

Робастное, адаптивное и сетевое управление

© 2020 г. П.Ш. ГЕЙДАРОВ, канд. техн. наук (plbaku2010@gmail.com)
(Институт систем управления НАН Азербайджана, Баку)

СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕЗУЛЬТАТОВ ОБУЧЕНИЙ НЕЙРОННОЙ СЕТИ С ВЫЧИСЛЕННЫМИ ВЕСОВЫМИ ЗНАЧЕНИЯМИ И С ГЕНЕРАЦИЕЙ ВЕСОВЫХ ЗНАЧЕНИЙ СЛУЧАЙНЫМ ОБРАЗОМ

Нейронные сети на основе метрических методов распознавания позволяют на основе начальных условий задачи распознавания, таких как количество образов и эталонов, определить структуру нейронной сети (количество нейронов, слоев, связей), а также позволяют аналитически вычислять значения весов связей нейронной сети. Будучи нейронными сетями прямого распространения, эти сети могут также обучаться классическими алгоритмами обучения. Возможность предварительного вычисления значений весов нейронной сети позволяет утверждать, что процедура создания и дообучения нейронной сети прямого распространения ускоряется по сравнению с классической схемой создания и обучения нейронной сети, в которой значения весов генерируются случайным образом. В статье выполняются два эксперимента на базе рукописных цифр MNIST, подтверждающие это утверждение.

Ключевые слова: нейронные сети, метрические методы распознавания, метод ближайшего соседа, алгоритм обратного распространения ошибки, случайная инициализация весов.

DOI: 10.31857/S0005231020070041

1. Введение

В современном мире применение нейронных сетей получило широкое применение, в особенности в задачах распознавания образов. Несмотря на это, на практике создание и обучение нейронных сетей остается сложной и часто непредсказуемой работой. Это главным образом связано с тем, что процесс создания и обучения нейронных сетей [1, 2] не является строго определенным, что создает ряд сложностей и делает этот процесс трудоемким. Сложности заключаются как в выборе самой структуры нейронной сети, так и в выборе параметров обучения.

В публикациях [3, 4] предложена архитектура нейронной сети, реализующая метрические методы распознавания [5]. Структура этих нейронных сетей: количества нейронов, связей, слоев строго определяются на основе начальных условий задачи метрических методов распознавания [5], таких

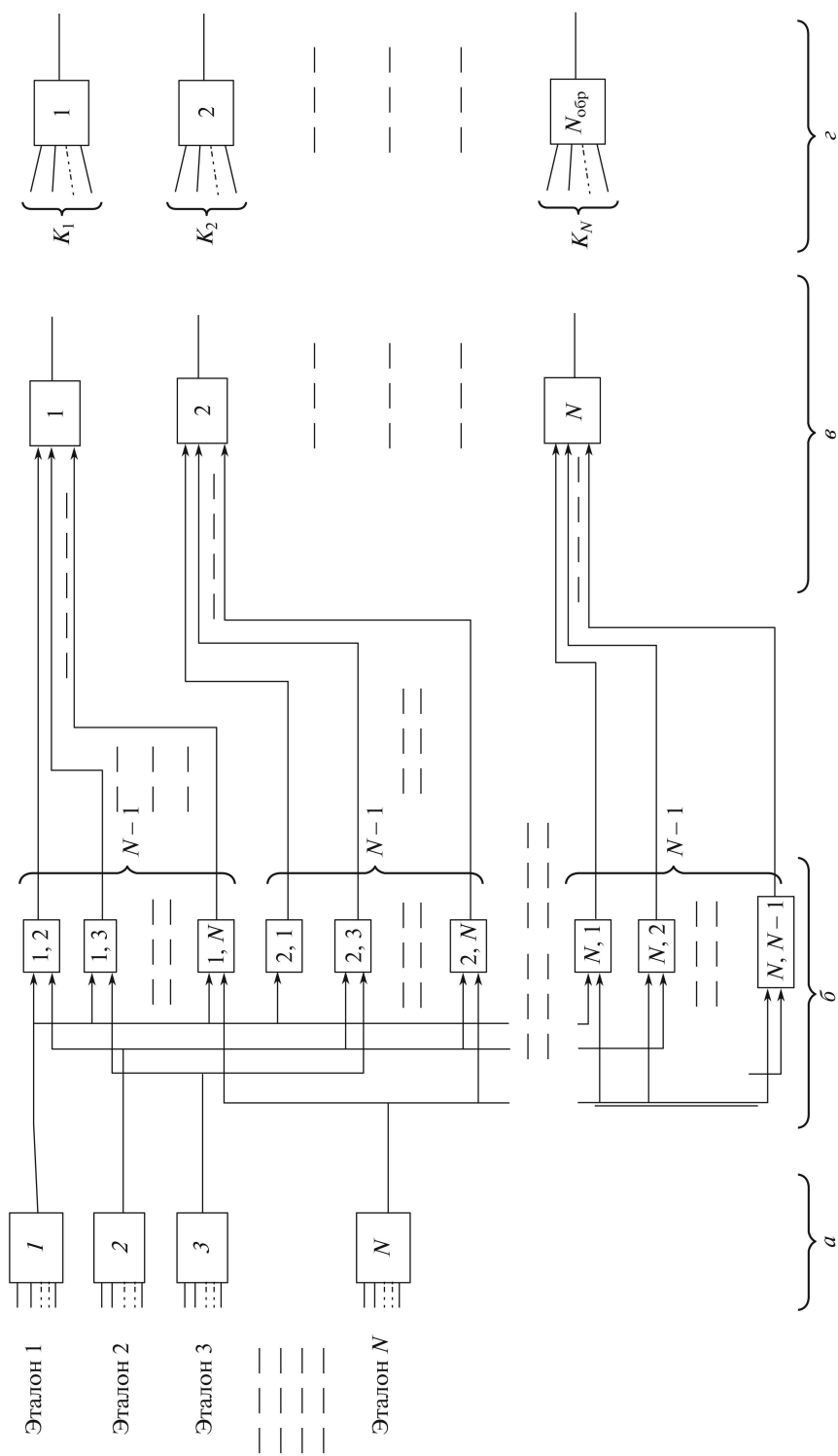


Рис. 1. Четырехслойная нейронная сеть для задачи распознавания $N_{\text{обр}}$ образов с N эталонами: a — нулевой слой; $б$ — первый слой; $в$ — второй слой; $г$ — третий слой.

как количество используемых эталонов и количество распознаваемых образов. Значения весов связей для этих сетей также вычисляются аналитически на основе метрических мер близости [5]. Такая возможность уже позволяет получить работающую нейронную сеть без применения алгоритмов обучения. Нейронные сети на основе метрических методов распознавания являются частным случаем классического трех–четырёхслойного многослойного перцептрона, но при этом архитектуры этих сетей позволяют определить структуру нейронной сети и аналитически определять значения весов. Кроме того, архитектура этих сетей позволяет каскадно добавлять в нейронную сеть новые эталоны и распознаваемые образы без изменения предыдущих весовых значений, что также выгодно отличает эти сети от классических сетей прямого распространения, в том числе и от глубоких сверточных сетей [6, 7].

