ВЕСТНИК ЮГОРСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
2019 г. Выпуск 1 (52). С. 41–54

DOI: 10.17816/byusu20190141-54
УДК 681.5:621.396.6

А. И. Бредихин

АЛГОРИТМЫ ОБУЧЕНИЯ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

В данной статье мы рассматриваем один из наиболее используемых классов нейронных сетей – сверточные нейронные сети (далее – СНС). В частности, описываются области их применения, алгоритмы распространения сигнала по СНС и обучения СНС и приводятся методы реализации алгоритмов функционирования СНС на языке программирования MATLAB. В статье приводятся результаты исследований эффективности алгоритма обучения СНС при решении с его помощью задач классификации. В ходе данных исследований рассматривается такая характеристика нейронной сети как динамика значений ошибки сети в зависимости от скорости обучения, а также проводится проверка корректности работы алгоритма обучения сверточной нейронной сети. В данном случае в качестве задачи классификации используется задача распознавания рукописных цифр на выборке MNIST.

Ключевые слова: искусственный интеллект, сверточная нейронная сеть, алгоритм обратного распространения ошибки, перцептрон, классификация, распознавание образов.

A. I. Bredikhin

TRAINING ALGORITHMS FOR CONVOLUTIONAL NEURAL NETWORKS

In this article we consider one of the most used classes of neural networks – convolutional neural networks (hereinafter CNN). In particular, the areas of their application, algorithms of signal propagation by CNN and CNN training are described and the methods of CNN functioning algorithms implementation in MATLAB programming language are given. The article presents the results of research on the effectiveness of the CNN learning algorithm in solving classification problems with its help. In the course of these studies, such a characteristic of the neural network as the dynamics of the network error values depending on the learning rate is considered, and the correctness of the algorithm of learning convolutional neural network is checked. In this case, the problem of handwritten digits recognition on the MNIST sample is used as a classification task.

Keywords: artificial intellect, convolutional neural network, algorithm of error backpropagation, perceptron, classification, pattern recognition.

Введение

СНС – одна из разновидностей нейронных сетей, предназначенных для эффективного анализа преимущественно двумерных и трехмерных (RGB-изображения) данных (например, распознавания объектов на изображениях). Архитектура СНС впервые была предложена французским ученым в области искусственного интеллекта и машинного обучения Яном Лекуном в конце 80-х годов XX века. Он же в 1989 году вместе с другими учеными впервые применил СНС для распознавания рукописных цифр, результаты которого описаны в статье [1].

Согласно данным результатам СНС имеет следующие преимущества:

- **бóльшая временнáя эффективность** по сравнению с перцептроном за счет меньшего количества настраиваемых параметров;
- **улучшенные способности выделения отдельных элементов на изображении** (углы, кривые, прямые, яркие области и т. д.) за счет использования нескольких карт признаков на одном слое;
- **способность формирования высокоуровневых признаков на основе низкоуровневых в пределах одного класса** за счет использования ядер свертки сравнительно не-

большого размера вместо соединения нейронов двух соседних слоев по принципу «каждый с каждым», как у полносвязного перцептрона;

В настоящее время СНС часто используются при распознавании и классификации объектов на двумерных изображениях, анализе спектрограмм и других двумерных наборах данных. Широкое использование СНС объясняется достаточно высоким качеством решения вышеуказанных задач.

Целью настоящей работы является проведение подробного анализа эффективности алгоритмов обучения СНС при решении с его помощью задач классификации. Анализ эффективности алгоритмов обучения СНС включает в себя исследование динамики значений ошибки сети в зависимости от скорости обучения и проверку корректности работы алгоритма обучения СНС. Алгоритмы реализованы на языке программирования MATLAB. В данном случае в качестве задачи классификации используется задача распознавания рукописных цифр на выборке MNIST. В качестве алгоритмов функционирования СНС будут рассматриваться алгоритмы, предложенные в работах [4] (полносвязные слои) и [6] (сверточные слои и слои пулинга).

Архитектура СНС

Рассмотрим типовую архитектуру СНС на для задач классификации объектов на рисунке.

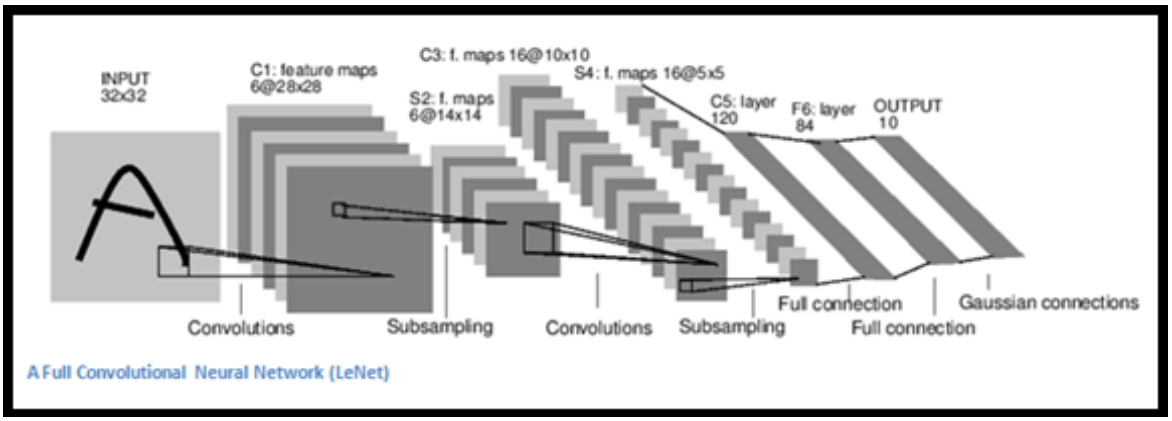

Рисунок 1 – Типовая архитектура СНС для задач классификации объектов на изображении [2]

Как можно увидеть из рисунка 1, СНС состоит из следующих типов слоев (слева направо):

1. **Сверточный** (convolutional) – данный слой производит свертку входной матрицы с **ядром свертки**. Количество ядер свертки определяет количество карт признаков – первое равно второму.

2. **Подвыборочный** (subsampling), или **слой пулинга** (pooling) – данный слой принимает результат свертки предыдущего слоя в виде матрицы и сжимает данную матрицу. Делается это с целью выделения низкоуровневых признаков и понижения размера данных. В качестве функции сжатия чаще всего используется **среднее арифметическое элементов** по окну или **максимальное значение** по окну.

3. **Полносвязный** (full connection). На данный слой подается одномерный вектор от стоящего перед ним сверточного/подвыборочного слоя, причем данный вектор получен из матрицы путем записи ее элементов построчно в одну строку.

Выделим основные особенности архитектуры СНС для задач классифкации объектов на изображении:

- Входной (input) слой СНС является сверточным, а выходной (output) – полносвязным.
- Сверточные и подвыборочные слои чередуются между собой, а после их чередования следуют полносвязные слои (не менее 1). Таким образом, конечная часть СНС представляет собой не что иное, как полносвязный перцептрон.

