Systems Theory principles for investigating the spiking neural network trained with the Hebbian rule

1 author:

Dmitry Antonov
Kotel'nikov Institute of Radio Engineering and Electronics of Russian Academy of Sciences (Ulyanovsk branch)

НЕЙРОКОМПЬЮТЕРЫ, НЕЙРОМОРФНЫЕ ВЫЧИСЛЕНИЯ И ИМПУЛЬСНЫЕ НЕЙРОННЫЕ СЕТИ

D.I. ANTONOV
Ulyanovsk State Technical University
d.antonov@ulireran.ru

SYSTEMS THEORY PRINCIPLES FOR INVESTIGATING THE SPIKING NEURAL NETWORK TRAINED WITH THE HEBBIAN RULE*

The article presents the results of research on the synaptic weights distribution obtained by applying a new local learning rule for spiking neural networks (SNNs). The developed method uses a combination of Hebbian rules: spike-time dependent plasticity (STDP) and long-term depression (all-LTD). The synaptic weights distribution of a 3-layer SNN trained according to the combined rule allowed, after training, to restructure the synaptic weights distribution of SNN based on the principles of Systems Theory.

Keywords: *spiking neural network, supervised learning, spike-timing-dependent plasticity, all-time long-term depression, learning algorithm, Hebbian rule, Systems Theory, information entropy.*

1. Introduction

1.1. Systems Theory principles

A neural network is a very relevant example of a system. The latest generation of neural networks, spiking neural networks (SNN), demonstrates the behavior that corresponds to the main concepts of Systems Theory and, as L. von Bertalanffy wrote [1], to the provisions of disciplines that have common goals and methods with Systems Theory (Combinatorics, Information Theory).

Cl. Shannon introduced the concept of information entropy [2] as a measure of the uncertainty of a system for a random variable x taking n independent random values x_i with probabilities p_i:

$$H(x) = -\sum_{i=1}^{n} p_i \log_2 p_i, \tag{1}$$

where $H(x)$ is the amount of information in one event.

* The reported study was funded by the Russian Science Foundation (project number 24-21-00470).

A.N. Kolmogorov introduced a formal analog of the concept of information entropy for a continuous random variable ξ distributed over $X \subseteq R^n$ (n<∞) [3], differential entropy, a functional defined on a set of absolutely continuous probability distributions:

$$H(\xi) = -\int_X f(x) \log_2 f(x) dx, \quad (2)$$

where $f(x)$ is the distribution density of a random variable.

Formulas (1) and (2) show that the minimum information is in a system with a simple structure, the ideal order corresponds to the minimum entropy.

In this way, a connection is established between the degree of information content and orderliness: the greater the degree of orderliness of information, the more it can be compressed. If it is possible to identify a regularity of data distribution, then you can significantly compress the amount of data.

One of the classical definitions describes a system (in Systems Theory) as a set of elements and their relationships, that is, a structure. The structure has a certain hierarchy: each component of the system can also be considered as a system (subsystem), and the properties of the individual components in total are not equivalent to the properties of the entire system.

Information in neural networks is contained in synaptic weights distributed in a certain structure. If the structure of synaptic weights is such that it can be described (in a whole or in a part) using some distribution function, then this means that it can be simplified without loss of system information.

1.2. Spiking Neural Networks learning rules

Research on SNN and the development of their learning rules are relevant today. Neuromorphic systems are bio-inspired and generally operate on the principles of SNN.

Neural network learning methods that have shown good performance for artificial neural networks (ANNs) cannot be directly used for spiking neural networks. The error backpropagation algorithm [4], which has become standard for ANNs, in the case of SNN requires special surrogates.

As we know, biological neurons learn locally rather than globally, hence gradient methods are biologically implausible. And what is more important, gradient methods require additional time for the correction signal to pass through.

Local learning rules based on Hebb's rule [5] are more suitable for training SNNs. Hebb's rule defines a learning process in which connections between neurons that fire one after another should be strengthened, and connections between neurons that fire independently of each other should be weakened. Local methods are bio-inspired but typically provide Unsupervised rather than Supervised Learning. Unfortunately, Unsupervised Learning is more suited to clustering problems and is poorly suited to distinguishing between data with fairly similar features.

We developed a Supervised Learning method for SNNs on combined Hebbian learning rules [6], in which no correction signal is required.