Нужно сказать, что существуют нейронные сети с предварительно определяемыми весами, такие, например, как сети Хопфилда и Хемминга [1, 2]. Но эти сети не являются сетями прямого распространения, имеют обратные связи и в связи с этим имеют ряд сложностей и нерешенных проблем, среди которых и такие, как проблема неустойчивости этих сетей. Помимо этого, предварительное фиксирование значений весов иногда применяется и в нейронных сетях прямого распространения. Это так называемый подход “замораживания весов” [8], что, например, применяется в случаях, когда необходимо сократить время обучения путем замораживания веса скрытого нейрона, если его выход не изменяется существенно в течение обучения, или когда необходимо воспользоваться преимуществами ранее обученной нейронной сети и использовать ее веса для дальнейшей настройки данной нейронной сети. Но эти подходы не определяют значения весов, а направлены на корректирование и ускорение процесса алгоритма обучения.

Поскольку нейронные сети на основе метрических методов распознавания являются сетями прямого распространения, то эти сети могут также обучаться и классическими алгоритмами обучения [1, 2]. В [9, 10] был приведен обобщенный алгоритм определения значений весов для второго и третьего слоя полносвязной нейронной сети, полученной на основе нейронной сети на основе метрических методов распознавания (рис. 1). Данный алгоритм позволяет вычислять всевозможные диапазоны значений весов и порогов нейронной сети второго и третьего слоев, при которых логика работы нейронной сети на рис. 1 сохраняется неизменной. Там же было сделано предположение, что процесс вычисления и последующее за этим дообучение нейронной сети будет выполняться быстрее, чем это происходит классическим способом, когда обучение нейронной сети выполняется на основе случайно сгенерированных значений весов. Но такое предположение могло быть и ошибочным, поскольку не исключает того, что процесс дообучения может, наоборот, уничтожить результативность, основанную на предварительно вычисленных значениях весов, и в результате, возможно, еще больше удлинять процесс обучения. Другими словами, для проверки предположения в публикациях [9, 10] необходимо наличие экспериментального подтверждения. С этой целью в данной статье выполняется сравнительный эксперимент на базе MNIST обучений одной и той же нейронной сети как с вычисленными, так и со случайно сгенерированными весовыми значениями. Кроме того, такой эксперимент также позволит

проверить работоспособность нейронных сетей на основе метрических методов распознавания на большой тестируемой базе, такой как база MNIST, а также проверить возможность работы данных сетей с непрерывными функциями активаций и соответственно возможность дообучения этих сетей алгоритмом обратного распространения ошибки. Принципиальная разница от классических методов при таком подходе будет заключаться в ускорении процедуры создания и обучения нейронной сети благодаря предварительному вычислению значений весов связей нейронов, тогда как в классических схемах создания и обучения нейронной сети ускорение реализуется при помощи использования более быстрых алгоритмов обучения нейронной сети со случайной инициализацией весов.

Отметим, что возможность ускорения создания и обучения нейронной сети посредством предварительного вычисления весов может быть особо актуальной для будущих нейронных сетей, близких к возможностям биологических нейронных сетей, где количество распознаваемых образов будет значительным или даже огромным.

Цель статьи — выполнить два эксперимента по обучению одной и той же нейронной сети с одинаковым количеством эпох обучения как с предварительно вычисленными весами, так и с инициализацией значений весов случайными числами. В качестве базы данных используется база MNIST изображений рукописных цифр. Окончательная цель заключается в сравнении полученных результатов двух экспериментов с целью оценки как по результативности распознанных цифр базы MNIST, так и по общему потраченному времени создания и обучения нейронной сети.

2. Основные положения нейронных сетей на основе метрических методов распознавания

Метрические методы распознавания — это методы, которые определяют принадлежность распознаваемого объекта к тому или иному образу в некоторой определенной признаковой системе координат на основе наименьшего значения метрической характеристики близости к эталону или к группе эталонов образа [5]. Здесь под “эталонами” понимаются выбранные (выделенные) образцы от каждого образа в существующей выборке данных. Работоспособность метрических методов распознавания основана на гипотезе компактности, которая предполагает, что элементы одного класса (образа) в некоторой признаковой системе координат находятся компактно друг к другу. В качестве метрических характеристик близости для одной точки (ячейки таблицы, пикселя) могут использоваться разные метрические выражения, например выражения среднеквадратичной разности (2.1), (2.2):

$$(2.1) \quad w_{ij,k} = (y_{etal} - j)^2,$$

где $w_{ij,k}$ — значение веса таблицы весов с координатами i, j для k -го нейрона нулевого слоя, y_{etal} — ближайшая по вертикали к координате j активная ячейка бинарной матрицы. Выражение (2.1) применимо, например, для за-

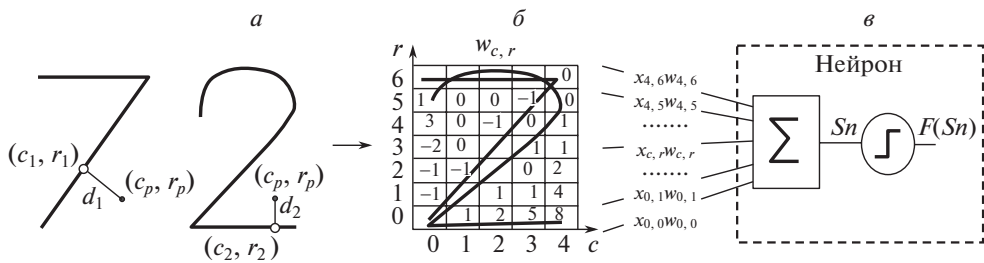


Рис. 2. Пример таблицы весов с размерностью 5×7 для пары эталонов $\{7, 2\}$: *a* — формирование таблицы весов для нулевого слоя; *б* — таблица весов для нейрона первого слоя; *в* — нейрон первого слоя.

дачи распознавания кривых

$$(2.2) \quad w_{c,r}^{(0)} = d_1^2 = \left((c_1 - c_p)^2 + (r_1 - r_p)^2 \right).$$

Формула (2.2) — еще одно возможное выражение определения значения веса для нейрона нулевого слоя, где (c_p, r_p) — координаты точки (ячейки таблицы весов), для которой вычисляется значение веса, (c_1, r_1) — ближайшие координаты активных точек (ячеек таблицы весов) для точки (ячейки) с координатами (c_p, r_p) (рис. 2, *a*).

К метрическим методам относятся такие методы, как: метод построения эталонов (метод эталона), метод ближайшего соседа, метод ближайших N соседей, метод потенциальных функций и т.д. [5]. Приведенная на рис. 1 схема нейронной сети реализует метод ближайшего соседа, алгоритм которого реализуется следующим образом:

- Определяется характеристика (коэффициент) близости, например по выражениям (2.1), (2.2), для каждого эталона;
- Определяется минимальное значение коэффициента близости;
- По принадлежности ближайшего эталона к тому или иному образу определяется наименование ближайшего образа (класса).

