сетях, используются в анализе данных [1], в компьютерном зрении [2], их применяют для создания модели оценки и управления рисками предприятия [3]. А также для аппроксимации различных математических функций [8, 9, 10]

В данной статье рассматривается возможность использования нейронных сетей для решения системы линейных алгебраических уравнений (СЛАУ). Задачу решения системы линейных алгебраических уравнений можно записать в матричном виде:

$$Ax = b \tag{1}$$

где A – основная матрица системы, x – вектор-столбец неизвестных, b – вектор-столбец свободных членов.

Для решения системы линейных уравнений разработано большое количество разнообразных прямых и итерационных методов для вычисления с помощью ЭВМ. При использовании прямых методов находят точное решение СЛАУ, а также заранее известно требуемое количество операций, которые необходимо выполнить для вычисления неизвестных. В итерационных методах решение СЛАУ находят с некоторой заранее заданной точностью, причем количество вычислений (итераций), необходимых для решения СЛАУ, заранее рассчитать невозможно. предлагаемый в статье метод решения СЛАУ позволит создать новые подходы в решении задач математического моделирования учитывая что СЛАУ является основной решения существенного класса задач математического моделирования.

Предлагаемый в данной работе способ решения заключается в построении нейронной сети с последующим ее обучением решению системы из n неизвестных. Входными сигналами данной сети будут матрица A и вектор-столбец свободных членов b, выходными – вектор-столбец неизвестных переменных x. Предполагается, что если сконструировать и обучить НС решать СЛАУ, то находить решение системы можно будет для матрицы любого вида за заранее известное число шагов.

Генерирование исходных данных

Исходные данные будут генерироваться двумя способами. Необходимо сгенерировать уравнения (1). Для этого требуется задать матрицу коэффициентов A, вектор-столбец неизвестных членов x, а также вектор-столбец свободных членов b. Количество уравнений N

1 Способ А

Для задания исходных данных пер
вым способом выполнялся следующий алгоритм:

1) Задается матрица A, где

$$A = \begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{matrix}, a_{ij} \in (0;100), i,j = 1..n$$

2) На втором шаге задается вектор x, где

$$X = \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{matrix}$$

, $x_i \in (0;100), i = 1..n$; 3) Следующий этап вычисляется вектор $b = A * x$ Шаги 1-3 выполняется N раз, пока не будет сгенерированы исходные данные.

2 Способ Б

Для второго случая необходимо была единичная матрица E:

$$E = \begin{matrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{matrix}$$

Во втором случае алгоритм был следующий: 1) Задается матрица A, где

$$A = \begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{matrix}, a_{ij} \in (0;100), i,j = 1..n$$

2) На втором шаге задается вектор x, который являлся вектором-столбцом единичной матрицы e_i, т.е. $x = e_i$ 3) Следующий этап вычисляется вектор $b = A * x$. Уравнение задано 4) Шаг 2-3 повторяется до тех пор, пока i не примет все значения от 1 до n Шаги 1-4 повторяются N/n раз.

3 Нормализация

После создания исходных данных, которые будут использоваться для обучения и проверки НС, их необходимо нормализовать так, чтобы они соответствовали следующим условиям:

1) принимали небольшие значения — как правило, в диапазоне 0–1;

2) были однородными — т.е. все признаки должны принимать значения из примерно одного и того же диапазона.

Нормализация проходила следующим образом:

За C обозначим расширенную матрицу СЛАУ, полученную из основной матрицы A, путем дописывания справа вектор-столбца свободных членов b

$$\bar{A} = \frac{1}{n^2+n}\sum_{i,j=1}^{n,n+1} C_{i,j}$$

$$C = \frac{C-\bar(A)}{S}$$

, где

$$S = \sqrt{\frac{1}{n^2+n}\sum_{i,j=1}^{n,n+1}(c_{i,j}-\bar{A}_{i,j})^2}$$

Для минимизации возможного переобучения исходные данные СЛАУ были разделены на 3 группы:

1. Тренировочный набор данных ($89,9\%$ СЛАУ) – для обучения модели.
2. Проверочный набор данных ($9,99\%$ СЛАУ) – для оценивания адекватности обучения модели.
3. Контрольный набор данных ($0,11\%$ СЛАУ) – для проверки обученной модели.