– В самом простом случае количество ядер свертки во всех сверточных слоях равно 1. Такая СНС, скорее всего, также сможет успешно анализировать числовые матрицы, вычисляемые при помощи математических методов (например, спектрограммы). **Данное предположение и будет доказываться экспериментальным путем в данной статье** (см. раздел «Исследование алгоритма обучения СНС на конкретном примере» данной статьи).

Функционирование СНС. Распространение сигнала

Как было сказано ранее, СНС получила свое название из-за операции свертки. При прямом распространении сигнала по сети в момент его прохождения через **сверточный слой** СНС выполняется valid-свертка [3] по следующей схеме:

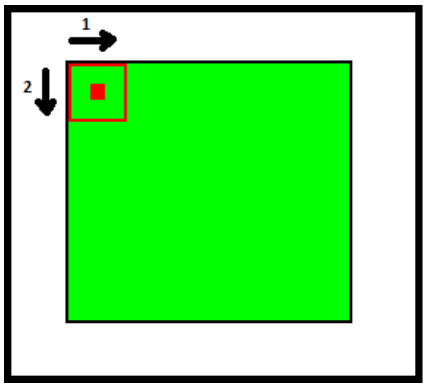

Рисунок 2 – Операция valid-свертки [3]

Замечание. На рисунке 2 цифрами 1 и 2 обозначены приоритеты операций сдвига ядра – оно сначала сдвигается слева направо до правой границы матрицы и, достигнув ее, сдвигается на 1 элемент сверху вниз (при этом перемещаясь к левой границе матрицы).

При valid-свертке ядро накладывается на область левого верхнего угла матрицы и затем производится поэлементное умножение соответствующих элементов матрицы и ядра с последующим суммированием полученных произведений. После выполнения указанных операций ядро смещается на 1 элемент слева направо и данные операции повторяются снова. То есть свертка матриц в данном случае выполняется следующим способом:

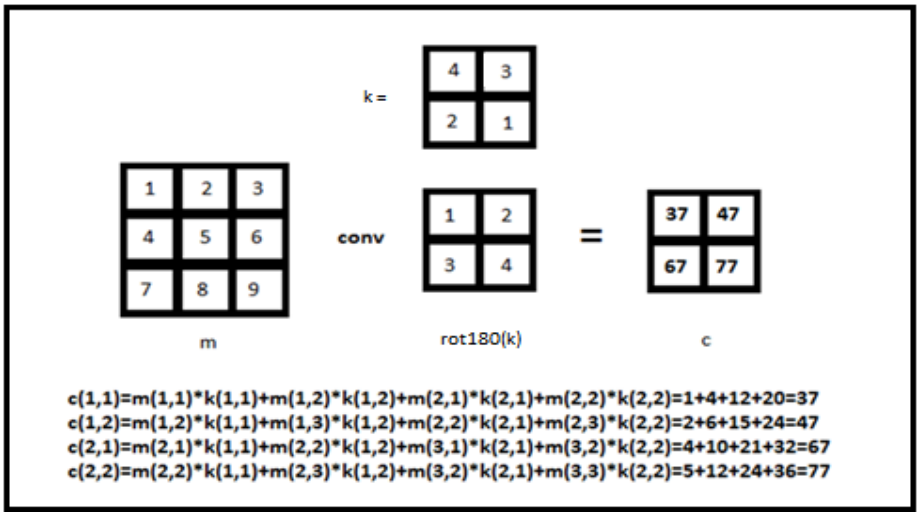

Рисунок 3 – Операция прямой свертки матриц более детально

При свертке матриц размер матрицы – результата свертки будет отличаться от размера исходной матрицы в меньшую сторону. Т. е. при свертке матрицы размером $m_1 * n_1$ с ядром размером $m_2 * n_2$ размер матрицы – результата свертки будет равен $(m_1 - m_2 + 1) * (n_1 - n_2 + 1)$. Данное выражение выполняется и для указанного на рисунке 3 случая *((3-2+1)*(3-2+1)=2*2).*

После выполнения операции свертки производится *вычисление значения функции активации* от каждого элемента получившейся матрицы.

Когда сигнал попадает на **подвыборочный слой**, производится разбиение матрицы-сигнала на области размером $m*n$ (числа m и n – целые). Для каждой области вычисляется среднее/максимум ее элементов и в итоге получается уменьшенная в $m*n$ раз матрица.

Перед помещением сигнала на **полносвязный слой** сигнал преобразуется из двумерных матриц в одномерные векторы. Данный вектор получается из матрицы путем записи ее элементов построчно в одну строку, т. е. следующим образом:

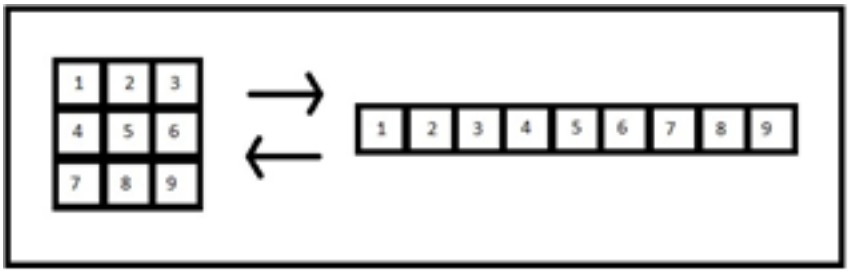

Рисунок 4 – Преобразование матрицы в одномерный вектор-строку

Рассмотрим структуру полносвязного слоя (пусть первый слой нейронов состоит из m нейронов, а второй – из n нейронов):

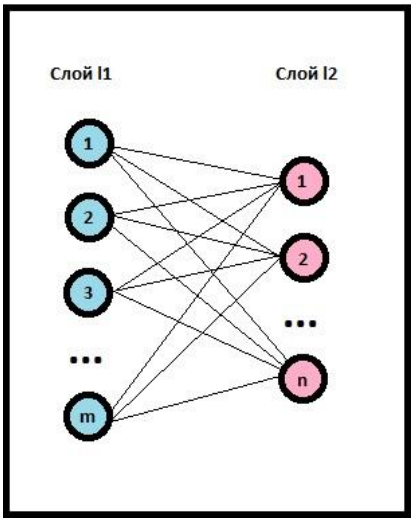

Рисунок 5 – Структура полносвязного слоя

В полносвязном слое НС каждый нейрон первого слоя нейронов связан с каждым нейроном второго слоя нейронов (т. е. по принципу «каждый с каждым»). Вычисление значений нейронов второго слоя производится по следующим формулам [4]:

$$l_{2j} = F(S_{2j}),\ j = \overline{1..n}, \tag{1}$$

$$S_{2j} = (\sum_{i=1}^{m} l_{1i} * \omega_{ij}) - T_{2j}, \tag{2}$$

где F – значение функции активации от взвешенной суммы S_{2j};

l_{1i} и l_{2j} – значения i-го нейрона первого слоя нейронов и j-го нейрона второго слоя соответственно;

ω_{ij} – значение связи между i-м нейроном первого слоя нейронов и j-м нейроном второго слоя соответственно;

T_{2j} – значение порога (смещения) j-го нейрона второго слоя.

