БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ . BULGARIAN ACADEMY OF SCIENCES

КИБЕРНЕТИКА И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ • Том **2**, № 2
CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **2**, No 2

София . 2002 . Sofia

Entropy Based Delta Rule for Supervised Training of Temporal Sequence Sensitive Neuron

Stefan Sarnev

Institute of Information Technologies, 1113 Sofia

Abstract: *A robust method for recognition of specific temporal correlations in the input pattern is presented. The method is based on a modified model of spiking neuron. A supervised single neuron training algorithm is proposed. The training rule could be used with both types of input patterns – rate-coded spatial pattern and temporal coded pattern. A Cascade-Correlation architecture enables complex temporal sequence recognition.*

Keywords: *spiking neuron model, entropy based Delta rule, supervised neuron training algorithm.*

1. Introduction

Spiking neuron is a common class of biologically inspired neuron models. The neural networks consisting of this type of neurons are known as third generation of neural networks. It is shown in [6] that the computational power of a single neuron could be reached at least with several traditional neurons organized in a neural network. We use the concept of a traditional neuron to refer to McCulloch-Pitts threshold gate neuron and its successor sigmoidal gate neuron. The traditional neuron is a powerful model for spatial coded data analysis. It corresponds to simplified spiking neuron using only rate coded inputs, but is irrelevant when the data is temporally coded. The ability to process temporal coded data is essential for spiking neuron model. From biological point of view, it makes the neuron see and analyze events in time. W. Maass showed in [5] that a sigmoidal gate neuron could be simulated successfully with temporal coded inputs by specific spiking neuron model. In Chapter 3 we define another modification of a spiking neuron model which could be used for analysis of specific temporal correlations described in Chapter 5. At the same time a neuron of this model could be used for rate based spatial analysis of the power of sigmoidal gate neurons (Chapter 4). We will see that there is no sharp boundary between rate based spatial and temporal data coding. In Chapter 6 we present a modification of delta-rule based learning for single neuron training. For the purpose of the temporal pattern recognition, some important features are discussed in Chapter 7.

2. Spiking neuron model

Let i is a neuron of Spiking Neuron Model that receives input in the form of spike trains from a set of neurons Γ_i. We call Γ_i a set of presynaptic neurons to the neuron i. Every synaptic conjunction that connects the input of neuron i (dendrite) with the output of presynaptic neuron (axon) has two parameters assigned:

– weight w_{ji} corresponding to synaptic efficiency,

– time delay coefficient Δ_{ij} to represent the synaptic delay,

where i is a current neuron and j is a presynaptic neuron to i.

Definition 2.1. If the value of synaptic weight w_{ij} is positive, we call this synapse *excitatory.*

Definition 2.2. If the value of synaptic weight w_{ij} is negative, we call this synapse *inhibitory.*

When a single fire passes through synapse, a postsynaptic potential (PSP) is generated. The PSP could be excitatory postsynaptic potential (EPSP) or inhibitory postsynaptic potential (IPSP) depending on the sign of synaptic weight (Fig. 2.1). The shape of PSP is very significant for the computational power of a neuron [1]. The neuron model presented further gives a different view over the significance of PSP shape.

Let $u_i(t)$ be a current neuron membrane potential. The set of all firing times of k-th neuron is denoted by F_k. The sum of all postsynaptic potentials forming the membrane potential is

$$u_i(t) = \sum_{t_i^{(f)} \in F_i} \eta_i(t - t_i^{(f)}) + \sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in F_j} w_{ij} \varepsilon_{ij}(t - t_j^{(f)}), \tag{2.1}$$

where $t_j^{(f)}$ represents firing time of presynaptic neuron j.

An additional $\eta_i(s)$ in the sum represents refractory period that limits the higher possible firing rate. When s is close enough to zero, $\eta_i(s)$ assumes strongly negative value. In any other case $\eta_i(s)$ is near to zero and could be ignored. As soon as membrane potential reaches a certain threshold, the neuron fires.