Конструирование нейронной сети

Следующий этап решения – конструирование НС. Как уже было сказано, доказанных правил конструирования нет, но есть правила, которые не всегда дают оптимальную структуру сети. Так, например, в [4] говорится, что количество скрытых слоев не должно быть больше, чем размерность входящих данных. Количество нейронов в скрытом слое должно быть столько же, сколько значений принимает вектор-столбец неизвестных. Т.е. если всё множество значений, которые принимает вектор-столбец x от 100 до 1100, то количество скрытых нейронов должно быть $1100-100=1000$ на каждом слое.

Например, для решения СЛАУ размером $2x2$, где неизвестные x могут принимать значения от 0 до 1000, необходима НС с 6 слоями и 1000 нейронов на каждом слое. На этом конструирование НС не закончено, т.к. еще не решен вопрос, какие именно функции активации использовать и на каком слое, а также какую функцию потерь выбрать. Т.к. задачу решения СЛАУ можно отнести к задаче регрессии, т. е. необходимо установить соответствие между «случайными» переменными – входными и выходными данными, то функции активаций можно выбрать, следуя правилам, предложенным в [4], при необходимости их можно будет подкорректировать.

Для первого слоя лучше использовать функцию активации $sigmoid$ или гиперболический тангенс. Для последнего слоя лучше вовсе не использовать функцию активации. Для остальных слоев применить $ReLU$. Функцию потерь выбрали MSE

в ходе перебора разнообразных функций потерь т.к. при использовании именно MSE скорость обучения была наивысшей:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x_i})^2$$

где x - ожидаемый результат нейронной сети, $\bar{x}$- полученный результат. Обучение и настройка весов функции проходит по методу обратного распространения ошибки [5, 6] а в качестве алгоритма оптимизации будет использоваться стохастический градиентный спуск (СГС). Гиперпараметры СГС задавались и изменялись автоматически.

Вычислительный эксперимент 1

Данные для первого эксперимента генерировались по способу A, СЛАУ состояла из 2-х неизвестных, которые принимали значения от 0 до 100. Исходя из этого нейросеть состояла из 6 слоев и 120 нейронов на каждом слое. Нейронов было больше на 20% чем рекомендуется, т.к. результаты тестирования показали, что увеличение нейронов позволяет увеличить скорость обучения.

Функции активации была выбраны согласно описанным выше правилам, но в ходе экспериментов было выявлено, что использовать в качестве первого слоя sigmoid или гиперболический тангенс не оптимально, т. к. скорость сходимости была ниже, чем при использовании на первом слое ReLu. В остальном же НС была оставлена без изменений.

НС обучалась 170 эпох после чего остановилось обучение и дальнейшее уменьшение гиперпараметров не позволяло обучать НС дальше. Точность с которой НС обучилась решать СЛАУ составляет 0.97625554.

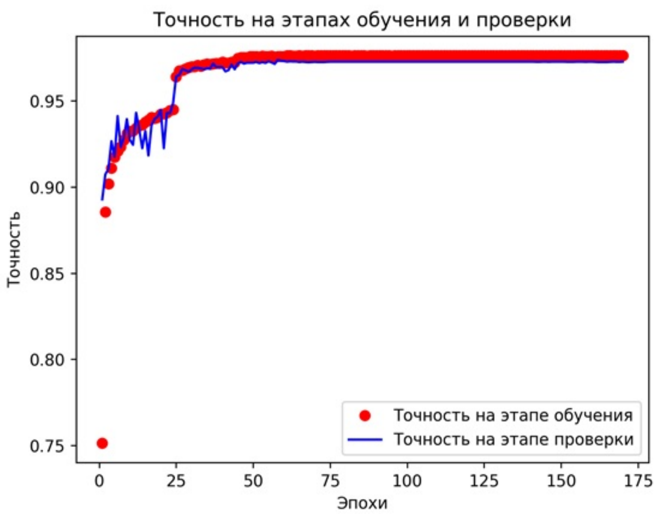


Рис. 1. Точность на этапах обучения и проверки.
[Figure 1 Accuracy in the training and verification phases.]

На рис.1 представлена точность на этапе обучения и проверки. Как видно из данного графика, уже к 75 эпохе точность перестала сильно изменяться. а к 110 эпохе и вовсе изменения были на уровне погрешности.