В нейронной сети на рис. 1 каждому выбранному эталону соответствует один нейрон в нулевом слое и согласно п. 1 алгоритма метода ближайшего соседа в каждом нейроне нулевого слоя определяется суммарное значение коэффициента близости между входным элементом X и эталоном, которому соответствует данный нейрон:

$$(2.3) \quad Sn_k^{(0)} = \sum_{i=1}^R \sum_{j=1}^C x_{ij} w_{i,j},$$

где $Sn_k^{(0)}$ — значение функции состояния k -го нейрона нулевого слоя, x_{ij} — значение ячейки бинарной матрицы входного распознаваемого элемента. R, C — число колонок и рядов таблицы весов и бинарной матрицы. При этом

здесь нейроны нулевого слоя являются линейными нейронами, т.е. нейронами, для которых функция активации равна функции состояния нейрона:

$$(2.4) \quad f(Sn_k^{(0)}) = Sn_k^{(0)}.$$

Далее, согласно п. 2 алгоритма метода ближайшего соседа определяется эталон (нулевой нейрон) с минимальным значением коэффициента близости $Sn_k^{(0)}$, который соответствует самому ближайшему эталону к распознаваемому элементу X . Для этого в первом слое выполняется попарное сравнение всех выходов нулевого слоя, рис. 1,б, например, для первого нейрона первого слоя функция состояния определяется выражением

$$(2.5) \quad Sn_1^{(1)} = w_2^{(1)} f(Sn_2^{(0)}) - w_1^{(1)} f(Sn_1^{(0)}),$$

где $w_2^{(1)} = w_1^{(1)} = 1$, а функция активации для этого нейрона определяется условиями:

$$(2.6) \quad \begin{aligned} f(Sn_1^{(1)}) &= 1, & \text{если } Sn_1^{(1)} < 0, \\ f(Sn_1^{(1)}) &= 0, & \text{если } Sn_1^{(1)} > 0. \end{aligned}$$

А в нейроне второго слоя (рис. 1,в) выполняется суммирование всех выходов первого слоя, соответствующих одному эталону (одному нейрону нулевого слоя):

$$(2.7) \quad Sn_k^{(2)} = \sum_{j=1, j \neq k}^N f(Sn_{k,j}^{(1)}),$$

где активный выход второго слоя выделяется пороговым значением $H^{(2)} = N - 1$ и определяет ближайший эталон к распознаваемому объекту X :

$$(2.8) \quad \begin{aligned} f(Sn_k^{(2)}) &= 1, & \text{если } Sn_k^{(2)} \geq (N - 1) = H^{(2)}, \\ f(Sn_k^{(2)}) &= 0, & \text{если } Sn_k^{(2)} < (N - 1) = H^{(2)}. \end{aligned}$$

Каждый k -й нейрон третьего слоя нейронной сети суммирует выходы нейронов второго слоя, принадлежащих к эталонам одного k -го образа

$$(2.9) \quad Sn_k^{(3)} = \sum_{i \in k}^{K_k} f(Sn_i^{(2)}),$$

где K_k — количество эталонов для k -го распознаваемого образа, и проверяет на активность хотя бы один вход нейрона при помощи функции активации:

$$(2.10) \quad \begin{aligned} f(Sn_k^{(3)}) &= 1, & \text{если } Sn_k^{(3)} > 0, \\ f(Sn_k^{(3)}) &= 0, & \text{если } Sn_k^{(3)} \leq 0. \end{aligned}$$

Значения весов для каждого входа нейронов второго и третьего слоев вычисляются либо по приведенному в [9, 10] обобщенному алгоритму определения значения весов второго и третьего слоя, либо в простейшем случае принимаются равными единице:

$$(2.11) \quad w_{i,j}^{(2)} = w_{i,j}^{(3)} = 1.$$

Таким образом, согласно п. 3 алгоритма метода ближайшего соседа порядковый номер активного выхода третьего слоя определяет ближайший образ для распознаваемого элемента X .

На рис. 1 приведена расширенная схема нейронной сети на основе метрических методов распознавания, реализующая метод ближайшего соседа. Нейронная сеть строится на основе набора выбранных эталонов и количества распознаваемых в задаче образов (классов).

На схеме рис. 1 количество нейронов второго слоя равно количеству используемых эталонов $n^{(2)} = N$, а количество нейронов третьего слоя соответствует количеству распознаваемых образов $n^{(3)} = N_{\text{обр}}$. Количество нейронов 1-го слоя в расширенной схеме на рис. 1,б определяется количеством всевозможных пар эталонов:

$$(2.12) \quad n^{(1)} = N(N - 1).$$

Нейронная сеть на рис. 1 может быть и без нулевого слоя, тогда для каждого нейрона первого слоя вычисляется таблица весов, где значения весов связей нейронов первого слоя вычисляются аналитически на основе метрических выражений близости двух нейронов нулевого слоя, например, выражением

$$(2.13) \quad w_{c,r}^{(1)} = d_1^2 - d_2^2 = \left((c_1 - c_p)^2 + (r_1 - r_p)^2 \right) - \left((c_2 - c_p)^2 + (r_2 - r_p)^2 \right),$$

где (c_p, r_p) — координаты точки (ячейки таблицы весов), для которой вычисляется значение веса, (c_1, r_1) и (c_2, r_2) — ближайшие координаты активных точек (ячеек таблицы весов) для точки (ячейки) с координатами (c_p, r_p) (рис. 2,а и 2,б). В качестве выражений меры близости могут использоваться и другие более простые или сложные выражения, отличные от (2.13). В данной статье при создании нейронной сети используется выражение (2.13).

Отметим, что на основе схемы на рис. 1 с незначительными дополнениями и поправками могут быть также реализованы другие метрические методы распознавания.

3. Создание нейронной сети и вычисление значений весов

В данной статье нейронная сеть создается на базе MNIST. Это означает, что набор эталонов набирается из базы MNIST (рис. 3), размерность таблицы весов определяется также на основе размерности матрицы изображений базы MNIST, превышая ее в два раза и будет равна 28×56 . Дальнейшее дообучение сети выполняется также на базе MNIST.

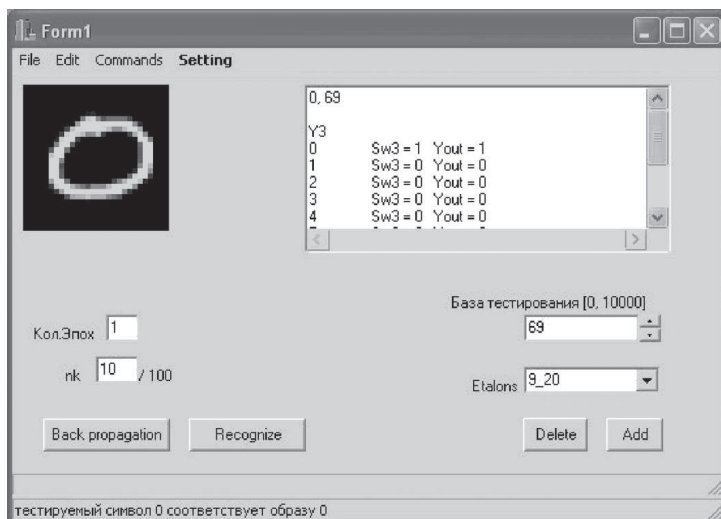


Рис. 3. Программный модуль для распознавания и обучения изображений базы MNIST, реализованный в среде Builder C++.

