The Concept and Properties of Sigma-if Neural Network

M. Huk, H. Kwasnicka
Department of Computer Science, Wroclaw University of Technology, Poland
E-mail: {maciej.huk, halina.kwasnicka}@pwr.wroc.pl

Abstract

Our recent works on artificial neural networks point to the possibility of extending the activation function of a standard artificial neuron model using the conditional signal accumulation technique, thus significantly enhancing the capabilities of neural networks. We present a new artificial neuron model, called Sigma-if, with the ability to dynamically tune the size of the decision space under consideration, resulting from a novel activation function. The paper discusses construction of the proposed neuron as well as training Sigma-if feedforward neural networks for well known sample classification problems.

1 Introduction

The basic constituent of a classic artificial neural network (ANN) is the neuron, used to process signals presented as input, through an activation function and a nonlinear output (threshold) function. The former function, hereafter referred to as *A*, determines the activation level of the neuron, while the latter (*F*) bases on the result returned by *A* to construct the neuron's output value. [1,2] The importance of the threshold function as well as of the weights attached to individual interneural connections is well discussed in numerous publications. [1,3,4,5] However almost all of them assume the activation function to be a simple weighted sum of neuron input values and connections weights. In fact, analyzing the behavior of artificial neurons with nonstandard activation functions may lead us to develop new classification structures, with interesting and useful properties. [6-13]

2 Sigma-if Neuron

The authors propose a special type of neuron, whose activation function can be clearly interpreted from a biological perspective. In the case of real neurons, individual dendrites differ in length, allowing a biological neural network to associate incoming signals with particular connections and processing areas. Such a property can be incorporated into classic feedforward neural network models through connections grouping and conditional signals accumulation technique. [11,14]

More specifically, the *M* dendrites of a Sigma-if neuron are divided into *K* distinct groups, by complementing each *i*-th input connection with an additional integer parameter $\theta_i \in \{0,1,..,K{-}1\}$, determining membership in one of the groups. This allows us to divide the process of signals accumulation into *K* steps, where *K* is a function of neuron's grouping vector $\theta^T=[\theta_1, \theta_2, ..., \theta_M]$:

$$K(\theta) = \max_{i=1}^{M}(\theta_i). \tag{1}$$

During each step *k* (from 0 to *K*-1) the neuron accumulates data belonging to one selected group, such that

$$\theta_i = k. \tag{2}$$

Within each *k*-th group, partial activation $\Delta g(k)$ is determined as a weighted sum of input signals and the appropriate Kronecker's delta:

$$\Delta g(k, w, x, \theta) = \sum_{i=1}^{M} w_i x_i \delta(k, \theta_i), \tag{3}$$

where w_i and x_i are coefficients of the neuron's weight vector *w* and input vector *x*. This process is repeated until the activation derived from respective groups exceeds a preselected activation threshold net^*. It can be described by the following recursive formula (vectors *w*, *x* and θ are omitted for clearness):

$$net(k) = \begin{cases} \Delta g(k) H(net^* - net(k-1)) + net(k-1) & : k \geq 0 \\ 0 & : k < 0 \end{cases} \tag{4}$$

where *H* is Heaviside's function. This sum is then treated as the neuronal activation value. Input from remaining (heretofore unconsidered) groups is neglected. Thus, the proposed form of activation function *A* is:

$$A(w, x, \theta) = net(K, w, x, \theta). \tag{5}$$

For completeness, it is also important to note that in the final stages of determining the output value *Y* of the neuron, function (5) serves as a parameter of the nonlinear threshold function *F*:

$$Y(w, x, \theta) = F(A(w, x, \theta)). \tag{6}$$

Neurons of the presented type can easily be used for building network structures. One can choose architectures similar to classic synchronous feedforward neural networks, but recurrent realizations are also possible. While there are no special restrictions regarding network architecture, Sigma-if neurons can work in fully-connected as well as in sparse structures.

3 Training the Sigma-if Neural Network

The Sigma-if network requires a suitable training procedure. To characterize the influence of neuron modifications on network capabilities, we have separated the problem of grouping vector coefficients selection from that of connections weights calculation.

While in our study we use fully-connected feedforward Sigma-if networks with one hidden layer, the weights vector can be determined through a slightly modified back propagation algorithm. Its modification involves adding an assertion stating that during each weight adjustment cycle, only those weights can be changed, which have recently influenced the output values of the network. The reason for such an approach is as follows: the training algorithm should limit the influence of immediately-recognizable patterns on network structures used in categorizing data which rely on connections with greater θ_i values.