Часть результаты (10%) тестирования на контрольном наборе данных представлена в Табл. 1. Из данных результатов было видно, что точность решения СЛАУ зависит от числа обусловленности.

Таблица 1

Результаты тестирования НС на неизвестных сети данных [Results of NN testing on unknown data networks].

Номер	СЛАУ	Ответ	Ответ НС	Точность	Число обусловленности
0	61x 52y = 3286 17x 9y = 762	x=30 y=28	x=33,0157 y=2	27,3548 2	20,23416
1	44x 5y = 435 91x 44y = 2347	x=5 y=438	x=6,02325 y=42,78008	79,53492 99,48855	8,09936
2	26x 39y = 2106 24x 74y = 2780	x=48 y=22	x=45,87919 y=20,56331	95,58165 93,46961	8,22765
3	x 36y = 1369 96x 77y = 9780	x=73 y=36	x=76,0724 y=34,31283	95,79115 95,31341	4.65093
4	33x 20y = 1484 83x 78y = 4926	x=18 y=44	x=20,1896 y=41,97576	87,83536 95,2631	15,7593
5	19x 70y = 3817 20x 39y = 2249	x=13 y=51	x=13,12835 y=50,51111	99,01268 99,04139	10,80579
6	15x 14y = 2179 87x 6y = 7299	x=79 y=71	x=79,86122 y=70.48585	98,90985 99,27584	6,97181
7	90x 5y = 1285 34x 45y = 2253	x=12 y=41	x=12,95244 y=42,23759	92,063 96,9815	2,51655
8	4x 14y = 828 48x 98y = 7136	x=67 y=40	x=61.52185 y=43,26383	91.82366 91,84042	43,2626
9	51x 43y = 7511 55x 3y = 5411	x=95 y=62	x=93,0355 y=62,36974	97,93218 99,40365	43,2626
10	25x 78y = 7036 17x 35y = 3215	x=10 y=87	x=7,73597 y=87,09838	77,35969 99,88692	18,1778

В связи с выше сказанным был построен график зависимости точности решения СЛАУ от числа обусловленности матрицы, рассчитанное как максимальное сингулярное число матрицы[7].

Как видно из рис.2 чем больше число обусловленности, тем ниже точность, кроме того на графики видны «дыры» НС, т.е. те места в которых она должна была решить СЛАУ с точностью выше чем это было сделано.

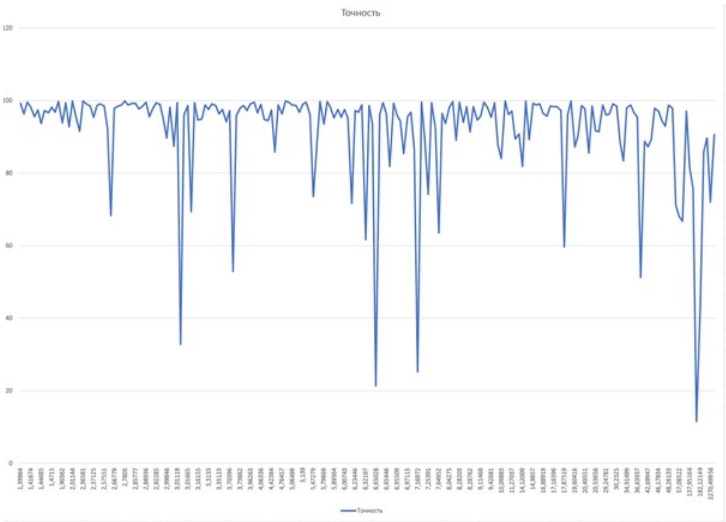


Рис. 2. Зависимость точности решения СЛАУ от числа обусловленности
[Figure 2 Dependence of the accuracy of system linear solution on the condition number.]

Вычислительный эксперимент 2

Для второго эксперимента данные генерировались способом Б. Структура нейронной сети осталось неизменной. В итоге за 280 эпох обучения точность составила 0.9999667.

