Determining Neural Network Weights Using an Electrostatic Field

Polad Geidarov
Institute of Control Systems, Ministry of Science and Education of Azerbaijan
Baku,Azerbaijan
plbaku2010@gmail.com

***Abstract*—This study explores an approach to determining the weights and thresholds of a neural network based on the potential of an electrostatic field, without the need for additional analytical computations or traditional training algorithms. The neural network follows a metric-based recognition method, and the electrostatic field simulation is implemented in the Builder C++ programming environment. The software computes the total electrostatic potential at designated points in the proposed model (corresponding to sensor locations—potentiometers). The same software module also enables the creation of a neural network based on metric recognition methods, where the weights of the first-layer neurons are assigned based on the computed potentials of the simulated electrostatic field. The effectiveness of the resulting neural network is evaluated using the MNIST dataset for the task of handwritten digit classification. Additionally, the possibility of applying this approach to semantic text understanding tasks is considered.**

***Keywords*—neural networks, image recognition, MNIST, training algorithms, electrostatic field potential, electrostatic field.**

I. Introduction

The use of artificial neural networks remains a promising approach for solving various tasks [1]–[4], including pattern recognition problems. Although artificial neural networks were originally designed to mimic biological neural networks, their capabilities are still significantly limited compared to their biological counterparts. For example, biological neural networks possess the ability to accumulate a large number of recognizable patterns and can rapidly memorize and recognize new patterns (classification tasks) without lengthy training procedures or large training datasets. In contrast, modern artificial neural networks typically lack these abilities.

Consequently, developing new neural network architectures that enhance the capabilities of conventional models—particularly by accelerating network creation and training—remains an important research objective. To this end, studies [5]–[8] have proposed and described neural network architectures based on metric recognition methods. These architectures exhibit several distinctive features, including:

- The ability to rapidly construct the network structure (number of layers, neurons, and connections) based on the initial problem parameters, such as the number of reference patterns and recognized classes.
- Transparency, meaning that the function of each neuron, layer, weight, threshold, and connection is well-defined and interpretable.
- Easy and flexible expansion of the network through cascade-based addition of new neurons, enabling new patterns (new recognized classes) to be incorporated into an operational network without modifying previously established weight values.
- The possibility of precomputing neural network weight values analytically using metric similarity measures.
- Compatibility with standard neural network training algorithms for additional fine-tuning.

While precomputing weight values accelerates neural network creation and training, it also requires computational time. As the number of recognized patterns and reference examples increases, the time required for weight calculations grows, particularly when higher-dimensional weight matrices are needed for the first or zeroth layer of the network.

To address this challenge, study [8] demonstrated that neural network weights can be determined using electrostatic field parameters. This method enables near-instantaneous weight assignment without analytical calculations, provided a relatively small reference dataset is available and each neuron has access to a model for reading electrostatic field parameters.

The primary objective of this study is to experimentally validate the proposed model) by assessing the functionality of a neural network whose weight values are determined based on electrostatic field parameters. In this approach, the potential of the electrostatic field serves as the key parameter.

To achieve this goal, a simulation model [8] was developed in the Builder C++ environment. This model computes the total electrostatic potential at each potentiometer sensor location (Figure 1). Additionally, a neural network based on the metric recognition method was implemented, allowing the first-layer neurons' weight values to be assigned according to the computed potentials of the simulated electrostatic field.

II. The Electrostatic Field Potential as the Weights of a Neural Network

The developed software module also facilitates performance evaluation using the MNIST (Modified National Institute of Standards and Technology) dataset. MNIST consists of handwritten digit images (0–9) and comprises two subsets: a training set (60,000 images) and a test set (10,000 images). Each image in MNIST is 28:28 pixels in size, with grayscale values ranging from 0 to 255.

In the experiments presented below, no training dataset was used, as the neural network did not undergo a training phase. Instead, only the test dataset (10,000 images) was utilized to assess the performance of the generated neural networks and compare the results. Here, the performance of the neural network is defined as the proportion of correctly classified test images.

The functionality of the generated neural network was evaluated using the Builder C++ software module.

The distance between two compared images in this example is set to 8 sm. That is, the plane for the simulated potentiometer sensors is positioned between two planes of charged surfaces of the images, at a distance of d_2 = 4 cm from each image (see Figure 1). Each active pixel on one image receives a charge of q = -10^{-9}, and q = 10^{-9} on the other image, thus creating an electrostatic field between the image panels. Each sensor on the sensor panel, located between them, measures the potential of the electrostatic field between the two images at the sensor's position.

