Statistical Control of RBF-like Networks for Classification

Norbert Jankowski[1] and Visakan Kadirkamanathan[2]

[1] Nicholas Copernicus University, Toruń, Poland, e-mail:norbert@phys.uni.torun.pl
[2] The University of Sheffield, UK, e-mail: visakan@acse.shef.ac.uk

Abstract. *Incremental Net Pro* (IncNet Pro) with local learning feature and statistically controlled growing and pruning of the network is introduced. The architecture of the net is based on RBF networks. *Extended Kalman Filter* algorithm and its new fast version is proposed and used as learning algorithm. IncNet Pro is similar to the *Resource Allocation Network* described by Platt in the main idea of the expanding the network. The statistical *novel criterion* is used to determine the growing point. The Bi-radial functions are used instead of radial basis functions to obtain more flexible network.

1 Introduction

The *Radial Basis Function* (RBF) networks [13,12] were designed as a solution to an approximation problem in multi–dimensional spaces. The typical form of the RBF network can be written as

$$f(\mathbf{x};\mathbf{w},\mathbf{p}) = \sum_{i=1}^{M} w_i G_i(||\mathbf{x}||_i, \mathbf{p}_i) \tag{1}$$

where M is the number of the neurons in hidden layer, $G_i(||\mathbf{x}||_i, \mathbf{p}_i)$ is the i-th Radial Basis Function, $\mathbf{p}_i$ are adjustable parameters such as centers, biases, etc., depending on $G_i(||\mathbf{x}||_i, \mathbf{p}_i)$ function which is usually choosed as a Gaussian ($e^{-||\mathbf{x}-\mathbf{t}||^2/b^2}$), multi-quadratics or thin-plate spline function[1]. In contrast to many *artificial neural networks* (ANNs) including well known *multi-leyered perceptrons* (MLPs) networks the RBF networks have well mathematical properties. Girosi and Poggio [6,12] proved the existence and uniqueness of best approximation for regularization and RBF networks. In the 1991 Platt published the article on the *Resource–Allocating Network* [11]. The RAN network is an RBF-like network that grows when two criteria are satisfied:

$$\mathbf{y}_n - f(\mathbf{x}_n) = e_n > e_{min}; \qquad ||\mathbf{x}_n - \mathbf{t}_c|| > \epsilon_{min} \tag{2}$$

e_n is equal the current error, $\mathbf{t}_c$ is the nearest center of a basis function to the vector $\mathbf{x}_n$ and e_{min}, ϵ_{min} are some experimentally choosen constants. The growing network can be described by $f^{(n)}(\mathbf{x},\mathbf{p}) = \sum_{i=1}^{k-1} w_i G_i(\mathbf{x},\mathbf{p}_i) + e_n G_k(\mathbf{x},\mathbf{p}_k) =$

[1] For a interesting review of many other transfer function see [3].

$\sum_{i=1}^{k} w_i G_i(\mathbf{x}, \mathbf{p}_i)$, where $\mathbf{p}_k$ includes centers $\mathbf{x}_n$ and others adaptive parameters which are set up with some initial values. If the growth criteria are not satisfied the RAN network uses the LMS algorithm to estimate free parameters. Although LMS algorithm is faster than *Extended Kalman Filter* (EKF) algorithm [1] we decided to used EKF algorithm because it exhibits fast convergence, use lower number of neurons in hidden layer [9] and gives some *tools* which would be useful in control of the growth and pruning process.

The Goal of IncNet Pro The main goal of our researche was to build a network which would be able to adjust the complexity of the network to complexity of the data shown to the network during the learning time.

The IncNet Pro tries to solve the above task in 4 ways: • ESTIMATION: The typical learning process is based on fast EKF algorithm. • GROWING: If the *novelty criterion* is satisfied then a new neuron is added to the hidden layer. • DIRECT PRUNING: IncNet algorithm checks whether or not a neuron should be pruned. If yes, then the neuron with the smallest saliency is removed. • BI-RADIAL FUNCTIONS: The Bi-radial transfer function estimate more complex density of input data through using the separate biases and separate slopes in each dimension and for each neuron.

Similar work has been done in recent years by several authors, but it is quite rare to combine growing and pruning in one network, which is quite important for optimal generalization of the network. Weigend, Rumelhart & Huberman [16] described weight-decay, pruning neurons with smallest magnitude of weights. LeCun et al. [10] described more effective pruning method, *Optimal Brain Damage*. Hassibi in 1993 [7] published the *Optimal Brain Surgeon* algorith, which works without assumption used by LeCun that the Hessian matrix is near diagonal.