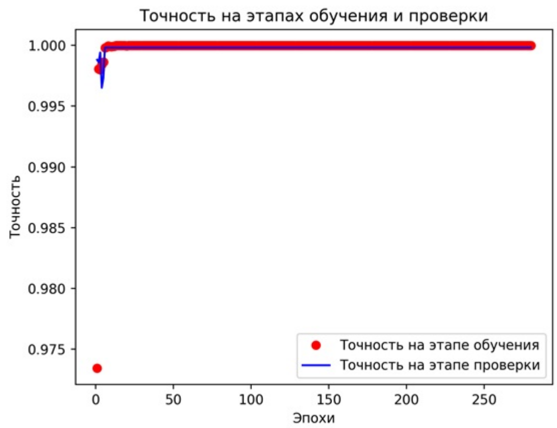


Рис. 3. Точность на этапах обучения и проверки.
[Figure 3 Accuracy during the training and verification phases.]

На рис. 3 представлен график точности на этапах обучения и проверки уже к 25 эпохе точность приблизилась к конечному результату и в дальнейшем изменялась на уровне погрешности, на 280 эпохе обучение было остановлено в виду того, что изменение гиперпараметров не помогало увеличить точность во время обучения.

Таблица 2

Результаты тестирования НС на неизвестных сети данных [Results of NN testing on unknown data networks].

Номер	СЛАУ	Ответ	Ответ НС	Точность	Число обусловленности
0	$\begin{matrix} 54x & 13y \\ 44x & 48y \end{matrix} = \begin{matrix} 54 \\ 44 \end{matrix}$	x=1 y=0	x=1.00621 y=0.00043	99.379 99.957	3,32553
1	$\begin{matrix} 54x & 13y \\ 44x & 48y \end{matrix} = \begin{matrix} 13 \\ 48 \end{matrix}$	x=0 y=1	x=0.00501 y=1.0026	99.499 99.74	3,32553
2	$\begin{matrix} 62x & 76y \\ 78x & 14y \end{matrix} = \begin{matrix} 62 \\ 78 \end{matrix}$	x=1 y=0	x=0.98781 y=0.01057	98.781 98.943	2,78296
3	$\begin{matrix} 62x & 76y \\ 78x & 14y \end{matrix} = \begin{matrix} 76 \\ 14 \end{matrix}$	x=0 y=1	x=0.003041 y=1.00927	99.696 99.073	2,78296
4	$\begin{matrix} 68x & 14y \\ 39x & 90y \end{matrix} = \begin{matrix} 68 \\ 39 \end{matrix}$	x=1 y=0	x=1.0031 y=0.00061	99.69 99.939	2,11882
5	$\begin{matrix} 68x & 14y \\ 39x & 90y \end{matrix} = \begin{matrix} 14 \\ 90 \end{matrix}$	x=0 y=1	x=0.00674 y=1.00448	99.326 99.552	2,11882
6	$\begin{matrix} 85x & 97y \\ 73x & 18y \end{matrix} = \begin{matrix} 85 \\ 73 \end{matrix}$	x=1 y=0	x=0.98766 y=0.00564	98.766 99.436	3.74815
7	$\begin{matrix} 85x & 97y \\ 73x & 18y \end{matrix} = \begin{matrix} 97 \\ 18 \end{matrix}$	x=0 y=1	x=0.0026 y=0.99604	99.926 99.604	3.74815
8	$\begin{matrix} 46x & 89y \\ 31x & 4y \end{matrix} = \begin{matrix} 46 \\ 31 \end{matrix}$	x=1 y=0	x=1.00613 y=0.00373	99.387 99.627	4.02909
9	$\begin{matrix} 46x & 89y \\ 31x & 4y \end{matrix} = \begin{matrix} 89 \\ 4 \end{matrix}$	x=0 y=1	x=0.0026 y=0.99839	99.74 99.839	4.02909
10	$\begin{matrix} 89x & 60y \\ 54x & 6y \end{matrix} = \begin{matrix} 89 \\ 54 \end{matrix}$	x=1 y=0	x=1.00949 y=0.00043	99.051 99.957	5.15448
11	$\begin{matrix} 89x & 60y \\ 54x & 6y \end{matrix} = \begin{matrix} 60 \\ 6 \end{matrix}$	x=0 y=1	x=0.00263 y=0.99986	99.737 99.986	5.15448
12	$\begin{matrix} 39x & 66y \\ 17x & 88y \end{matrix} = \begin{matrix} 39 \\ 17 \end{matrix}$	x=1 y=0	x=0.99245 y=0.00766	99.245 99.234	5,85073
13	$\begin{matrix} 39x & 66y \\ 17x & 88y \end{matrix} = \begin{matrix} 66 \\ 88 \end{matrix}$	x=0 y=1	x=0.01194 y=0.98705	98.806 98.705	5,85073
14	$\begin{matrix} 28x & 51y \\ 72x & 13y \end{matrix} = \begin{matrix} 28 \\ 72 \end{matrix}$	x=1 y=0	x=0.98997 y=0.00627	98.997 99.373	2,18349
15	$\begin{matrix} 28x & 51y \\ 72x & 13y \end{matrix} = \begin{matrix} 51 \\ 13 \end{matrix}$	x=0 y=1	x=0.0116 y=0.99968	98.84 99.968	2,18349
16	$\begin{matrix} 84x & 51y \\ 42x & 56y \end{matrix} = \begin{matrix} 84 \\ 42 \end{matrix}$	x=1 y=0	x=0.99646 y=0.00109	99.646 99.891	5.50007
17	$\begin{matrix} 84x & 51y \\ 42x & 56y \end{matrix} = \begin{matrix} 51 \\ 56 \end{matrix}$	x=0 y=1	x=0.00121 y=0.99964	99.879 99.964	5.50007
18	$\begin{matrix} 75x & 18y \\ 37x & 94y \end{matrix} = \begin{matrix} 75 \\ 37 \end{matrix}$	x=1 y=0	x=1.00885 y=0.00914	99.115 99.086	2.04025
19	$\begin{matrix} 75x & 18y \\ 37x & 94y \end{matrix} = \begin{matrix} 18 \\ 94 \end{matrix}$	x=0 y=1	x0.00889 y=1.00487	99.111 99.513	2.04025