Thus, all the obtained potential values on the sensor panel are further used as weights for the first-layer neurons, whose output separates these two images. Similarly, other reference images are compared pairwise in the same manner. The technology for instant weight determination based on the electrostatic field potential was described in more detail in [8].

III. Experimental Results

Table I presents the recognition results on the MNIST test dataset using 30 reference patterns from Figure 2a. The table shows both the results for each individual digit (where i_j represents the total number of digit j images in the MNIST test dataset, and s_j represents the number of correctly identified digit j images) and the overall result, which amounted to 5,047 correctly identified images (approximately 50%).

Table II presents the recognition results for the MNIST test set after reducing the distance between the two image planes to 4 cm and the distance to the sensor plane to d_2 = 2 cm.

Based on the results in Table II, it can be observed that the final recognition accuracy 53% improved by 3% compared to the previous experiment. This indicates that changes in the initial physical parameters of the simulated system—such as the value of the point charge q, the distance between the weight matrix cells d_1 (sensor spacing), and the distance between the image plane and the potentiometer sensor plane d_2 –also affect recognition performance.

Table III shows the recognition results for the MNIST test set after adding two additional reference patterns (Figure 2a) to each image, as shown in Figure 2c. After the new reference patterns were added, new first-layer neurons were cascaded into the neural network, and their corresponding weight tables were computed. The weight tables for the remaining neurons were not altered. Testing on the MNIST test set with the updated reference patterns increased the performance to approximately 70%.

In Table IV, the results of recognizing the MNIST test set after adding 20 new reference patterns, as shown in Figure 2c, are presented. This increased the total number of reference patterns to 70 (4 patterns for each class). The recognition results for the MNIST test set improved, with 7,721 images correctly identified (77%).

From the results in the tables, it can be observed that increasing the number of reference patterns also increases the percentage of correctly recognized images. Adding reference patterns to the neural network based on metric recognition methods does not change the previous weight values of the network.

IV. Determining Weights Based on Electrostatic Field Parameters for Semantic Tasks

This approach can theoretically also be applied to semantic tasks and natural language understanding tasks. As is known, there are various ways to encode language, implemented in different word embedding libraries. For example, Word2Vec, GloVe, BERT, GPT, and others. Language elements such as words, syllables, and word combinations are represented as vectors of numbers (embeddings). The number of digits in one word vector can be in the hundreds. To digitally represent a sentence, the text is broken down into words, syllables, or entire word combinations. For some word encoding methods, such as the BERT method, not only words or syllables are encoded but also the positional number of the word in the sentence, which allows for a more accurate understanding of the meaning of the text. For example, if for the sentence "Cat caught mouse" each word is encoded as a vector with three numbers, like this:

- "Cat" → [23, -67, 89]
- "Caught" → [12, 45, -34]
- "Mouse" → [34, .78, -23]

We end up with a matrix of size 3x3. If each number in this matrix is converted into binary code, we get a matrix with binary codes:

[00010111] [10111111] [01011001]
[00001100] [00101101] [11111110]
[00100010] [00000001] [11101001]

As a result, by analogy with black-and-white images, we obtain a binary table of the sentence, which is

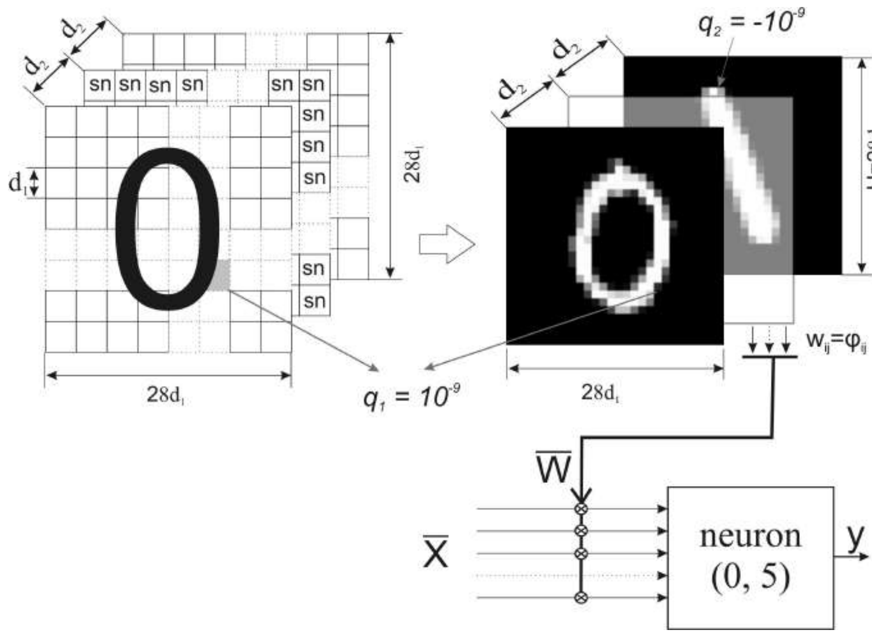

Figure 1. A scheme for obtaining neuron weight values based on the potential of an electrostatic field.