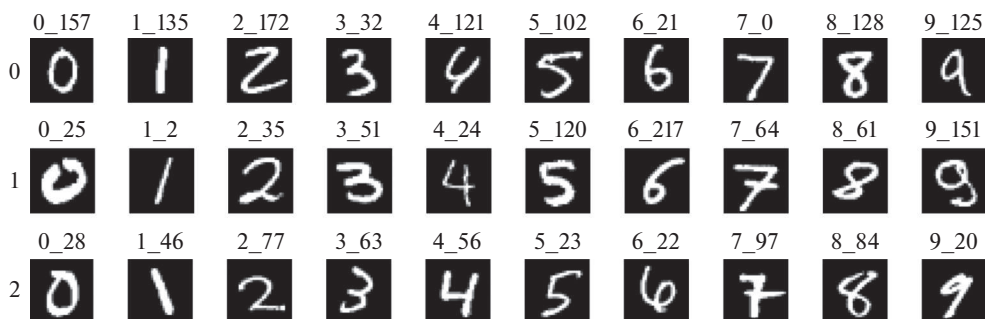


Рис. 4. Выбранные эталоны из контрольной базы MNIST, наименование изображения включает название образа и порядковый номер изображения в базе MNIST.

Напомним, что база MNIST состоит из обучающей базы, в который входят 60000 изображений рукописных цифр, и контрольной (тестовой) базы, состоящей из 10000 образцов изображений цифр. Каждой базе также прилагается своя база наименований цифр, порядок расположения которых такой же, как и порядок расположения изображений цифр в обучающей и контрольной базах. Изображения цифр в базах описаны в виде матрицы цифр размерностью 28×28 . Каждая цифра матрицы определяет значение тональности одного пикселя изображения цифры в диапазоне $[0, 255]$.

В качестве эталонов было выбрано по три образца изображений цифр от каждого образа. Итого использовалось 30 эталонов, приведенных на рис. 4. Эталоны были выбраны интуитивно из первых 250 изображений цифр контрольной базы MNIST. На рис. 4 над каждым выбранным эталоном приве-

дено наименование изображения, например 2_1, в котором первая цифра указывает образ, которому принадлежит изображение цифры, вторая цифра указывает порядковый номер данного изображения цифры в базе MNIST. Далее по тексту изображения базы MNIST будут обозначаться так, как представлено. Поскольку количество эталонов равно 30, то в соответствии с архитектурой нейронных сетей, реализующих метрические методы распознавания, количество нейронов второго слоя будет равно тоже 30, где каждый i -й выход соответствует i -му эталону и определяет принадлежность распознаваемого изображения к i -му эталону.

Для эталонов в рассматриваемой задаче выбран порядок расположения в той последовательности, который приведен на рис. 4. Сначала столбец с изображениями цифры “0”, потом — цифры “1” и т.д. В соответствии с этим схема сети на рис. 1 определяет и выходы второго слоя. Например, эталону 0_157 соответствует $k = 0$ выход второго слоя, а эталону 5_23 соответствует $k = 5 \cdot 3 + 3 - 1 = 17$ выход второго слоя и т.д. Количество нейронов третьего слоя равно количеству распознаваемых образов цифр, $n^{(3)} = 10$. Каждый i -й выход третьего слоя определяет принадлежность распознаваемого элемента к i -му образу цифры. Порядок образов цифр определен последовательно от 0 до 9.

Количество нейронов первого слоя определяется выражением (2.12):

$$(3.1) \quad n^{(1)} = 30 \cdot 29 = 870.$$

Для того чтобы упростить и ускорить вычисление таблиц весов первого слоя, вычисляются таблицы весов нулевого слоя [4] по формуле (2.2), на основе которых далее вычисляются таблицы весов первого слоя (2.13).

В процессе распознавания или обучения для каждого входного распознаваемого изображения составляется бинарная матрица, которая состоит из двух частей. В первой части бинарной матрицы единицы определяют светлые пиксели изображения, значения которых >150 и, наоборот, нули определяют затемненные пиксели изображения, значения которых <150 . Другая часть матрицы зеркально противоположна, определяет активными ($=1$) темные пиксели изображений (<150) и неактивными ($=0$) — светлые пиксели изображений (>150). Размерность бинарной матрицы равна 28×56 . Соответственно каждая таблица весов нулевого слоя соразмерна с бинарной матрицей и состоит тоже из двух частей, приведенных на рис. 5. На рис. 5 можно видеть, что в отличие от входной бинарной матрицы распознаваемых изображений в таблицах весов нулевого слоя активным пикселям изображения соответствуют нули. Для каждого эталона вычисляется своя таблица весов нулевого слоя, подобная таблице на рис. 5.

Таблица весов первого слоя определяется на основе таблиц весов нулевого слоя путем вычитания двух попарно сравниваемых таблиц весов нейронов нулевого слоя выражением

$$(3.2) \quad \overline{W}_{i,j}^{(1)} = \overline{W}_i^{(0)} - \overline{W}_j^{(0)}.$$



Рис. 8. Образцы изображений цифр из контрольной базы MNIST.

второго слоев и равно $H^{(2)} = N - 1 = 30 - 1 = 29$, для третьего слоя $H^{(3)} = 0$. Пороговое значение, представленное в качестве веса, противоположно по знаку пороговому значению H , ($Wh2 = -H = -29$).

В качестве функции активации нейронов использовалась сигмоидная функция активации

$$(3.3) \quad f(Sw) = \frac{1}{1 + e^{-Sw}}.$$

Поскольку для схемы сети на рис. 1 все весовые значения нейронов второго и третьего слоев являются положительными, то соответственно и все выходы нейронов третьего слоя, вычисленные с сигмоидной функцией активации,

Таблица 1. Выходы третьего слоя нейронной сети с пороговой и сигмоидальной функцией активации при распознавании символа 2_174 (рис. 8) из контрольной базы MNIST

с пороговой функцией активации		
0	Sw3 = 0	Yout = 0
1	Sw3 = 0	Yout = 0
2	Sw3 = 1	Yout = 1
3	Sw3 = 0	Yout = 0
4	Sw3 = 0	Yout = 0
5	Sw3 = 0	Yout = 0
6	Sw3 = 0	Yout = 0
7	Sw3 = 0	Yout = 0
8	Sw3 = 0	Yout = 0
9	Sw3 = 0	Yout = 0
с сигмоидной функцией активации		
0	Sw3 = 0,000270242677595017	Yout = 0,500067560668988
1	Sw3 = 3,43951083407716E-6	Yout = 0,500000859877708
2	Sw3 = 0,0526576299057448	Yout = 0,513161366435398
3	Sw3 = 0,0109778057331013	Yout = 0,502744423871947
4	Sw3 = 9,28464982946175E-6	Yout = 0,500002321162457
5	Sw3 = 0,000934588149375769	Yout = 0,500233647020337
6	Sw3 = 0,000250751415905857	Yout = 0,500062687853648
7	Sw3 = 1,20144980966343E-7	Yout = 0,500000030036245
8	Sw3 = 0,014626235919069	Yout = 0,503656493794839
9	Sw3 = 0,00149247436075622	Yout = 0,50037311852093

Таблица 2. Результаты распознавания контрольной базы MNIST (10000 символов) без обучения

s0 = 834	i0 = 980	p0 = 85 %
s1 = 968	i1 = 1135	p1 = 85 %
s2 = 530	i2 = 1032	p2 = 51 %
s3 = 454	i3 = 1010	p3 = 44 %
s4 = 410	i4 = 982	p4 = 41 %
s5 = 411	i5 = 892	p5 = 46 %
s6 = 586	i6 = 958	p6 = 61 %
s7 = 556	i7 = 1028	p7 = 54 %
s8 = 773	i8 = 974	p8 = 79 %
s9 = 750	i9 = 1009	p9 = 74 %
Итого: i = 10000, s = 6272, p = 62 %		

больше 0,5. Это можно наблюдать в табл. 1, где приведены выходы третьего слоя при распознавании изображения 2_174 (рис. 8). В данных экспериментах в качестве правила сравнения на выходе нейронной сети было использовано определение активности выхода нейрона третьего слоя по наибольшему значению выхода Y_i нейронной сети, например в табл. 1 наибольший выход соответствует выходу 2.