В табл. 2 представлены часть результатов тестирования НС на контрольном наборе данных. Так же, как и в первом случае число обусловленности равно максимальному сингулярному числу матрицы.

Так же, как и в первом эксперименте был построен график зависимости точности от числа обусловленности. Данный график представлен на рис. 4

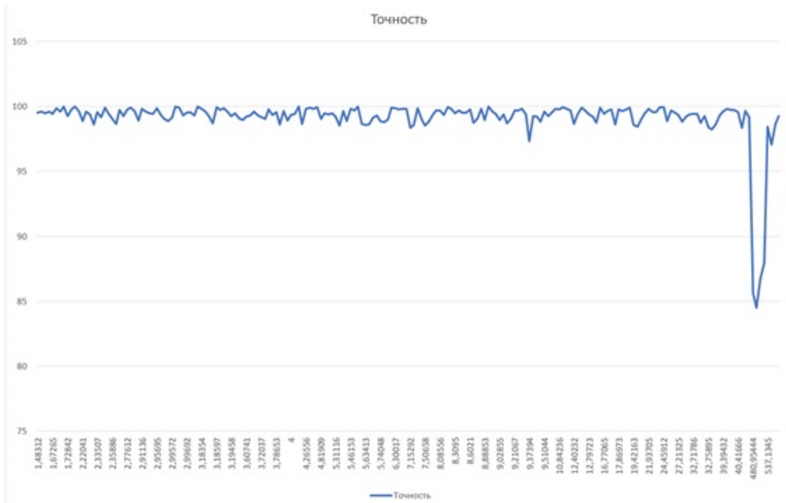


Рис. 4. Зависимость точности решения СЛАУ от числа обусловленности
[Figure 4 Dependence of the accuracy of system linear solution on the condition number.]

Использование способа Б для генерации исходных данных позволил избавиться от «дыр», но тем не менее при числе обусловленности выше 40 образуется резкое уменьшение точности решения СЛАУ. Кроме того, данный способ позволил увеличить точность обучения НС.

Попытка использовать обученную НС на данных, сгенерированных способом Б, для решения СЛАУ сгенерированных способом А успеха не принесло.

Заключение

В данной статье была рассмотрена возможность решения СЛАУ с использованием нейронных сетей. Выдвинуто предположение о том, что данный подход даст возможность решать СЛАУ независимо от вида матрицы. В ходе вычислительных экспериментов было показано, что данное предположение имеет под собой основание, но требует доработки структуры НС, а также проведения большего числа экспериментов.