The selection of coefficients of the grouping vector should also be adjusted to the problem considered by the network. Yet, looking for an optimal grouping vector is generally a very difficult task. In practice it involves computationally expensive multidimensional and multimodal optimization. Thus use of reasonable heuristics is hence justified.

Following preliminary experiments we have decided to use the random walk technique. When the random grouping vector selected at the beginning of the learning process doesn't enable the backpropagation algorithm to reduce the network classification error below an assumed target level, we randomly select all theta values again. This process is repeated for every hundred backpropagation cycles until the network is successfully trained. Experience shows that the typical number of grouping vector selection attempts required to properly train the network is on the order of 10. Considering that, and remembering that our main goal was to check the basic properties of the Sigma-if model, the proposed solution seems to be acceptable.

4 Properties of the Sigma-if network

It is worth mentioning that proper selection of connection θ_i parameters between the input and hidden layers is very important. If the selection of grouping vector coefficients is highly disadvantageous, some highly important data attributes served through connections with high θ_i values, may end up not being considered at all. This is a particularly pressing issue when the low-theta connections carry strong noise, exceeding the activation threshold net^* of the neuron. Nevertheless, it is not a hopeless situation, due to another property of the Sigma-if network. It can perform reactivation of inactive and important attributes through minimization of active connection weights. This is achieved through the back propagation algorithm by lowering the weights of dendrites which distort the classification process.

The above property suggests that diversification of theta values may have other positive consequences. It enables neurons to separate disruptive input signals from those that carry useful information, thus increasing the signal-to-noise ratio. This can act as very effective noise filter, but only when noised connections have greater theta values than inputs required for proper classification

The next consequence of neuronal inputs grouping is that simple analysis of the activity of input to hidden layer connections in a properly trained network may yield information about subsets of data attributes important for the classifier. This analysis is reduced to observing which neuronal inputs are considered when establishing the network output as a response to a particular test input patterns.

However the most important feature of the proposed network is the ability to discriminate the input space in an adaptive manner. The conditional signal accumulation technique enables it to partition the data space with hypersurfaces using an increasing number of dimensions, where the number of attempts is determined by the number of distinct theta values assigned to neuronal inputs. This can easily be observed in the case of a single Sigma-if neuron. Despite the use of a sigmoid threshold function in its body, the Sigma-if neuron - unlike the classic neuron - can solve simple linearly inseparable problems, since it is able to use (depending on the circumstances) all or just some of the information present on its inputs.

It should, however, be noted that the presented approach fails for the XOR function. In this case, each straight line perpendicular to one of the dimensions of the data space and passing through a selected point corresponding to a training pattern from class zero contains a point belonging to a class different than zero. It is therefore impossible to separate both points by means of straight lines perpendicular to selected normal vectors of data spaces (hypersurfaces reduced to one dimension). In this case, a different solution may be utilized: rotating the coordinate set by a preselected acute angle. Following such a transformation, a single Sigma-if neuron will be able to properly classify points defining the XOR function.

5 Results of Experiments

The main goal of our experimental study was to find evidence for the basic theoretical expectations concerning the presented Sigma-if neuron model. The second goal was to compare classification capabilities of Sigma-if and classic neural networks, and to show that conditional signal accumulation technique can be useful in data mining applications. At the end we wanted to check if there are any differences between classic and Sigma-if networks in the context of knowledge extraction.

The tested networks were trained using the previously described modified back propagation algorithm, with randomly selected theta values, controlled by a separate algorithm which oversaw the space of convergence of the learning process. Behind tests with single neuron, the number of neurons in the hidden layer was set to the value assuring best training results for the classic network model. All neurons used a bipolar sigmoid threshold function and the activation threshold level was set to a value of net^*=0.4.

5.1 Single Sigma-if Neuron

The basic functionality test of the Sigma-if network involved gauging the properties of a single neuron. According to theoretical analysis, this neuron is not able to properly dissect the data space of the XOR problem (over the real number space). However, even a slight repositioning on one of the points defining the data space enables rapid training of the neuron. Such a modification allows the Sigma-if neuron to dissect the decision space with two different hypersurfaces, one of which must be a straight line perpendicular to one of the dimensions.

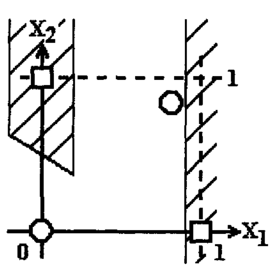

Fig. 1. Sample shape of trained Sigma-if neuron decision borders for the modified XOR function.