Функционирование СНС. Обучение сети

Для обучения сверточной нейронной сети используется специфическая версия **алгоритма обратного распространения ошибки (АОРО)**, который относится к методам обучения с учителем.

Рассмотрим **обучение полносвязных слоев СНС**. Ошибка формируется на последнем слое нейронов СНС и определяется как разность между выходной реакцией сети (значениями нейронов последнего слоя нейронов) *y* и эталоном *t* [4, с. 63]:

$$\gamma_j = y_j - t_j. \tag{3}$$

Далее происходит изменение значений весов и порогов по следующим формулам [4, с. 63]:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha\gamma_j F'(S_j) y_i, \tag{4}$$

$$T_j(t+1) = T_j(t) + \alpha\gamma_j F'(S_j), \tag{5}$$

где α – скорость обучения сети;

t и *t+1* – моменты времени до и после изменения весов и порогов соответственно;

индексы *i* и *j* обозначают нейроны первого и второго слоя нейронов соответственно.

Ошибка для скрытого слоя с индексом *i* вычисляется через ошибки следующего за ним слоя с индексом *j* следующим образом [4, с. 63]:

$$\gamma_i = \sum_j \gamma_j F'(S_j)\omega_{ji}. \tag{6}$$

Полносвязные слои обучаются по *процедуре обучения Розенблатта*, согласно которой значение скорости обучения постоянно в процессе всего времени обучения и принимает значения в промежутке (0;1] [5, с. 43].

Перед попаданием на сверточный слой или слой пулинга *одномерный сигнал преобразовывается в двумерный* по той же схеме (см. рис. 4).

Рассмотрим обучение **сверточного** и **подвыборочного слоев СНС**. Далее в качестве алгоритмов обучения данных слоев будут рассмотрены алгоритмы, предложенные в статье [6].

Обратное распространение ошибки **по подвыборочному слою** зависит от функции пулинга. Если функция пулинга – **среднее**, то ошибка равномерно распространяется по $m*n$ нейронам блока предыдущего слоя, причем она должна быть умножена на $\frac{1}{m*n}$. Т. е. для каждого из $m*n$ нейронов блока **значение ошибки одинаково**. Если же функция пулинга – **минимум**, то ошибка присваивается тому нейрону блока, с которого было взято максимальное значение по блоку.

Основой обратного распространения ошибки **по сверточному слою** служит **операция свертки**. При передаче матрицы ошибок от слоя пулинга ко сверточному слою производится **обратная свертка** матрицы ошибок слоя пулинга с ядром матрицы по следующей схеме:

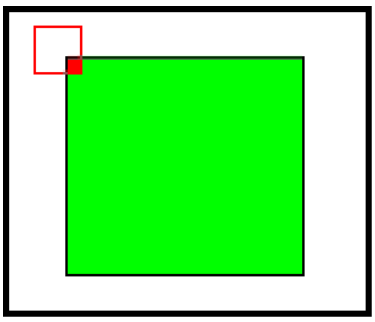

Рисунок 6 – Схематичное изображение операции обратной свертки[3]

При выполнении обратной свертки на выходе получается матрица большего размера, чем входная матрица. Т. е. размер выходной матрицы становится равным $(m_1 + m_2 - 1) * (n_1 + n_2 - 1)$.

Затем производится свертка входа сверточного слоя с матрицей ошибок данного слоя, повернутой на 180 градусов, по следующей схеме:

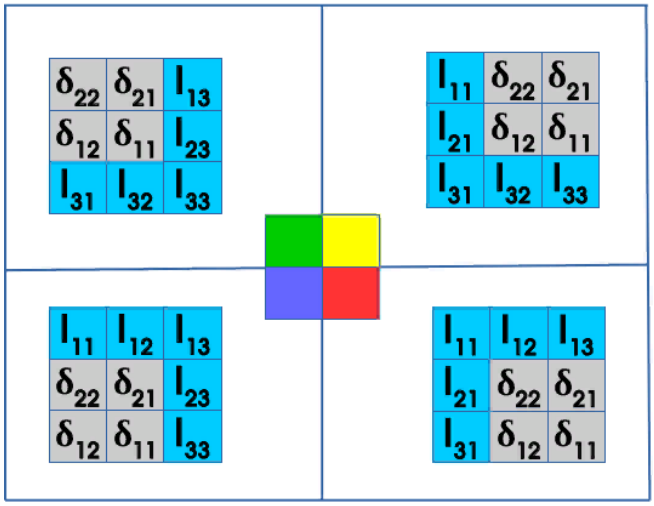

Рисунок 7 – Схема свертки входа сверточного слоя с матрицей ошибок[6]

Затем полученная матрица умножается на скорость обучения α, и вычитается из ядра свертки данного слоя. Таким образом производится изменение весов в ядре свертки сверточного слоя СНС.

Исследование алгоритма обучения СНС на конкретном примере

В целях выполнения данной работы была реализована собственная СНС по вышеописанным алгоритмам функционирования и обучения СНС. Данная СНС была применена для решения **задачи классификации**. Это одна из задач, для решения которых применяются СНС.

Задача классификации заключается в следующем [7]: имеется множество *объектов*, разделённых некоторым образом на *классы*. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется **обучающей выборкой.** Классовая принадлежность остальных объектов неизвестна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

Помимо обучающей выборки, на которой обучается нейронная сеть, используется **тестовая выборка** – множество объектов, не входящее в обучающую выборку и служащее для валидации нейронной сети.

Для решения задачи классификации с помощью нейронной сети в качестве функции потерь рекомендуется использовать **перекрестную энтропию**, а в качестве функции активации последнего слоя СНС – функцию **Softmax** [8].

Перекрестная энтропия применяется для вычисления ошибки между выходной реакцией сети и эталонным вектором. Она задается следующей формулой:

$$CE(\vec{y}, \hat{\vec{y}}) = -\sum_{i=1}^{k} \hat{y}_i \log y_i \qquad (7)$$

где y – вектор выходных значений сети, а $\hat{y}$ – вектор эталонных значений сети для определенного входного образца.

Функция Softmax и ее производная для некоторого выходного вектора z вычисляются по следующим формулам:

$$\sigma(\vec{z}_j) = \frac{\exp(z_j)}{\sum_{i=1}^{n} \exp(z_i)} \tag{8}$$

$$\sigma'(\vec{z}_j) = \sigma(\vec{z}_j) * (1 - \sigma(\vec{z}_j)) \tag{9}$$

При применении функции Softmax к выходному сигналу на выходе НС получаются значения вероятностей принадлежности входного образа к тому или иному классу.

