INTELLECT SENSING OF NEURAL NETWORK THAT TRAINED TO CLASSIFY COMPLEX SIGNALS

Reznik A. Galinskaya A.

Abstract: *An experimental comparison of information features used by neural network is performed. The sensing method was used. Suboptimal classifier agreeable to the gaussian model of the training data was used as a probe. Neural nets with architectures of perceptron and feedforward net with one hidden layer were used. The experiments were carried out with spatial ultrasonic data, which are used for car's passenger safety system neural controller learning. In this paper we show that a neural network doesn't fully make use of gaussian components, which are first two moment coefficients of probability distribution. On the contrary, the network can find more complicated regularities inside data vectors and thus shows better results than suboptimal classifier. The parallel connection of suboptimal classifier improves work of modular neural network whereas its connection to the network input improves the specialization effect during training.*

1. Task description

Experience shows that learning speed and decision's accuracy of complex tasks can be improved using different methods of neural networks combination in the multimodular systems [Sharkey. 1996], [Sharkey et al, 1997]. Great amount of papers is dedicated to understanding the potential capabilities and practical application of modular neural networks. Most of these papers describe homogeneous multimodular structures based on feedforward neural networks. Methods of input data preprocessing and different types decision fusion modules are discussed [Giacinto et al, 2001], [Crepet et al, 2000]. Multimodular architectures based on different neural network types are investigated [Crepet et al, 2000], [Tang et al], [Happel et al, 1994]. Most of researches incline to common conclusion that multimodular neural networks act like a number of experts, which consider task from different positions. Due to such organization decisions based on local reactions of individual neural ensembles are more infallible.

Unfortunately, simple mechanical transfer of collective human expert behavior to the neural ensembles, level of complexity (and intellect) of which could be compared may be only with worm's neural system, hardly appropriate. Of course, neural network is able to learn and take self-dependent decisions, and so we can consider that it has artificial intellect that characterized by outer world models presence. But such interpretation cannot be used for understanding inner procedure of forming and co-ordination of neural modules' decisions.

In our investigation we used sensing method for clarifying factors that define behavior of trained neural network. For this purpose statistically optimal receiver solving the same problem that neural network was used. We proceed from that neural network during training tends to statistically optimal behavior for defined training set and errors criteria. Having statistical distribution for training data it's possible to create statistically optimal device that satisfies defined criteria of work quality and estimates the characteristics of trained network's behavior. Combining such optimal device with neural net in modular structure it's possible to investigate their interaction during training and decision-making. Nature of information features of initial data, which neural network uses for making decision, can be understood by varying different combination methods.

Unfortunately, it's necessarily to have consistent estimates for every combination of training set elements' values to create statistically optimal device. It's almost impossible for big data sets. But if data vectors' elements have weak statistical dependence we can assume that the conditions of central limit theorem are met. In this case the gaussian model of distribution for training set can be a good approximation. This model considers first two moments of distribution: average and covariation. It's easy to obtain the estimations for their values. Having such estimations we can create suboptimal receiver that reflect the main features of training data set.

The goal of this work is experimental investigation of described approach to the analysis of real neural net behaviour. Comparison of behaviour of suboptimal classifier and neural net was carried out with spatial ultrasound signals used in car's passenger safety system. Analysing these signals neural net should estimate level of safety for passenger position and block air-bag deployment if passenger can be damaged. All experiments were carried out with a help of MNN CAD software [Kussul et al, 2002] using data provided by Automotive Technologies International (ATI Inc. Danville, New-Jersey, USA).

2. Description of investigated neural architectures

Five base classifier models were used during experiments:

1. Feed-forward neural network with one hidden layer;
2. Simple perceptron;
3. Linear suboptimal classifier;
4. Quadratic suboptimal classifier;
5. Combination of linear and quadratic suboptimal classifiers.

Experiments were carried out with a help of MNN CAD [Kussul et al, 2002] software that allows creating multimodular structures using different types of neural networks and additional modules. Figure 1 shows the architecture of one of the used methods of modules combination. It's composed of feedforward neural net (module B3) and suboptimal classifier (B4). Fusion module (B5) consists of one neuron. Module B1 is used for input vector normalization, module B2 forms target vectors of all modules during training.

