ISSN 1407-7493 COMPUTER SCIENCE DATORZINĀTNE 2008-7493

INFORMATION TECHNOLOGY AND MANAGEMENT SCIENCE

INFORMĀCIJAS TEHNOLOĢIJA UN VADĪBAS ZINĀTNE

INVESTIGATION OF REFRACTORINESS PRINCIPLE IN RECURRENT NEURAL NETWORKS

Sergey Jakovlev, Mg.sc.comp., Institute of Information Technology, Riga Technical University, 1 Meza Street, Riga LV-1048, Latvia, e-mail: sergeyk@fis.lv, tac@inbox.lv

Keywords: perceptron, recurrent neural networks, refractory period

1. Introduction

The kind of an artificial neural network (ANN) presented in this work - perceptron RP-2 is the development of ideas about recurrent perceptron stated earlier in [1]. The basic achievement of the previous work was the fact that it became possible to use the error-correction algorithm as in Rosenblatt's perceptron, but for recurrent networks. It, consequently, allowed refusing *the error back-propagation algorithm* [2, 3] which collides with serious high-speed restrictions, therefore it cannot be used in problems with a big dimension.

In the present article *modelling the principle of relative refractoriness* is considered. In electrophysiology the relative refractory period is a time interval during which the raised neurons gradually restore the ability to form actions' potential (stimulus). During the relative refractory period, a stimulus stronger than the one which has caused the first activation of a neuron, can lead to formation of a repeated activation of a neuron. The qualitative model of refractoriness principle has earlier been discussed in the description of ideas about synergistically active neural models [4]. Hereafter, the principle of relative refractoriness described in the above-mentioned work is considered and it is shown that it cannot be used in isolation from the architecture of ANN. As refractoriness has an important role in biological organisms, it will be shown under what conditions the modelling of this principle and at what architecture of ANN it can be favourable informatively and how it is used in *perceptron RP-2.*

2. The structure of perceptron RP-2

It makes sense to start with reminding the reader of the architecture of recurrent perceptron RP-1 as in perceptron RP-2 it does not differ schematically (Fig. 1b). Such architecture shapes can be observed if one supplies Rosenblatt's perceptron [5] with the feedback and contextual receptors of the Jordan network [6]. Thus training does not occur by the error back-propagation algorithm as in Jordan network, but by the error correction algorithm as in the Rosenblatt's perceptron. The basic feature of *perceptron RP-2* is that the principle of refractoriness is realized in it. Below some formal definitions supplementing the ones described earlier in [1] are listed.

Definition 1. *Recurrent perceptron RP-1* is a system that meets the following conditions:

1. The system consists of binary S, S_{in}, A and R - elements;

2. The system represents perceptron with consecutive connections only going from S-elements (and S_{in} – elements whose groups are divided into modalities) to A - elements and from A- elements to R-elements;
3. The system possesses feedback with an individual time lag from a R-element to S_{in}-elements by means of which R-elements can influence the S_{in}-elements during the following moment of time;
4. Weights of all connections of S-elements and S_{in} – elements to A - elements are fixed (do not change in training) and are selected once, in stochastic process with a certain function of distribution;
5. Weights of all connections from A - elements to R-elements are variable and are adjusted during training by the method of error correction;
6. The number of connections of S-elements and S_{in} – elements to A - elements is defined by number d which is considerably less than the total number of sensing elements and is chosen for each of the modalities. The total number of input connections of A - element will be equal to the sum of numbers d chosen for each of the modalities. Connections of A - elements to R – elements are defined by the strength of connection which is defined at training and in some cases can be equal to zero, i.e. be absent;
7. The time of internal processing of a signal is synchronized with a certain frequency τ, and the internal processing takes some time, which corresponds to the length of the remembered sequence of pairs of stimulus-reaction. And the internal processing of one stimulus-reaction pair is equal to one cycle of network work;
8. Training can be divided into phases (see Definition 3).

Definition 2. *Perceptron RP-2* is a system satisfying all conditions of *recurrent perceptron RP-1,* but in addition each associative element possesses a refractory period, which means that after activation of A – elements, the threshold for the period of internal processing a signal increases by the certain size. And then after each time unit of the internal processing decreases by exponential law (an example of the refractory period fading see in Fig. 1b below on the right). At the reception of a new external signal with frequency τ the thresholds of all associative elements are established to be equal to zero.