Сконструирована и обучена НС, способная решать СЛАУ с некоторой погрешностью. Несмотря на то, что НС для некоторых систем выдает полностью неверное решение, дальнейшее изучение поднятой темы имеет смысл. Остается нерешенная проблема, связанная с правильным конструированием и обучением нейронных сетей, однако, найдя оптимальный вид НС и обучив её до требуемой точности, любое СЛАУ можно будет решить, заранее зная, сколько времени требуется для этого.

Дальнейшее исследование возможности решения СЛАУ с использованием нейронных сетей позволит создать новые подходы в решении задач математического моделирования.

Конкурирующие интересы. Конфликтов интересов в отношении авторства и публикации нет.

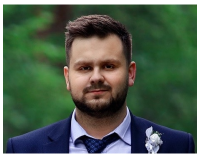
Авторский вклад и ответственность. Каждый из авторов участвовал в написании статьи и полностью несет ответственность за предоставление окончательной версии статьи в печать.

Список литературы

1. Манжула В. Г., Федяшов Д. С. Нейронные сети Кохонена и нечеткие нейронные сети в интеллектуальном анализе данных, *Фундаментальные исследования*, 2011. № 4, С. 108-114.
2. Долотов Е. А. Кустикова В. Д. Сравнение некоторых методов решения задачи детектирования лиц на изображениях, *GraphiCon*, 2017. № 2017, С. 202-207.
3. Корнеев Д. С. Использование аппарата нейронных сетей для создания модели оценки и управления рисками предприятия, *Управление большими системами*, 2007. № 17, С. 81-102.
4. Франсуа Ш. *Глубокое обучение на Python*. СПб: Питер, 2018. 400 с.
5. Попова Ю. Б., Яцынович С. В. *Обучение искусственных нейронных сетей методом обратного распространения ошибки*. Минск: БНТУ, 2016.
6. Ивановский М. Н., Шафеев О. П. Применение метода обратного распространения ошибки для обучения нейронной сети / *Информационные технологии в науке и производстве*, V Всероссийская молодежная научно-техническая конференция, 2018, С. 39–43.
7. Уткин П. С. *Лекция по теме: Введение в решение СЛАУ Нормы векторов и матриц. Число обусловленности матрицы СЛАУ*. М.: МФТИ, 2014 mipt.ru/education.
8. Cybenko, G. V. Approximation by Superpositions of a Sigmoidal function, *Mathematics of Control Signals and Systems*, 1989. vol. 2, pp. 303–314.
9. Shiyu Liang, R. Srikant Why deep neural networks for function approximation?, *Published as a conference paper at ICLR*, 2017.
10. Hanin B. Universal Function Approximation by Deep Neural Nets with Bounded Width and ReLU Activations, *Mathematics*, 2019. vol. 7, pp. 992.

Галкин Валерий Алексеевич – доктор физико-математических наук, профессор, директор, Сургутский филиал ФГУ «ФНЦ НИИСИ РАН», БУ ВО «Сургутский государственный университет», г. Сургут, Россия, ORCID 0000-0002-9721-4026.

Гавриленко Тарас Владимирович – кандидат технических наук, доцент, заместитель директора, Сургутский филиал ФГУ «ФНЦ НИИСИ РАН», БУ ВО «Сургутский государственный университет», г. Сургут, Россия, ORCID 0000-0002-3243-2751.



Смородинов Александр Денисович – аспирант кафедры прикладной математики, преподаватель кафедры АСОИУ, БУ ВО «Сургутский государственный университет»; инженер отдела биофизики, Сургутский филиал ФГУ «ФНЦ НИИСИ РАН» Инженер отдела биофизики, г. Сургут, Россия, ORCID 0000-0002-9324-1844.

