



Исследование устойчивости многослойного персептрона с вычисляемыми весами синапсов к меньшим объемам обучающей выборки

П. Ш. Гейдаров^а, доктор техн. наук, доцент, orcid.org/0000-0002-3881-0629, plbaku2010@gmail.com

^аИнститут систем управления НАН Азербайджана, Бахтияра Вагабадзе ул., 68, AZ 1141, Баку, Азербайджан

Введение: процесс создания искусственных нейронных сетей требует от разработчиков наличия большого объема обучающей базы данных, что усложняет процесс создания и обучения искусственной нейронной сети. При этом в биологических нейронных сетях живых организмов процесс обучения успешно выполняется с сравнительно малыми объемами обучающих баз данных. Поэтому задача создания новых архитектур искусственных нейронных сетей, позволяющих эффективно уменьшать необходимый объем обучающей базы данных, является весьма актуальной задачей. **Цель:** проверить устойчивость результативности нейронной сети, реализованной на основе метрических методов распознавания, к меньшим объемам обучающих баз данных. **Результаты:** для решения поставленной задачи выполнены сравнительные эксперименты обучения нейронной сети с разными объемами базы MNIST для нейронной сети как с предварительно вычисленными весовыми значениями, так и со случайной генерацией весов. Сравнительный анализ итогов экспериментов показал, что результативность нейронной сети с вычисленными весовыми значениями более устойчива к уменьшению объема обучающей выборки. Также предложена общая схема и основные положения алгоритма реализации представленной технологии для создания вычисляемых сверточных нейронных сетей. **Практическая значимость:** нейронные сети с вычисленными весовыми значениями требуют меньшего объема обучающей базы данных, что позволяет ускорить и упростить процедуру создания, обучения и дообучения нейронной сети.

Ключевые слова – многослойный персептрон, метрические методы распознавания, обучение нейронных сетей, сверточные нейронные сети.

Для цитирования: Гейдаров П. Ш. Исследование устойчивости многослойного персептрона с вычисляемыми весами синапсов к меньшим объемам обучающей выборки. *Информационно-управляющие системы*, 2023, № 2, с. 2–14. doi:10.31799/1684-8853-2023-2-2-14, EDN: WOHNNE

For citation: Geidarov P. Sh. Investigating stability of a multilayer perceptron with calculated synaptic weights to smaller training sample sizes. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2023, no. 2, pp. 2–14 (In Russian). doi:10.31799/1684-8853-2023-2-2-14, EDN: WOHNNE

Введение

В настоящее время искусственные нейронные сети (НС) широко применяются в различных сферах техники и науки [1–14]. Это объясняется тем, что из всех алгоритмов машинного обучения современные возможности искусственных НС позволяют получить наилучшие результаты в задачах распознавания образов. Тем не менее нужно отметить, что на сегодня возможности искусственных НС по многим критериям уступают возможностям биологических НС живых организмов. В частности, реализация искусственных НС требует наличия очень большой обучающей базы данных, исчисляемой десятками и сотнями тысяч (а иногда и более) элементов, что усложняет процедуру реализации НС для их разработчиков. При этом в биологических НС процесс обучения не требует таких огромных обучающих баз данных. Например, наблюдения за детьми показывают, что те обучаются сравнительно небольшими наборами данных [15]. Следовательно, разработка новых архитектур искусственных НС, которые могли бы позволить НС обучаться

с меньшими объемами обучающей выборки по аналогии с тем, как это делают биологические НС, остается весьма актуальным вопросом.

Для улучшения возможностей искусственных НС прямого распространения были предложены архитектуры НС многослойного персептрона на основе метрических методов распознавания (НСММП) [16, 17], которые реализуют алгоритмы метрических методов распознавания [18], таких как метод ближайшего соседа, метод ближайших n соседей, метод потенциалов и др., и при этом имеют архитектуру многослойного персептрона, что делает возможным эти сети дополнительно дообучать, как обычный многослойный персептрон. И как было также показано в работах [16, 17, 19], значения весов для этих НС можно вычислять аналитически по формулам с применением небольшого набора эталонов, что позволяет без использования алгоритмов обучения и обучающих баз данных сразу получить работоспособную НС. Процесс предварительного аналитического вычисления значений весов НС по формулам без обучения НС выполняется очень быстро — за доли секунд и минут [20, 21].

В работе [21] за полсекунды были вычислены весовые значения НС и таким образом сразу получена работоспособная НС с результативностью 64 % правильно идентифицированных символов контрольной базы MNIST. Такая способность очень похожа на способность биологического мозга — запоминать и узнавать объекты сразу с использованием малого набора данных.

Полученные преимущества реализуются благодаря внедрению алгоритмов метрических методов распознавания в классические НС, что позволяет объединить возможности метрических методов распознавания с НС в единой архитектуре НС многослойного перцептрона. В частности, в отличие от классических НС алгоритмы метрических методов распознавания позволяют распознавать образы с небольшим количеством эталонов. В метрических методах распознавания для распознавания входного объекта на основе метрических выражений близости вычисляются коэффициенты близости распознаваемого объекта к каждому выбранному эталону. По наименьшему или наибольшему значению (в зависимости от метрического выражения) выбирается ближайший к распознаваемому объекту эталон и соответственно выбирается и образ, которому принадлежит данный эталон. Именно эти возможности позволяют аналитически вычислять значения весов НС, реализованной на основе метрического метода распознавания, и ускорять весь процесс создания и обучения НС.

Отметим также, что результативность НС с вычисленными весами повторяет результативность используемого метрического метода, на основе которого НС и была реализована. Соответственно, согласно метрическим методам распознавания, чем больше количество используемых эталонов и чем результативнее используемый метрический метод, тем выше будет результативность предварительно аналитически вычисленной НС.

В настоящее время существуют разработанные методы и подходы начальной инициализации весов [22–24], отличные от классических методов. Но по этим методам не вычисляют точные значения весов, а они являются разновидностями более «умной» случайной инициализации весов. Методами [22–24] вычисляются не сами значения весов НС, а значения пределов диапазонов для случайной инициализации весов. Предназначение использования методов инициализации весов является также иным. Оно направлено на то, чтобы решить проблему обучения глубоких сетей с очень большим количеством слоев (более десяти), поскольку в процессе обучения таких глубоких НС выходы нейронов в слоях образуют слишком большие или, наоборот, слишком малые значения, что сильно замедляет обу-

чение глубокой НС и приводит ее в паралич или в локальный минимум. Методы начальной инициализации весов [22–24] позволяют предотвратить этот процесс и ускорить алгоритм обучения глубокой НС. Но нужно отметить разницу этих методов с предлагаемым в данной работе подходом аналитического вычисления весов. Методы начальной инициализации весов [22–24] не способны создавать работоспособную НС без использования алгоритма обучения, тогда как рассматриваемый метод позволяет сразу получить НС, способную распознавать образы без обучения. При этом сам процесс обучения полученной НС заменяется на процесс дообучения. В работах [20, 21] показано, что полученная НС может дополнительно дообучаться алгоритмом обратного распространения ошибки “backpropagation”. Также на основе проведенных экспериментов показано [21], что процесс дообучения НС выполняется значительно быстрее, чем полное обучение НС классическим образом — путем начальной случайной генерации значений весов.