Definition 3. The *training phase* is the name we give to the process of realization of network training with one subset of training sample which is given by the experimenter for an agent's training. Division of training into phases enables the trainee to receive the information in understandable portions. At each following phase the network is already in partially trained condition.

3. An example of using perceptron RP-2

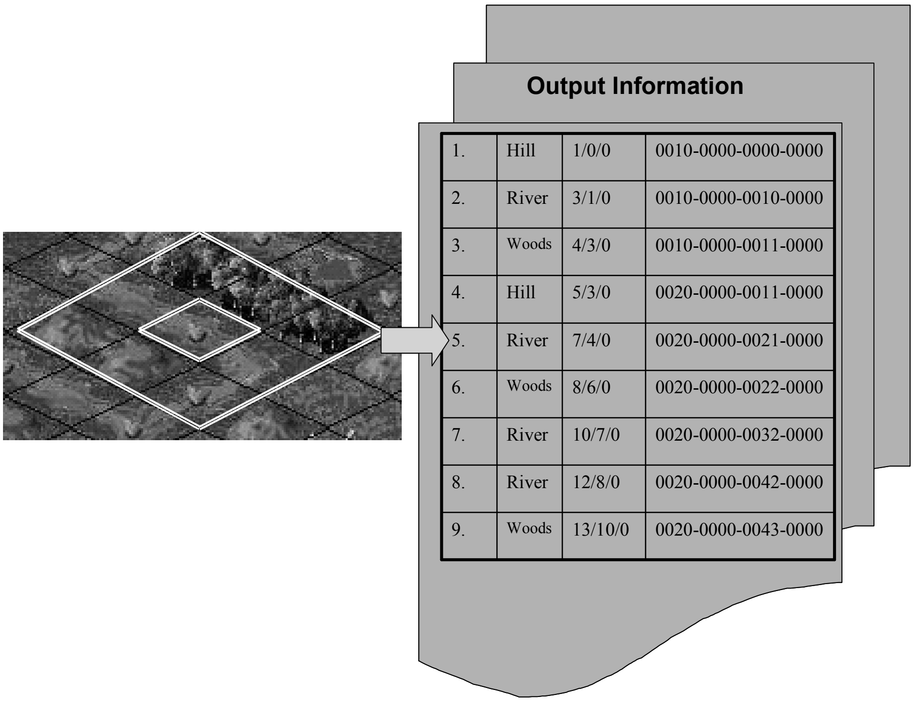
3.1. The goal of agent's training

In an artificial intellect there is a concept of a trained intellectual agent - in our case, it is the program system observing the model environment, capable of training and adaptation to changing conditions. To carry out its functions actively, the agent usually has a hierarchical structure including many "subagents". Only the work of sensor subagent (further called simply "agent") which on the basis of sensor signals creates its " picture of the world" will be considered.

The agent uses the mechanism of *perceptron RP*, which enables it to respond to the information in the model environment and remember the important parts of it. In experiments we shall compare the first version of *recurrent perceptron RP* and that of *refractory perceptron RP* described here below. The behavior of the agent on the basis of the analysis of the information received by it is not considered in this paper. Instead, attention is only paid to the process of training with reinforcement, its speed and some of its characteristics.

The model environment represents a map of a district (Fig. 1b, below on the left), divided into 276 squares (sites) of various types –meadow, plain, hill, river, woods, etc. (20 kinds in all). Each type of territory differs in the quantity of resources which can be received while cultivating this territory. There are four kinds of resources – food, metal, gold and the speed of moving an object on the territory.

In the training process the agent passes all positions on the map. During the first stage it passes through (one after another) 10 external positions on axis X and 15 positions on axis Y – 1350 positions in total (the coordinates are coloured white in Fig. 1b). And during the second stage the agent passes through (one after another) 9 internal positions on axis X and 14 positions on axis Y – 1134 positions in total (the coordinates are coloured grey in Fig. 1b).