Все вычисления, описанные в данной статье, проводились на одном компьютере в программном модуле, приведенном на рис. 3, реализованном в среде Builder C++. На весь процесс создания нейронной сети и вычисления всех весов в программном модуле на рис. 3 было зафиксировано общее потраченное время $t_{\text{созд}} = 0,5469$ с, т.е. меньше секунды.

В табл. 2 приведены результаты распознавания контрольной базы MNIST (10000 изображений) на основе полученной нейронной сети с использованием как пороговой, так и сигмоидной функции активации. Также приведены количество и процент правильно идентифицированных объектов контрольной базы MNIST отдельно для каждого образа цифры (sj, pj, где j — наименование образа), а также приводится общее количество изображений ij для каждого j образа в контрольной базе MNIST.

По данным табл. 2 можно видеть, что общее количество правильно идентифицированных изображений MNIST составило 62 % для нейронной сети с пороговой функцией активации. Результат вычисленных значений весов сохраняется и с сигмоидной функцией активации, что подтверждается также распознаванием контрольной базы MNIST нейронной сети с сигмоидной функцией активацией и с проверкой на выходе нейронной сети по наибольшему значению выхода Y_i . В этом случае результат нейронной сети с сигмоидной функцией активации идентичен результату нейронной сети с пороговой функцией активации и равен также 62 %.

Нужно также сказать, что в данных экспериментах для сигмоидной функции активации значение веса-порога $Wh2 = -29$ увеличивалась до значения $Wh2 = -27$, что было сделано для того, чтобы повысить пропускную способность нейронов второго слоя, поскольку известно, что в отличие от пороговой

функции активации сигмоидная функция стремится к единице, но не достигает ее.

Возможно, в этом изменении и не было никакой необходимости, поскольку процент распознавания по контрольной базе MNIST с сигмоидной функцией активации и с проверкой на выходе нейронной сети по наибольшему значению выхода сети одинаков по результату и для случая $Wh2 = -27$ и для случая $Wh2 = -29$ и равен 62 %, т.е. равен результату тестирования нейронной сети с пороговой функцией активации.

Обучение нейронной сети выполнялось стохастическим алгоритмом обратного распространения ошибки (*back propagation*) с использованием обучающей выборки MNIST (60000 изображений). Поправки вносились после каждого представления нового объекта на входы сети в случае наличия ошибки распознавания на выходе нейронной сети. В случае отсутствия ошибки распознавания правки не вносились. При обратном распространении в качестве правильного активного выхода нейронной сети рассматривалось значение $y^{corr} = 0,7$, а для правильного неактивного выхода — значение $y^{corr} = 0,2$, т.е. в процессе обучения значения выходов нейронной сети подтягивались к этим значениям. Для каждого эксперимента обучения нейронной сети использовались три эпохи, первые две из которых обучались со скоростью $pk = 0,1$, а последняя эпоха — со скоростью $pk = 0,02$. Ошибка обучения S_{err} вычислялась для каждой эпохи по формуле:

$$(3.4) \quad S_{err} = \frac{1}{2} \sum_{i=0}^P \sum_{k=0}^{N_{обп}-1} \left(y_k^{(corr)} - f \left(Sn_k^{(3)} \right) \right)^2,$$

где $y_k^{(corr)}$ — правильное значение k -го выхода третьего слоя для активного выхода $y_k^{(corr)} = 0,7$, а для неактивного $y_k^{(corr)} = 0,2$; P — количество неправильно идентифицированных изображений обучающей базы MNIST, для которых в процессе эпохи делались правки весов алгоритмом обратного распространения ошибки. Во втором эксперименте также выполнялось обучение полученной выше нейронной сети, но на этот раз значения всех весов генерировались случайным образом в диапазоне $[-0,5; 0,5]$.

Стохастический алгоритм обратного распространения ошибки, реализованный в программном модуле рис. 3, выполняется следующим образом:

1. В последовательном порядке выбирается из обучающей базы MNIST изображение обучающей выборки, для которой создается бинарная матрица изображения, которая подается на вход нейронной сети с предварительно вычисленными или сгенерированными случайными весовыми значениями;
2. Выполняется прямое распространение по нейронной сети с последовательным вычислением значений функций состояний и активаций нейронов по выражениям (2.5)–(2.10), начиная с первого слоя и кончая третьим;
3. Проверяются выходы нейронной сети по правилу наибольшего значения выхода сети. Если активный выход нейронной сети соответствует установленному выходу образа, которому принадлежит изображение, то выполняется возвращение к п. 1, если не соответствует, то реализуется обратное распро-

странение ошибки по сети от третьего слоя к первому, для чего выполняется переход к п. 4;

4. Для каждого i -го нейрона третьего слоя вычисляется новое весовое пороговое значение (смещение):

$$(3.5) \quad \text{Wh3}[i][0] = \text{Wh3}[i][0] + d\text{Wh3}[i][0],$$

где $d\text{Wh3}[i][0]$ — приращение весового порогового значения i -го нейрона третьего слоя

$$(3.6) \quad d\text{Wh3}[i][0] = nk \cdot \text{Sig3}[i],$$

здесь nk — коэффициент скорости обучения (на рис. 3 поле nk), а $\text{Sig3}[i]$ — ошибка для i -го выхода третьего слоя нейронной сети, определяемая по выражению:

$$(3.7) \quad \text{Sig3}[i] = (Y_{\text{corr}} - Y_{\text{out}}[i]) \cdot Y_{\text{out}}[i] \cdot (1 - Y_{\text{out}}[i]),$$

где $Y_{\text{out}}[i]$ — текущее i -е выходное значение третьего слоя, Y_{corr} — ожидаемое i -е выходное значение третьего слоя. Для ожидаемого правильного выхода $Y_{\text{corr}} = 0,7$, для ожидаемого неправильного выхода нейронной сети $Y_{\text{corr}} = 0,2$. На этом же этапе вычисляется также поправка суммарной квадратичной ошибки (3.4) S_{err} для данного изображения и добавляется к предыдущему значению S_{err} :

$$(3.8) \quad S_{\text{err}} = S_{\text{err}} + \sum_{i=0}^{N_{\text{обп}}-1} (Y_{\text{corr}} - Y_{\text{out}}[i])^2;$$