2. The combined Hebbian learning rule and SNN architecture

2.1. Combined Hebbian learning rule, theory

The developed method for SNNs [9] is based on two bio-plausible learning rules: canonical Hebbian spike-timing-dependent plasticity (STDP) and all-time long-term depression (all-LTD).

The STDP rule adjusts the strength of synaptic connection w changes depending on the time difference in the excitation of spikes in the pre- and postsynaptic neurons $\Delta t = t_{post} - t_{pre}$ (Fig. 1a):

- if the presynaptic spike precedes the postsynaptic one (in a period of less than 50 ms) $\Delta t>0$ and the connection is strengthened,
- if, on the contrary, the postsynaptic spike precedes the presynaptic one (in the same period) $\Delta t<0$ and the synaptic connection decreases.

The change in synaptic connection is determined by the following expression:

$$\Delta w(\Delta t) = \begin{cases} A_{pre} \cdot \exp(-\Delta t/\tau_{pre}), & \Delta t > 0 \\ A_{post} \cdot \exp(\Delta t/\tau_{post}), & \Delta t < 0 \end{cases}, \tag{3}$$

where $A_{pre} > 0$, $A_{post} < 0$.

The STDP mechanism provides the sensitivity of postsynaptic neurons to the specific features of the input signal and thus teaches a neuron to respond to a specific input.

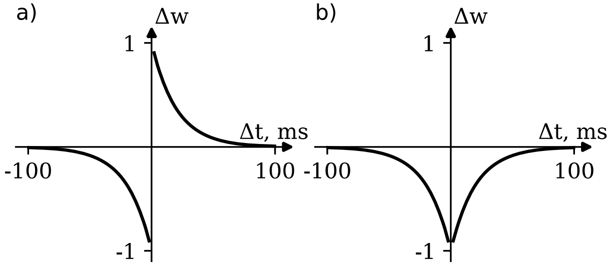

Fig. 1. Canonical Hebbian STDP (a) and all-LTD (b)

On the contrary, the all-LTD rule provides the ignoring the specific features of the input signal and thus teaches a neuron to remain silent to a specific input:

if pre- and postsynaptic spikes both fall within the interval $|\Delta t|$ (within the same period of 50 ms), the connection strength between the neurons decreases. This mechanism of synaptic connection change is determined by the following expression (Fig. 1b):

$$\Delta w(\Delta t) = \begin{cases} -A_{pre} \cdot \exp(-\Delta t/\tau_{pre}), & \Delta t > 0 \\ A_{post} \cdot \exp(\Delta t/\tau_{post}), & \Delta t < 0 \end{cases}. \quad (4)$$

2.2. SNN's architecture

The rule was initially tested on a network based on the architecture proposed in [7] and later revised in [8].

The developed SNN uses the leaky integrate-and-fire model of neuron.

The original network in [6] has two layers: the first input layer contains Poisson neurons and the second layer consists of an equal number of excitatory and inhibitory neurons. To improve classification accuracy, a third layer was added to the network (Fig.2) with an equal number of excitatory and inhibitory neurons.

Each training image at the SNN input causes the next steps (Fig.2):

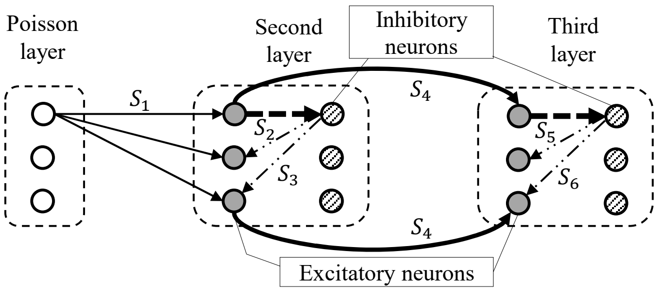

Fig. 2. SNN architecture: the first layer contains 784 Poisson neurons, the second layer contains 100 excitatory and 100 inhibitory neurons, the third layer has 30 excitatory and 30 inhibitory neurons; S1, S2, S3, S4, S5, and S6 are the synaptic connections

- Each Poisson layer neuron receives one pixel of input data, after which the neuron generates a spike train with an average frequency equal to the intensity of a given pixel (rate coding).
- The signal passes from Poisson neurons to excitatory neurons of the second layer through synapses of the S_1 group, connecting Poisson neurons with excitatory neurons of the second layer in one-to-all manner.
- The spike trains arrive at the excitatory neurons of the second layer and increase their membrane potential V, when potential V reaches the threshold level V_{thres}, the excitatory neuron generates spikes. Through the synapses

of the S_2 group, the spikes pass to the inhibitory neurons of the second layer, inducing spikes in them, since the synapses of the S_2 group connect them in a one-to-one manner. The weights of synapses in group S_2 remain constant.