RAN network using EKF learning algorithm (RAN-EKF) was proposed by [9]. The M-RAN net [17] is based on RAN-EKF with pruning based on removing neurons with smallest normalized output from hidden layer. The previous version of the IncNet [8] is a RAN-EKF network with statistically controlled growth criterion. Another very good example, derived from MLP network, is the *Cascade–Correlation* algorithm [4]. *Feature Space Mapping* (FSM) system is the system which joins two strategies: growing and pruning, see [2] for more information. For more exhaustive description of ontogenic neural network see [5].

2 The IncNet Pro Framework

Fast EKF: We introduce new fast version of the EKF learning algorithm, described in [1]. The EKF was chosen because it can estimate not only adaptive parameters, but also some others values which will be used in novelty criterion and in pruning.

Covariance matrix $\mathbf{P}_n$ can be quite large for real data because its size is the square of the total number of adaptive parameters. Assuming that correlations between parameters of different neurons are not very important we can simplify the matrix $\mathbf{P}_n$ assuming block-diagonal structure of $\widetilde{\mathbf{P}}_n$ with $\widetilde{\mathbf{P}}_n^i$, $i = 1 \ldots M$.

Diagonal elements represents correlations of adaptive parameters of the i-th neuron.

Let m be the number of adaptive parameters per neuron and M the number of neurons. The size of matrix $\mathbf{P}_n$ is $m \cdot M \times m \cdot M$, but matrix $\widetilde{\mathbf{P}}_n$ has only $m^2 M$ elements not equal to zero. For a given problem $\mathcal{P}$ the complexity of matrix $\mathbf{P}_n$ is $O(M^2)$, and matrix $\widetilde{\mathbf{P}}_n$ just $O(M)$ (m is constant in $\mathcal{P}$)! Using this approximation the fast version of the EKF algorithm is:

$$\begin{aligned}
e_n &= y_n - f(\mathbf{x}_n; \mathbf{p}_{n-1}) \\
\mathbf{d}_n^i &= \tfrac{\partial f(\mathbf{x}_n; \mathbf{p}_{n-1})}{\partial \mathbf{p}_{n-1}^i} \\
R_y &= R_n + \mathbf{d}_n^{1\,T} \widetilde{\mathbf{P}}_{n-1}^1 \mathbf{d}_n^1 + \cdots + \mathbf{d}_n^{M\,T} \widetilde{\mathbf{P}}_{n-1}^M \mathbf{d}_n^M \qquad i = 1, \ldots, M \\
\mathbf{k}_n^i &= \widetilde{\mathbf{P}}_{n-1}^i \mathbf{d}_n^i / R_y \\
\mathbf{p}_n^i &= \mathbf{p}_{n-1}^i + e_n \mathbf{k}_n^i \\
\widetilde{\mathbf{P}}_n^i &= [\mathbf{I} - \mathbf{k}_n^i \mathbf{d}_n^{i\,T}] \widetilde{\mathbf{P}}_{n-1}^i + Q_0(n) \mathbf{I}
\end{aligned} \tag{3}$$

the suffixes $n-1$ and n denote the priors and posteriors. $\mathbf{p}_n$ consists of all adaptive parameters: weights, centers, biases, etc. To prevent too quick convergence of the EKF, which leads to data overfitting, the $Q_0\mathbf{I}$ adds a *random* change, where Q_0 is scalar (sometimes decreasing to small values around 10^{-5}) and $\mathbf{I}$ is the identity matrix.

Novelty Criterion: Using methods which estimate during learning covariance of uncertainty of each parameter, the network output uncertainty can be determined and the same criterion as in the previous version of IncNet [8] may be used. Then the hypothesis for the statistical inference of model sufficiency is stated as follows:

$$\mathcal{H}_0 : \quad \frac{e^2}{\mathrm{Var}[f(\mathbf{x}; \mathbf{p}) + \eta]} = \frac{e^2}{R_y} < \chi^2_{n,\theta} \tag{4}$$

where $\chi^2_{n,\theta}$ is θ% confidence on χ^2 distribution for n degree of freedom, $e = y - f(\mathbf{x}; \mathbf{p})$ is the error and $R_y = \mathrm{Var}[f(\mathbf{x}; \mathbf{p}) + \eta]$ (part of EKF) — see Eq. 3.

If this hypothesis is satisfied the current model is sufficient and the IncNet network continues learning using the fast EKF algorithm. Otherwise, a new neuron $(M+1)$-th should be added with some initial parameters. For Gaussian functions $G_{M+1}(\cdot)$ these parameters are: $w_{M+1} := e_n, \quad \mathbf{t}_{M+1} := \mathbf{x}_n, \quad b_{M+1} := b_0, \quad \mathbf{P}_n := \begin{bmatrix} \mathbf{P}_n & 0 \\ 0 & P_0 \mathbf{I} \end{bmatrix}$, where e_n is the error for given input vector $\mathbf{x}_n$, b_0 and P_0 are some initial values for bias (depending on a given problem) and covariance matrix elements (usually 1).