The postsynaptic potential $\varepsilon_{ij}(s)$ is represented as function of time interval s since the moment $t_j^{(f)}$ when a presynaptic spike is received:

$$\varepsilon_{ij}(s) = \frac{1}{1 - (\tau_s / \tau_m)} [\exp(-\frac{s - \Delta_{ij}}{\tau_m}) - \exp(-\frac{s - \Delta_{ij}}{\tau_s})] H(s - \Delta_{ij}), \tag{2.2}$$

where τ_s and τ_m are time constants and $0 < \tau_s < \tau_m$;

$$H_s = \begin{cases} 0, s < 0, \\ 1, s \geq 0. \end{cases}$$

For more detailed description see [2, 3].

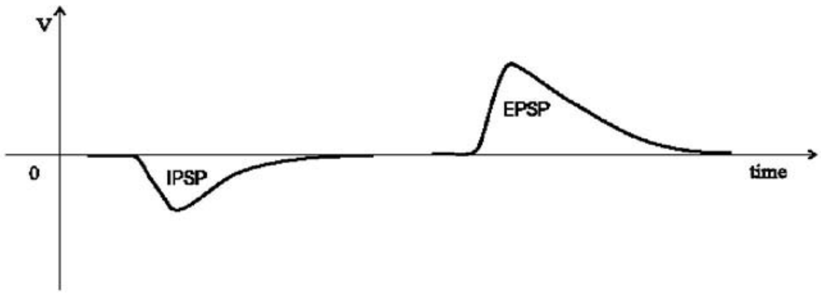


Fig. 2.1. Typical shapes of response functions of a biological neuron

3. The Temporal Sequence Sensitive Model (TSSM) neuron

The neuron model used further in this paper is a modification of Spiking Neuron Model. The goal is to achieve an easy to train neuron model, which is capable to recognize some specific temporal correlations in the input pattern and at the same time can be trained to recognize rate-based spatial pattern simulating sigmoidal gate neuron. Of course, this simplification is not mandatory, but it will help the basic idea of the representation and for problem understanding. The form of a single excitatory postsynaptic potential (EPSP) is shown in Fig. 3.1.

As we saw in Chapter 2, every synaptic conjunction has two properties – synaptic weight w_{ij} and associated time-delay Δ_{ij}. We assume $\Delta_{ij} = \Delta t^{(s)}$, where $\Delta t^{(s)}$ is some constant minimal, but not negative value. This way we keep as adjustable neuron properties through the training time only neuron weights w_{ji} and its threshold. The threshold accepts positive values. We will also simplify the response function $E_{ji}(s)$ defined in (2.2):

$$\varepsilon_{ij}(s) = \exp(-(s - \frac{\Delta t^{(s)}}{\tau_m}))H(s - \Delta t^{(s)})\,. \tag{3.1}$$

We can skip $\Delta t^{(s)}$ from $\varepsilon_{ij}(s)$ equation and add it to $t_i^{(t)}$ if neuron fires.
Now we have

$$\varepsilon_{ij}(s) = \exp(-\frac{s}{\tau_m})H(s)\,.$$

We see that derivative $\varepsilon'_{ij}(s)$ is strong negative for $S > 0$.
It is essential for the purpose of temporal sequence recognition that

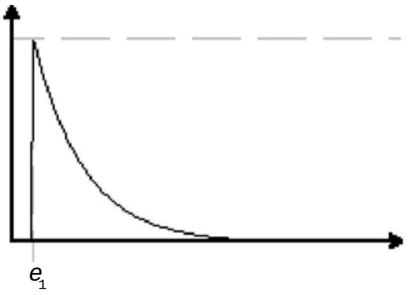


Fig. 3.1. The form of a single EPSP

$$-\varepsilon'_{ij}(s_1) > -\varepsilon'_{ij}(s_2),\ \ 0 < s_1 < s_2.$$

For the purpose of training we present a N-dimensional vector $\overline{V}_i$, called t-Buffer, where N is the number of all neurons in Γ_i – presynaptic to neuron i. Every element V_{ij} of $\overline{V}_i$ keeps the time track to the last firing of presynaptic neuron V_j, where

$$\overline{V}_i : \{V_{ij} = t - t_j^{(f)} \mid t_j^{(f)} \in F_j, \forall p_j^{(f)} \in F_j \Rightarrow t_j^{(f)} \geq p_j^{(f)}, j \in \Gamma_i\}.$$

Parameter t corresponds to the time of current training cycle.