1.	Hill	1/0/0	0010-0000-0000-0000
2.	River	3/1/0	0010-0000-0010-0000
3.	Woods	4/3/0	0010-0000-0011-0000
4.	Hill	5/3/0	0020-0000-0011-0000
5.	River	7/4/0	0020-0000-0021-0000
6.	Woods	8/6/0	0020-0000-0022-0000
7.	River	10/7/0	0020-0000-0032-0000
8.	River	12/8/0	0020-0000-0042-0000
9.	Woods	13/10/0	0020-0000-0043-0000

Fig. 1a. Example of input and output data (stimulus-reaction pairs)

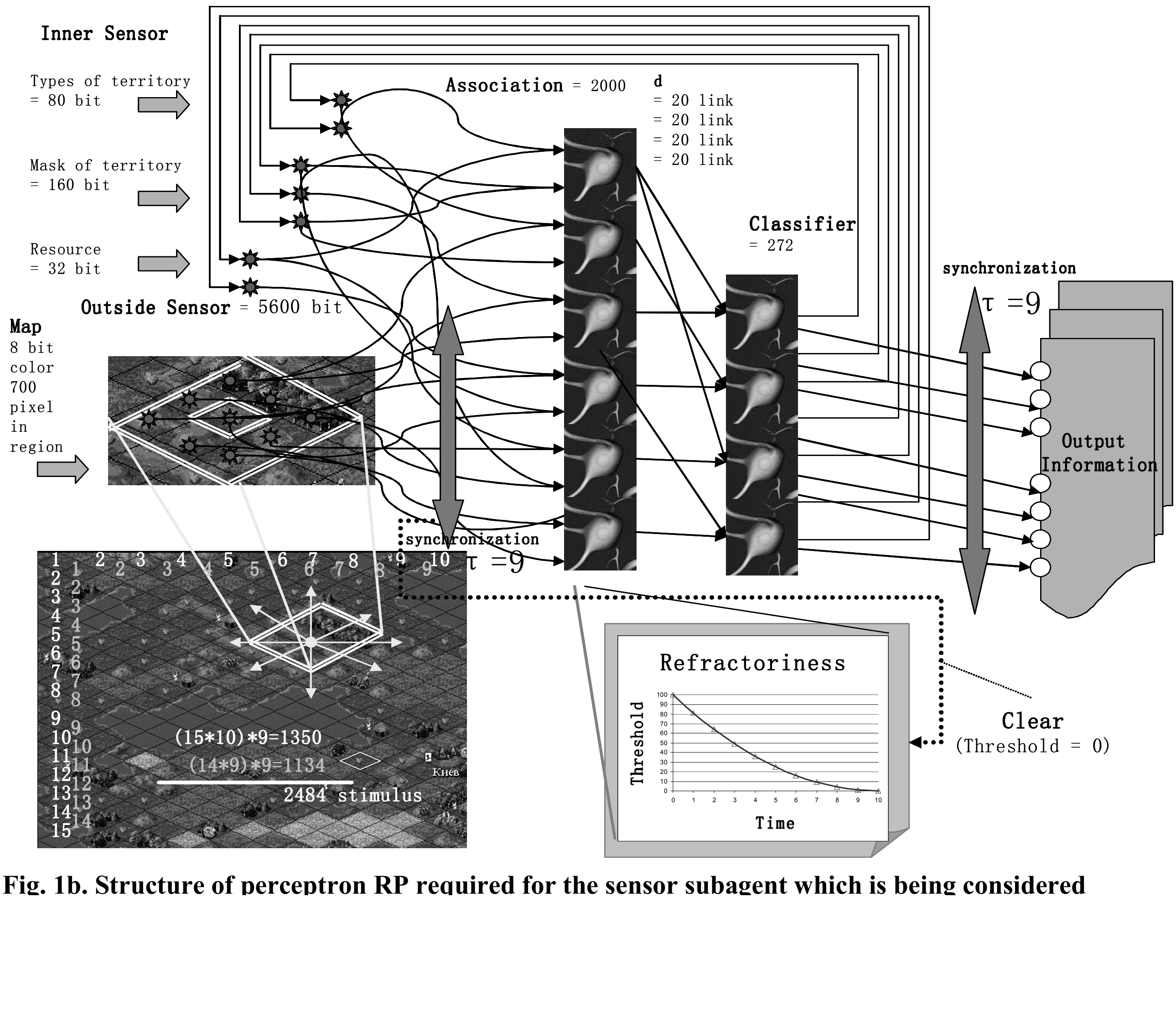


Fig. 1b. Structure of perceptron RP required for the sensor subagent which is being considered