Pruning: As a result of the learning process a neuron can become completely useless and should be pruned. Assume the structure of vector $\mathbf{p}_n$ and the covariance matrix as:

$$\mathbf{p}_n = [w_1, \ldots, w_M, \ldots]^T \qquad \mathbf{P} = \begin{bmatrix} \mathbf{P}_w & \mathbf{P}_{wv} \\ \mathbf{P}_{wv}^T & \mathbf{P}_v \end{bmatrix} \tag{5}$$

where $\mathbf{P}_w$ is a matrix of correlations between weights, $\mathbf{P}_{wv}$ between weights and other parameters, $\mathbf{P}_v$ **only** between others parameters (excluding all weights).

Then by checking the inequality $\mathcal{P}$ presented below we can decide whether to prune or not and find the neuron for which value L has smallest saliency and should be pruned.

$$\mathcal{P}: \; L/R_y < \chi^2_{1,\vartheta} \qquad\qquad L = \min_i w_i^2/[\mathbf{P}_w]_{ii} \tag{6}$$

where $\chi^2_{n,\vartheta}$ is $\vartheta\%$ confidence on χ^2 distribution for one degree of freedom.

Neurons are pruned if the saliency L is too small and/or the uncertainty of the network output R_y is too big.

Bi-radial Transfer Functions: To obtain greater flexibility the bi-radial transfer functiona [3] are used instead of Gaussians. These functions are build from products of pairs of sigmoidal functions for each variable and produce decision regions for classification of almost arbitrary shapes.

$$Bi(\mathbf{x};\mathbf{t},\mathbf{b},\mathbf{s}) = \prod_{i=1}^{N} \sigma(e^{s_i}\cdot(x_i - t_i + e^{b_i}))(1-\sigma(e^{s_i}\cdot(x_i - t_i - e^{b_i}))) \tag{7}$$

where $\sigma(x) = 1/(1+e^{-x})$. The first sigmoidal factor in the product is growing for increasing input x_i while the second is decreasing, localizing the function around t_i. Shape adaptation of the density $Bi(\mathbf{x};\mathbf{t},\mathbf{b},\mathbf{s})$ is possible by shifting centers $\mathbf{t}$, rescaling $\mathbf{b}$ and $\mathbf{s}$, see Fig. 1. The number of adjustable parameters per processing unit is in this case (excluding weights w_i) $3N$. Dimensionality reduction is possible as in the *gaussian bar case* [3], but we can obtain more flexible density shapes, thus reducing the number of adaptive units in the network. Exponentials e^{s_i} and e^{b_i} are used instead of s_i and b_i to prevent oscillations during learning procedure (learning becomes more stable).

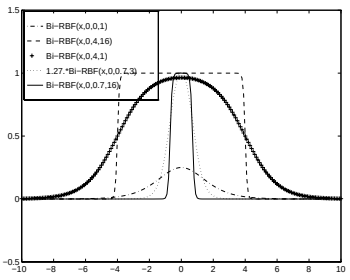
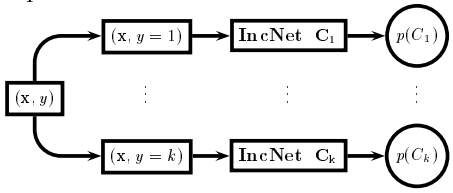


Fig. 1: A few shapes of the bi-radial functions in two dimensions.

Classification using IncNet Pro: k independed IncNet network are used for k-class problem. Each of them receives input vector $\mathbf{x}$ and 1 if index of i-th IncNet is equal to desired number of class, otherwise 0. The output of i-th IncNet Pro network is equal to probability that the vector belongs to i-th class. See figure on the right.



3 Results

The two-spiral problem. The data consists of two sets (training and testing) with 194 patterns each for two spirals. After 10,000 iterations (it took about 35

minutes on PC Pentium 150MHz) we got result which fit 192 points out of 194 (99%) for training set and 191 (98.5%) for the test set. Final net has 79 neurons. The fast version of EKF accelerates computation 50 times in comparision with standard EKF learning. There are other nets which are able to solve the two-spiral problem too, for example one of the best is an MLP using a global optimization algorithm by Shang and Wah [14]. Their network is able to get 100% correct results for the training set but never more than 95.4% for the test set. Although it used only 6 neurons, it takes about 200 minutes to train.