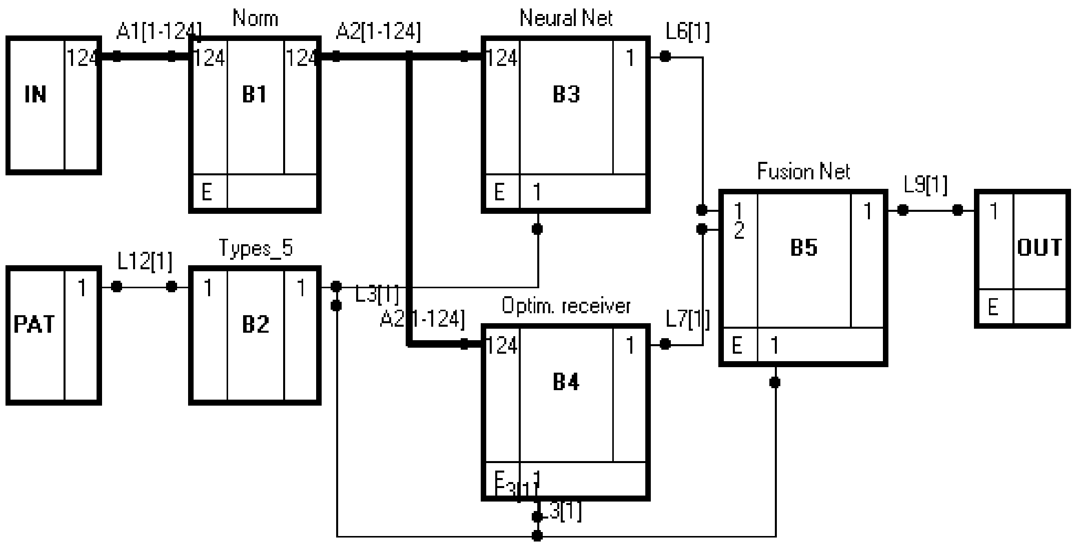


Fig. 1. Multimodular classifier architecture (Var2)

We examined 12 different variants of classifier including 5 base models, combination of feedforward network with perceptron and 6 types of hybrid modular network that combined feedforward net with suboptimal classifier. Two ways of construction of such hybrid network were used. In the first net's variant (Var1) suboptimal classifiers were connected to additional input of neural network that is additional component of input vector was created. In the second structure type (Var2) we used additional fusion neuron with output of neural network and normalized suboptimal classifier's output as its inputs. In both variants neural network was trained after connection of suboptimal classifier that is its reaction took part in the neural network's decision forming.

3. Structure of suboptimal classifier

In our task training data set consists of vectors *X* for two classes "0" and "1". We consider that data is distributed normally. Frequency distribution of probability for *i –th* class is:

$$W_i(X) = \frac{1}{\sqrt{(2\pi)^N |\Psi_i|}} \exp\left[-\frac{1}{2}(X - A_i)\Psi_i^{-1}(X - A_i)^T \right]. \qquad (1)$$

Here Ψ_i is the covariance matrix for data of *i*-th class; $|\Psi_i|$ is the determinant of the matrix Ψ_i;

A_i is the average of distribution of *X*; X_i^T is the transposed vector; N is the dimension of the vectors A_i, X_i.

A statistically optimal solution that provides a minimum risk of error is based on a threshold estimate of the value *T(X)* that defines structure of optimal classifier [Middleton, 1960] :

$$T(X) = \ln W_1(X) - \ln W_0(X)\,, \tag{2}$$

Decision of the situation either "0" or "1" is made after results of comparison of value *T(X)* with the threshold that depends on the relationship between the costs of losses caused by errors to one or the other side. After substitution of expression (1) to (2) obtain:

$$T(X) = \underline{const} + X\left(A_1\Psi_1^{-1} - A_0\Psi_0^{-1}\right)^T - \frac{1}{2}X\left(\Psi_1^{-1} - \Psi_0^{-1}\right)X^T \tag{3}$$

$$\underline{const} = \frac{1}{2}\left(\ln\left|\Psi_0\right| - \ln\left|\Psi_1\right| + A_0\Psi_0^{-1}A_0^T - A_1\Psi_1^{-1}A_1^T\right).$$