В данной работе ставилась цель показать устойчивость результативности (Rv) НС прямого распространения с вычисленными весами к уменьшению объема обучающей выборки (V_{BD}) по сравнению с НС того же класса со случайной инициализацией весов. Здесь под понятием «результативность» имеется в виду количество (или процент) правильно идентифицированных изображений контрольной (тестируемой) базы MNIST.

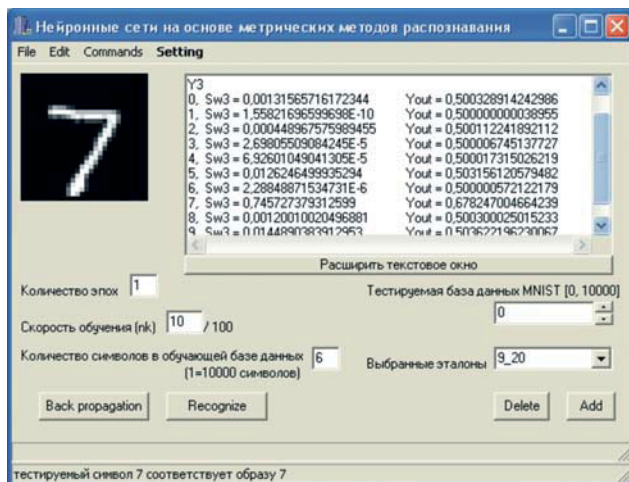
Для реализации этой цели проведены эксперименты обучения НС на базе MNIST с вычисленными весовыми значениями и случайно сгенерированными весовыми значениями для разных объемов обучающей выборки. По результатам этих экспериментов выполнен сравнительный анализ. Если функция зависимости результативности обученной НС с вычисленными весовыми значениями $Rvi(V_{BD})$ больше результативности НС со случайной инициализацией весов $Rsi(V_{BD})$ для всех проведенных экспериментов i и при всех прочих равных условиях ($Rvi(V_{BD}) > Rsi(V_{BD})$), то в этом случае можно говорить, что результативность НС с вычисленными весовыми значениями весов более устойчива (стабильна) к уменьшению объема обучающей выборки по сравнению с результативностью НС с классической случайной инициализацией весов.

Начальные параметры и условия эксперимента

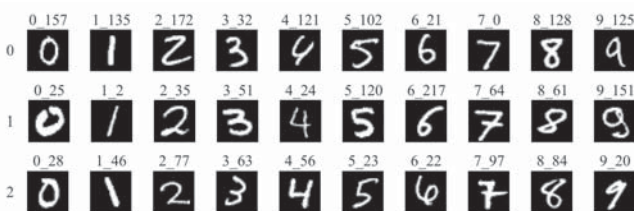
В работе [21] проведены два эксперимента, на основе которых был выполнен сравнительный анализ результатов обучения трехслойной

НС с вычисленными весами и со случайной генерацией весов. Была создана НС на основе метрического метода распознавания – метода ближайшего соседа по схеме, приведенной на рис. 3 в работе [19]. Для реализации НС был разработан программный модуль (рис. 1), позволяющий создавать и тестировать НС на основе метрических методов распознавания по выбранному набору эталонов. В качестве эталонов использовался набор из 30 образцов цифр (рис. 2) – по три от каждого образа, случайным образом выбранных из контрольной базы MNIST.

Для входных изображений использовалась бинарная входная матрица с размерностью 28 столбцов и 56 строк, где первая часть таблицы (28:28) определяет активные пиксели входного изображения, для которых значение пикселя >150, и неактивные пиксели, для которых значение пикселя <150. Во второй части бинарной таблицы наоборот: активными считаются пиксели со значениями <150, неактивными – со значениями >150. Размерность таблиц весов нейрона первого слоя определяется также размерностью бинарной входной таблицы (28:56). Пример вычисленной таблицы весов первого слоя приведен



■ **Рис. 1.** Программный модуль на базе Builder C++
 ■ **Fig. 1.** A program module based on Builder C++



■ **Рис. 2.** 30 выбранных эталонов из контрольной базы MNIST
 ■ **Fig. 2.** 30 selected samples from the MNIST control database

для эталонов 7_0 и 3_{32} (рис. 3). Wh – смещение (bias) нейрона слоя. В качестве выражения меры близости используется формула, по которой вычисляются значения всех весов первого слоя:

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

где d_1, d_2 – кратчайшие расстояния от ячейки таблицы весов с координатами (c_p, r_p) до ближайших ячеек таблицы весов с координатами (c_1, r_1) и (c_2, r_2) , в которых изображения эталонов 1 и 2 активны.

Алгоритм определения ближайшей активной ячейки эталонного изображения приведен в работе [19]. Для всех нейронов НС использовалась сигмоидная функция активации

$$f(Sw) = \frac{1}{1 + e^{-Sw}}, \quad (2)$$

где Sw – функция состояния нейрона. Для нейрона первого слоя, выполняющего сравнение i -го и j -го эталонов, функция состояния нейрона $Sw_{i,j}^{(1)}$ определяется по выражению

$$Sw_{i,j}^{(1)} = \sum_{r=0}^R \sum_{c=0}^C (x_{c,r} w_{i,j,c,r}^{(1)}), \quad (3)$$

где C, R – количество столбцов и строк в таблице весов; c, r – координаты пикселя или ячейки изображения i , соответственно, координаты таблицы весов; i и j – порядковые номера двух сравниваемых эталонов.

Для нейронов второго и третьего слоя полносвязной НС [19, рис. 3] все значения весов принимаются равными $w_{ij}^{(2)} = w_{ij}^{(3)} = 1$. Поскольку в данной задаче используется полносвязный персептрон, то весовые значения всех добавленных связей в НС принимаются равными 0. В работе [17] показано, что такой подход не меняет логику работы НС. На рис. 4 представлен фрагмент значений весов для второго слоя, а на рис. 5 приведены все значения весов третьего слоя.

Функции состояния нейрона второго слоя $Sw_k^{(2)}$ и третьего слоя $Sw_k^{(3)}$ определяются по выражениям

$$Sw_k^{(2)} = \sum_{j=1, j \neq k}^N f(Sw_{k,j}^{(1)}); \quad (4)$$

$$Sw_k^{(3)} = \sum_{i \in k}^{K_k} f(Sw_i^{(2)}). \quad (5)$$

Для всех нейронов первого слоя значения весов смещения $w_0^{(1)} = Wh1 = 0$. Для всех нейронов

инициализированными весовыми значениями к уменьшению объема обучающей выборки MNIST были проведены сравнительные эксперименты с той же НС и с теми же начальными параметрами алгоритма обучения, что описаны в предыдущем разделе.