5. Вычисляются новые весовые значения для третьего слоя по выражению:

$$(3.9) \quad \text{W3}[i][k] = \text{W3}[i][k] + d\text{W3}[i][k],$$

здесь $d\text{W3}[i][k]$ — приращение значения веса $\text{W3}[i][k]$, связывающего k -й нейрон второго слоя и i -й нейрон третьего слоя, и определяется по выражению:

$$(3.10) \quad d\text{W3}[i][k] = nk \cdot \text{Sig3}[i] \cdot \text{F2}[k],$$

где $\text{F2}[k]$ — выход k -го нейрона второго слоя. Вычисляются весовые пороговые значения нейронов второго слоя:

$$(3.11) \quad \text{Wh2}[k][0] = \text{Wh2}[k][0] + d\text{Wh2}[k][0],$$

где $d\text{Wh2}[k][0]$ — приращение весового порогового значения $\text{Wh2}[k][0]$ для k -го нейрона второго слоя, определяемое выражением:

$$(3.12) \quad d\text{Wh2}[k][0] = nk \cdot \text{Sig2}[k],$$

здесь ошибка

$$(3.13) \quad \text{Sig2}[k] = \sum_{i=0}^{N_{\text{обп}}-1} (\text{Sig3}[i] \cdot \text{W3}[i][k]);$$

6. Вычисляются новые весовые значения для второго слоя по выражению:

$$(3.14) \quad W2[i][k][k1] = W2[i][k][k1] + dW2[i][k][k1],$$

где $dW2[i][k][k1]$ — приращение значения веса $W2[i][k][k1]$ для связи, связывающего i -й нейрон второго слоя и нейрон первого слоя, который выполняет идентификацию пары эталонов с порядковыми номерами k и $k1$. $dW2[i][k][k1]$ вычисляется выражением:

$$(3.15) \quad dW2[i][k][k1] = nk \cdot \text{Sig2}[i] \cdot F1[k][k1],$$

где $F1[k][k1]$ — выход нейрона первого слоя, выполняющий сравнение k и $k1$ эталонов.

Вычисляются весовые пороговые значения для нейронов первого слоя:

$$(3.16) \quad \text{Wh1}[k][k1][0] = \text{Wh1}[k][k1][0] + d\text{Wh1}[k][k1][0],$$

где $d\text{Wh1}[k][k1][0]$ — приращение весового порогового значения веса $\text{Wh1}[k][k1][0]$ для нейрона первого слоя, выполняющее распознавание k и $k1$ эталонов

$$(3.17) \quad d\text{Wh1}[k][k1][0] = nk \cdot \text{Sig1}[k][k1],$$

где ошибка

$$(3.18) \quad \text{Sig1}[k][k1] = \sum_{i=0}^{N-1} (\text{Sig2}[i] \cdot W2[i][k][k1]);$$

7. Вычисляются новые весовые значения отдельно для двух частей таблиц весов первого слоя по выражениям:

$$(3.19) \quad W1[k][k1][0][r][c] = W1[k][k1][0][r][c] + dW1[k][k1][0][r][c],$$

$$(3.20) \quad W1[k][k1][1][r][c] = W1[k][k1][1][r][c] + dW1[k][k1][1][r][c],$$

где $dW1[k][k1][0][r][c]$ — приращение весового значения $W1[k][k1][0][r][c]$ для связи, связывающего ячейку первой части таблицы весов с координатами (r, c) с нейроном первого слоя, который выполняет попарную идентификацию эталонов с порядковыми номерами k и $k1$, а $dW1[k][k1][1][r][c]$ — приращение весового значения $W1[k][k1][1][r][c]$ для связи, связывающего ячейку второй части таблицы весов с координатами (r, c) с нейроном первого слоя, который выполняет попарную идентификацию эталонов с порядковыми номерами k и $k1$. Значения $dW1[k][k1][0][r][c]$ и $dW1[k][k1][1][r][c]$ определяются выражениями:

$$(3.21) \quad dW1[k][k1][0][r][c] = nk \cdot \text{Sig1}[k][k1] \cdot \text{BinX}[0][r][c],$$

$$(3.22) \quad dW1[k][k1][1][r][c] = nk \cdot \text{Sig1}[k][k1] \cdot \text{BinX}[1][r][c],$$

где $\text{BinX}[0][r][c]$, $\text{BinX}[1][r][c]$ — значения ячеек с координатами (r, c) для первой и второй частей бинарной матрицы входного изображения обучающей базы MNIST;

Таблица 3. Сравнение результатов обучения нейронной сети по обучающей выборке MNIST (60000 изображений) для каждой эпохи обучения

		Обучение нейронной сети с предварительно вычисленными весами			
№ эпохи	Скорость обучения	Количество узнанных изображений	Процент узнанных изображений	S_{err}	время в минутах
1	0,1	43932	73 %	1199	159
2	0,1	49748	83 %	737	98
3	0,02	52285	87 %	545	72
Общее время обучения в минутах					329
		Обучение нейронной сети с случайной инициализацией весов в диапазоне $[-0,5; 0,5]$			
№ эпохи	Скорость обучения	Количество узнанных изображений	Процент узнанных изображений	S_{err}	время в минутах
1	0,1	35370	59 %	1935	256
2	0,1	46033	76 %	1051	139
3	0,02	49195	82 %	784	104
Общее время обучения в минутах					499

Таблица 4. Сравнение результатов обучения нейронной сети с проверкой на контрольной выборке MNIST (10000 изображений) для каждой эпохи обучения

№ эпохи	Скорость обучения в эпохе	Обучение нейронной сети с предварительно вычисленными весами	Обучение нейронной сети с случайной инициализацией весов
1	0,1	9145	8894
2	0,1	9282	9116
3	0,02	9449	9256

8. Если были последнее, 60000-е, изображение обучающей базы MNIST и последняя заданная эпоха (рис. 3), то алгоритм на этом завершается, в противном случае выполняется переход к п. 1.

Отметим, что при программной реализации нейронных сетей на основе метрических методов распознавания в какой-либо программной среде, целесообразно и удобно нумерацию нейронов первого слоя вести не по количеству нейронов, как это делается при реализации классических схем, а по номерам двух эталонов, которые сравниваются в данном нейроне, как это было показано выше, например $W1[k][k1][0][r][c]$. Такой подход позволяет удобно и понятно реализовать структуру схемы на рис. 1, а также функции: предварительного вычисления значений весов, распознавания входного изображения, обучения нейронной сети, вывода полученных результатов и т.д.

В табл. 3 приведены результаты обучений полученной нейронной сети как с предварительно вычисленными значениями весов нейронной сети, так и классическим образом — со случайной инициализацией весов.

В табл. 4 приведены результаты проверки полученной нейронной сети после каждой эпохи обучения на контрольной базе MNIST (10000 изображений).