The second and the third terms of the expression that depend on *X* define the linear and the quadratic components of the classifier. The function of the former can be represented by the following sum:

$$S_l = \sum_{i=1}^{N} u_i x_i\,, \tag{4}$$

where u_i are components of the optimum filter vector:

$$U = A_1\Psi_1^{-1} - A_0\Psi_0^{-1}. \tag{5}$$

The quadratic component function of the classifier is:

$$S_s = \frac{1}{2}X\left(\Psi_0^{-1} - \Psi_1^{-1}\right)X^T. \tag{6}$$

Using expressions (4)-(6) we can create suboptimal classifier with sample estimates of distribution average and covariation matrices for training set. If the covariation matrices values for both classes are similar then optimal classifier consists of only linear part. Also it is possible that linear component (5) is zero so optimal is the quadratic classifier (6).

4. Statistical characteristics of data set

The data vectors used are sequences of 124 short integer values of output signals of four ultrasound sensors. Data was divided into three groups: Train – 128000 vectors; Test – 38400 vectors and Valid – 16800 vectors, which were used for training, testing during training and network validation correspondingly. Each array consists of "0" class vectors (safe passenger position) and "1" class vectors (air-bag deploy should be blocked) with approximately equal amount.

Statistical characteristics of vectors obtained on Train data set are shown on Figure 2. On this figure diagonal elements of the matrices and the boundaries of the sensor areas can be clearly seen. Differences of covariation matrices (on the right window) are the most pronounced classification features of spatial signals.

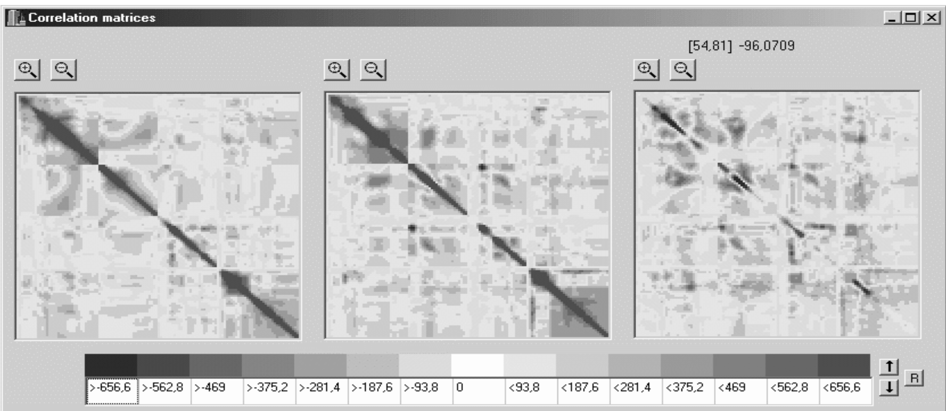


Fig. 2. Covariance matrices for classes "0" and "1" and their difference (on the right).

The results of data analysis shown on Fig. 2 were used to suboptimal classifier's components calculation (4-6). Figure 3 shows values of linear part *U* of filter.

5. Neural modules

Before investigation of modular network with suboptimal classifier in it we chose the best architecture of neural network. Best results were obtained using three-layer neural network with 15 neurons in hidden layer. Sigmoid activation function and EDBD learning algorithm [Reed et al, 1999] were used. Input data was normalized in [0,1] diapason. Normalization was made independently for each input using training data set. Initial values for neural network module were defined by random values. Each experiment was carried out 5 times and results were averaged.

Perceptron module consisted of one neuron with sign activation function and was trained using Hebb learning rule. Initial weights for this module were zero. Weight coefficients obtained during training are shown on Fig. 3. The same figure shows weights of suboptimal linear filter (5). Position of extreme values is similar for both charts. Suboptimal filter has strong extremes only in the range of 64-67 inputs. In contrast to this values of input's weights for perceptron distributed more equally.