Результаты двух процедур обучений НС как с вычисленными весовыми значениями, так и со случайно сгенерированными начальными значениями весов приведены в табл. 2. На всех трех эпохах обучения НС количество правильно идентифицированных изображений больше для НС с вычисленными весовыми значениями, а общее время обучения НС с вычисленными

весовыми значениями на 38 % меньше ($p_{\text{ускор}} = (471 - 293) \times 100/471 \approx 38 \%$). В табл. 3, где приводятся результаты тестирования обученной НС на контрольной базе MNIST, видно, что результативность после каждой эпохи обучения выше для НС с вычисленными весовыми значениями.

Были также проведены обучение НС на базе 20 000 изображений и тестирование по 10 000 изображений. По результатам (табл. 4, 5) видно, что на всех трех эпохах обучения количество узнанных изображений больше для НС с вычисленными весовыми значениями, общее время обучения НС с вычисленными весовыми значениями выполняется на 34 % быстрее ($p_{\text{ускор}} =$

■ **Таблица 2.** Результаты обучения НС на 40 000 изображений базы MNIST с вычисленными весами и со случайной генерацией весов

■ **Table 2.** Results of training a neural network using 40,000 images of the MNIST database with calculated weights and with random generation of weights

№ эпохи	Скорость обучения	Обучение с предварительно вычисленными весами				Обучение со случайной инициализацией весов в диапазоне [-0,5; 0,5]				
		Количество узнанных изображений	Процент узнанных изображений	S_{err}	Время, мин	Количество узнанных изображений	Процент узнанных изображений	S_{err}	Время, мин	
1	0,1	32 815	82,00	765,7	137,6	27 346	68,30	1314,5	228,6	
2	0,1	35 705	89,20	411	88,47	32 578	81,44	729,25	140,4	
3	0,02	36 899	92,20	295,4	67,5	34 851	87,10	498,7	102,8	
Общее время обучения					293,57	Общее время обучения				471,8

■ **Таблица 3.** Сравнение результатов обучения НС на 40 000 изображений с проверкой на контрольной выборке MNIST (10 000 изображений) для каждой эпохи обучения

■ **Table 3.** Comparison of training results of a neural network on 40,000 images with a tested MNIST control set (10,000 images) for each training epoch

№ эпохи	Скорость обучения	Количество узнанных изображений (в скобках – процент)	
		с предварительно вычисленными весами	с начальной случайной инициализацией весов
1	0,1	8731 (87,31)	6661 (66,61)
2	0,1	8910 (89,10)	6869 (68,69)
3	0,02	9194 (91,94)	8533 (85,33)

■ **Таблица 4.** Результаты обучения НС на 20 000 изображений базы MNIST с вычисленными весами и случайной генерацией весов

■ **Table 4.** Neural network training results using 20,000 MNIST images with computed weights and random weight generation

№ эпохи	Скорость обучения	Обучение с предварительно вычисленными весами				Обучение со случайной инициализацией весов в диапазоне [-0,5; 0,5]				
		Количество узнанных изображений	Процент узнанных изображений	S_{err}	Время, мин	Количество узнанных изображений	Процент узнанных изображений	S_{err}	Время, мин	
1	0,1	15564	77,82	498,87	90,27	12 497	62,48	798,8	142,23	
2	0,1	17412	87,06	254,61	58,8	15 552	77,76	437,9	90,39	
3	0,02	18098	90,49	183,77	47,13	17 015	85,07	290,07	65,51	
Общее время обучения					196,2	Общее время обучения				298,13

■ **Таблица 5.** Сравнение результатов обучения НС на 20 000 изображений с проверкой на контрольной выборке MNIST (10 000 изображений) для каждой эпохи обучения

■ **Table 5.** Comparison of training results of a neural network on 20,000 images with a tested MNIST control set (10,000 images) for each training epoch

№ эпохи	Скорость обучения	Количество узанных изображений (в скобках – процент)	
		с предварительно вычисленными весами	с начальной случайной инициализацией весов
1	0,1	8644 (86,44)	7051 (70,51)
2	0,1	8774 (87,74)	7716 (77,16)
3	0,02	9078 (90,78)	8025 (80,25)

= $(298 - 196) \times 100/298 \approx 34\%$ и количество правильно узанных символов по контрольной базе MNIST с вычисленными весовыми значениями после каждой эпохи обучения также выше.

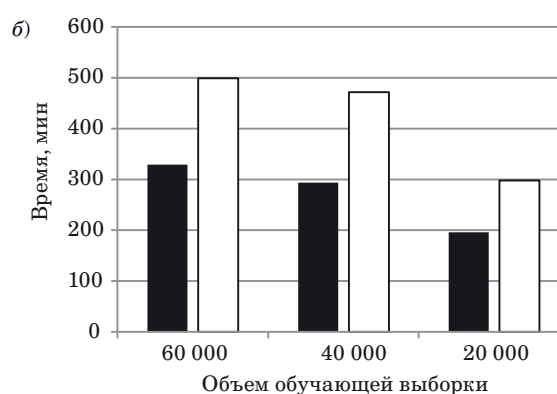
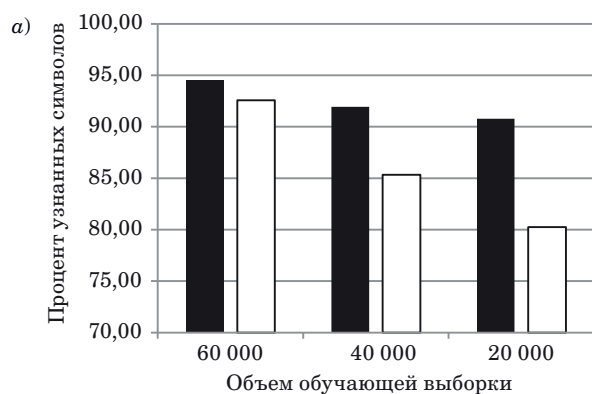
Диаграмма (рис. 6, а) показывает зависимость количества правильно идентифицированных символов контрольной базы MNIST от используемого объема обучающей выборки для обученной НС как с предварительно вычисленными весами, так и со случайной инициализацией весов. На диаграмме также видно, что НС с вычисленными весовыми значениями ведет себя более стабильно (устойчиво) к уменьшению объема обучающей выборки, количество правильно распознанных символов по контрольной базе MNIST сохраняет значения более чем 90 %, тогда как для НС со случайной инициализацией весов количество правильно распознанных символов падает до 80 %. На второй диаграмме (рис. 6, б) процедура обучения НС при разных объемах обучающей базы данных MNIST выполняется

быстрее для НС с предварительно вычисленными весовыми значениями.

Приведенные выше результаты показывают, что после трех эпох обучения количество правильно идентифицированных символов многослойного персептрона с вычисленными весовыми значениями и объемом обучающей выборки 20 000 изображений приблизительно равно количеству правильно узанных изображений НС со случайной начальной генерацией весов и с объемом обучающей выборки 60 000 изображений ($9078 \approx 9256$). При этом по табл. 1 из работы [21] и табл. 4 можно также наблюдать, что время, потраченное на обучение НС с предварительно вычисленными значениями весов и с использованием 20 000 изображений обучающего набора, составляет 196 мин, тогда как время, потраченное на обучение НС с начальной случайной генерацией весов и с использованием 60 000 изображений обучающего набора базы MNIST, составляет 499 мин. Таким образом, процедура обучения многослойного персептрона с вычисленными весовыми значениями еще более ускоряется до значения $p_{\text{ускор}} = (499 - 196) \times 100/499 \approx 60,72\%$.