There are two alternative definitions of our t-Buffer.

Definition 3.1. The following is similar to the already defined:

$$\overline{V}_i : \{V_{ij} = \exp(-t + t_j^{(f)}) \mid t_j^{(f)} \in F_j, \forall p_j^{(f)} \in F_j \Rightarrow t_j^{(f)} \geq p_j^{(f)}, j \in \Gamma_i\}.$$

Definition 3.2. This definition is a generalization, which allows better results in some cases depending on the definition of training process.

$$\overline{V}_i : \{V_{ij} = \sum \exp(-t + t_j^{(f)}) \mid t_j^{(f)} \in F_j, j \in \Gamma_i\}.$$

Definition 3.3. We call every element V_{ij} of $\overline{V}_i$ tiredness of synaptic conjunction j that connects presynaptic neuron j with neuron i. In the following chapters we will refer to the t-Buffer with the meaning here defined. An additional assumption is that at single synaptic conjunction rate-code will be based on higher rates than temporal-code. See examples of rate-code and temporal-code in Fig. 3.2.

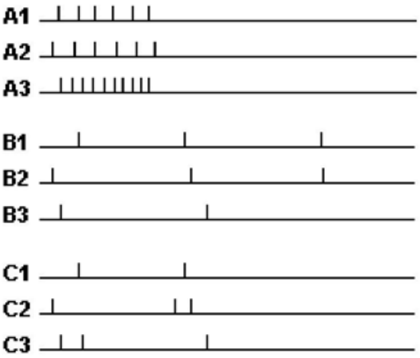


Fig. 3.2

Fig. 3.2 shows examples of output spikes of three groups of neurons A ,B and C. Group A demonstrates rate based coding. Group B demonstrates pure temporal-code. Group C uses both types of coding simultaneously.

Let P be a spatial or temporal pattern.

Definition 3.4. We call a pattern positive recognized by the neuron if in a small interval Δt_p triggered from pattern appearance to the input of neuron, at least one output spike is generated.

Definition 3.5. We call a pattern P negative recognized by the neuron if in a sufficiently enough interval Δt_n triggered from pattern appearance to the input of the neuron, there is no output spike generated.

Further we will accept that $\Delta t_p = \Delta t_n$.

4. Rate-based coding of spatial pattern

There are two common ways to simulate sigmoidal gate neuron with spiking neuron dependent on the way potential amplitude is coded – rate based coding and temporal based coding. Both coding types have their advantages and faults. There is evidence that the simple rate based coding of analog value is not adequate when complex computations should be done in a short time interval over 150 ms when typically firing rates are below 100 Hz. Please, refer to [2, 4, 6].

There are three definitions for rate based coding. These definitions correspond to different procedures for estimation of the average rate – average over a time interval, average over several repetitions and average over population of neurons. Fast computations could be done based on space-rate coding with population of neurons [3].

We have not presented here a novel method for rate-based coding, but have just tried to show the ability of our model neuron to analyze spatial rate-coded pattern and to be trained to do this. Furthermore we will see that there is no sharp transition between rate-code and temporal-code. For temporal based sigmoidal neuron simulation see [5].

Suppose that a presynaptic neuron j generates regular spikes with time interval Δt_j.

Let the spikes generated by presynaptic neuron j be the only ones that excite the neuron membrane.

