The Mixture of Neural Networks Adapted to Multilayer Feedforward Architecture

Joaquín Torres-Sospedra, Carlos Hernández-Espinosa,
and Mercedes Fernández-Redondo

Departamento de Ingenieria y Ciencia de los Computadores, Universitat Jaume I, Avda. Sos Baynat s/n, C.P. 12071, Castellon, Spain
{jtorres, espinosa, redondo}@icc.uji.es

Abstract. The *Mixture of Neural Networks* (*MixNN*) is a Multi-Net System based on the *Modular Approach*. The *MixNN* employs a neural network to weight the outputs of the expert networks. This method decompose the original problem into subproblems, and the final decision is taken with the information provided by the expert networks and the gating network. The neural networks used in *MixNN* are quite simple so we present a mixture of networks based on the *Multilayer Feedforward* architecure, called *Mixture of Multilayer Feedforward* (*MixMF*). Finally, we have performed a comparison among *Simple Ensemble*, *MixNN* and *MixMF*. The methods have been tested with six databases from the *UCI repository* and the results show that *MixMF* is the best performing method.

1 Introduction

The most important property of an artificial neural network is the ability to correctly respond to inputs which were not used in the learning set. One technique commonly used to increase this ability consist on training some Multilayer Feedforward networks with different weights initialization. Then the mean of the outputs are applied to get the output of the ensemble. This method, known as *Simple Ensemble* or *Basic Ensemble Machine*, increases the generalization capability [1,2,3]. The diagram of a *Simple Ensemble* is shown in Figure 1.

Although most of the methods to create a Multi-Net System are based on the *ensemble approach* [4,5], we also focus on a *Mixture of Neural Networks* (*MixNN*) because is one of the most known *modular* methods and we think it could be improved.

Mixture of Neural Networks is a method to build a Modular Network which consist on training different neural networks with a gating network. The method divides the problem into subproblems, each subproblem tends to be solved by one network. The gating network is used to combine the outputs of the neural networks to get the final output.

The original *Mixture of Neural Networks* (*MixNN*) [6] is based on a quite simple neural network architecture. We think that *MixNN* could perform better if the method was based on *Multilayer Feedforward* networks. In this paper we present a *Mixture of Multilayer Feedforward Networks* (*MixMF*) which is a modular approach based on Multilayer Feedforward networks trained with Backpropagation.

In section 2 we describe the basic concepts of the MixMF neural network model. We have built multiple classification systems of 3, 9, 20 and 40 networks on six databases

D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNCS 4113, pp. 488–493, 2006.

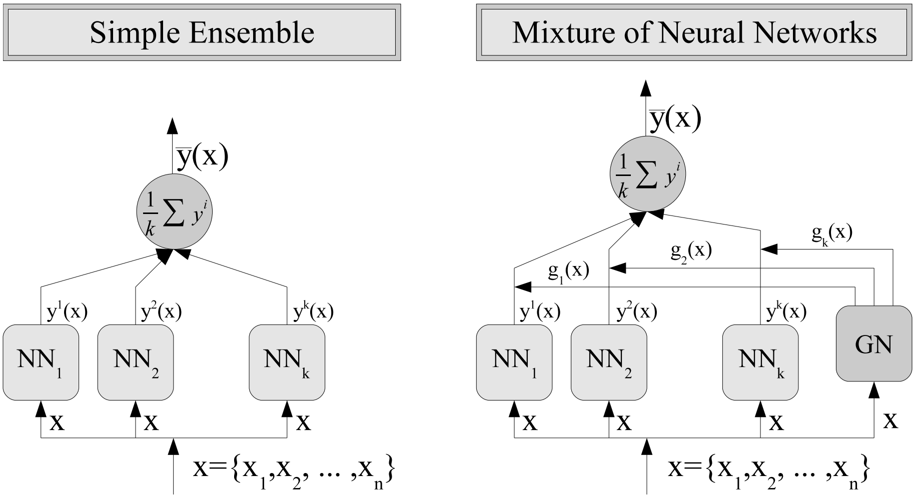

Fig. 1. Simple Ensemble and Mixture of Neural Networks diagrams

from the *UCI repository* to test the performance of *Simple Ensemble*, *MixNN* and *MixMF*. The results we have obtained on these six databases are in subsection 3.1. We have also calculated the mean percentage of error reduction, PER, of the different networks models to compare the methods, these results appear in subsetion 3.2.