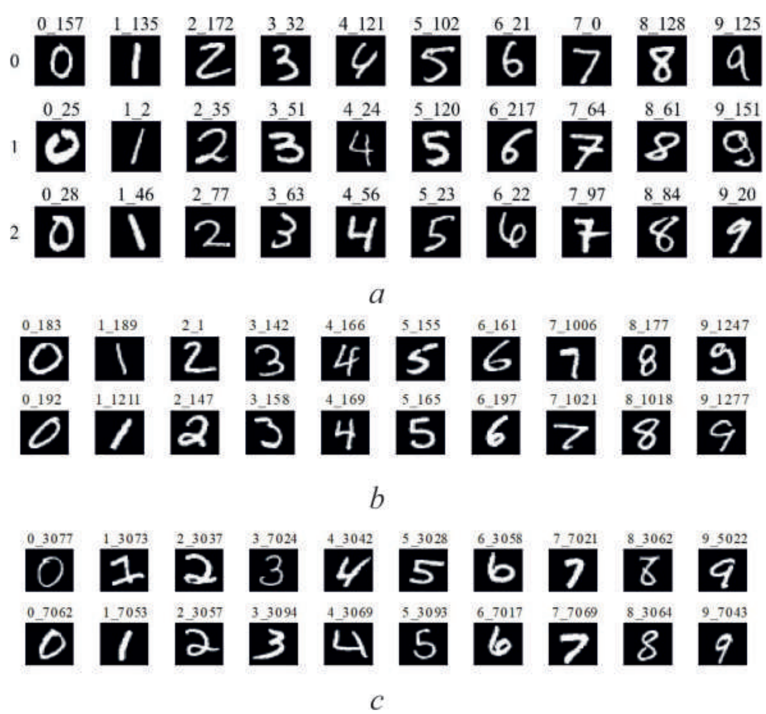

Figure 2. (a) 30 reference patterns, (b) Added reference set for the experiment with 50 reference samples, (c) 20 additional reference patterns for the experiment with 70 reference patterns.

Table I

Test results of the neural network on the MNIST test set with initial conditions: N = 30 reference patterns, q = 10^{-9} C, d_2 = 4 sm, d_2 = 2 sm. The number of recognized symbols for each digit class (s_j) The total number of characters in the database for each digit class (i_j).Percentage of correctly recognized symbols (p_j)

The number of recognized symbols for each digit class (s_j)	The total number of characters in the database for each digit class (i_j).	Percentage of correctly recognized symbols (p_j)
s0 = 701	i0 = 980	p0 = 71%
s1 = 936	i1 = 1135	p1 = 82%
s2 = 296	i2 = 1032	p2 = 28%
s3 = 361	i3 = 1010	p3 = 35%
s4 = 434	i4 = 982	p4 = 44%
s5 = 357	i5 = 892	p5 = 40%
s6 = 413	i6 = 958	p6 = 43%
s7 = 504	i7 = 1028	p7 = 49%
s8 = 500	i8 = 974	p8 = 51%
s9 = 545	i9 = 1009	p9 = 54%
In summary		
5047	10000	50%

Table II

Testing Results of the Neural Network on the MNIST Test Set with Initial Conditions: N = 30 References, q = 10^{-9} C, d_2 = 2 sm, d_1 = 2 sm. The number of recognized characters for each digit class (s_j). The total number of characters in the dataset for each digit class (i_j) Percentage of correctly recognized characters (p_j)

The number of recognized characters for each digit class (s_j)	The total number of characters in the dataset for each digit class (i_j)	Percentage of correctly recognized characters (p_j)
s0 = 729	i0 = 980	p0 = 74%
s1 = 987	i1 = 1135	p1 = 86%
s2 = 307	i2 = 1032	p2 = 29%
s3 = 387	i3 = 1010	p3 = 38%
s4 = 468	i4 = 982	p4 = 47%
s5 = 385	i5 = 892	p5 = 43%
s6 = 484	i6 = 958	p6 = 50%
s7 = 532	i7 = 1028	p7 = 51%
s8 = 477	i8 = 974	p8 = 48%
s9 = 608	i9 = 1009	p9 = 60%
In summary		
5364	10000	53%