While passing through different positions in the model environment, the agent thus forms a training sample which consists of a set of stimulus–reaction conformities (Fig. 1a). As a stimulus, a visual image of 9 sites surrounding the agent (one site has the size of 64x32 pixels with 256 colour scale) (see Fig. 1b, on the left, in the middle – an enlarged part of the map of the district) is used, and the teacher trains the agent to create corresponding semantic reactions. The information given by the teacher in training and then required at the output consists of three components:

1. **Types of the territory** surrounding the agent;
2. **Quantity of identical territories** of a certain type (the number of meadows, plains, oceans, rivers, etc.) surrounding the agent. This kind of a set forms 20 numbers – one number for each type of territory;
3. **The quantity of resources** existing in the area investigated by the agent; thus, a mask of resources is created consisting of 3 numbers (quantity of food/metal/gold), describing the total quantity of resources in the adjoining area where the agent is located.

During the internal processing it is necessary to accumulate information keeping in memory, for a simultaneous analysis, up to 9 types of territories, thus having full information on the state of the whole area surrounding the agent. An example of such input-output data is shown in Fig. 1a.

3.2. The results of modelling

The program developed is executed on one virtual processor of four-nuclear processor Xeon E5310 under OS Windows Vista x86. The system used 800-950 Mb of operative memory from 2 Gb available.

A is the number of elements which depends on the total number of stimuli necessary for storing (2484) and makes up, as a rule, 60-80 % of that number. In a test problem, training is carried out in two stages (1350+1134 stimulus) and tested on two cases with 1500 A - elements (the bottom border) and 2000 A - elements (the top border). Thus, four various experiments were carried out whose results are summed up in Fig. 2 – 5.

By virtue of networks' recurrence during its work, 9 internal conditions are formed, which is defined by the length of sequences defining the type of territories, summation of the quantity of identical types of territories and quantities of resources on them (Fig. 1a) is formed as well. And then the whole available complex of the information is given at the output. Later on such information will enable the agent to choose the direction in search of an optimum place on the map of the district (this process is not considered in the present paper).

Thus, recurrence is used as if for an opportunity of internal processing of the information – restoration of 9 similar sets of information about each site remembered before. To be more precise, for each step of internal processing there may be not simply restoration of separately collected information on a site, but a certain operation (for example, summation) taking into consideration the background of the analysis of those 9 sites. Certainly, such operations are simply specified by training, they are not formed in a network. On external sensor controls (on the retina) there is the same stimulus, and by virtue of the internal nine-cycle work a ready-made result is achieved.

Refractoriness principle is realized as follows: if A – element got activated, its threshold increases by 100 units. During the following time unit of the internal processing (step) the threshold will go down to 81 units, i.e. in accordance with square-law dependence ($9^2 = 81$). This will go on further up to 64, 49 and 36 until it reaches zero.

Comparing Figures 2 with 3 and 4 with 5 it is possible to analyze a difference: whether refractoriness principle is realized at A – element or not. And comparing Figures 2 with 4 and 3 with 5 is possible to see the difference between different quantities of A - elements. There are four curves in each figure which enable one to see how various characteristics change in relation to each other depending on the number of iterations of the training process. The number of iterations is shown on axis X, and the number has various senses - on axis Y. For the convergence curve that is the number of errors on the given iteration; for the speed of training curve that is the time spent on modelling (in seconds) before the given iteration; for the curve of training the classifiers it is the number of the classifiers trained during the iterations expired; for the curve of increase in the weight factors – it is the total strength of connections (in thousands) available for the given iteration.

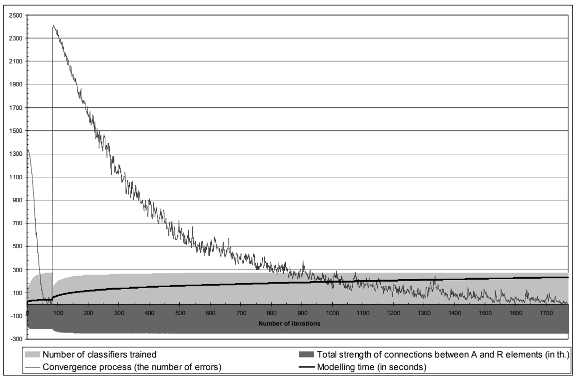


Fig. 2. Characteristics of training of perceptron RP-2 realizing refractoriness principle, at a minimally necessary (60% of the total number of stimuli) number of A – elements