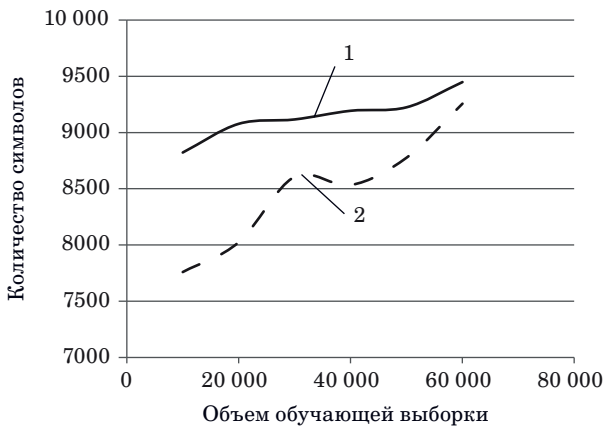
Аналогичным образом были также проведены сравнительные эксперименты на наборах обучающих выборок с 10 000, 30 000 и 50 000 символов базы MNIST с теми же начальными условиями (табл. 6).

Две кривые (рис. 7) отражают зависимость значений, приведенных в табл. 6. Из табл. 6 и рис. 7 делаем вывод, что для всех объемов обучающей базы данных значения правильно идентифицированных символов больше для НС с предварительно вычисленными значениями, и чем меньше количество обучающей выборки, тем больше эта разница проявляет себя. Для обученной НС с предварительно вычисленными весовыми значениями кривая коли-



■ **Рис. 6.** Зависимость количества узанных символов по контрольной базе MNIST (а) и времени обучения НС (б) с вычисленными весовыми значениями (черный цвет) и случайной генерацией весов (белый цвет) от объема обучающей выборки

■ **Fig. 6.** Percentage of recognized characters in the MNIST control base (а) and training time (б) with calculated weights (black color) and random generation of weights (white color) in relation to the volume of the training set



■ **Рис. 7.** Зависимость количества правильно идентифицированных символов контрольной базы MNIST от объема обучающего набора базы MNIST для НС с предварительно вычисленными весами 1 и со случайной генерацией весов 2

■ **Fig. 7.** Functions of the number of correctly identified symbols of the MNIST base to the volume of the training set of the MNIST base for a neural network with computed weights 1 and with random generation of weights 2

■ **Таблица 6.** Итоговые значения количества правильно идентифицированных изображений контрольной базы MNIST (10 000 изображений) для НС, обученных на трех эпохах с разными наборами обучающих баз данных, со случайной инициализацией весов и с вычисленными весами

■ **Table 6.** The final values of the number of correctly identified images of the MNIST control database (10,000 images) for the neural network with random initialization of weights and with calculated weights trained on three epochs with different sets of training databases

Нейронная сеть	Количество изображений при объеме символов обучающей базы данных					
	10 000	20 000	30 000	40 000	50 000	60 000
Со случайными значениями весов	7760	8025	8607	8533	8776	9256
С вычисленными значениями весов	8823	9078	9117	9194	9225	9449

чества правильно идентифицированных символов более стабильна, тогда как кривая обученной НС с начальной случайной генерацией весов ведет себя не стабильно, и, кроме того, значения количества правильно идентифицированных символов для этой кривой сильно уменьшаются с уменьшением объема обучающей выборки.

График первой функции (см. рис. 7, кривая 1) определяется полиномом третьей степени

$$Y_1 = 2 \times 10^{-11}x^3 - 2 \times 10^{-6}x^2 + 0,0654x + 8333,3, \quad (7)$$

а график второй функции (см. рис. 7, кривая 2) определяется полиномом пятой степени

$$Y_2 = -2 \times 10^{-19}x^5 + 5 \times 10^{-14}x^4 - 3 \times 10^{-09}x^3 + 10^{-04}x^2 - 1,2896x + 13730, \quad (8)$$

что также говорит об устойчивости НС с вычисленными весами к уменьшению объемов обучающей базы данных.

Схемы применения предлагаемых архитектур нейронных сетей для сверточных нейронных сетей

Алгоритм аналитического вычисления значений весов можно применить не только для трех-четырёхслойных НС, но и для глубоких НС [25, 26]. Если на схеме [19, рис. 3] изображения эталонов сразу использовались для вычисления значений весов полносвязной НС, то для сверточных НС предварительно для каждого изображения эталона будут создаваться слои с каналами и ядрами, каждый — на основе выделенного признака эталона.

Фрагмент эталонного изображения 0_157 (см. рис. 2) показан на рис. 8. Для простоты рассматривается однотонное черно-белое изображение, для которого пиксель принимается равным 255, если значение пикселя >127, и равным 0, если значение пикселя <127. Выделенный участок одного признака изображения — это матрица зна-

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	255	0	0	0	0	0	0	0	0
0	0	0	0	0	0	255	255	0	0	0	0	0	0	0
0	0	0	0	255	255	255	255	255	255	0	0	0	0	0
0	0	0	255	255	255	255	255	255	255	255	0	0	0	0
0	0	255	255	255	255	0	0	0	0	0	255	255	0	0
0	0	255	255	0	0	0	0	0	0	0	0	255	255	0
0	255	255	0	0	0	0	0	0	0	0	0	255	255	0
0	255	255	0	0	0	0	0	0	0	0	0	255	255	0
0	255	255	0	0	0	0	0	0	0	0	0	255	255	0
0	255	255	0	0	0	0	0	0	0	0	0	255	255	0
0	255	255	0	0	0	0	0	0	0	0	0	255	255	0
0	255	255	0	0	0	0	0	0	0	0	0	255	255	0
0	255	255	0	0	0	0	0	0	0	0	0	255	255	0
0	0	0	255	255	255	255	255	255	255	255	0	0	0	0
0	0	0	0	0	255	255	255	255	255	255	0	0	0	0

■ **Рис. 8.** Фрагмент эталонного изображения 0_157 с выделенным участком одного признака

■ **Fig. 8.** Fragment of the sample image 0_157 with a selected area of one feature

чений пикселей размерностью 5:5, на основе которой и будет вычисляться ядро первого канала.

Каждое значение веса матрицы ядра $w_{i,j}^{(ядро)}$ может вычисляться на основе соответствующего ему значения таблицы свертки $C_{i,j}$ (для первого сверточного слоя это будет значение пикселя изображения), масштабированного соотношением разницы максимального ($w_{max}^{(ядро)} = 1$) и минимального ($w_{min}^{(ядро)} = -1$) значения весов к разнице максимального и минимального значения таблицы свертки (для первого сверточного слоя значения $C_{max} = 255, C_{min} = 0$ определяются максимальным и минимальным значением пикселя изображения). Например, значение веса $w_{ij}^{(ядро)}$ для данного признака на рис. 8 может определяться из выражения

$$w_{i,j}^{(ядро)} = \frac{w_{max}^{(ядро)} - w_{min}^{(ядро)}}{C_{max} - C_{min}} \cdot C_{i,j} - 1 = (2C_{i,j} / 255) - 1. \quad (9)$$

Таким образом вычисляются все значения весов в матрице ядра (рис. 9).