Fig. 1 presents a fragment of the decision space of a sample Sigma-if neuron trained to properly classify points belonging to the following function:

$$D(x_1, x_2) = \begin{cases} 0 & :(x_1, x_2) \in \{(0,0),(0.8,0.8)\} \\ 1 & :(x_1, x_2) \in \{(0,1),(1,0)\} \end{cases} \quad (7)$$

The parameters of this trained neuron are as follows: weight vector w =[1.3, 2.5] and grouping vector θ=[0, 1]. Thus since θ_1 is less than θ_2, input x_2 will be considered only if the partial neuron activation $\Delta g(0)$ (equal to the product of w_1 and x_1) is less than the activation threshold net^*. It can be seen that when the neuron's output values less or equal to 0.5 and greater than 0.5 are treated as classes 0 and 1 respectively, the Sigma-if neuron with the presented parameters correctly solves the linearly inseparable problem defined by (eq. 7).

This experimental confirmation of the theoretical potential of the Sigma-if neuron is further strengthened by achieving positive results for a training set which includes XOR function points, rotated by 45^o around point (1,0). It is , however, important to note that training results depend on the activation threshold level. For the presented problem positive results were obtained only for net^* between 0 and 1. Outside that range, successful training was not possible.

5.2 Sigma-if performance for artificial problems

The promising results described in the previous section have led us to check the performance of the proposed model for more complicated artificial problems. For this purpose we have chosen the well known nested spirals testbed, and the two-class 10x10 checkerboard classification task. [15,16] Both problems have been used to train a fully-connected classic neural network with two inputs, 50 neurons in one hidden layer and two outputs. The obtained results have been compared with the outcome of analogous tests of a Sigma-if neural network with an identical architecture.

Test results have shown that in the case of the nested-spirals testbed, the Sigma-if network acts very similarly to the standard feedforward neural network. Both networks have been able to properly solve the two-spirals problem at similar computational cost. The differences between both types of neural networks only became apparent during the second test. While the standard model was unable to reduce the classification error below 50%, the Sigma-if network reached a stage where almost 70% of patterns were classified correctly.

5.3 Minimization of the number of active attributes

Another experiment involved training the Sigma-if network to minimize the number of attributes required for proper classification of selected data sets from the UCI Machine Learning Repository. This necessitated extending the mechanism which controlled the random selection of grouping vector parameters with facilities for analyzing changes in the number of active input connections. Thus, when the number of active input attributes exceeds 50%, the training algorithm forces random selection of the grouping vector.

Such a mode of operation, for a limited number of neurons in the hidden layer (2 to 4, depending on the training set), enabled us to extract subsets of data attributes which had the greatest impact on the classification process. Table 1 presents the capabilities of the process, assuming at least 80% classification accuracy with the limited attribute set. The best results have been achieved for those training sets which can be expected to contain a lot of redundant data (i.e. Sonar and Breast-Cancer-W). Surprisingly, for the latter set it was possible to reduce the number of active attributes to just two, while retaining a 92% accuracy (the two relevant attributes are Uniformity-of-Cell-Shape and Single-Epithelial-Cell-Size).

Table 1. Trimming data attribute sets with Sigma-if network.

Training set	No. of attributes (total)	No. of attributes (limited set)	Gain [%]
Heart	13	9	30.7
Hypothyroid	29	19	34.5
Iris	4	3	25.0
Breast-Cancer-W	9	2	77.7
Monk1	6	6	0
Monk2	6	6	0
Monk3	6	6	0
Sonar	60	10	83.3
Vote	16	9	43.7
		Mean	32.7

On average, the conducted tests permitted a 30% reduction in the number of attributes used in classification. We can therefore conclude that the proposed solution is well adapted to real-life applications, where data gathering can often be costly.

5.4 Extraction of knowledge from the Sigma-if network

The promising results of the above experiments raise questions about whether such networks process and store knowledge in a way that differs from classic neural networks. To shed some light on that problem, it is necessary to use a knowledge extraction method. The task of extracting knowledge from such an atypical neural network as the Sigma-if network essentially limits the selection of applicable algorithms to black-box-type methods. Such methods are scarce, so we have come to rely on the Trepan algorithm, since the available sources strongly recommend it. [17, 18]

Just like in the previous experiments, the results obtained are very interesting. It appears that, when compared to decision trees extracted from classic neural networks, the Sigma-if trees are much more readable - they are less complex and use a smaller number of decision attributes. They also more accurately represent the functioning of the network from which they have been derived as well as the properties of data being classified. It seems advisable to further study the possibility of applying Sigma-if networks in data mining applications.