According to equation (2.1) the potential maximum $u_i(t)$ will be as high as the spike rate is. In this way $u_i(s)$ represents spikes count. It could be shown that $u_i(s)$ (disregarding $\eta_i(s)$, represents IIR (recursive) low-pass filter. This fact is the key connection of current neuron model to leaky integrate-and-fire neuron model [2]. Potential fluctuation between its minimal and maximal value for a given interval of time will be the lower the higher the spike rate in the same interval is. For small synaptic weights the potential fluctuations are small too. This is due to the exponential decay nature of the response function ε_{ij}. The following dependence could be examined – high fluctuations in membrane potential correspond to temporal coding, small fluctuations correspond to rate-based coding. It could be important that the neuron can generate the same type of output as the input it accepts. In this case we talk about rate-based output. If the neuron has constant excitation in the form of regular spike sequence, its potential sum u should grow to a value s for a time interval Δt. For a given Δt_1 we can find such threshold p, that the sum u grows to value p for time interval Δt_1. A good precondition is the summary input spike rate to be higher than the output spike rate. This will minimize output frequency fluctuations.

5. Temporal sequence recognition

The temporal sequence recognition is very important for wide range of tasks. Here we will examine several types of temporal correlations, which could be successfully recognized by our model neuron.

Theorem 5.1. A single neuron of our model could recognize the following temporal correlations:

a) temporal closeness of two or more presynaptic potentials.

See “element distinctness function” EDn in [5, 6].

b) temporal order of appearance of two presynaptic potentials.

Proof of a):

Let us consider what properties a certain TSSM neuron i must have in order to recognize if two intput spikes are accepted in a sufficiently short time interval.

We have to find such weights w_{ij} that if both firings are close enough, the accumulated membrane potential represented by $u_i(t)$ has to cross an apriori fixed threshold θ_i. Notice that the value of θ_i is actually determined through the neuron training process, but for simplicity we consider it fixed.

We will use (2.1) next, ignoring $\eta_i(s)$ in the sum.

In the definition of TSSM (see Chapter 3) we examine post-synaptic potential $\varepsilon_{ij}(s)$ as a function of the time interval $t - t_j^{(f)}$.

In the proof we will base the reasoning only on EPSP, but it can be generalized in the case of IPSP too.

According to the definition of a response function $\varepsilon_{ij}(s)$ (3.1), it represents unlinear, but monotone potential decay for $s > 0$. The decay rate is as high as the potential amplitude is. This characteristic of the response function $\varepsilon_{ij}(s)$ is not mandatory in case of firing closeness recognition task.

Due to the symmetry of input spikes sequence requrement, we accept that the weights of both synaptic inputs are equal (i.e. close enough not to consider this difference significant). So we search temporal closeness only and any sequence of presynaptic potentials appearance is satisfactory.

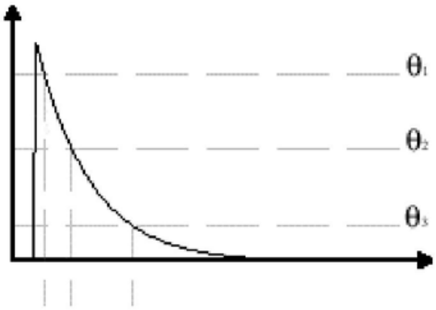


Fig. 5.1. It could be seen that the higher threshold we take, the shorter period of time potential value stays above it

Let us have a fixed threshold θ and the corresponding synaptic potential stays above $\theta/2$ for a time period Δt. Then Δt is the maximal period of time when both spikes should appear on the neuron input, so that the sum of corresponding EPSPs reaches θ. (Fig. 5.1).

Via setting the threshold height we may tune the coefficient of simultaneousity, which characterizes the input temporal pattern so that it can be recognized by the neuron. We must note that because of the definition of our training rule (see 6.1.2), which we will further use for neuron training, we do not need to normalize the weight vector w_i and threshold θ. This allows us to choose large enough weights in order to easily satisfy the above requirements.

Proof of b):

Let us denote by e_1e_2 the temporal sequence of postsynaptic potential of the input e_1, which precedes the one of input e_2 and both potentials are in a close enough time interval Δt. We want to recognize e_1e_2 and not to recognize the "wrong" sequence e_2e_1 at the same time.

We apply the similar approach as in the part a, but in this case we will use different suitable weights for the corresponding inputs..