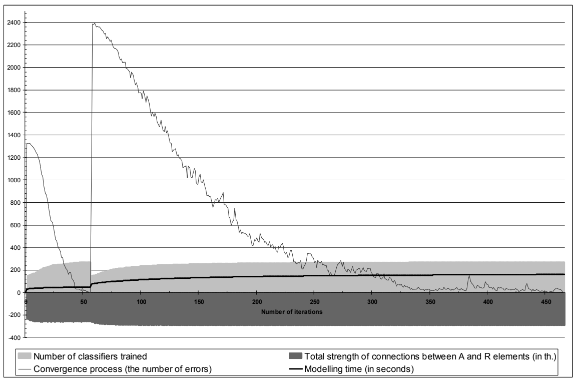


Fig. 4. Characteristics of training perceptron RP-2 realizing refractoriness principle, at a practically sufficient (80% of the total number of stimuli) number of A – elements

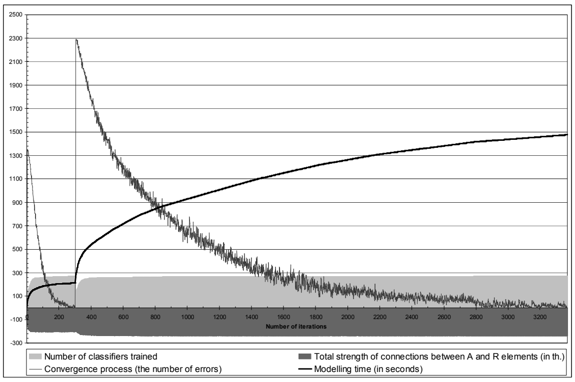


Fig. 3. Characteristics of training of recurrent perceptron RP-1 in which refractoriness principle is not realized, at a minimally necessary (60% of the total number of stimuli) number of A – elements

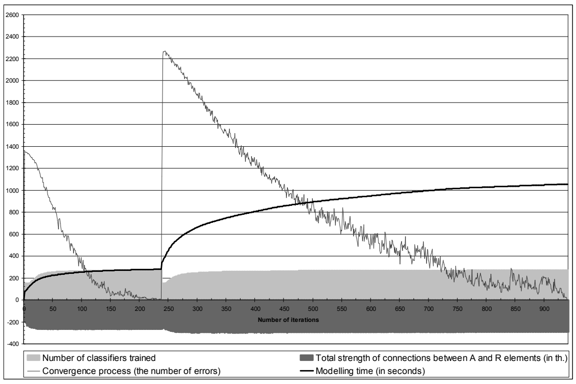


Fig. 5. Characteristics of training recurrent perceptron RP-1 in which refractoriness principle is not realized, at a practically sufficient (80% of the total number of stimuli) number of A – elements

It should be noted that the schedules clearly show the difference between the first and second grade levels (see definition 3). At the first stage half of the information is remembered. And at the beginning of the second grade level practically everything that was learnt at the first grade is forgotten. On the curve, two jumps with the subsequent convergence up to zero correspond to convergence to grade levels. But if information at each stage was not added in such big amounts, then there would be less forgetting. But on the other hand, training with a greater number of stages would require more time. Therefore it is more effective to receive the information all at once in several portions (subsets of training sample) which is similar to the agent having short-term memory which then forms long-term memory in the form of neuro-network models of the agent's "world". One can also notice that the speed of training at the first stage is always higher than at the second. That is because at the first stage there is still a lot of free memory (the number of A-elements is twice than the necessary quantity) – that is where prompt training comes from.

One of the curves on the schedules shows the number of the classifiers trained. The fact is that each classifier is trained absolutely independently (in parallel) of another. And when one of the classifiers has been trained, that is its output always corresponds to the value required at all stimuli, it can be excluded from the training procedure. Various classifiers take different time to train which is due to the complexity of dependence of reaction to stimulus. And the greater the number of classifiers trained, the less time will be spent on subsequent iteration, which can be seen on the schedule. The curve of speed of training (in seconds) depends on the number of the classifiers trained more than on the number of errors.

Also it makes sense to track the total strength of connections between A and R elements. There is a hypothesis that it should increase constantly depending on the number of iterations. That assumption is based on the fact that Minsky has shown on a separate example for a problem on a predicate parity [7] that the weight factors will increase beyond all bounds. Here one can see from a problem at which experiments were made that the total strength of connections only increases at the beginning, and then for a long time it is practically in a stable condition, having only insignificant fluctuations. Also it is worth noting that the total strength of connections has a negative sign. It means that the number of connections raised during each moment of time is small but their strength is big enough to overcome the general background of braking influence that is during the moment of receipt of the stimulus there is a sharp dot splash in activity. It is for this reason that such method of training is stable. If the total strength of connections were positive, any insignificant stimulus would activate a big number of elements and it would be difficult to differentiate between various stimuli in such conditions.