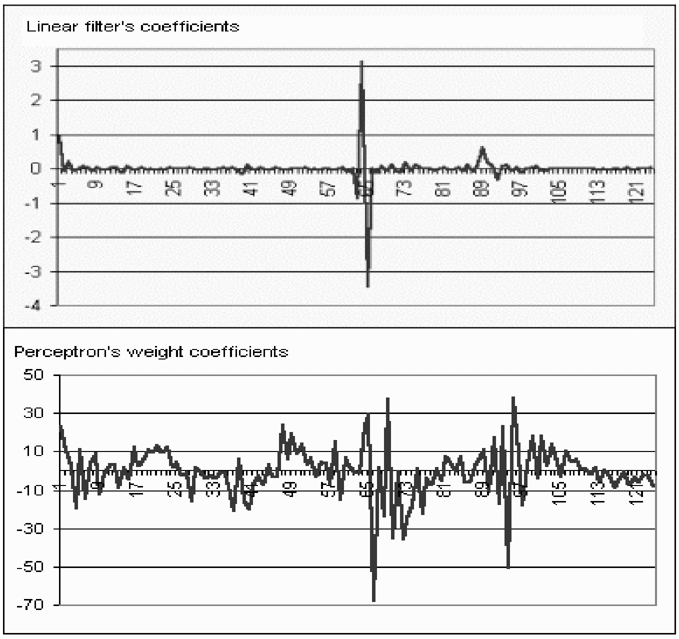


Fig. 3. Weight coefficients of perceptron and suboptimal linear filter

6. Results of independent modules' testing

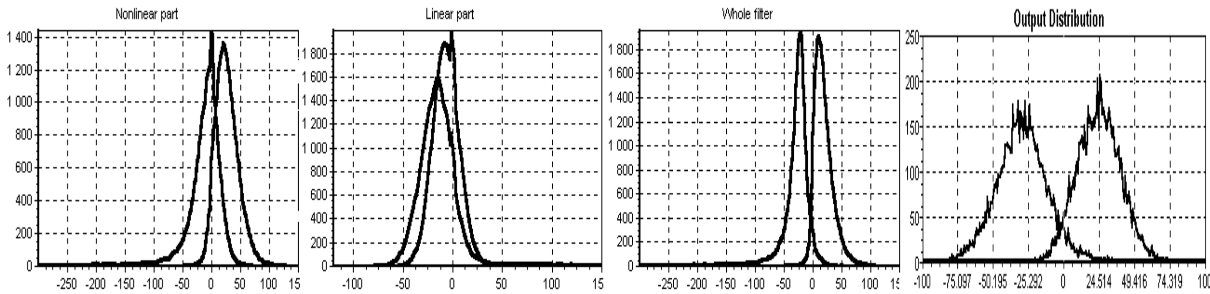
Table 1 shows comparative testing results for base classifier's models on Train and Valid data sets.

Table 1. Error rate for base modules.

Type	Train %	Valid %
SO/Linear	42.15	37.92
SO/Square	18.25	26.85
SO/Lin. + Sq.	4.81	5.92
Perceptron	6.02	4.83
Neural network	1.98	3.17

Table shows that feedforward neural network gives best results. Perceptron and suboptimal classifier (even with both linear and quadratic components) show much worse success rates. Suboptimal classifier shows better results than perceptron on training data set but much worse on validation set. So we can conclude that neural network during training finds more complex associative relations than contained in average values and mutual correlation functions. To some extent the same suggestions can be applied to perceptron, decisions of which are based on linear transformation of input data. This transformation is similar to that linear suboptimal classifier uses.

Figure 4 shows histograms of reactions of suboptimal classifier and perceptron for two data classes. Diagrams correspond (left to right) to the nonlinear component, linear component, full suboptimal classifier and postsynaps values of perceptron. Under each diagram there is the threshold value and the average number of correct decisions for each situation class. It can be seen that results for separate linear and quadratic classifiers are much worse than for full classifier. Distribution of potential at the perceptron's input is similar to the distribution of reaction of full suboptimal classifier.



0 / 1	19.29/ 17.81	25.55 / 58.75	5.48 / 4.14	6.23 / 5.82
Total %	18.55	42.15	4.81	6.02

Fig. 4. Histograms of reactions for suboptimal classifier and perceptron. Left to right: linear, quadratic, full suboptimal classifier, perceptron (Train data set).

7. Experiments with hybrid networks.

The goal of experiments was to understand the importance of classification features used by suboptimal classifier. We tested hybrid networks Var1 and Var2 with linear, quadratic and full suboptimal classifiers. Thus neural network before training had all useful information that suboptimal classifier extracts from input data. Also we investigated variant of hybrid network Var2 where already trained perceptron was used instead of suboptimal classifier.