В рамках данной работы была рассмотрена следующая задача: распознавание рукописных цифр собственной СНС. В качестве набора данных использовалась широко известная в кругах специалистов по машинному обучению выборка рукописных цифр MNIST [9]. Обучающая выборка MNIST содержит 60000 образцов цифр, а тестовая – 10000 образцов цифр. Каждый образец представляет собой двумерную матрицу чисел от 0 до 255 размером 28*28 пикселей. Примеры цифр из выборки MNIST представлены ниже.

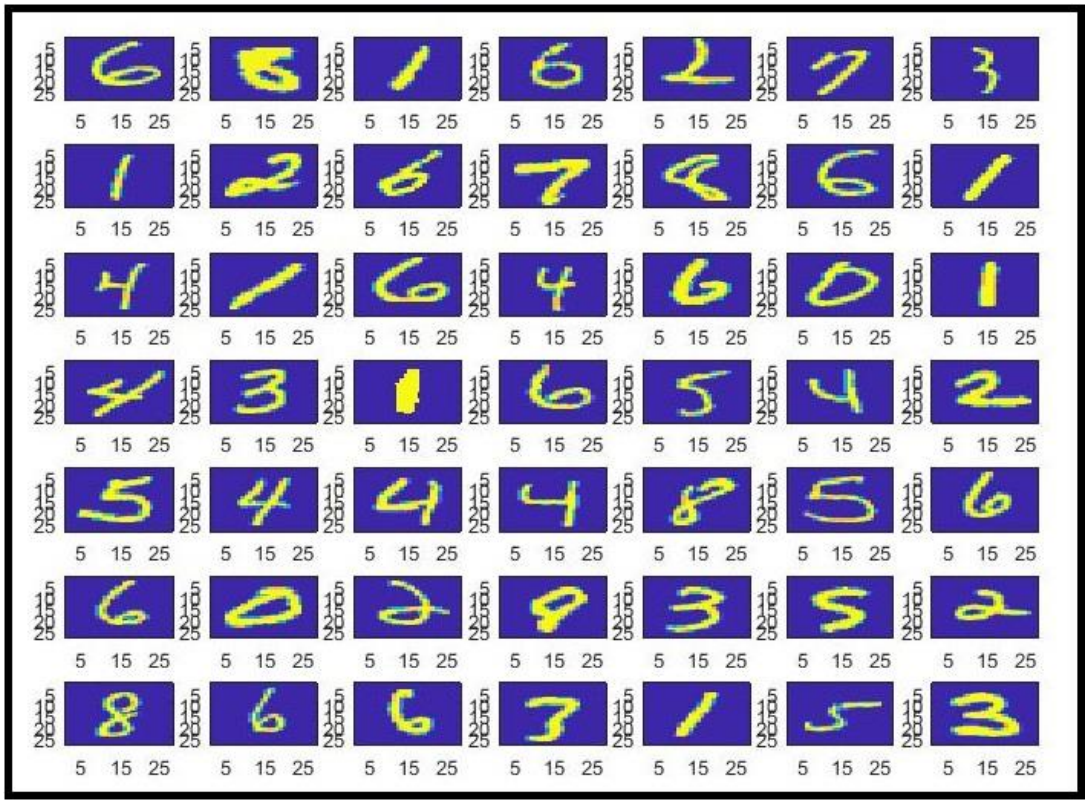

Рисунок 8 – Примеры цифр из выборки MNIST

Выборка MNIST перед обучением была подвергнута **нормализации**, т. е. все значения матриц были поделены на 256 с целью их приведения в промежуток [0;1].

Затем была сконструирована нейронная сеть следующей архитектуры:

Архитектура собственной СНС

Номер слоя	Тип слоя	Размер слоя (входа)	Размер ядра свертки	Функция активации
1	Сверточный	28*28	7*7	ReLU
2	Подвыборочный	22*22	-	-
3	Сверточный	11*11	4*4	ReLU
4	Подвыборочный	8*8	-	-
5	Плоский	4*4	-	-
6	Полносвязный	16	-	ReLU
7	Полносвязный	30	-	Softmax
8	Полносвязный	10	-	-

Замечание. Количество карт признаков для всех сверточных слоев СНС принято равным 1 (упрощенная СНС), опция пулинга – среднее, весовые коэффициенты задаются случайными значениями в промежутке [–0.5;0.5].

После этого были сформулированы следующие **задачи исследования**:

1. Определить, при каких значениях скорости обучения выполняется обучение СНС без переобучения с наиболее заметной сходимостью. Длительность обучения – 250 эпох, количество образцов, проходящих через СНС за 1 эпоху, – 15, значения скорости обучения – {1, 0.4, 0.2, 0.1, 0.04, 0.02, 0.01, 0.004, 0.002, 0.001} (10 значений). **Цель задачи** – определение по динамике ошибки сети оптимального значения скорости обучения СНС по процедуре обучения Розенблатта. **Критерий** оптимального значения скорости обучения СНС – постепенно убывающее значение ошибки СНС в процессе обучения без резких скачков.

2. Выбрать одно из оптимальных значений скорости обучения и продолжить обучение СНС с уменьшенными значениями скорости обучения из того же списка. **Цель задачи** – определить наивысшую точность распознавания цифр данной СНС.

Замечание. Точность СНС проверялась на тестовой выборке цифр MNIST (10000 образцов).

Результаты выполнения **задачи 1**:

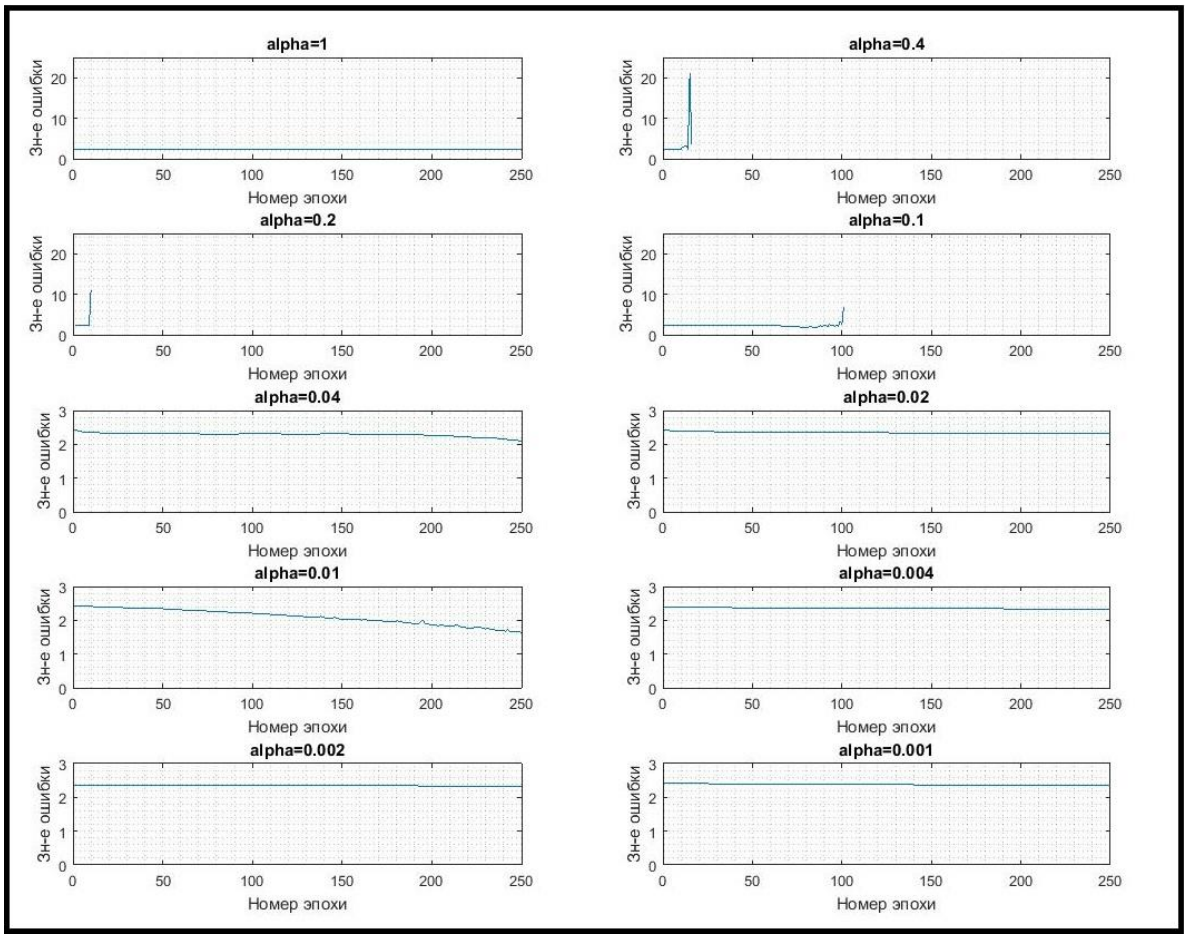

Рисунок 9 – Результаты выполнения задачи 1

Процесс обучения СНС при различных значениях скорости обучения:

- 1 – сходимость не наблюдается, ошибка сети постоянна (ок. 2,3);
- 0,4 – на 15–20 эпохе ошибка СНС становится равной NaN – произошло переобучение сети: она стала выдавать один и тот же ответ вне зависимости от входного сигнала. Перед этим наблюдается резкий скачок ошибки;
- 0,2 – на 10–12 эпохе ошибка СНС становится равной NaN – произошло переобучение сети с предшествующим резким скачком ошибки;

- 0,1 – на 100–105 эпохе ошибка СНС становится равной NaN – произошло переобучение сети с предшествующим резким скачком ошибки;
- 0,04 – переобучения не происходит, наблюдается медленная сходимость сети (уменьшение ошибки);
- 0,02 – переобучения не происходит, наблюдается незначительная сходимость сети;
- 0,01 – переобучения не происходит, наблюдается **значительная** сходимость сети;
- 0,004 – переобучения не происходит, наблюдается незначительная сходимость сети;
- 0,002 – переобучения не происходит, наблюдается незначительная сходимость сети;
- 0,001 – переобучения не происходит, наблюдается незначительная сходимость сети.

Из этого следует, что при неизменной скорости обучения сеть необходимо обучать при значениях скорости обучения **в районе 0,01**.

При значениях скорости обучения 0,04 и 0,02 наблюдается меньшая сходимость, чем при скорости обучения, равной 0,01, несмотря на то, что данные значения больше. Это дает основания предположить, что при данных значенияхдовольно часто происходят значительные «перескоки» сети через локальные минимумы таким образом, что нейронная сеть становится «дальше» от локального минимума, чем была до смены весовых коэффициентов.

А при значениях скорости обучения 0,004–0,001 также наблюдается меньшая сходимость, чем при скорости обучения, равной 0,01. Но объяснение в данном случае более банально: сходимость к локальному минимуму происходит гораздо медленнее из-за меньшего значения скорости ошибки.

Таким образом, в рамках выполнения **задачи 2** была обучена сеть при начальном значении скорости обучения 0,01 и количестве эпох, равном 1500. В случае переобучения сети выбирался снимок сети на определенной эпохе с наименьшим значением ошибки. Затем процесс обучения продолжался уже при уменьшенной скорости обучения (0,004). В случае повторного переобучения сети данные действия повторялись (значения скорости обучения 0,002 и 0,001).

В процессе выполнения **задачи 2** фиксировались значения следующих наиболее значимых выходных параметров обучения: ошибка сети, средние значения F-меры, точности и полноты.