2 Theory

In this section, we describe the *Mixture of Neural Networks* and the *Mixture of Multilayer Feedforward*

2.1 Mixture of Neural Networks

The *Mixture of Neural Networks* (*MixNN*) is a method to build a Multi-Net System based on the *Modular approach*. It consist on training *k* expert networks and a gating network. The input *x* is applied to the expert networks and the gating network. The modular network output is:

$$\overline{y}_{class} = \sum_{net=1}^{k} y_{class}^{net} \cdot g_{net} \tag{1}$$

Where the output of the expert networks is described in equation 2 and the output of the gating networks is described in equation 3

$$y_{class}^{net} = x^T \cdot w_{class}^{net} \tag{2}$$

$$g_{net} = \frac{\exp\left(x^T \cdot a_{net}\right)}{\sum_{j=1}^{k} \exp\left(x^T \cdot a_j\right)} \tag{3}$$

To adapt the weights of the expert networks and the gating network, we have used the objective function described in equation 4.

$$L = \log\left(\sum_{net=1}^{k} g_{net} \cdot \exp\left(-\frac{1}{2} \cdot \left\|d - y^{net}\right\|^2\right)\right) \tag{4}$$

The equations used to adapt the weights of the expert networks w and the gating network a are:

$$w_{class}^{net}(ite+1) = w_{class}^{net}(ite) + \eta \cdot h_{net} \cdot \left(d - y^{net}\right) \cdot x \tag{5}$$

$$a_{net}(ite+1) = a_{net}(ite) + \eta \cdot h_{net} \cdot (h_{net} - g_{net}) \cdot x \tag{6}$$

where:

$$h_{net} = \frac{g_{net} \cdot \left(-\frac{1}{2}\left|d - y^{net}\right|^2\right)}{\sum_{j=1}^{k}\left(g_j \cdot \left(-\frac{1}{2}\left|d - y^j\right|^2\right)\right)} \tag{7}$$

2.2 Mixture of Multilayer Feedforward Networks

Mixture of Multilayer Feedforward Networks (MixNF) is a method to build a modular network. *MixMF* is an approach of *MixNN* where the expert networks are *Multilayer Feedforward* networks with one hidden layer and threshold nodes. *Multilayer Feedforward* networks are more accurate than the expert networks used in *MixNN* but the training process is slower. In [7] it can be found that MLP network with one hidden layer and threshold nodes can solve any function with a specified precision.

In order to adapt the weights of the expert networks and the gating network we have used the objective function described in equation 4. The equations used to adapt the input layer weights of the expert networks wi, the hidden layer weights of the expert networks wh and the gating network a the following ones:

$$wh_{j,k}^{net}(ite+1) = wh_{j,k}^{net}(ite) + \eta \cdot h_{net} \cdot \delta_k \cdot ho_j \tag{8}$$

$$wi_{i,j}^{net}(ite+1) = wi_{i,j}^{net}(ite) + \eta \cdot h_{net} \cdot \delta_j^{'} \cdot x_i \tag{9}$$

$$a_{net}(n+1) = a_{net}(n) + \eta \cdot h_{net} \cdot (h_{net} - g_{net}) \cdot x \tag{10}$$

where:

$$h_{net} = \frac{g_{net} \cdot \left(-\frac{1}{2}\left|d - y^{net}\right|^2\right)}{\sum_{j=1}^{k}\left(g_j \cdot \left(-\frac{1}{2}\left|d - y^j\right|^2\right)\right)} \tag{11}$$

$$\delta_k = (d_k - y_k) \cdot (1 - y_k) \cdot (y_k) \tag{12}$$

$$\delta_j^{'} = ho_j \cdot (1 - ho_j) \cdot \sum_{h=1}^{m} \delta_h \cdot wh_{j,h} \tag{13}$$

3 Experimental Testing

In this section, we describe the experimental setup, the datasets we have used in our experiments and we show the results we have obtained. Finally we compare the results we have obtained with *Simple Ensemble*, *MixNN* and *MixMF* on the different datasets.