Let us choose such weights w_1, w_2 and activation threshold p, so that the following requirements hold:

1. The potential of the input e_1 stays above the threshold p_1 for time interval $\Delta t_1 = \Delta t$.
2. The potential of the input e_2 stays above the threshold p_1 for some minimal time interval Δt_2, $\Delta t_2 < \Delta t_1$.
3. $w_1 < w_2$.

The idea is to juxtapose to the first input e_1 a slowly decreasing potential, which can stay above the threshold p_1 for time interval Δt_1. It will assure the base potential level.

To the second input e_2 we juxtapose a quickly decreasing potential, which exceeds the threshold p_2 only for a very short time Δt_2.

So, if we choose threshold $p = p_1 + p_1$, then for a short time interval near Δt_2 we will have the sum of potentials exceeding the threshold p in case the input of e_2 arrives in time interval Δt_1 after the first one.

What will happen if the input spikes arrive in a reverse order, i.e. e_1e_2 sequence? We have two cases:

1. Both spikes arrive in a time interval Δt_2(minimal chosen). We denote Δt_2 as interval of simultaneousity. The sequence e_2e_1 will be positive recognized.

2. Let us take time interval $\Delta t_3 \geq \Delta t_2$. The potential on the input e_2 decays below p_2 before the arrival of a spike on the input e_1, but still it is possible the sum of maximum of postsynaptic potential on e_1 with the postsynaptic potential on e_2 to exceed p. The interval Δt_3 is as smaller as faster is the potential decay of synaptic input e_2 compared to this of e_1.

If not both potentials are met in Δt_3, the sequence e_2e_1 is negative recognized. The interval Δt_3 is an extension of the interval of simultaneousity definition.

We saw that the neuron is sensible to the temporal sequences in a certain interval Δt_1 or to a simultaneousity in $\Delta t_3 \geq \Delta t_2$. Finding suitable weights w_1, w_2 and the threshold p we may attain arbitrary minimality of the interval. During the training process these weights as well as the threshold can be found via appropriate training examples.

Two types of synaptic potential decay are demonstrated in both firing sequences (Fig. 5.2). In case of exponential decay, a different maximal potential sum is reached depending on the input sequence – a) and b). In case of a linear synaptic potential decay the neuron has poor sensitivity to the firing sequence – c) and d).

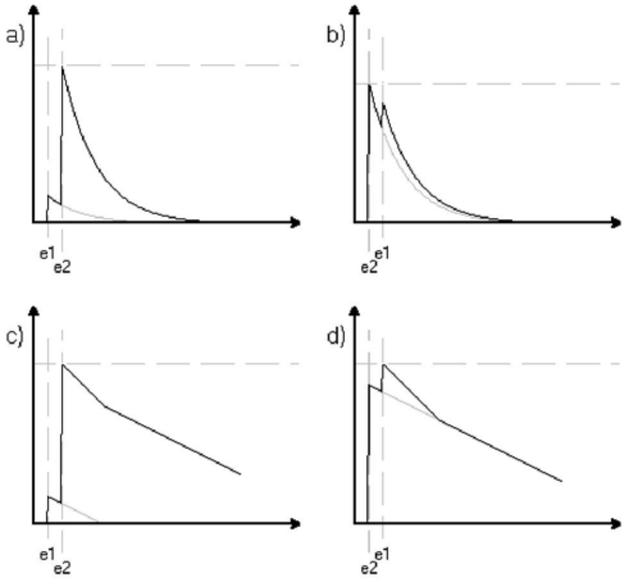


Fig. 5.2

6. Entropy based Delta rule for supervised training

6.1. Modified Delta rule for spiking neuron training

The Entropy Based Delta Rule (EDR) for supervised training originates from the Delta rule, which is also known as the least mean squared error rule (LMS). See Adaline and the Adaptive Linear Combiner in [7].

Since our goal is to train a single spiking neuron, LMS rule is near to what we need. We have adapted this algorithm to the specifics of our task. For the purpose of supervised training, a set of training patterns and corresponding desired outputs are necessary.