Training was carried out using Save Best method until 1.5M vectors training depth. Best result was saved. Each experiment was carried out 5 times with different weight initialization. In each series of experiments average and minimum error rates were calculated.

Experimental results are shown in Table 2. Data in the last row of table for Var2 were obtained while autonomous testing of neural network within hybrid network Var2.

Table 2. Experimental results for two types of hybrid network

Mode	Test mode	Var1		Var2	
		Avg.%	Min%	Avg.%	Min%
SO/ Linear	Train	2.02	1.78	1.88	1.75
	Test	4.39	4.10	4.34	4.29
	Valid	3.59	3.32	3.53	3.33
SO/ Square	Train	2.13	1.74	1.92	1.79
	Test	4.35	4.17	4.39	4.18
	Valid	3.49	3.18	3.46	3.20
SO/ Lin. + Sq.	Train	1.87	1.64	1.74	1.63
	Test	4.31	4.23	3.98	3.77
	Valid	3.56	3.38	2.97	2.67
Perceptron	Train			1.89	1.76
	Test			4.28	4.22
	Valid			3.26	3.13
Neural network	Train	2.23	1.98	2.09	1.94
	Test	4.31	4.29	4.49	4.15
	Valid	3.48	3.17	3.67	3.28

Testing results on Train data set show that connection of suboptimal classifier leads to decreasing of classification errors for both types of hybrid network. Best results were obtained using both linear and quadratic components of suboptimal classifier. But for Test array there is no such improvement and error rate on Valid data set even increased for Var2 network. Testing of Var2 network on this array gives opposite result – error rate decreases almost on one third.

Connection of perceptron in Var2 network also decreases error rate for all data arrays. But success rate is worse than with connection of suboptimal classifier.

Quantitative estimation of information level of suboptimal classifier's decision can be obtained by comparison of weights values for the inputs of fusion neuron in Var2 network. Table 3 shows these estimations. We used such symbols: W_{NN} – weight value for neural network's output; W_{SO} – weight value for suboptimal classifier's output; $k=W_{SO}/(W_{NN}+W_{SO})$ – information level for corresponding component. Information level is about 15% for linear component of suboptimal classifier, for quadratic it's much higher – more than 25%. Naturally highest information level was obtained for sum of components – more than 37%. Estimation of information level for perceptron appears enough unexpected – less than 3%.

Table 3. Estimation of information level for gaussian features in hybrid neural network

Mode	W_{NN}	W_{SO}	$k = W_{SO}/ (W_{NN}+ W_{SO})$
SO/ Lin.	4.92	0.91	0.156
SO/ Sq.	4.74	1.63	0.256
SO/ Lin. + Sq.	4.15	2.49	0.376
Perceptron	3.95	0.11	0.027

8. Conclusion

Distribution of used experimental material is close enough to multidimensional gaussian distribution. So we could expect that neural network will use mainly the same classification features that suboptimal classifier does. But neural network behavior shown is far from that. Neural network finds complex associative relations between data elements during training. But it does not fully make use of more simple correlation dependencies used by suboptimal classifier. As it can be seen in Table 2, it's obvious that connection of suboptimal classifier always decreases error rate of neural network. Success rate improvement is the most when suboptimal classifier is connected to fusion neuron (Var2). Improvement effect in this case is the strongest for Valid data set. This is a very important result. It shows that generalization ability was improved. Also we can conclude that decision-making criteria used by neural network and suboptimal classifier are relatively independent.

If suboptimal classifier is connected to neural network input (Var1) success rate increases only for Train data set. It shows that specialization effect becomes stronger due to suboptimal classifier's connection. We found such effect for the first time and have no explanation for it yet. We can only suppose that neural network's hidden layer blocks information flow from suboptimal classifier to network's output. The nature of this phenomenon is unclear but we can make practical suggestion that the most informational features should be connected closer to output of hybrid neural network.

Low information level of perceptron connected to fusion neuron appeared quite unexpected. Its weight is less than 3% that is much less than for linear and quadratic suboptimal classifiers (15% and 25%). This is opposite to results shown in Table 2. We can suppose that in this case information features formed by neural network have the same character that for perceptron but they are much more powerful. So perceptron contribution to the final decision is insignificant.