For this reason we have trained multiple classification systems of 3, 9, 20 and 40 MF networks with *Simple Ensemble*, *MixNN* and *MixMF* on eight different classification problems from the *UCI repository of machine learning databases* [8] to test the performance of methods. The databases we have used are: Cylinder Bands Database (band), Australian Credit Approval (cred), Solar Flare Database (flare), Glass Identification Database (glas), The monk's problem 1 (mok1), Congressional Voting Records Database (vote), Wisconsin Breast Cancer Database (wdbc). In addition, we repeated ten times the whole learning process, using different partitions of data in training, validation and test sets. With this procedure we can obtain a mean performance of the ensemble for each database and an error in the performance calculated by standard error theory.

3.1 Results

In this subsection we present the experimental results we have obtained with the ensembles of MF networks trained with *Simple Ensemble* and the modular networks.

Table 1 shows the results we have obtained with ensembles of 3, 9, 20 and 40 networks trained with *Simple Ensemble*. Table 2 shows the results we have obtained with a modular network of 3, 9, 20 and 40 networks trained with *Mixture of Neural Networks* and *Mixture of Multilayer Feedforward Networks*.

Table 1. Simple Ensemble results

Database	3 Nets	9 Nets	20 Nets	40 Nets
band	73.5±1.2	72.9±1.5	73.8±1.3	73.8±1.3
cred	86.5±0.7	86.4±0.7	86.6±0.7	86.5±0.7
flare	81.8±0.5	81.6±0.4	81.5±0.5	81.6±0.5
glas	94±0.8	94±0.7	94±0.7	94.2±0.6
mok1	98.3±0.9	98.8±0.8	98.3±0.9	98.3±0.9
survi	74.3±1.3	74.2±1.3	74.3±1.3	74.3±1.3
vote	95.6±0.5	95.6±0.5	95.6±0.5	95.6±0.5
wdbc	96.9±0.5	96.9±0.5	96.9±0.5	96.9±0.5

3.2 Interpretations of Results

Comparing the results showed in tables 1 and 2 we can see that the improvement in performance using our method depends on the database and the number of networks used in the ensemble. We can also see that, in general, *Mixture of Multilayer Feedforward Networks* is better than *Mixture of Neural Networks*.

We have also calculated the percentage of error reduction (PER) of the ensembles with respect to a single network to get a general value for the comparison among the methods we have studied. We have used equation 14 to calculate the ensemble PER value.

Table 2. Mixture Methods PER

	MixNN				MixMF			
DB	**3 Nets**	**9 Nets**	**20 Nets**	**40 Nets**	**3 Nets**	**9 Nets**	**20 Nets**	**40 Nets**
band	72.7±2.2	74.4±1.3	74±1.9	75.5±1.3	75.5±1.9	74.2±2	74.7±1.7	73.8±1.6
cred	86.8±0.5	86.9±0.5	86.5±0.6	86±0.5	85.9±0.5	86.7±0.7	86.5±0.7	86.8±0.5
flare	81.5±0.5	81.7±0.5	81.7±0.6	81.8±0.6	82.1±0.6	81.9±0.6	81.6±0.6	81.7±0.6
glas	89.4±1	91.2±1.1	90.2±1.3	91±1.1	94.6±1	94.6±1.2	94.2±1.3	95±1.2
mok1	87.8±2.2	93.6±2.6	93.6±2.1	93.9±2.5	99.3±0.8	99.3±0.8	98.8±0.9	100±0
survi	72.3±1.2	72.6±0.9	73.8±0.9	73.6±1.2	74.6±1.3	74.9±1.2	74.6±1.1	75.1±1.2
vote	95±1.2	96.1±0.6	96.1±0.6	96.5±0.7	96.1±0.6	96.1±0.6	96.1±0.6	95.8±0.6
wdbc	94.7±0.5	94.9±0.4	95.1±0.6	94.6±0.5	96.9±0.5	96.9±0.5	96.9±0.5	96.9±0.5