6.1.1. LMS rule

Let $\{x_1, ..., x_L\}$ represents a set of training input vectors. For each x_k we have the desired output value d_k.

For each step of training process, the following is performed:

1. An input vector x_k is applied to the Adaline inputs. Output y is computed.
2. The value of the error is determined, using the current value of the weight vector.
3. Then we have $\Delta w_i = 2\mu x_k(d_k - y)$.
4. Update the weight vector

$$w_i^{(\text{new})} = w_i^{(\text{current})} + \Delta w_i.$$

5. Steps 1 through 4 are repeated with the next input vector, until the error reaches an acceptable value.

6.1.2. The EDR definition

The output of our model neuron is binary, i.e. the output is {0, 1}. That is why we have three possibilities for the error e_k determined in step 3 of the above process, i.e. $e_k \in \{0, -1, 1\}$:

- $e_k = 0$ – we do not need to perform further weight correction;
- $e_k = -1$ – this corresponds to iteration of negative training;
- $e_k = 1$ – this corresponds to iteration of positive training.

A positive training with EDR of neuron i is called the increasing of the values of synaptic weights, which is calculated as follows:

$$\begin{aligned} \Delta w_i &= \mu \overline{V}_i^{(R)} \\ w^{(\text{new})}{}_i &= w_i + \Delta w_i, \end{aligned} \tag{6.1}$$

where $\overline{V}_i$ is the vector of synaptic tiredness of i defined in Chapter 3, which is time restricted to the moment of neuron firing time (if neuron fires) or to the moment training stimulus, μ is the learning rate, which we will discuss further.

In parallel we decrease the value of the threshold θ, which in our case is equivalent to the negative value of neuron bias.

$$\Delta\theta = -\mu\gamma,$$
$$\theta^{(\text{new})} = \theta + \Delta\theta,$$

where γ is an appropriately chosen constant.

When a negative training iteration is performed, we decrease the synaptic weight values using

$$\begin{aligned} \Delta w_i &= -\mu \overline{V}_i^{(R)}, \\ w^{(\text{new})}{}_i &= w_i + \Delta w_i. \end{aligned} \tag{6.2}$$

We decrease the value of threshold θ as

$$\Delta\theta = \mu\gamma,$$
$$\theta^{(\text{new})} = \theta + \Delta\theta.$$

After the synaptic weights are updated performing positive or negative training cycle we need to initialize vector $\overline{V}$ to $\overline{0}$.

We perform positive training in case of positive stimulus and neuron doesn't fire in a short interval of time Δt after the pattern is presented to its input, i.e. the input pattern is negative recognized by the neuron.

We perform negative training when the input pattern is positive recognized by the neuron, but negative stimulus is applied.

The stimulus represents the desired output and is applied by the training system immediately after the Δt interval ends.

Note that Δt interval comes from the nature of the training couple – input pattern and desired output. It is believed that the stimulus applied is a result of some event in the world of training, where the world state is visible to the neuron by the input pattern presented.

In case of rate-coded input pattern, the time interval Δt could be dependent on the desired output rate.

The coefficient μ is within the range of (0, 1). We know the value of μ could be tuned in a way to increase the learning ability of the neuron. As long as μ is close to 1, fast changes are made to the weight vector and the opposite if μ is close to 0. We need fast weight changes when the learning process is monotone positive or negative. We need fine weight tuning in case of rapid changes of learning direction.

We can count the learning direction changes for a limited interval of time in the current training cycle.

Let $f^{(m)}$ be the mean learning direction changes rate, i.e. $f^{(m)} \in (0, 1)$. Then we define μ as follows:

$$\mu = \mu_0 + (\gamma - \mu_0)(1 - f^{(m)}),$$

where μ_0 is the initial learning rate, $-\ 0 < \mu_0 < \gamma < 1$.

6.2. Oriented training towards temporal recognition

Let us suppose that two presynaptic potentials are received by the neuron always in a certain sequance. Then the weight of the late activated synaps grows faster than the weights of the other according to (6.1). This is very important with respect to temporal sequence recognition and to the corresponding training.