Table III

Test Results on the MNIST Test Set Using 50 References The number of recognized characters for each digit class (s_j) The total number of characters in the dataset for each digit class (i_j) Percentage of correctly recognized characters (p_j)

The number of recognized characters for each digit class (s_j)	The total number of characters in the dataset for each digit class (i_j)	Percentage of correctly recognized characters (p_j)
s0 = 852	i0 = 980	p0 = 87%
s1 = 1123	i1 = 1135	p1 = 99%
s2 = 516	i2 = 1032	p2 = 50%
s3 = 585	i3 = 1010	p3 = 58%
s4 = 775	i4 = 982	p4 = 79%
s5 = 588	i5 = 892	p5 = 66%
s6 = 555	i6 = 958	p6 = 58%
s7 = 616	i7 = 1028	p7 = 60%
s8 = 633	i8 = 974	p8 = 65%
s9 = 726	i9 = 1009	p9 = 72%
In summary		
6969	10000	70%

Table IV

Testing results on the MNIST test set using 70 reference patterns. The number of recognized characters for each digit class (s_j) The total number of characters in the dataset for each digit class (i_j) Percentage of correctly recognized characters (p_j).

The number of recognized characters for each digit class (s_j)	The total number of characters in the dataset for each digit class (i_j)	Percentage of correctly recognized characters (p_j)
s0 = 852	i0 = 980	p0 = 87%
s1 = 1123	i1 = 1135	p1 = 99%
s2 = 516	i2 = 1032	p2 = 50%
s3 = 585	i3 = 1010	p3 = 58%
s4 = 775	i4 = 982	p4 = 79%
s5 = 588	i5 = 892	p5 = 66%
s6 = 555	i6 = 958	p6 = 58%
s7 = 616	i7 = 1028	p7 = 60%
s8 = 633	i8 = 974	p8 = 65%
s9 = 726	i9 = 1009	p9 = 72%
In summary		
7721	10000	77%

widely used in image recognition with neural networks. If we change the order of words, for example: "Mouse caught cat" then in this case, the binary image table also changes:

[00100010] [00000001] [11101001]
[00010111] [10111111] [01011001]
[00001100] [00101101] [11111110]

Accordingly, the neural network will be able to distinguish between these two sentences. The figure 3 shows the binary images of the two sentences: "Cat caught mouse" and " Mouse caught cat" constructed from the two matrices above, where 0 corresponds to the white area and 1 corresponds to the black area. Similarly to Fig. 1, the figure 3 presents the scheme for obtaining the first layer neuron weights based on the electrostatic field parameter – potential. The value of each pixel in the first binary image equal to 1 has a charge q = 10(-9), and the value of each pixel in the second binary image equal to 1 has the opposite charge q = -10(-9). As a result, an electrostatic field is formed between the two images. The figure 3 also shows that between the two binary images is a layer of 3*24=72 potentiometer sensors measuring the potential of the electric field at their location. The values of the obtained potentials are immediately fed as the weights of the first-layer neuron. As a result, this neuron can immediately distinguish between the two sentences without training. Similarly, for other pairs of images, the weights for other neurons in the first layer can also be directly defined.

V. Conclusion

Thus, the experiments conducted above show that a neural network with weight values determined by the characteristics of the electrostatic field is feasible. This indicates that the neural network can instantly determine weight values with a small reference set.

This means that the neural network can significantly faster memorize and recognize new patterns compared to standard neural network architectures. Furthermore, if needed, the resulting neural networks can be further trained using conventional training algorithms. As shown in [7], even in this case, the process of creating a neural network, calculating its weight and threshold values, and further training occurs faster than training a neural network using traditional methods with randomly initialized weight values. The model for determining the weight and threshold values of the neural network based on the electrostatic field parameters can further accelerate this entire process.

The reasoning presented above in this work shows that the capabilities of neural networks are even broader and allow not only computation but also instant memorization of patterns without training and calculations, provided there is a small set of reference patterns. If we consider the fact that electric charges and electric fields are omnipresent in the living cells of neurons in living organisms, as well as in the retinas of living organisms, etc., it can be hypothesized that living organisms could likely possess similar abilities.

References

[1] Krizhevsky, A., Sutskever, I., Hinton, G. E. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 2012, V. 25, pp. 1097-1105.