4. Conclusions

Perceptron RP-2 owes its characteristics first of all to the refractory effect which can be observed at the results of modelling. But refractoriness principle can only be applied in recurrent networks which possess a number of characteristics of perceptron RP-1. Thus, that principle can only be good in perceptron with certain architecture, but used all by itself, it cannot do any good: on the contrary, it may make the network unstable, and because of that the time of training will increase or there will be no convergence.

If we apply ideas about refractoriness principle to the artificial neural network, then irrespective of its architecture the network during training will be unstable, and it would need to be retrained constantly as simultaneously two parameters will change iteratively– thresholds and strengths of connections. And it is clear, that theoretically at best it will take more time (as at the following iteration the strength of connections will steal up at other thresholds), and practically the probability of convergence of a network will be the smallest and will worsen when the dimension of the network increases. Besides, such a network will not be indifferent towards the order perceives external stimulus, which means that such network will not be universal.

Here the researcher faces a choice: on the one hand, it is necessary to reflect the dependence on the previous conditions, and on the other hand – it should not matter for classification in what order the stimuli came. This paradox can only be removed by recurrent networks and if the stimuli are divided into external and internal ones. Besides, synchronization of processing the internal stimuli is important, and, as a consequence, reception of external stimuli with a certain frequency and generating some reaction only during certain moments of time (the 7th condition of definition of perceptron RP-1) is obligatory. That is the basic difference from architecture of the ANN by Jordan.

However, it is not enough to achieve stability at network training and it is therefore necessary to note two more nuances:

1. The first layer should not be trained, which means that connections from the first layer should not change during the training (the fourth condition of definition of perceptron RP-1). That is necessary because training – display of the nonlinear data which have acted from sensor controls on linear representation becomes unreasonably complicated, greater dimension has already occurred, and any change will lead to a situation in which it will be a totally different display (in no way better than the initial one), which will be another factor of instability at training. Besides it will take some extra time of modelling without getting the quality required.
2. Refractoriness principle can only be applied during the internal processing of the stimuli. At reception of an external stimulus, the thresholds are set in the initial condition (the condition of perceptron RP-2 definition). That must be done because external stimuli should be perceived no matter in what sequence they have come or what internal activity there is in the network now.

Why does the refractory effect influence the speed of training so much? It is not only because the A - elements alternate thanks to which the active elements have time for a phase of rest after excitation. Another reason is that such alternation allows us to reduce, firstly, the number of active A - elements and, accordingly, to correct a smaller number of weight factors. Secondly, alternation enables us to better sort A – elements, both used and unused. As a rule, after a random choice of fixed connections of the first layer, very close groups with greater crossing are activated (as the external stimulus remains unchanged during internal processing, and the differentiation only occurs due to internal stimuli). It is the refractory effect that reduces such crossing, enables us to involve even those neurons whose activity would not normally make a serious contribution, but as the activity of the strongest neurons is muffled, some other weight factors are adjusted. Any reduction in crossing, as a consequence, leads to reducing the effect of forgetting during the training. Therefore fewer iterations (and not just less time) is required in perceptron RP-1.

As a matter of fact, that is a more successful realization of the principle of neurons' competition in the middle layer. For this purpose in other architectures, as a rule, braking connections are introduced between neurons of the middle layer which reduce the excitability of neurons. Then braking leads to increasing the output of the most highly active neurons at the expense of the next ones. Such systems increase contrast, raising the level of the activity of the neurons connected to the "bright" area of the retina, at the same time weakening the outputs of neurons connected to the "dark" areas still more. But introduction of such connections requires more resources and is ineffective in the long run.

That happens because if one introduces such connections casually, it is necessary to make sure that closed contours (feedback) are not formed, otherwise it is necessary to control the stability of the system. And the main thing is that introduction of such connections in the middle layer at best doubles the total number of connections in a neural network. As a result, the effect gained is lost in calculation of neurons' activities. At the same time for modelling the refractory effect it is enough to change the threshold of A – element, without introducing excessive connections and calculations connected with them.