Acknowledgments

The work is supported by INTAS grant 2001-0257 "Smart Sensors for Field Screening of Air Pollutants". The authors wish to thank David Breed and ATI Inc. for the giving data sets.

Bibliography

[Sharkey, 1996] Amanda J.C. Sharkey, On combining artificial neural nets. - Connection Science. - 1996. - V. 8. - N. 3/4. - P. 299-314.

[Sharkey et al, 1997] Amanda J.C. Sharkey, Noel J. Sharkey, Combining diverse neural networks. - The Knowledge Engineering Review. - 1997. - V. 12: 3. -P. 1-17.

[Giacinto et al, 2001] Giorgio Giacinto, Fabio Roli, Design of effective neural network ensembles for image classification purposes. - Image and Vision Computing. -Elsevier Science. - 2001. - V. 19. - P. 699-707.

[Crepet et al, 2000] Agnes Crepet, Helen Paugam-Moisy, Emanuelle Reynaud, Didier Puzenat, A modular neural model for binding several modalities/ In Proc. of IC-AI'2000, Int. Conf. of Artificial Intelligence. - Las Vegas, USA. - 2000. - P. 921 – 928.

[Tang et al] Bin Tang, Malcolm I. Heywood, Michael Shepherd, Input Partitioning to Mixture of Experts

[Happel et al, 1994] Bart L.M. Happel, Jakob M.J. Murre, The design and evolution of modular neural network architecture. - Neural Networks. – 1994. – V. 7. – P. 985-1004.

[Kussul et al, 2002] Kussul M., Riznyk A., Sadovaya E., Sitchov A., Chen T.Q. A visual solution to modular neural network system development // Proc. of the 2002 International Joint Conference on Neural Networks IJCNN'02. - Honolulu, HI, USA. - 2002. - 12-17 May. - Vol.1. - N749.

[Middleton, 1960] D. Middleton, An Introduction to Statistical Communication Theory, V. 2., 1960.

[Reed et al, 1999] R.D. Reed and R.J. Marks, Neural Smithing. MIT Press. - 1999. – P. 346.

Author information

Reznik A., Galinskaya A. - Department of neural technologies, Institute of Mathematical Machines and systems NAS Ukraine. neuro@immsp.kiev.ua

APPLICATION OF THE SUFFICIENCY PRINCIPLE IN ACCELERATION OF NEURAL NETWORKS TRAINING

Krissilov V.A., Krissilov A.D., Oleshko D.N.

Abstract: One of the problems in AI tasks solving by neurocomputing methods is a considerable training time. This problem especially appears when it is needed to reach high quality in forecast reliability or pattern recognition. Some formalised ways for increasing of networks' training speed without loosing of precision are proposed here. The offered approaches are based on the Sufficiency Principle, which is formal representation of the aim of a concrete task and conditions (limitations) of their solving [1]. This is development of the concept that includes the formal aims' description to the context of such AI tasks as classification, pattern recognition, estimation etc.

Keywords: neural networks

Introduction

Nowadays developers have a lot of different models of neural networks and algorithms of their training [2, 3] for disposal. Though the scientific researches are permanently carried on in this field, the theory of neural networks is still feebly formalised. However, even now two stages of creation of artificial neural systems could be defined: structural and parametric synthesis. At the first stage, developer has to do the following: choose the model for the network, define its structure and choose the algorithm for its training. The parametric synthesis includes training processes of the created network and verification of the obtained results. Then, depending on verification results, there can be a necessity of return to one of the stages of structural or parametric synthesis. Thus, becomes obvious that creation of the neural system is an iterative process.

Feeble formalisation of these stages results in necessity for the developer of the neural system to solve a number of problems. E.g., at the structural synthesis stage, in case of solving a non-standard task, it is necessary to spend a lot of time for choosing the corresponding model for the network, choosing its structure and training method. The problem of the parametrical synthesis is a considerable training time. If real tasks are being solved without any simplification, then duration of training process for created network could be too long. However, some tasks require spending as less training time as it is possible, e.g., real-time tasks.

The aim of the given article is to offer possible methods to reduce the training time for neural networks with back propagation training algorithm. As such methods are offered: control of procedures of modification and