- Through synapses of the S_3 group, the spikes generated by inhibitory neurons of the second layer return to excitatory neurons of the second layer according to the one-to-all-except-initiator manner. Synaptic connections of the S_3 group provide lateral inhibition. The weights of group S_3 remain constant.
- The spike trains arrive from the excitatory neurons of the second layer to the excitatory neurons of the third layer (increasing their membrane potential V to the threshold level V_{thres}) through the synapses of the S_4 group connecting them in a one-to-all manner.
- Subsequently, the signal goes through the synapses of groups S_5 and S_6, providing communication similarly to the synapses of groups S_2 and S_3, respectively:
  - through the synapses of the S_5 group, the spikes pass to the inhibitory neurons of the third layer, connected in a one-to-one manner (with constant synapse weights),
  - through synapses of the S_6 group, the spikes of inhibitory neurons of the third layer return to excitatory neurons of the third layer, connected in a one-to-all-except-initiator manner (with constant synapse weights).

During training, images are presented sequentially, after the spike train by a 350 ms (triggered by a single image) followed by a 150 ms period with no inputs. In the 150 ms resting period, the potential of the excitatory neurons drops to a lower threshold, after which the loop is repeated in the next image.

During the testing process, the class is determined by the highest activity of the neuron populations assigned to each class.

2.3. Training process

Training is carried out on S_1 and S_4 synaptic groups: S_1 group is trained according to the canonical STDP rule, which leads to the clustering of data features, and group S_4 is trained according to the combined Hebbian rule *‘STDP + all-LTD’*.

To implement Supervised Learning, we divided the excitatory neurons of the third layer into subsets, the number of subsets is equal to the number of data classes, and the sizes of these subsets are the same. The subset number coincides with the class number that this subset classifies. If the spike reaches a subset of excitatory neurons in the third layer with the same number as the data class number, then this action leads to a correct classification.

Synapses of the S_4 group associated with excitatory neurons of the third layer are trained according to the rule *'STDP + all-LTD'*. When input data of a certain class (target) are given, synapses associated with a subset of neurons recognizing the target class are trained according to the STDP rule. The remaining synapses associated with neurons recognizing other classes are trained according to the all-LTD rule.

Two details are an important part of the learning rule implementation: lateral inhibition and adaptive threshold. Inhibitory neurons carry out lateral inhibition and thereby reveal a variety of class features: synapses associated with neurons corresponding to unique features are strengthened, and synapses associated with neurons corresponding to general features are weakened. The adaptive threshold allows the activity of excitatory neurons to be coordinated with the strength of the signs; it changes in proportion to the number of activations of a given neuron. Due to the adaptive threshold, not only strongly expressed features are highlighted, but also less pronounced features.

3. Experiment

3.1. Conditions of experiments

We used a computer with an Intel Core i9 processor (3.1 GHz), 16 GB RAM, PyTorch 1.8.0 and Ubuntu 18.04 to run the code. The SNN was implemented in the Brian 2.0 package, an open-source framework for SNN modeling.

The network was trained to classify data from the MNIST set. Due to the low speed of training of SNN on complete MNIST set, we used just 5000 images for the 1-st training step (S_1 training), 900 images for the 2-nd training step (S_4 training), and 300 images for the testing. In the experiments, we used only 3 classes of images out of 10 possible.

The SNN used consists of three layers, the first layer contains 784 Poisson neurons (one neuron per pixel of a 28×28 image), the second layer consists of 100 excitatory and 100 inhibitory neurons, the third layer has 30 excitatory and 30 inhibitory neurons (10 excitatory neurons per class).

In [6], the following classification accuracy was obtained on 2-layer SNN:

- (74.9±19.8)% without adaptive threshold,
- (80.0±13.0)% with an adaptive threshold.