Значение смещения для данного канала первого слоя

$$w_0 = K\% \left(\sum_{i=1}^5 \sum_{j=1}^5 (C_{i,j} \cdot w_{i,j}^{(ядро)}) \right), \quad (10)$$

где K – выбранное значение процента от произведения матрицы вычисленного ядра (см. рис. 9) и матрицы признака изображения (см. рис. 8), на основе которого ядро было создано.

Далее на основе полученного ядра и смещения определяются все схожие признаки с участками изображений, близкие к изображению отмеченного признака (см. рис. 8):

$$C_{n,m}^{(1)} = \sum_{i=1}^5 \sum_{j=1}^5 (C_{n+i,m+j} \cdot w_{i,j}^{(ядро)}) - w_0, \quad (11)$$

где $C_{n,m}^{(1)}$ – значение первого канала первого слоя свертки с координатами значений канала n и m .

-1	-1	-1	-1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	1
-1	1	1	1	1
1	1	1	1	1

■ **Рис. 9.** Аналитически вычисленное ядро для признака, выделенного на рис. 8

■ **Fig. 9.** Analytically calculated kernel for the selected feature in fig. 8

Для данного примера n и m меняются последовательно в диапазоне [0, 24]. Согласно функции активации *ReLU*, наиболее часто используемой в слоях свертки сверточных сетей, значение канала будет определяться следующим образом:

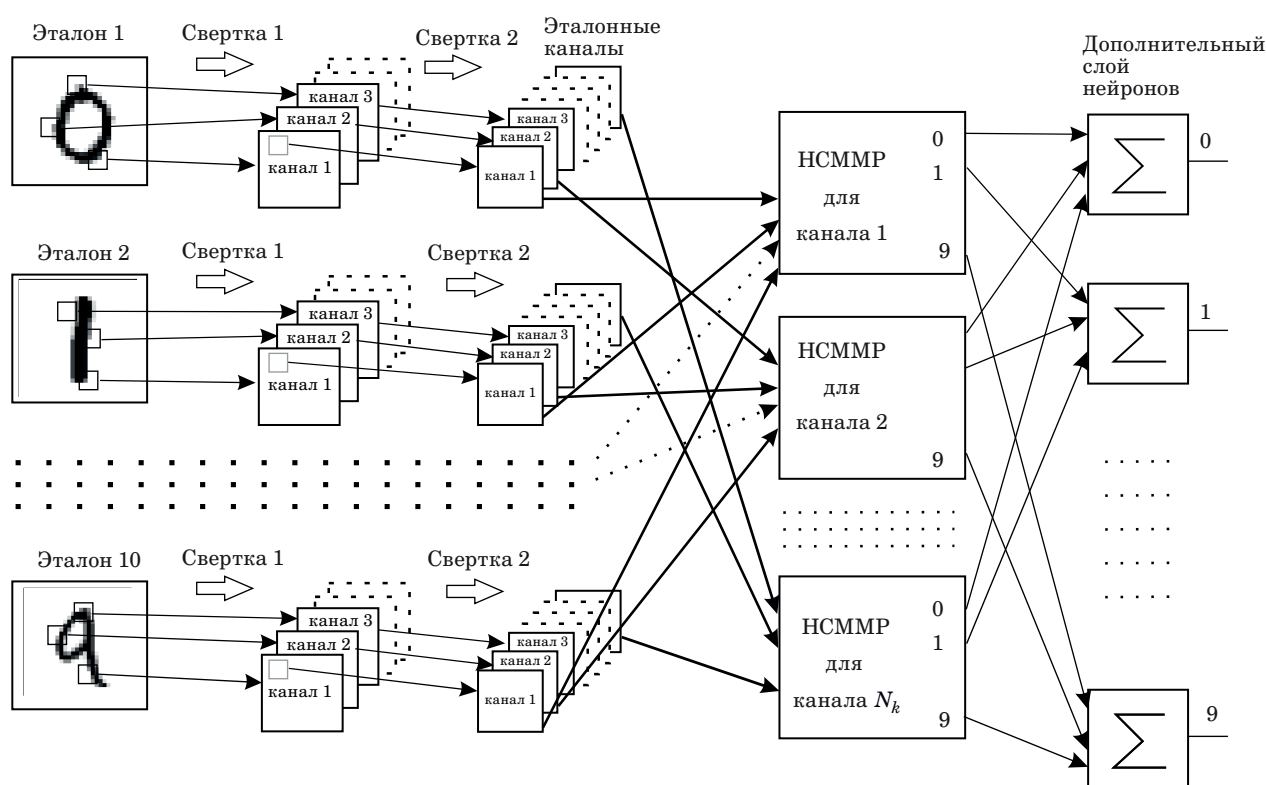
$$\begin{aligned} \text{если } C_{n,m}^{(1)} > 0, f(C_{n,m}^{(1)}) &= C_{n,m}^{(1)}; \\ \text{если } C_{n,m}^{(1)} \leq 0, f(C_{n,m}^{(1)}) &= 0. \end{aligned} \quad (12)$$

Так выполняется свертка слоя и вычисление первого канала для эталона 0_157. Таким же образом на основе ядра на рис. 9 и смещения w_0 вычисляются первые каналы для всех эталонов на рис. 2.

Далее алгоритм выбирает следующий признак (с учетом исключения повторений признаков) на изображении эталона (см. рис. 8), по описанному выше образцу вычисляет ядро и смещение для второго канала и на основе полученного ядра и смещения вычисляет следующие каналы для всех эталонов. Чем больше выбирается значение K , тем больше будут значения смещений w_0 и тем точнее выделяются признаки, схожие с признаком, на основе которого вычислялось ядро [см. формулу (10)], но при этом будет увеличиваться и количество каналов, поскольку в таком случае увеличивается и количество признаков эталонных изображений.

Процесс повторяется до полного завершения признаков на всех эталонных изображениях. Для вычисления следующих сверточных слоев процесс повторяется аналогичным образом с той только разницей, что признаки будут выбираться уже не на основе изображения эталона, а на основе вычисленного канала в предыдущем слое, а само ядро для этих слоев будет не двумерным, как в первом слое (см. рис. 9), а трехмерным. Третье измерение ядра будет равно количеству каналов предыдущего слоя, что связано с тем, что признак $C_{i,j}$ для всех слоев > 1 будет также трехмерным. Значения каждого k -го слоя трехмерного ядра будут вычисляться аналогично по выражению (9).