Let us assume now that the input potentials are met in both possible sequences with equal probability. Then they will obtain approximately the same weights for sufficiently large number of training iterations. In this way we can train the neuron to recognize temporal closeness.

Based on the EDR we can make unsupervised training. Now we have to train the neuron to foresee the appearance of a specific event, which is a result of certain factors. The factors should be visible to the neuron as temporal sequence input, which precedes the event appearance. It may be seen as well that in temporal recognition of sequences the neuron reacts simultaneously or almost so with the arrival of the last potential needed for the activation. So it is not delayed till the time when “usually” the positive training stimulus is passed. This forestallment would be very useful for a biological organism, which has to react before dangerous events appears. Such events previously give it a training stimulus only.

6.3. Training with respect to both types of patterns in parallel

We examine two types of patterns – rate based spacial and temporal based code.

As mentioned in Chapter 4, every intensity is represented by a relevant frequency and higher intensity corresponds to higher frequency.

Statement:

On absence of correlation (temporal or rate based spatial correlation) between the input vector and the desired output and sufficiently large set of training examples, the weights of synaptic connections are near to zero.

Note:

To make weights inclining to zero, we use the additional rule based on the coefficient of surprise.

The idea for this statement is the fact that if there is no correlation and the set of training examples (positive and negative) are regularly dispersed during the training process then the neuron returns a wrong result with probability of 50%. Thus for a sufficiently long period it can be considered that the number of iterations for positive and negative training is approximately equal.

Let us assume that the values of synaptic weights are shifted to the positive direction. Then the neuron will return positive results more often than negative. Due to the condition for absence of correlation in the training set of examples, the error rate in recognition of a positive example as a negative one, will be higher than the error rate in recognition of a negative example as positive. Negative training iterations will be more than the positive training ones. This leads again to weights getting close to zero.

Similar dependence is observed in the case of strong negative synaptic weights.

The coefficient of correlation in the training set of examples may be reviewed evaluationg the average rate of the neuron error. We should take into consideration the rate f of appearance of a positive error after a negative one and its reverse.

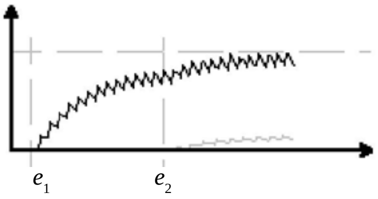
As mentioned in training rule definition (see Chapter 6.1.2.) we use this value to reduce the correction coefficient μ. This helps to stabilize synaptic weight values near to zero.

This feature allows smooth switching between rate based pattern analysis and temporal based pattern analysis depending on the case where the correlation is. Furthermore they may be simultaneously presented.

Let us assume that we consider pure rate based coding. In the cell body we have to convert every frequency into potential intensity scaled with the corresponding synapse weight.

Remember that we have defined an integrator in the neuron body as a lowpass IIR filter. If we use small values for the synaptic weights, then the sum will change relevantly smooth, which will give a more stable result on the integrator output. If synaptic weights grow rapidly then the result is a serrated curve. With a certain probability some of the maximums can cross the threshold boundary and an output spike will be generated. It is clear that this case is getting close to the temporal coding nature. If the generated output spike is considered as wrong answer, then the weights will be decreased, which brings us back to the rate based coding. In the other case it is possible a temporal correlation to be presented too.

Fig. 6.1. A sum of two rate-coded intensities. The potential is stabilizing, but some maximums could cross the threshold



7. Characteristics

7.1. Features that helps neuron training. Temporal similarity

Let us examine two cases of training with the same pure temporal coded pattern, but positive stimulus applied after different time delays Δt_1 and Δt_2. Let denote by t_1 and t_2 the corresponding times of stimulus arrival.

We want to compare the weigth corrections applied through both training iterations. Note, we assume that a single fire to every synaps is received, what we mean pure temporal code.