A significant feature of such a network is its ability to learn rapidly: classification accuracy of about 80% is achieved when training the network with just 100 images.

To improve the quality of the network, we decided to add another layer. The first testing of the new 3-layer network showed classification accuracy close to the accuracy obtained as a result of random guessing of equally likely outcomes (for the 3-class option, the accuracy of random guessing is 1/3). The selection of hyperparameters did not lead to a significant increase in accuracy, which was the reason for an in-depth analysis.

3.2. Preliminary analysis

Based on the results of the initial analysis, the main object of subsequent research was the synaptic weights of the S_4 group. The weight distribution of S4 groups allows a clear visual representation on the graph (Fig. 3).

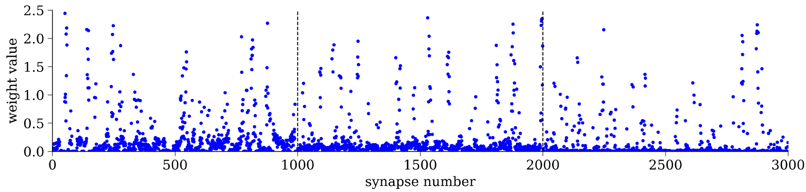

Fig. 3. Distribution of S_4 synaptic weights group after training; weight subgroups are separated by the vertical dashed lines

In the experiments, excitatory neurons of the third layer, according to the *'STDP + all-LTD'* rule, were divided into 3 subsets, each of which was responsible for classifying one of the 3 classes of data (the subset number coincided with the class number).

The synapses of S_4 group are associated to the three 10-element subsets ($10 \times 3 = 30$) of the third layer excitatory neurons.

Visually, Fig. 3 shows a densely distributed subsystems of lighter weights ('light' weights at the bottom of the graph) and a less densely distributed subsystems of heavier weights ('heavy' weights at the top).

The densities of the 'light' weight subsystem and the 'heavy' weight subsystem within the subgroup are quite constant. The analysis showed that different distributions of weights in subgroups are directly related to different degrees of class definition accuracy.

Each weights subgroup is associated with the definition of one class, and the best classification accuracy was shown by the subgroup in which:

- 'light' weights are distributed so that they occupy a relatively wider range,
- the distribution of 'light' weights (without taking into account weights with 0 value) is close to a uniform distribution,
- the sum of the 'heavy' weights is maximal,
- dividing the weights into 'heavy' and 'light' ones is most effective if the number of 'heavy' weights is about 10% of the number of all weights (the

rest are the ‘light’ ones). This result is consistent with our previous research regarding the importance of weights for classification accuracy [9].

The best value for the point separating ‘light’ and ‘heavy’ weights was determined as the inflection point of the weight density distribution curve of the subgroup that had the maximum sum of ‘heavy’ weights (the inflection point mentioned below refers specifically to this class).

3.3. Confirming assumptions based on the principles of Systems Theory

Based on the results of the analysis, we suggested that ‘light’ and ‘heavy’ weights play different roles in the classification process (taking into account the probabilistic nature of the process): ‘light’ weights maintain the potential level in excitatory neurons, ‘heavy’ weights directly fire excitatory neurons.

The suggestion about the different roles of weights allowed us to draw an analogy and assume using the base concepts of Systems Theory: the division of weights into ‘light’ and ‘heavy’ corresponds to their level of information content. The ‘heavy’ weights subsystem contains almost all the information of the weights system; in the ‘light’ weights subsystem, information entropy is close to 0, which makes it possible to radically simplify the ‘light’ weights subsystem.

To confirm the assumption of information redundancy (on 3-layer SNN with adaptive threshold), two series of experiments were carried out, during which, on the one hand, the number of ‘heavy’ weights in subgroups and the total sum of ‘heavy’ weights in subgroups were equalized (by compressing their structure upward), and on the other hand, a simplification of ‘light’ weight subsystems was carried out:

- in the 1-st series, ‘light’ weights (obtained as a result of training) were replaced by random variables uniformly distributed from 0 to the inflection point (Fig. 4); on the graph, ‘light’ weights are presented in the form of an equal-wide ‘foaming’ strip obtained accuracy (68.4±4.9)%,