[2] He, K., Zhang, X., Ren, S., Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. Attention is all you need. Advances in Neural Information Processing Systems, 2017, V. 30, pp. 5998-6008.

[4] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171-4186.

[5] Geydarov, P. Sh. Algorithm for calculating the synapse weights of the first layer of a neural network based on metric methods of recognition. Part 1. Information and Control Systems, 2020, №. 2, pp. 20-30.

[6] Geydarov, P. Sh. Algorithm for calculating the synapse weights of the first layer of a neural network based on metric methods of recognition. Part 2. Information and Control Systems, 2020, №. 3, pp. 25-38.

[7] Geidarov, P. Sh. Comparative Analysis of the Results of Training a Neural Network with Calculated Weights and with Random Generation of the Weights. Automation and Remote Control, 2020, V. 81, №. 7, pp. 1211-1229.

[8] Geidarov P. Sh. (2022) Experiment for Creating a Neural Network with Weights Determined by the Potential of a Simulated Electrostatic Field. Scientific and Technical Information Processing, 2022, V.49, № 6, pp. 1–13.

[9] Zhaksybaev, D. O., Mizamova, G. N. Algorithms for natural language processing for text semantics understanding. Trudy ISP RAN, 2022, V. 34, №. 1, pp. 141-150.

[10] Naidenova, K. A., Nevzorova, O. A. Machine learning in natural language processing tasks: A review of the current state of research. Proceedings of Kazan State University. Series: Physical and Mathematical Sciences, 2008, V. 150, №. 4, pp. 5-24.

[11] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. Language models are few-shot learners. Proceedings of Advances in Neural Information Processing Systems, 2020, V. 33, pp. 1877-1901.

[12] Zhelyabin, D. V. Application of machine learning methods for solving the NLP text classification problem based on natural language semantics analysis. Bulletin of the Altai Academy of Economics and Law, 2020, №. 6, pp. 229-235.

[13] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171-4186.

[14] Cer, D., Yang, Y., Kong, S.-Y., et al. Universal Sentence Encoder. arXiv preprint arXiv:1803.11175, 2018.

[15] Reimers, N., Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 2019, pp. 3982-3992.

[16] Pennington, J., Socher, R., Manning, C. D. GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532-1543.

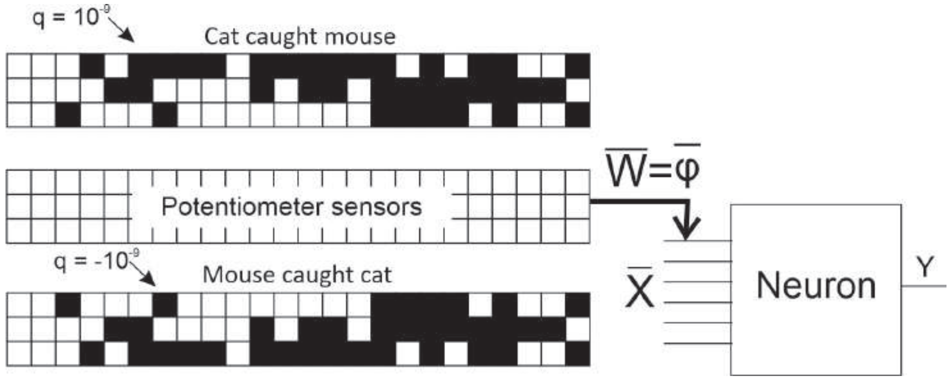

Figure 3. Scheme for determining neuron weights based on the electrostatic field potential for a semantic task.

ОПРЕДЕЛЕНИЕ ВЕСОВ НЕЙРОННОЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ ЭЛЕКТРОСТАТИЧЕСКОГО ПОЛЯ

Гейдаров Полад

В данном исследовании рассматривается подход к определению весов и порогов нейронной сети на основе потенциала электростатического поля, без необходимости в дополнительных аналитических вычислениях или использовании традиционных алгоритмов обучения. Нейронная сеть работает по методу метрического распознавания, а моделирование электростатического поля реализовано в среде программирования Builder C++. Программное обеспечение вычисляет общий электростатический потенциал в заданных точках предложенной модели (соответствующих расположению датчиков – потенциометров). Этот же программный модуль позволяет создавать нейронную сеть на основе методов метрического распознавания, в которой веса нейронов первого слоя назначаются на основе вычисленных потенциалов смоделированного электростатического поля. Эффективность полученной нейронной сети оценивается с использованием набора данных MNIST для задачи классификации рукописных цифр. Также рассмотрена возможность применение этого подхода в задачах семантического понимания текста.

Received 30.03.2025