4. Сравнение полученных результатов двух экспериментов

По результатам табл. 3 и 4 построим диаграммы как для контрольной базы MNIST (рис. 9), так и для обучающей базы MNIST (рис. 10). На диаграммах можно видеть, что результативность нейронной сети с предварительно вычисленными весами выше после каждой эпохи обучения по сравнению с результативностью обученной сети, для которой начальные значения весов генерировались случайным образом. По-видимому, это связано с тем, что у

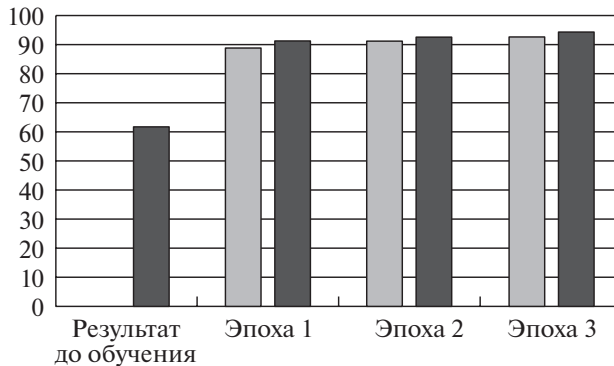


Рис. 9. Диаграмма процентов количества правильно узнанных изображений контрольной базы MNIST (10000 изображений): светлые столбцы — результаты обучения нейронной сети с нуля без предварительных вычислений значений весов; темные столбцы — результаты обучения нейронной сети с предварительно вычисленными значениями весов.

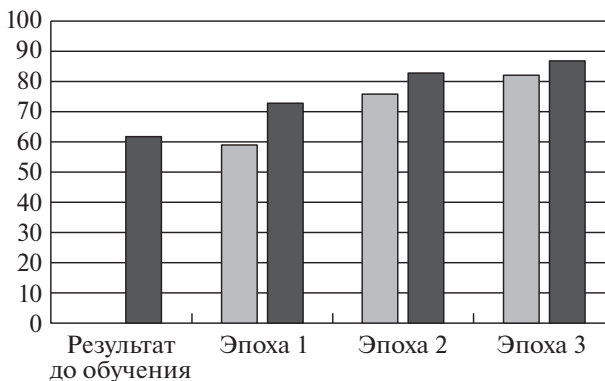


Рис. 10. Диаграмма процентов количества правильно узнанных изображений, выполненных по обучающей выборке MNIST (60000 изображений) для двух экспериментов: светлые столбцы — результаты обучения нейронной сети с нуля без предварительных вычислений весов; темные столбцы — результаты обучения нейронной сети с предварительно вычисленными весами.

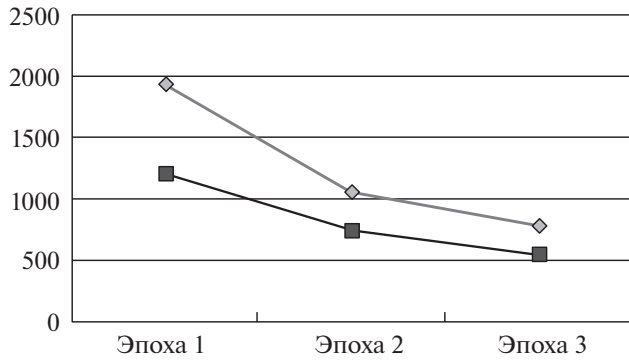


Рис. 11. Графики изменений ошибки S_{err} после каждой эпохи для двух экспериментов: светлая кривая — ошибка S_{err} при обучении с нуля без предварительных вычислений весов; темная кривая — ошибка S_{err} до обучения с предварительно вычисленными весами.

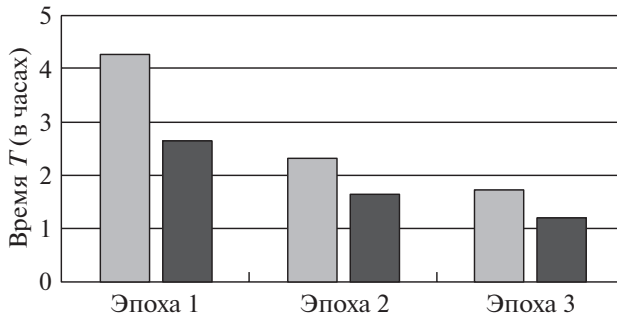


Рис. 12. Диаграмма сравнения затраченного времени (в часах) на каждую эпоху для двух экспериментов: светлые столбцы — обучение с нуля без предварительных вычислений весов; темные столбцы — до обучение с предварительно вычисленными весами.

нейронной сети с вычисленными весовыми значениями, согласно графикам на рис. 9 и 10, в запасе больше времени и в то же время короче путь, который требуется для того, чтобы добраться до более лучших результатов как по количеству правильно распознанных изображений базы MNIST, так и для достижения более низких значений ошибок S_{err} (рис. 11).

Время, потраченное на обучение нейронной сети с вычисленными весовыми значениями, также меньше на всех эпохах по сравнению с нейронной сетью, обучаемой с нуля (рис. 12). При этом на диаграмме (рис. 12) можно также наблюдать, что основное преимущество по времени накапливается в первой эпохе. В целом, на обучение нейронной сети с вычисленными весовыми значениями было затрачено $329 \text{ мин} = 5 \text{ ч. } 29 \text{ мин}$, а при обучении нейронной сети с нуля общее время обучения составило $499 \text{ мин} = 8 \text{ ч. } 19 \text{ мин}$, т.е. для второго эксперимента понадобилось почти на 3 часа больше.

Отметим также, что приведенные результаты были получены при наборе эталонов, приведенных на рис. 4, который по-видимому не является самым

a

Wh2 = -26,9798
1,02273 1,03707 1,01161 1,01429 1,01668 1,09436 1,04434 1,00577 1,05321 1,03789 1,01096 1,04362 1,02
Wh2 = -27,0031
0,00653327 -0,00412966 -0,00308971 -0,00309406 -0,00309341 -0,0225603 0,00632404 -0,0030341 0,0019
Wh2 = -26,9888
0,0346238 0,0520874 0,00711415 0,0081946 0,00931319 0,110453 0,0461116 0,00551403 0,0534076 0,08
Wh2 = -26,976
0,00955242 0,00718852 0,00442882 0,00583441 -0,0221693 -0,00520425 -0,000323037 -0,0199936 -0,008
Wh2 = -26,9727
0,0125543 -0,00108514 0,0293263 -0,0103038 -0,00784111 -0,0202156 -0,0139231 -0,0355479 0,0290557
Wh2 = -26,9823
0,00514448 0,0116094 -0,00815483 -0,00703385 0,00181616 -0,00108448 -0,0140935 -0,011053 0,004342
Wh2 = -26,9982
0,0114983 0,0531569 -0,000970677 0,026209 -0,0192114 0,0790814 0,0235482 0,0105859 -0,0275643 0,05
Wh2 = -27,0017
0,0658932 0,0589804 -0,00999116 0,0468242 -0,0438653 0,0331221 0,0517183 0,0197762 0,0212233 0,09

б

Wh3 = -1,21389
1,09088 0,917265 0,764771 -0,0506185 -0,043322 -0,0267967 -0,222947 -0,223477 0,0288491 -0,0778204 -0,12255
Wh3 = -1,21495
-0,0784049 -0,0627303 -0,107165 1,31762 1,52215 1,31522 0,00396852 0,0381043 0,108136 -0,0740865 -0,078349
Wh3 = -1,21087
0,0808409 0,0209968 -0,111898 -0,0983802 -0,234842 -0,034789 1,27359 1,2942 0,825619 -0,0772912 -0,0878193
Wh3 = -1,01762
-0,296377 -0,148033 -0,139417 -0,181047 -0,291219 -0,103179 -0,0403369 -0,20726 -0,127507 1,05603 0,979885 1
Wh3 = -1,13877
-0,0386596 -0,0405599 -0,0744336 -0,0665303 -0,0492572 -0,0661714 -0,31225 -0,146512 -0,0254047 -0,0572122 -
Wh3 = -1,13345
-0,124775 -0,205603 -0,140333 -0,134727 -0,13084 -0,207992 -0,0317918 -0,121255 -0,0549948 -0,436682 -0,0869
Wh3 = -1,24501
-0,0342267 0,200686 -0,0743375 -0,0776128 0,0077351 -0,0174852 -0,187996 -0,101407 -0,0414697 -0,0512284 -0

Рис. 13. Фрагменты значений весов после третьей эпохи обучения для нейронной сети с предварительно вычисленными значениями весов: *a* — для второго слоя; *б* — для третьего слоя.