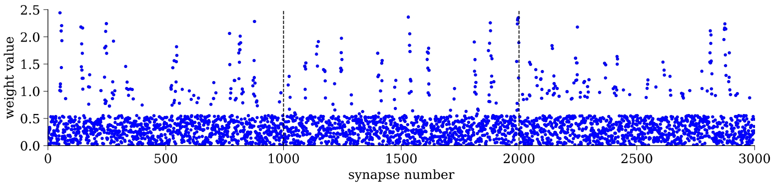

Fig. 4. Distribution of S4 group, where ‘heavy’ weights were equalized, ‘light’ weights were replaced by uniformly distributed random variables

- in the 2-nd series, ‘light’ weights were replaced by a single value (Fig. 5); on the graph, ‘light’ weights are represented as a horizontal line accuracy (67.8±3.1)%.

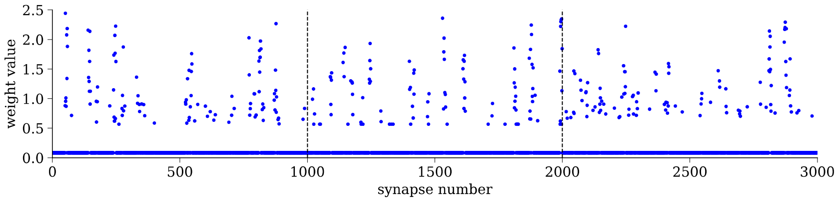

Fig. 5. Distribution of S_4 group, where ‘heavy’ weights were equalized, values of ‘light’ weights were replaced with a single value

Simplification of the ‘light’ weights structure together with the compression of the ‘heavy’ weights structure led to an increase in accuracy, which shows the correctness of the assumption about the information content levels of subgroups.

We think that this indicates that ‘light’ weights, having minimal information content, play a significant role in maintaining a certain level of potential in the excitatory neurons of the third layer. To confirm this assumption about the role of ‘light’ weights, we conducted two more series of experiments (on SNN with adaptive threshold) in which:

- the values of the ‘light’ weights were replaced with the 0-th value, and the structure of the ‘heavy’ was compressed, which led to a drop in accuracy to (64.0±4.5)%,
- all ‘light’ weights were replaced by one value, and all ‘heavy’ weights were also replaced by one value, which led to a drop in accuracy to (50.5±6.7)%.

The results of the new experiment series, in addition to confirming the role of ‘light’ weights, showed that the ‘heavy’ weights subsystem contains a significant part of system information, but the help of ‘light’ weights in maintaining a certain level of potential of excitatory neurons significantly increases classification accuracy.

4. Conclusion

The system of synaptic weights was obtained as a result of training a 3-layer SNN with the *‘STDP + all-LTD’* learning rule. Only the first part of the 3-layer SNN research was carried out. For the present this approach has allowed us to obtain only a quality understanding of the synaptic weights system operation and, based on the principles of systems theory, to develop the general scheme. Further research will focus on practical aspects of network design (improving SNN accuracy).

The author would like to thank Dr. S. Sukhov for careful reading of this manuscript and for constructive comments.

References

1. Bertalanffy L.V. General System Theory — A Critical Review // General Systems. 1962. V. VII. P. 1–20.
2. Shannon C.E. A Mathematical Theory of Communication // Bell System Technical Journal. 1948. V. 27, N 3. P. 379–423.
3. Kolmogorov A.N. Information theory and algorithm theory. Moscow: Nauka. 1987.
4. Werbos P.J. Applications of advances in nonlinear sensitivity analysis // Drenick R.F., Kozin F. (eds) System Modeling and Optimization. Lecture Notes in Control and Information Sciences. Springer. Berlin. Heidelberg. 1982. V. 38. P. 762–770.
5. Hebb D.O. Organization of behavior. New York : Wiley and Sons. 1949.
6. Antonov D.I., Sukhov S. Spiking neural networks-classifiers training with local rules // Proceedings of XXV International Conference on Artificial Neural Networks Neuroinformatics-2023. M.: National Research Nuclear University MEPhI. 2023. P. 116–125.
7. Diehl P.U., Cook M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity // Front. Comput. Neurosci. 2015. V. 9, N 99.
8. https://www.kaggle.com/code/dlarionov/mnist-spiking-neural-network/notebook
9. Antonov D.I., Sviatov K.V., Sukhov S. Continuous learning of spiking networks trained with local rules // Neural Networks. 2022. V. 155. P. 512–522.