Breast Cancer, Hepatitis, Pima Indians Diabetes, Heart Disease are medical diagnosis benchmarks considered in [15]. Short summary of the data: Breast Cancer – 2 classes, 9 attributes, 699 instances; Hepatitis – 2 classes, 19 attributes, 155 instances; Diabetes – 2 classes, 8 attributes, 768 instances. Heart – 2 classes, 13 attributes, 303 instances.

method	Breast	Hepat.	Diab.	Heart
IncNet	**97.1**	**82.3**	**77.6**	**90.0**
BP	96.7	82.1	76.4	81.3
LVQ	96.6	83.2	75.8	82.9
CART	94.2	82.7	72.8	80.8
Fisher	96.8	84.5	76.5	84.2
LDA	96.0	86.4	77.2	84.5
QDA	34.5	85.8	59.5	75.4
KNN	96.6	85.3	71.9	81.5
LFC	94.4	81.9	75.8	75.1
ASI	95.6	82.0	76.6	74.4

Table 1: Accuracies (%) for medical benchmarks

Breast Cancer problem used 49 neurons and 3000 iterations, the accuracies on training and test sets was very similar: 97.7%, 97.1%, computation time: 5150 sec. Hepatitis data used 97 neurons and 500 iterations, the accuracies on training and test sets was: 98.6%, 82.3%, computation time: 3100 sec. Diabetes data used 100 neurons and 5000 iterations, the test accuracy was better on test set (77.6%) than on the training set (77.2%), computation time: 11200 sec. Heart data used 117 neurons and 1000 iterations, the training accuracy was 92.6% and test was 90.0%, computation time: 7400 sec.

4 Conclusions

The IncNet network is able to control the complexity of its structure by growing and pruning the network. In spite of incremental character of the algorithm, the *pruning time* is determined by theoretical criterion — not in random time moment or by checking the error on the whole training/test data set. Another advantage of the direct pruning is reduction of the time of computation. Nearly all parameters of the network are controlled automatically by EKF algorithm, the other parameters are very similar for different benchmark problems (excluding the biases and slopes, which are defined by the *resolution* of data). Another positive feature of IncNet Pro is the capacity of uniform generalization. In many benchmarks (see section 3) the errors on testing and training sets are much more similar than for other networks.

In some classification problems it would be useful to add the possibility of merging two neurons G_i and G_j which can be replaced by another neuron G_{new}

with a confidence α, for example using the criterion:

$$\int_{d \subseteq \mathcal{D}} |G_i(\mathbf{x}) + G_j(\mathbf{x}) - G_{new}(\mathbf{x})| < \alpha$$

Acknowledgments I'm grateful to prof. W. Duch for many valuable comments and to the Polish Committe for Scientific Research, grant 8T11F00308 for partial support.

References

1. J. V. Candy. *Signal processing: The model based approach*. McGraw-Hill, New York, 1986.
2. W. Duch and G. H. F. Diercksen. Feature space mapping as a universal adaptive system. *Computer Physics Communications*, 87:341–371, 1994.
3. W. Duch and N. Jankowski. New neural transfer functions. *Jour. of Applied Math. and Computer Science*. submitted.
4. S. E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In D. S. Touretzky, editor, *NIPS*. Morgan Kaufmann, 1990.
5. E. Fiesler. Comparative bibliography of ontogenic neural networks. In *Proceedings of the International Conference on Artificial Neural Networks*, 1994.
6. F. Girosi and T. Poggio. Networks and the best approximation property. AI Lab. Memo, MIT, 1989.
7. B. Hassibi and D. G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *NIPS*, 1993.
8. V. Kadirkamanathan. A statistical inference based growth criterion for the RBF network. In *Proc. IEEE. Workshop on Neural Networks for Signal Processing*, 1994.
9. V. Kadirkamanathan and M. Niranjan. A function estimation approach to sequential learning with neural networks. *Neural Computation*, 5(6):954–975, 1993.
10. Y. LeCun, J. Denker, S. Solla, R. E. Howard, and L. D. Jackel. Optimal brain damage. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems II*. Morgan Kauffman, 1990.
11. J. Platt. A resource-allocating network for function interpolation. *Neural Computation*, 3:213–225, 1991.
12. T. Poggio and F. Girosi. Network for approximation and learning. *Proc. IEEE*, 78:1481–1497, 1990.
13. M. J. D. Powell. Radial basis functions for multivariable interpolation: A review. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation of Functions and Data*, pages 143–167. Oxford University Press, 1987.
14. Y. Shang and W. Wah. Global optimization for neural network training. *IEEE Computer*, 29, 1996.
15. B. Šter and A. Dobnikar. Neural networks in medical diagnosis: Comparison with other methods. In A. B. B. et al., editor, *Proceedings of the International Conference EANN '96*, pages 427–430, 1996.
16. A. S. Weigend, D. E. Rumelhart, and B. A. Huberman. Back–propagation, weight elimination and time series prediction. In *Proceedings of the 1990 Connectionist Models Summer School*, pages 65–80. Morgan Kaufmann, 1990.
17. L. Yingwei, N. Sundararajan, and P. Saratchandran. A sequential learning scheme for function approximation using minimal radial basis function neural networks. *Neural Computations*, 9:461–478, 1997.