лучшим ни по качеству, ни по количеству. Другими словами, при более качественном и при большем количестве эталонов результаты могли бы быть еще более заметными. Итоговая результативность обучения может также быть выше при использовании алгоритмов обучения [1], которые дают лучшие результаты по сравнению с использованным в экспериментах стохастическим алгоритмом обратного распространения ошибки.

Результаты на рис. 9–12, в целом, показывают также, что алгоритм обучения обратного распространения не уничтожает результативность нейронной сети с предварительно вычисленными весовыми значениями, а ищет решение еще лучше, отталкиваясь от уже существующего начального результата.

Это же подтверждают и весовые значения второго и третьего слоев, приведенные на рис. 13. На рис. 13 можно видеть, что изначальные значения

Таблица 5. Фрагмент (9 из 30) выходов второго слоя при распознавании символа 2_1 (рис. 8) из контрольной выборки MNIST. Сеть дообучена в трех эпохах

№ Y2	Эталоны	Sw2	F2
0	0_157	Sw2 = -2,25757884565147	F2 = 0,0946977310982947
1	0_25	Sw2 = -14,7120863080901	F2 = 4,07964075748757E - 7
2	0_28	Sw2 = -2,45988516820271	F2 = 0,0787186646034626
3	1_135	Sw2 = -16,1987387416131	F2 = 9,22522804629833E - 8
4	1_2	Sw2 = -21,1151249699638	F2 = 6,75799304608105E - 10
5	1_46	Sw2 = -14,8256166236525	F2 = 3,64180217856173E - 7
6	2_172	Sw2 = 1,78349180153865	F2 = 0,85612749736249
7	2_35	Sw2 = -0,45514766242243	F2 = 0,388137562543613
8	2_77	Sw2 = -6,77250084528418	F2 = 0,00114351889764061
9	3_32	Sw2 = -0,268545411318517	F2 = 0,433264229225887

весов — единиц и нулей, а также установленных порогов второго и третьего слоев в процессе всего обучения изменились незначительно в положительную или отрицательную сторону, сохранив при этом основную логику сети. То же самое подтверждают и выходы второго слоя нейронной сети, например, приведенных в табл. 5 для изображения 2_1 из контрольной базы (рис. 8). В табл. 5 можно видеть, что активный выход второго слоя для распознаваемого изображения 2_1 соответствует тому же выходу, который был изначально установлен в схеме сети для эталона 2_172 на рис. 1, в. Другими словами, каких-либо значительных изменений в структуре и логике обученной нейронной сети не наблюдается.

5. Заключение

На основе проведенных экспериментов и полученных результатов можно сделать следующие заключения:

1. Время создания и вычисления всех весовых значений нейронной сети, включая веса нулевого слоя, для формата изображений базы MNIST составляет доли секунд (для рассмотренного примера $t_{\text{созд}} = 0,5469$ с). Можно сказать, что по сравнению с затратами времени на обучение сети создание нейронной сети и вычисление всех весовых значений выполняется практически мгновенно;

2. Нейроны нейронной сети на основе метрических методов распознавания могут иметь и непрерывную функцию активации, например сигмоидную функцию активации;

3. Нейронная сеть на основе метрических методов распознавания обучается алгоритмом обратного распространения ошибки (*back propagation*);

4. Процесс дообучения нейронной сети с вычисленными весами требует меньше времени по сравнению с классическим обучением нейронной сети с генерированием случайных значений весов. В рассмотренном примере выигрыш по времени составил 2 ч. 50 мин;

5. Приведенный пример показал также, что результативность нейронной сети с вычисленными весами на всех трех эпохах имеет более хороший результат распознавания по базе MNIST. Наибольший результат распознавания контрольной базы MNIST по результатам трех эпох обучения нейронной сети с вычисленными весами составил 94 %;

6. На основе приведенных результатов экспериментов можно предположить, что при более качественном подборе эталонного набора, а также при большем количестве выбранных эталонов в приведенных на рис. 9–12 диаграммах и графиках преимущественные отличия могут быть еще более значимыми;

7. Приведенная технология предварительного вычисления значений весов может быть предположительно использована и в других архитектурах нейронных сетей прямого распространения, например в глубоких сетях [6, 7], что может позволить ускорить процедуру создания и обучения этих сетей.

СПИСОК ЛИТЕРАТУРЫ

1. *Круглов В.В., Борисов В.В.* Искусственные нейронные сети. Теория и практика. М.: Горячая линия–Телеком, 2001.
2. *Уоссерман Ф.* Нейрокомпьютерная техника. Теория и практика. М.: Мир, 1992.
3. *Geidarov P.Sh.* Neural Networks on the Basis of the Sample Method // *Automat. Contr. Comput. Sci.* N.Y.: Alerton Press, 2009. V. 43. No. 4. P. 203–210.
4. *Гейдаров П.Ш.* Многозадачное применение нейронных сетей, реализующих метрические методы распознавания // *АиТ.* 2013. № 9. С. 53–67.
Geidarov P.Sh. Multitasking Application of Neural Networks Implementing Metric Methods of Recognition // *Autom. Remote Control.* 2013. V. 74. No. 9. P. 1474–1485.
5. *Бургер И.А.* Техническая диагностика. М.: Машиностроение, 1978.
6. *LeCun Y., Bengio Y., Hinton G.* Deep learning // *Nature.* 2015. No. 521. P. 436–444.
7. *Schmidhuber J.* Deep Learning in Neural Networks: An overview // *Neural Networks.* 2015. No. 61. P. 85–117.
8. *Srivastava N. et al.* A Simple Way to Prevent Neural Networks from Overfitting // *J. Mach. Learn. Res.* 2014. V. 15. No. 1 P. 1929–1958.
9. *Geidarov P.Sh.* Clearly Defined Architectures of Neural Networks and Multilayer Perceptron // *Opt. Mem. Neural Network.* 2017. V. 26. P. 62–76.
10. *Гейдаров П.Ш.* Алгоритм реализации метода ближайшего соседа в многослойном персептроне // *Тр. СПИИРАН.* Вып. 51. 2017. С. 123–151.

Статья представлена к публикации членом редколлегии О.П. Кузнецовым.

Поступила в редакцию 03.12.2018

После доработки 24.10.2019

Принята к публикации 28.11.2019