В итоге на выходе последнего канала свертки вычисляемой сверточной НС мы получаем точно определенное количество вычисленных каналов ($N_{канал}$), которое одинаково для каждого эталона. Для каналов разных эталонов, но с одинаковыми порядковыми номерами, строится полносвязная трехслойная НС по схеме на основе НСММР, где количество эталонов также будет равным 30. Аналогичным образом создаются полносвязные НС для каждого канала выходного сверточного слоя. На рис. 10 приведена схема создания вычисляемой сверточной НС с использованием 10 эталонов (по одному эталону от каждого образа на рис. 2). Значения всех одинаковых по номеру выходов полносвязных НС НСММР подаются на соответствующие сум-



■ **Рис. 10.** Схема создания сверточной НС с вычисляемыми весами на основе НСММР

■ **Fig. 10.** Scheme for creating a deep convolutional neural network with calculated weights based on neural networks of metric methods for recognizing

мирующие линейные нейроны. Окончательный выход НС определяется по правилу наибольшего значения.

Заключение

Проведенные выше эксперименты и полученные результаты позволяют сделать следующие выводы.

Значения весов многослойного персептрона могут сразу вычисляться аналитически с использованием небольшого набора выбранных эталонов, что дает возможность сразу создать работоспособную НС без обучения, которая может дополнительно дообучаться классическими алгоритмами обучения.

Процедура дообучения многослойного персептрона с вычисленными весовыми значениями выполняется существенно быстрее, чем обучение НС с начальной случайной генерацией весов. Следовательно, приведенный в работе [21] результат сравнительного эксперимента был не случайным, а закономерным. Более того, за счет уменьшения объема обучающей выборки время обучения НС еще более сокращается.

Результативность НС с аналитически вычисленными весовыми значениями более устойчива к уменьшению объема обучающей базы данных.

Рассматривается возможность применить предлагаемые архитектуры НСММР для реализации аналитически вычисляемых сверточных НС (см. рис. 10).

Литература

1. Кузнецов О. П., Базенков Н. И., Болдышев Б. А., Жилиякова Л. Ю., Куливец С. Г., Чистопольский И. А. Асинхронная дискретная модель химических взаимодействий в простых нейронных системах. *Искусственный интеллект и приня-*

тие решений, 2018, № 2, с. 3–20. doi:10.14357/20718594180201

2. Осипов В. Ю., Никифоров В. В. Возможности рекуррентных нейронных сетей с управляемыми элементами по восстановлению потоков кадров. *Информационно-управляющие системы*, 2019, № 5, с. 10–17. doi:10.31799/1684-8853-2019-5-10-17

3. **Baymurzina D., Golikov E., Burtsev M.** A review of neural architecture search. *Neurocomputing*, 2021, no. 474, pp. 82–93. doi:10.1016/j.neucom.2021.12.014
4. **Peng P., Pan J., Xu H., Feng X.** Rectified-physics informed neural networks for solving stationary partial differential equations. *Computers & Fluids*, 2022, vol. 245, iss. 1. doi:10.1016/j.compfluid.2022.105583
5. **Yi Lin, Po-Jang Hsieh.** Neural decoding of speech with semantic-based classification. *Cortex*, 2022, no. 154, pp. 231–240. doi:10.1016/j.cortex.2022.05.018
6. **Gaba S., Budhiraja I., Kumar V., Garg S., Kad-doum G., Hassan M. M.** A federated calibration scheme for convolutional neural networks: Models, applications and challenges. *Computer Communications*, 2022, vol. 192, pp. 144–162. doi:10.1016/j.comcom.2022.05.035
7. **Guangyuan Liao, Limin Zhang.** Solving flows of dynamical systems by deep neural networks and a novel deep learning algorithm. *Mathematics and Computers in Simulation*, 2022, vol. 202, no. 1, pp. 331–342. doi:10.1016/j.matcom.2022.06.004
8. **Pawlicki M., Kozik R., Choraś M.** A survey on neural networks for (cyber-) security and (cyber-) security of neural networks. *Neurocomputing*, 2022, vol. 500(1), pp. 1075–1087. doi:10.1016/j.neucom.2022.06.002
9. **Yunfei Yang, Zhen Li, Yang Wang.** Approximation in shift-invariant spaces with deep ReLU neural networks. *Neural Networks*, 2022, vol. 153, pp. 269–281. doi: 10.1016/j.neunet.2022.06.013
10. **Hola A., Czarnecki S.** Brick wall moisture evaluation in historic buildings using neural networks. *Automation in Construction*, 2022, vol. 141. doi:10.1016/j.autcon.2022.104429
11. **Jaberi M., Ravanmehr R.** Human activity recognition via wearable devices using enhanced ternary weight convolutional neural network. *Pervasive and Mobile Computing*, 2022, vol. 83. doi:10.1016/j.pmcj.2022.101620
12. **Yetian Fan, Wenyu Yang.** A backpropagation learning algorithm with graph regularization for feedforward neural networks. *Information Sciences*, 2022, vol. 607, pp. 263–277. doi:10.1016/j.ins.2022.05.121
13. **Aki Koivu, Joonas-Pekko Kakko, Mikko Sairanen.** Quality of randomness and node dropout regularization for fitting neural networks. *Expert Systems with Applications*, 2022, vol. 207, iss. 3. doi:10.1016/j.eswa.2022.117938
14. **Tao Zhang, Hui Xu, Xinlong Feng.** A non-intrusive neural network model order reduction algorithm for parameterized parabolic PDEs. *Computers & Mathematics with Applications*, 2022, vol. 119, iss. 1, pp. 59–67. doi:10.1016/j.camwa.2022.05.035
15. **Cummaudo A.** What's the minimum amount of data needed to teach a neural network? *Applied Artificial Intelligence Institute's Blog*. <https://a2i2.deakin.edu.au/2018/02/21/whats-the-minimum-amount-of-data-needed-to-teach-a-neural-network/> (дата обращения: 05.12.2022).
16. **Geidarov P. Sh.** Neural networks on the basis of the sample method. *Automatic Control and Computer Sciences*, 2009, vol. 43, no. 4, pp. 203–210. doi:10.3103/S0146411609040063
17. **Гейдаров П. Ш.** Алгоритм реализации метода ближайшего соседа в многослойном перцептроне. *Тр. СПИИРАН*, 2017, т. 2, № 51, с. 123–151. doi:10.15622/sp.51.6
18. **Биргер И. А.** *Техническая диагностика*. М.: Машиностроение, 1978. 240 с.
19. **Гейдаров П. Ш.** Алгоритм вычисления значений весов синапсов первого слоя нейронной сети на основе метрических методов распознавания. Ч. 1. *Информационно-управляющие системы*, 2020, № 2, с. 20–30. doi:10.31799/1684-8853-2020-2-20-30
20. **Гейдаров П. Ш.** Алгоритм вычисления значений весов синапсов первого слоя нейронной сети на основе метрических методов распознавания. Ч. 2. *Информационно-управляющие системы*, 2020, № 3, с. 25–38. doi:10.31799/1684-8853-2020-3-25-38
21. **Geidarov P. Sh.** Comparative analysis of the results of training the neural network with calculated weights and with random generation of the weights. *Automation and Remote Control*, 2020, vol. 81, no. 7, pp. 1211–1229. <https://doi.org/10.3103/S0146411609040063>
22. **Glorot X. and Bengio Y.** Understanding the difficulty of training deep feedforward neural networks. *Intern. Conf. on Artificial Intelligence and Statistics*, 2010, pp. 249–256.
23. **Glorot X., Bordes A., Bengio Y.** Deep sparse rectifier networks. *Proc. of the 14th Intern. Conf. on Artificial Intelligence and Statistics*, 2011, pp. 315–323.
24. **Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.** Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. *arXiv:1502.01852*.
25. **LeCun Y., Bengio Y., Hinton G.** Deep learning. *Nature*, 2015, vol. 521, no. 7553, pp. 436–444. doi:10.1038/nature14539
26. **Schmidhuber J.** Deep learning in neural networks: An overview. *Neural Networks*, 2015, vol. 61, pp. 85–117. doi:10.1016/j.neunet.2014.09.003