Результаты выполнения **задачи 2:**

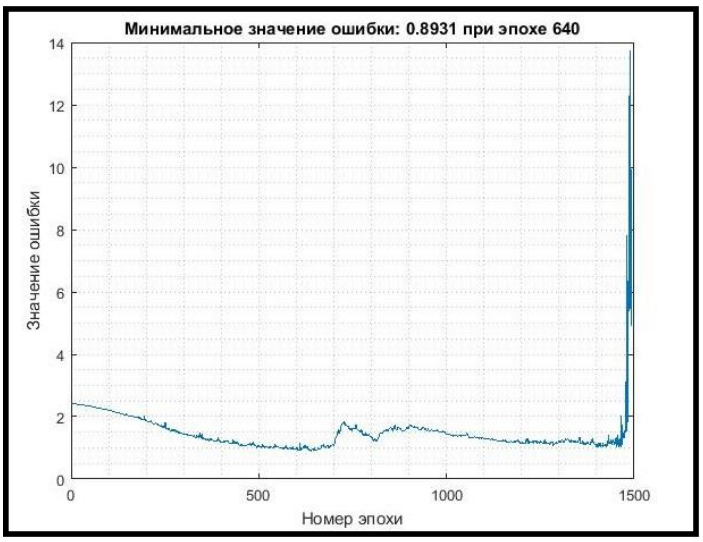

Рисунок 10 – Динамика ошибки сети при значении скорости обучения 0,01

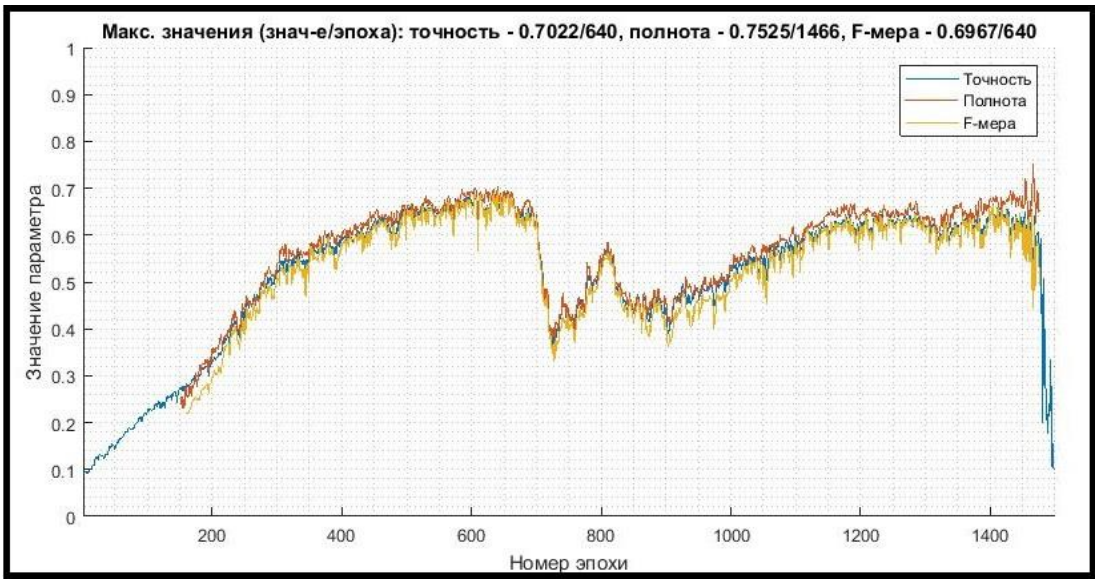

Рисунок 11 – Значения наиболее значимых выходных параметров обучения при скорости обучения 0,01

При значении скорости обучения сети, равном 0,01, наименьшая ошибка сети составила 0,8931; наибольшая точность на тестовой выборке составила 70,22 %.

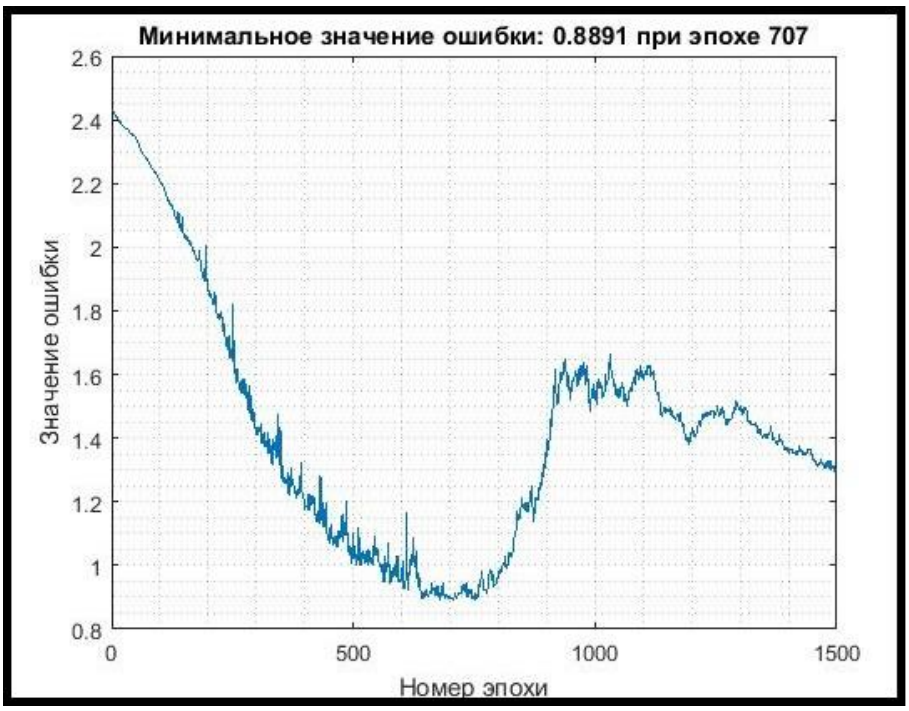

Рисунок 12 – Динамика ошибки сети при значении скорости обучения 0,004

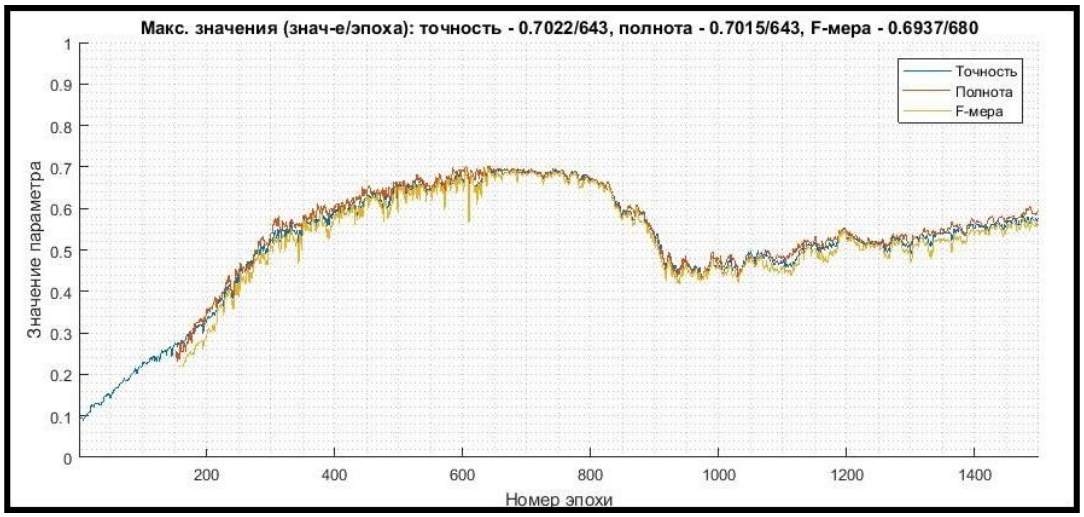

Рисунок 13 – Значения наиболее значимых выходных параметров обучения при скорости обучения 0,004

В процессе дообучения сети при значении скорости обучения сети, равном 0,004, наименьшая ошибка сети составила 0,8891 (незначительное снижение); наибольшая точность на тестовой выборке составила 70,22 % (без изменений).