According to definitions 3.2, it is simplified to

$$\bar{V}_i : \{V_{ij} = \exp(-t + t_j^{(f)}) \mid t_j^{(f)} \in F_j, j \in \Gamma_i\} .$$

Using (6.1), for the case of Δt_1 we have

$$\Delta w_{ij}^{(1)} = \frac{\mu \exp(t_1)}{\exp(t_j^{(f)})}$$

and in case of Δt_2 we have

$$\Delta w_{ij}^{(2)} = \frac{\mu \exp(t_2)}{\exp(t_j^{(f)})} .$$

Now for every synaptic input k, which receives spikes, we can calculate the following ratio:

$$\frac{\Delta w_{ik}^{(1)}}{\Delta w_{ik}^{(2)}} = \frac{\exp(t_1)}{\exp(t_2)} = C_{\text{const}} .$$

This shows that the neuron “sees” a certain pattern no matter of the time shifting. We have $C_{\text{const}} = 1$ in case of using normalized $\bar{V}_i$, which leads to identical training for both iterations. So we have an advantage vs. the Time-Delay Neural Network (TDNN). In TDNN the time shifting is simulated with change of the set of activated inputs. The TDNN architecture is often used in sequence input processing tasks as speech processing [8, 9].

7.2. Multi layer achitecture training

The EDR algorithm is not suitable enough to be generalized with respect to multi-layer training.

Although it can be used quite successfully with constructive algorithms based on single neuron traning at time. For example the Cascade-Correlation Neural Network

(CCNN) architecture [10] could be applied, which in many cases outperforms backpropagation based decision. Complex temporal correlations could be detected using CCNN.

References

1. Maass, W., B. Ruf. On the relevance of the shape of postsynaptic potentials for the computational power of networks of spiking neurons. – In: Proc. of the International Conference on Artificial Neural Networks ICANN, Paris, 1995, EC2&Cie, 515-520.
2. Gerstner, W. Basic Concepts and Models.Spiking Neurons. – In: Pulsed Neural Networks (Wolfgang Maass, Christopher M. Bishop, eds). Massachusettes Institute of Technology Press, 1999, 3-54.
3. Maass, W. Paradigms for computing with spiking neuron. – Models of Neural Networks, vol. 4. Berlin, Springer, 1999.
4. Maass, W. On the relevance of time in neural computation and learning. – In: Proc. of the 8th International Conference on Algorithmic Learning Theory in Sendai, Japan (M. Li and A. Maruoka, eds.). Lecture Notes in Computer Science, Vol. 1316. Springer, Berlin, 1997, 364-384.
5. Maass, W. Fast sigmoidal networks via spiking neurons. – Neural Computation, **9**, 1997, 279-304.
6. Maass, W. Networks of spiking neurons: the third generation of neural network models. – Neural Networks, **10**, August 4, 1997, 1659-1671.
7. Freeman, J. A., D. M. Skapura. Neural Networks. Algorithms, Applications and Programming Techniques. Addison-Wesley Pub Co, June 1991, 55-64.
8. Sada Siva Sarma, A., H. W. Strube, S. S. Agrawal. Hindi Phoneme Recognition Using Time Delay Neural Network, Fortschritte der Akustik. – DAGA 97, CSIR Complex, New Delhi, India, Oldenburg, DEGA, 1997, 556-557.
9. Karaali, O., G. Corrigan, I. Gerson, eds. Speech Synthesis with Neural Networks. – In: World Congress on Neural Networks, San Diego, September 1996, 45-50.
10. Scott, E., Fahlman and Christian Lebiere. The Cascade-Correlation Learning Architecture. School of Computer Science, Carnegie Mellon University, Pittsburg, PA 15213, Aug. 29, 1991.

Ентропийно Делта правило за обучение с учител на чувствителен към времеви последователности неврон

Стефан Сърнев

Институт по информационни технологии, 1113 София

(Резюме)

Представен е мощен метод за разпознаване на специфични времеви корелации във входните образци. Методът е основан на модифициран модел на spiking неврон. Предложен е алгоритъм за обучение с учител на единичен неврон. Обучаващото правило може да бъде използвано при двата вида входни образци – пространствени честотно кодирани и времево кодирани. Каскадно-корелационната архитектура позволява разпознаване на сложни времеви последователности.