6 Summary

The theoretical considerations presented in this paper and confirmed by experimental results clearly point to the fact that conditional signal accumulation is a useful technique in the area of neuronal data processing. In light of its potential benefits, the proposed modifications of classic artificial neural networks require further extensive study. It is, for example, difficult to explain why Monk training sets have proven more difficult to classify, even though they also include redundant data attributes.

It is also worth to underline that full exploitation of the Sigma-if model's potential is possible only when a suitable method of selecting grouping vector coefficients is used. The heuristic approach, mentioned earlier, has yielded some interesting results, but no definitive solution can yet be presented. This problem determines further directions of Sigma-if network research.

While the Sigma-if neuron is only a simple computational model, not designed for modeling biological neurons, it would nevertheless be interesting to research the relations between the properties of Sigma-if networks and those of biological processing systems.

7 Acknowledgements

The above research in the area of intelligent information systems and machine learning is funded by the Polish Ministry of Scientific Research and Information Technology within the 4 T11C 02425 research grant. This support is gratefully acknowledged.

References

[1] Duch, W., Jankowski, N. (1999) Survey of neural transfer functions. Neural Computing Surveys 2: 163-212

[2] Anthony, M., Bartlett, P.L. (1999) Neural networks learning: Theoretical foundations, 1st ed. Cambridge University Press, Cambridge

[3] Hammer, B., Tino, P. (2003) Recurrent neural networks with small weights implement definite memory machines. Neural Computation 15(8): 1897-1929

[4] Murray, A.F., Edwards, P.J. (1992) Synaptic weight noise during MLP learning enhances fault-tolerance,

generalisation and learning trajectory. Advances in Neural Information Processing Systems 5: 491-498

[5] Hornik, K., Stinchcombe, M., White, H. (1989) Multilayer feedforward networks are universal approximators. Neural Networks 2: 359-366

[6] Durbin, R., Rumelhart, D. (1989) Product units: A computationally powerful and biologically plausible extension to backpropagation networks. Neural Computation 1: 133-142

[7] Schmitt, M. (2002) On the complexity of computing and learning with multiplicative neural networks. Neural Computation 14: 241-301

[8] Duch, W., Jankowski, N. (2001) Transfer functions: hidden possibilities for better neural networks. In Verleysen, M. (ed.) 9th European Symposium on Artificial Neural Networks. D-Facto, Brugge, pp. 81-94

[9] Cohen, S., Intrator, N. (2002) A hybrid projection based and radial basis function architecture: initial values and global optimization. Pattern Analysis & Applications 5: 113-120

[10] Schmitt, M. (2002) Neural networks with local receptive fields and superlinear VC dimension. Neural Computation 14(4): 919-956

[11] Huk, M. (2004) The Sigma-if neural network as a method of dynamic selection of decision subspaces for medical reasoning systems. Journal of Medical Informatics & Technologies 7: 65-73

[12] Ridella, S., Rovetta, S., Zunino, R. (1999) Representation and generalization properties of Class-Entropy Networks. IEEE Transactions on Neural Networks 10(1): 31-47

[13] Banarer, V., Perwass, C., Sommer, G. (2003) The hypersphere neuron. In: Verleysen, M. (ed.) 11th European Symposium on Artificial Neural Networks. D-Side, Brugge, pp. 469-474

[14] Huk, M. (2003) Determining the relevancy of attributes in classification tasks through nondestructive elimination of interneuronal connections in a neural network (in Polish). Pozyskiwanie Wiedzy z Baz Danych 975: 138-147

[15] Fahlman, S.E., Lebiere, C. (1990) The Cascade-Correlation learning architecture. Advances in Neural Information Processing Systems 2: 524-532

[16] Kaufman, L., (1999) Solving the quadratic programming problem arising in support vector classification. In: Scholpkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in Kernel Methods - Support Vector Learning. MIT Press, Boston, pp. 146-167

[17] Tickle, A., Andrews, R., Golea, M., Diederich, J. (1998) The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks. IEEE Transactions on Neural Networks 9(6): 1057-1068.

[18] Craven, M.W., Shavlik, J.W. (1996) Extracting tree-structured representations of trained networks. Advances in Neural Information Processing Systems 8: 24-30