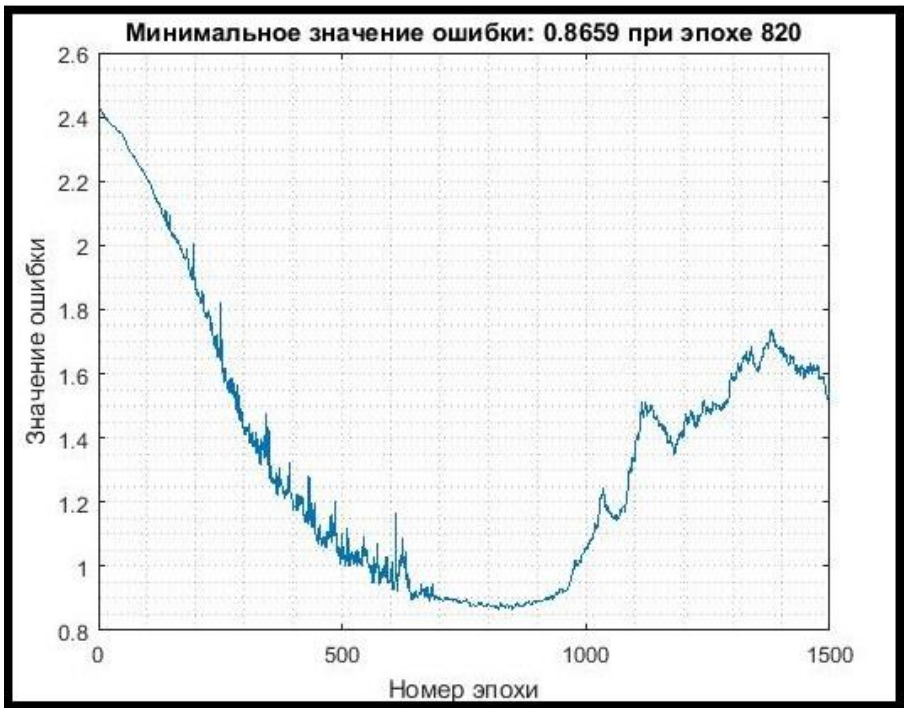

Рисунок 14 – Динамика ошибки сети при значении скорости обучения 0,002

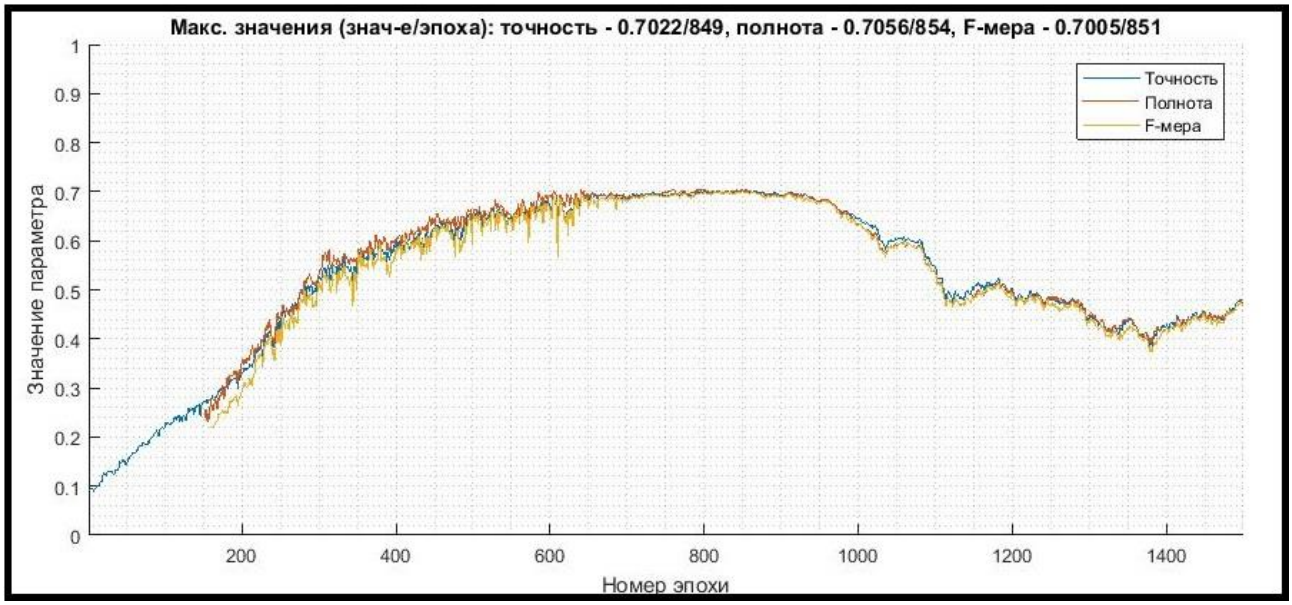

Рисунок 15 – Значения наиболее значимых выходных параметров обучения при скорости обучения 0,002

В процессе дообучения сети при значении скорости обучения сети, равном 0,002, наименьшая ошибка сети составила 0,8659 (ощутимое снижение); наибольшая точность на тестовой выборке составила 70,22 % (без изменений).