Vestnik KRAUNC. Fiz.-Mat. Nauki. 2022. vol. 40. no. 3. pp. 153–164. ISSN 2079-6641

INFORMATION AND COMPUTATION TECHNOLOGIES

MSC 68T07 Research Article

Approaches to solving systems of linear algebraic equations using neural networks

V. A. Galkin[1,2], *T. V. Gavrilenko*[1,2], *A. D. Smorodinov*[1,2]

[1] Surgut Branch of SRISA 628426, Surgut, Energetikov st., 4, Russia
[2] Surgut State University, 628412, Surgut, Lenina st., 1, Russia
E-mail: Sachenka_1998@mail.ru

System linear is the main solution for an essential class of mathematical modeling problems. The study of the possibility of solving system linear using neural networks will allow creating new approaches to solving problems of mathematical modeling. A new way of solving systems of linear equations using neural networks is presented. Feedforward networks and a stochastic gradient descent algorithm are used. The stages of designing a neural network are described, as well as the process of choosing the optimal NN structure, based on the computational experiments performed. The results of using neural networks for solving systems of linear equations are presented. The expediency of using NN for problems of this type is substantiated.

Key words: systems of linear algebraic equations, Neural networks, gradient descent

DOI: 10.26117/2079-6641-2022-40-3-153-164

Original article submitted: 03.10.2022 Revision submitted: 25.11.2022

For citation. Galkin V. A., Gavrilenko T. V., Smorodinov A. D. Approaches to solving systems of linear algebraic equations using neural networks. *Vestnik KRAUNC. Fiz.-mat. nauki.* 2022, **40**: 3, 153-164. DOI: 10.26117/2079-6641-2022-40-3-153-164

Competing interests. The authors declare that there are no conflicts of interest regarding authorship and publication.

Contribution and Responsibility. All authors contributed to this article. Authors are solely responsible for providing the final version of the article in print. The final version of the manuscript was approved by all authors.

Funding. The publication was made within the framework of the state task of the Federal State Institution FNTs NIISI RAS (Performance of fundamental scientific research GP 47) on topic No. 0580-2021-0007 «Development of methods for mathematical modeling of distributed systems and corresponding calculation methods»

References

[1] Manzhula V. G., Fedyashov D. S. Kohonen Neural Networks and Fuzzy. Fundamental'nyye issledovaniya, 2011, 4, 108-114.

[2] Dolotov E. A. Kustikova V. D. Comparison of some methods for solving the problem of detecting faces in images, GraphiCon, 2017, 2017, 202-207.

[3] Korneev D. S. Using the apparatus of neural networks to create a model for assessing and managing enterprise risks, Upravleniye bol'shimi sistemami, 2007, 17, 81-102

[4] Francois Ch. Glubokoye obucheniye v Python [Deep Learning in Python], Moscow, Mir, 2018, 400.

[5] Popova Yu. B., Yatsynovich S. V. Obucheniye iskusstvennykh neyronnykh setey metodom obratnogo rasprostraneniya oshibok, Minsk, BNTU, 2016.

[6] Ivanovsky M. N., Shafeev O. P. Primeneniye metoda obratnogo rasprostraneniya oshibok dlya obucheniya neyronnoy seti, Informatsionnyye tekhnologii v nauke i proizvodstve, V Vserossiyskaya molodezhnaya nauchno-tekhnicheskaya konferentsiya, 2018, 39–43.

[7] Utkin P.S. Lektsiya po teme: Vvedeniye v resheniye SLAU Normy vektorov i matrits. Chislovyye znacheniya matritsy SLAU, Moscow, MFTI, 2014, mipt.ru/education

[8] Cybenko G. V. Approximation by Superpositions of a Sigmoidal function, Mathematics of Control Signals and Systems, 1989, 2, 303–314.

[9] Liang S., Srikant R. Why deep neural networks for function approximation? Published as a conference paper at ICLR, 2017.

[10] Hanin B. Universal Function Approximation by Deep Neural Nets with Bounded Width and ReLU Activations, Mathematics, 2019, 7, 992.

Galkin Valery Alekseevich – D. Sci. (Phys. & Math.), Professor, Surgut State University; Director, Branch of SRISA, Surgut, Russia, ORCID 0000-0002-9721-4026.

Gavrilenko Taras Vladimirovich – PhD (Tech.), docent, Surgut State University; Deputy Director, Branch of SRISA, Surgut, Russia, ORCID 0000-0002-3243-2751.

Smorodinov Aleksandr Denisovich – Postgraduate Student of the Department of Applied Mathematics, Lecturer of the Department of ASOIU, Surgut State University; Engineer of the Department of Biophysics and Neurocybernetics, Branch of SRISA, Surgut, Russia, ORCID 0000-0002-9324-1844.