To sum up, we can state that the additional realization of refractoriness principle in perceptron RP-2 enables one to accelerate training of a network on average five times. At the same time no additional elements are introduced into the network, simply the available associative elements are used more effectively. Besides, realization of refractoriness principle for internal processing does not destabilize the functioning of the network. That means that this principle can only be applied to recurrent networks at which there are inwardnesses. Otherwise, the network would not be steady, and consequently, will be unsuitable for practical use.

References

1. Яковлев С., Борисов А. (2008) Исследование принципа рекуррентности в перцептроне Розенблата. http://www.tachome.times.lv/Articles/Yak_1.pdf
2. Rumelhart D.E., Hinton G.E., Williams R.J. Learning internal representations by error propagation // Parallel distributed processing, Cambridge, MIT Press, 1986.
3. Вассерман Ф. Нейрокомпьтерная техника: теория и практика, Москва, «Мир», 1992.
4. Шамис А.Л. Пути моделирования мышления, Москва, «КомКнига», 2006.
5. Проект Википедия – энциклопедия в интернете http://ru.wikipedia.org/wiki/Перцептроны_с_обратной_связью – Visit date May 2008.
6. Jordan, M. I. Serial order: A parallel distributed processing approach // Institute for Cognitive Science Report 8604. University of California, San Diego, 1986.
7. Минский М. Перцептроны, Москва, «Мир», 1971.

Jakovļevs Sergejs. Refrakteritātes principa pētīšana rekurentos neironu tīklos

Darbā ir parādīta Rozenblata perceptrona un Džordana tīkla apvienošanas iespēja, kura rezultātā perceptronam parādās atgriezeniskās saites, bet apmācība paliek tik pat ātra un efektīva ka apmācība Rozenblata perceptronā. Darbā parādīta arī refrakteritātes modelēšanas īpaša loma asociatīvos elementos, kas ļauj ātri iegaumēt iekšējā stāvokļa secību. Tas notiek pateicoties tam, ka refrakteritātes process automātiski regulē slodzi, kas gulstas uz asociatīviem elementiem, izvēlējoties grupas, kas vēl nebija izmantotas. Tas savukārt nosaka tīkla diferenciāciju uz savā starpā maz saistītiem neirona ansambļiem un tādejādi izceļ to no klasiskā perceptrona, kam trūkst vienlaicīgi aktīvo asociatīvo elementu skaitļa vadīšanas. Tādā veidā jauniegūtais rekurenta perceptrons RP-2 ir notestēts aģenta, kurš ir ievietots imitācijas vidē, apmācības uzdevumā.

Jakovlev Sergey. Investigation of refractoriness principle in recurrent neural networks

This paper describes a possibility of combining Rosenblatt's perceptron with Jordan network. As a result, the perceptron gets the ability to have feedbacks. At the same time, teaching remains as fast and effective as in Rosenblatt's perceptron. The paper also emphasises a special role of refractoriness modelling in associative elements, which enables the network to rapidly remember sequences of internal states. This occurs because the process of refractoriness automatically regulates the load the associative elements experience when choosing the groups that have not been activated yet, which, in its turn, determines network differentiation into a series of ill-connected neural ensembles and distinguishes it from the classical perceptron that does not have control over the number of simultaneously active associative elements. The recurrent perceptron RP-2 obtained in the above way is tested on the problem of teaching the agent which is placed into the simulation environment.

Яковлев Сергей. Исследование принципа рефрактерности в рекуррентных нейронных сетях

В работе показана возможность совмещения перцептрона Розенблата и сети Джордана, в результате чего перцептрон приобретает обратные связи, а обучение остается таким же быстрым и эффективным, как в перцептроне Розенблата. В работе также показана особая роль моделирования рефрактерности в ассоциативных элементах, что позволяет сети быстро запоминать последовательности внутренних состояний. Это происходит благодаря тому, что процесс рефрактерности автоматически регулирует нагрузку, которая ложится на ассоциативные элементы, выбирая группы, которые не еще были задействованы. Это, в свою очередь, и определяет дифференциацию сети на ряд малосвязанных нейронных ансамблей и отличает от классического перцептрона, у которого отсутствует управление числом одновременно активных ассоциативных элементов. Полученный таким образом перцептрон RP-2 протестирован на задаче обучения агента, помещенного в модельную среду.