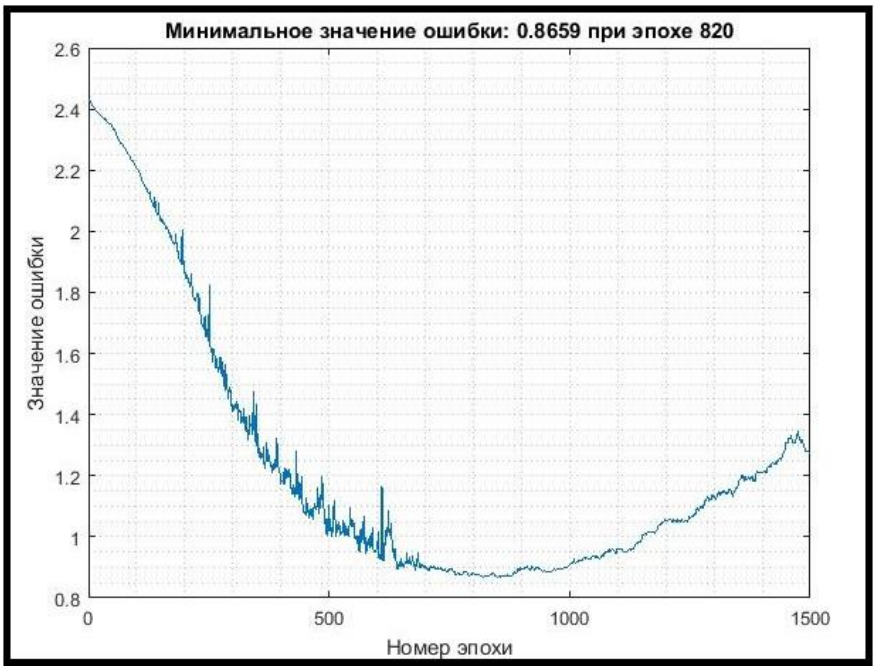

Рисунок 16 – Динамика ошибки сети при значении скорости обучения 0,001

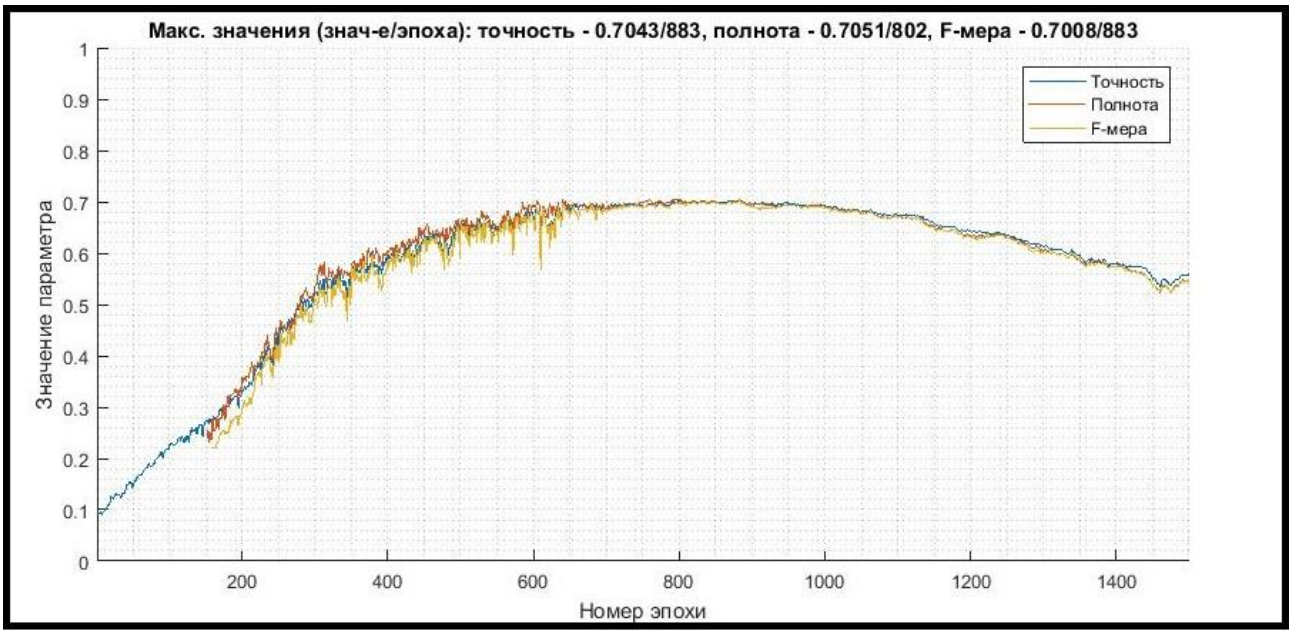

Рисунок 17 – Значения наиболее значимых выходных параметров обучения при скорости обучения 0,001

В процессе дообучения сети при значении скорости обучения сети, равном 0,001, наименьшая ошибка сети составила 0,8659 (без изменений); наибольшая точность на тестовой выборке составила 70,43 % (незначительное увеличение).

В процессе обучения значение точности распознавания СНС достигло близкого к максимально возможному значению при используемых в данном исследовании алгоритмах и настройках обучения, поскольку при снижении скорости обучения сети с 0,002 до 0,001 и дальнейшем обучении СНС значение ошибки сети не снизилось.

Максимальное значение точности СНС составило всего 70,43 %. Можно выделить несколько основных **причин**, из-за которых точность распознавания цифр СНС в данном случае невысока:

1. **Упрощенная архитектура СНС**. В данном случае количество карт признаков равно 1 для всех сверточных слоев. Из-за этого СНС обретает способность эффективно обнаруживать только один какой-нибудь признак цифры (прямые, углы, закругления), что негативно сказывается на точности распознавания цифр. Этому есть логическое объяснение – некоторые цифры могут иметь различное начертание (например, цифры 2 могут иметь как низ в виде прямой линии, так и петлю в левой нижней части). Применение более сложной архитектуры СНС с увеличенным количеством карт признаков не входило в планы данной работы.

2. **Обучение с постоянной скоростью обучения.** Для большей сходимости обучения необходимо использовать переменную скорость обучения, зависящую от значения ошибки.

3. **Несовершенство самого алгоритма обучения СНС.** В статье [6] не описано использование порогов для сверточных слоев, так же, как и для полносвязных слоев, и, следовательно, там отсутствует правило изменения порогов для сверточных слоев.

Заключение

В результате выполнения данной работы был проанализирован один из алгоритмов обучения СНС и произведена его реализация на ЯП программирования MATLAB с целью проверки корректности его работы. Исходя из результатов проведенного исследования, данный алгоритм имеет способность к обучению, что наблюдается по динамике ошибки в процессе обучения СНС.

Реализованная СНС была обучена на проверенном наборе данных – наборе рукописных цифр MNIST. Сеть в данном случае показала невысокую точность на тестовой выборке –

70,43 %. По результатам исследования выделены следующие возможные причины невысокой точности распознавания:

- упрощенная архитектура СНС (используется всего по одной карте признаков на каждом сверточном слое);
- обучение с постоянной скоростью обучения.

Литература:

1. Backpropagation Applied to Handwritten Zip Code Recognition / Y. LeCun [et al.] // Neural Computation. – 1989. – Vol. 1, iss. 4. – P. 541–551.

Борисов, Е. С. Классификатор изображений на основе сверточной нейронной сети [Электронный ресурс] / Е. С. Борисов. – Электрон. текстовые дан. // Дом-страница Евгения Сергеевича Борисова. – [Россия], 2016. – Режим доступа: http://mechanoid.kiev.ua/ml-lenet.html (дата обращения 14.12.2018).

2. Татьянкин, В. М. Подход к формированию архитектуры нейронной сети для распознавания образов / В. М. Татьянкин // Вестник Югорского государственного университета. – 2016. – № 2 (41). – С. 61–64.

3. Головко, В. А. Нейронные сети: обучение, организация и применение. Кн. 4 : учеб. пособие для вузов / В. А. Головко ; общ. ред. А. И. Галушкина. – Москва : ИПРЖР, 2001 – 256 с. – (Нейрокомпьютеры и их применение).

4. Backpropagation In Convolutional Neural Networks [Electronic resource]. – Electronic text data // DeepGrid. – [S. l.], 2016. – Режим доступа: https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/ (дата обращения 24.12.2018).

5. Классификация [Электронный ресурс]. – Электрон. текстовые дан. // MachineLearning.ru : проф. информ.-аналит. ресурс, посвящ. машин. обучению, распознаванию образов и интеллект. анализу данных. – [Б. и.], 2011. – Режим доступа: http://www.machinelearning.ru/wiki/ index.php?title=Классификация (дата обращения 24.12.2018).

6. Величкович, П. Глубокое обучение для новичков: распознаем рукописные цифры [Электронный ресурс] : [перевод] / П. Величкович. – Электрон. текстовые дан. // Хабр. – [Москва], 2016. – Режим доступа: https://habr.com/company/wunderfund/blog/314242/ (дата обращения 24.12.2018).

7. LeCun, Y. The MNIST database of handwritten digits [Electronic resource] / Y. LeCun, C. Cortes, C. Burges. – Electronic text data // Yann LeCun. – New York, [201?]. – Режим доступа: http://yann.lecun.com/exdb/mnist/ (дата обращения 25.12.2018).