$$PER = 100 \cdot \frac{Error_{singlenetwork} - Error_{ensemble}}{Error_{singlenetwork}} \tag{14}$$

The PER value ranges from 0%, where there is no improvement by the use of a particular ensemble method with respect to a single network, to 100%. There can also be negative values, which means that the performance of the ensemble is worse than the performance of the single network. This new measurement is relative and can be used to compare more clearly the different methods.

Moreover we have calculated the mean increase of performance and the mean percentage of error reduction across all databases with respect to the single network. Table 3 shows these general measurements.

Table 3. Global Measures

	Mean PER				Mean Increase of Performance			
DB	**3 Nets**	**9 Nets**	**20 Nets**	**40 Nets**	**3 Nets**	**9 Nets**	**20 Nets**	**40 Nets**
Simple Ens.)	20.96	20.63	20.98	21.1	5.17	5.1	5.19	5.21
MixNN	-0.12	8.58	8.92	8.51	3.67	3.99	3.93	4.17
MixMF	23.77	23.93	23.11	23.43	5.62	5.63	5.48	5.69

According to this global measurement *MixMF* is the best performing method. The highest difference between *MixMF* and *Simple Ensemble* is in the 9-network ensemble where the mean PER increase is 3.3%. The highest difference between original *MixMF* and *MixNN* is in the 3-network ensemble where the mean PER increase is 23.90%.

4 Conclusions

In this paper we have presented *Mixture of Multilayer Feedforward Networks*, a modular method based on *Mixture of Neural Networks* and *Multilayer Feedforward*. We have trained Multiple Classification Systems of 3, 9, 20 and 40 networks with *Simple Ensemble*, *MixNN* and *MMixMF* to cover a wide spectrum of the number of networks in the classification system. The results showed that in general the improvement by the use of *MixMF* depends on the database.

Finally, we have obtained the mean percentage of error reduction across all databases. According to the results of this measurement *MixMF* performs better than *MixNN*. In general, *MixMF* is the best performing method.

We can conclude that the *Mixture of Neural Networks* variation we have presented in this paper uses a better expert networks so the performance of the final classification system is, in general, better.

Acknowledgments

This research was supported by project *P1·1B2004-03* of Universitat Jaume I - Bancaja in Castellón de la Plana, Spain.

References

1. Tumer, K., Ghosh, J.: Error Correlation and Error Reduction in Ensemble Classifiers. Connection Science vol.8(3-4) (1996) 385–403
2. Raviv, Y., Intratorr, N.: Bootstrapping with Noise: An Effective Regularization Technique. Connection Science, Special issue on Combining Estimators vol.8 (1996) 356–372
3. Freund, Y., Schapire, R.E.: Experiments with A New Boosting Algorithm. In: International Conference on Machine Learning. (1996) 148–156
4. Hernandez-Espinosa, C., Fernandez-Redondo, M., Torres-Sospedra, J.: Ensembles of Multilayer Feedforward for Classification Problems. In: Neural Information Processing, ICONIP 2004. Volume 3316 of Lecture Notes in Computer Science. (2005) 744–749
5. Hernandez-Espinosa, C., Torres-Sospedra, J., Fernandez-Redondo, M.: New Experiments on Ensembles of Multilayer Feedforward for Classification Problems. In: Proceedings of International Conference on Neural Networks, IJCNN 2005, Montreal, Canada. (2005) 1120–1124
6. Sharkey, A.J., ed.: Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems. (1999)
7. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
8. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases (1998)