UDC 004.85

doi:10.31799/1684-8853-2023-2-2-14

EDN: WOHNNE

Investigating stability of a multilayer perceptron with calculated synaptic weights to smaller training sample sizesP. Sh. Geidarov^a, Dr. Sc., Tech., Associate Professor, orcid.org/0000-0002-3881-0629, plbaku2010@gmail.com^aInstitute of Control Systems of the National Academy of Sciences of Azerbaijan, Bakhtiyar Vahabzade St., 68, AZ 1141, Baku, Azerbaijan

Introduction: The process of creating artificial neural networks demands that developers have big samples of data sets, which complicates the process of creating and training an artificial neural network. At the same time, in biological neural networks of living organisms, the learning process is successfully performed with a relatively small samples of training data. Therefore, the task of creating new architectures of artificial neural networks that can effectively reduce the required samples of data in the training database is a very urgent task. **Purpose:** To test the stability of the performance of a neural network implemented on the basis of metric recognition methods with smaller training sample sizes. **Results:** We carry out comparative experiments to train a neural network using different samples of the MNIST database, the experiments being conducted with both precalculated weights and weighted random numbers. A comparative analysis of the results of the experiments shows that the performance of a neural network with precalculated weight values is more resilient to the size reduction of a training sample. We also propose a general scheme and the main provisions of the algorithm for implementing the presented technology for creating computed convolutional neural networks. **Practical relevance:** Neural networks with calculated weights require a smaller training data set, which allows one to speed up and simplify the procedure of creating, training and retraining a neural network.

Keywords – multilayer perceptron, metric recognition methods, neural network training, convolutional deep neural networks.

For citation: Geidarov P. Sh. Investigating stability of a multilayer perceptron with calculated synaptic weights to smaller training sample sizes. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2023, no. 2, pp. 2–14 (In Russian). doi:10.31799/1684-8853-2023-2-2-14, EDN: WOHNNE

References

- Kuznetsov O. P., Bazenkov N. I., Boldyshev B. A., Zhilyakova L. Yu., Kulivets S. G., Chistopolsky I. A. Asynchronous discrete model of chemical interactions in simple neuronal systems. *Scientific and Technical Information Processing*, 2018, no. 2, pp. 3–20 (In Russian). doi:10.14357/20718594180201
- Osipov V. Yu., Nikiforov V. V. Recurrent neural networks with controlled elements in restoring frame flows. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2019, no. 5, pp. 10–17 (In Russian). doi:10.31799/1684-8853-2019-5-10-17
- Baymurzina D., Golikov E., Burtsev M. A review of neural architecture search. *Neurocomputing*, 2021, no. 474, pp. 82–93. doi:10.1016/j.neucom.2021.12.014
- Peng P., Pan J., Xu H., Feng X. Rectified-physics informed neural networks for solving stationary partial differential equations. *Computers & Fluids*, 2022, vol. 245, iss. 1. doi:10.1016/j.compfluid.2022.105583
- Yi Lin, Po-Jang Hsieh. Neural decoding of speech with semantic-based classification. *Cortex*, 2022, no. 154, pp. 231–240. doi:10.1016/j.cortex.2022.05.018
- Gaba S., Budhiraja I., Kumar V., Garg S., Kaddoum G., Hassan M. M. A federated calibration scheme for convolutional neural networks: Models, applications and challenges. *Computer Communications*, 2022, vol. 192, pp. 144–162. doi:10.1016/j.comcom.2022.05.035
- Guangyuan Liao, Limin Zhang. Solving flows of dynamical systems by deep neural networks and a novel deep learning algorithm. *Mathematics and Computers in Simulation*, 2022, vol. 202, no. 1, pp. 331–342. doi:10.1016/j.matcom.2022.06.004
- Pawlicki M., Kozik R., Choraś M. A survey on neural networks for (cyber-) security and (cyber-) security of neural networks. *Neurocomputing*, 2022, vol. 500(1), pp. 1075–1087. doi:10.1016/j.neucom.2022.06.002
- Yunfei Yang, Zhen Li, Yang Wang. Approximation in shift-invariant spaces with deep ReLU neural networks. *Neural Networks*, 2022, vol. 153, pp. 269–281. doi: 10.1016/j.neunet.2022.06.013
- Hoła A., Czarnecki S. Brick wall moisture evaluation in historic buildings using neural networks. *Automation in Construction*, 2022, vol. 141. doi:10.1016/j.autcon.2022.104429
- Jaberi M., Ravanmehr R. Human activity recognition via wearable devices using enhanced ternary weight convolutional neural network. *Pervasive and Mobile Computing*, 2022, vol. 83. doi:10.1016/j.pmcj.2022.101620
- Yetian Fan, Wenyu Yang. A backpropagation learning algorithm with graph regularization for feedforward neural networks. *Information Sciences*, 2022, vol. 607, pp. 263–277. doi:10.1016/j.ins.2022.05.121
- Aki Koivu, Joonas-Pekko Kakko, Mikko Sairanen. Quality of randomness and node dropout regularization for fitting neural networks. *Expert Systems with Applications*, 2022, vol. 207, iss. 3. doi:10.1016/j.eswa.2022.117938
- Tao Zhang, Hui Xu, Xinlong Feng. A non-intrusive neural network model order reduction algorithm for parameterized parabolic PDEs. *Computers & Mathematics with Applications*, 2022, vol. 119, iss. 1, pp. 59–67. doi:10.1016/j.camwa.2022.05.035
- Cummaudo A. What's the minimum amount of data needed to teach a neural network? *Applied Artificial Intelligence Institute's blog*. Available at: <https://a2i2.deakin.edu.au/2018/02/21/whats-the-minimum-amount-of-data-needed-to-teach-a-neural-network/> (accessed 5 December 2022).
- Geidarov P. Sh. Neural networks on the basis of the sample method. *Automatic Control and Computer Sciences*, 2009, vol. 43, no. 4, pp. 203–210. doi:10.3103/S0146411609040063
- Geidarov P. Sh. Algorithm for implementing the nearest neighbor method in a multilayer perceptron. *Proc. of SPIRAS*, 2017, vol. 51, p. 123–151 (In Russian)
- Birger I. A. *Tekhnicheskaya diagnostika* [Technical diagnostics]. Moscow, Mashinostroenie Publ., 1978. 240 p. (In Russian).
- Geidarov P. Sh. Algorithm for calculating synapse weights of the first layer of a neural network on the base of metric recognition methods. Part 1. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2020, no. 2, pp. 20–30 (In Russian). doi:10.31799/1684-8853-2020-2-20-30
- Geidarov P. Sh. Algorithm for calculating synapse weights of the first layer of a neural network on the base of metric recognition methods. Part 2. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2020, no. 3, pp. 25–38 (In Russian). doi:10.31799/1684-8853-2020-3-25-38
- Geidarov P. Sh. Comparative analysis of the results of training the neural network with calculated weights and with random generation of the weights. *Automation and Remote Control*, 2020, vol. 81, no. 7, pp. 1211–1229. <https://doi.org/10.3103/S0146411609040063>

22. Glorot X. and Bengio Y. Understanding the difficulty of training deep feedforward neural networks. *Intern. Conf. on Artificial Intelligence and Statistics*, 2010, pp. 249–256.
23. Glorot X., Bordes A., Bengio Y. Deep sparse rectifier networks. *Proc. of the 14th Intern. Conf. on Artificial Intelligence and Statistics*, 2011, pp. 315–323.
24. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. *arXiv:1502.01852*.
25. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*, 2015, vol. 521, no. 7553, pp. 436–444. doi:10.1038/nature14539
26. Schmidhuber J. Deep learning in neural networks: An overview. *Neural Networks*, 2015, vol. 61, pp. 85–117. doi:10.1016/j.neunet.2014.09.003

Уважаемые авторы!

При подготовке рукописей статей необходимо руководствоваться следующими рекомендациями.

Статьи должны содержать изложение новых научных результатов. Название статьи должно быть кратким, но информативным. В названии недопустимо использование сокращений, кроме самых общепринятых (РАН, РФ, САПР и т. п.).

Текст рукописи должен быть оригинальным, а цитирование и самоцитирование корректно оформлено.

Объем статьи (текст, таблицы, иллюстрации и библиография) не должен превышать эквивалента в 20 страниц, напечатанных на бумаге формата А4 на одной стороне через 1,5 интервала Word шрифтом Times New Roman размером 13, поля не менее двух сантиметров.

Обязательными элементами оформления статьи являются: индекс УДК, заглавие, инициалы и фамилия автора (авторов), ученая степень, звание (при отсутствии — должность), полное название организации, аннотация и ключевые слова на русском и английском языках, ORCID и электронный адрес одного из авторов. При написании аннотации не используйте аббревиатур и не делайте ссылок на источники в списке литературы. Предоставляйте подрисовочные подписи и названия таблиц на русском и английском языках.

Статьи авторов, не имеющих ученой степени, рекомендуется публиковать в соавторстве с научным руководителем, наличие подписи научного руководителя на рукописи обязательно; в случае самостоятельной публикации обязательно предоставляйте заверенную по месту работы рекомендацию научного руководителя с указанием его фамилии, имени, отчества, места работы, должности, ученого звания, ученой степени.

Простые **формулы** набирайте в Word, сложные с помощью редактора MathType или Equation. Для набора одной формулы не используйте два редактора; при наборе формул в формульном редакторе знаки препинания, ограничивающие формулу, набирайте вместе с формулой; для установки размера шрифта в MathType никогда не пользуйтесь вкладкой Other, Smaller, Larger, используйте заводские установки редактора, не подгоняйте размер символов в формулах под размер шрифта в тексте статьи, не растягивайте и не сжимайте мышью формулы, вставленные в текст; пробелы в формуле ставьте только после запятой при перечислении с помощью Ctrl+Shift+Space (пробел); не отделяйте пробелами знаки: + = - ×, а также пространство внутри скобок; для выделения греческих символов в MathType полужирным начертанием используйте Style → Other → bold.

Для набора формул в Word никогда не используйте вкладки: «Уравнение», «Конструктор», «Формула» (на верхней панели: «Вставка» — «Уравнение»), так как этот ресурс предназначен только для внутреннего использования в Word и не поддерживается программами, предназначенными для изготовления оригинал-макета журнала.

При наборе символов в тексте помните, что символы, обозначаемые латинскими буквами, набираются светлым курсивом, русскими и греческими — светлым прямым, векторы и матрицы — прямым полужирным шрифтом.

Подробнее см. pdf-файл «Правила подготовки рукописей» (стр. 11) на сайте <https://guar.ru/ric>

Иллюстрации:

— рисунки, графики, диаграммы, блок-схемы предоставляйте в виде отдельных исходных файлов, поддающихся редактированию, используя векторные программы: Visio (*.vsd, *.vsdx); Adobe Illustrator (*.ai); Coreldraw (*.cdr, версия не выше 15); Excel (*.xls); Word (*.docx); AutoCad, Matlab (экспорт в PDF, EPS, SVG, WMF, EMF); Компас (экспорт в PDF), веб-портал DRAW.IO (экспорт в PDF);
— фото и растровые — в формате *.tif, *.png с максимальным разрешением (не менее 300 pixels/inch).

Наличие подрисовочных подписей и названий таблиц на русском и английском языках обязательно (желательно не повторяющих дословно комментарии к рисункам в тексте статьи).

В редакцию предоставляются:

— сведения об авторе (фамилия, имя, отчество, место работы, должность, ученое звание, учебное заведение и год его окончания, ученая степень и год защиты диссертации, область научных интересов, количество научных публикаций, домашний и служебный адреса и телефоны, e-mail), фото авторов: анфас, в темной одежде на белом фоне, должны быть видны плечи и грудь, высокая степень четкости изображения без теней и отблесков на лице, фото можно представить в электронном виде в формате *.tif, *.png, *.jpg с максимальным разрешением — не менее 300 pixels/inch при минимальном размере фото 40×55 мм;

— экспертное заключение.

Список литературы составляется по порядку ссылок в тексте и оформляется следующим образом:

— для книг и сборников — фамилия и инициалы авторов, полное название книги (сборника), город, издательство, год, общее количество страниц, doi;

— для журнальных статей — фамилия и инициалы авторов, полное название статьи, название журнала, год издания, номер журнала, номера страниц, doi;

— ссылки на иностранную литературу следует давать на языке оригинала без сокращений;

— при использовании web-материалов указывайте адрес сайта и дату обращения.

Список литературы оформляйте двумя отдельными блоками по образцам lit.dot на сайте журнала (<http://i-us.ru/paperrules>): Литература и References.

Более подробно правила подготовки текста с образцами изложены на нашем сайте в разделе «Руководство для авторов».

Контакты

Куда: 190000, г. Санкт-Петербург, ул. Большая Морская, д. 67, лит. А, ГУАП, РИЦ

Кому: Редакция журнала «Информационно-управляющие системы»

Тел.: (812) 494-70-02

Эл. почта: ius.spb@gmail.com

Сайт: www.i-us.ru