

The Mixture of Neural Networks as Ensemble Combiner

Mercedes Fernández-Redondo¹, Joaquín Torres-Sospedra¹,
and Carlos Hernández-Espinosa¹

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

Abstract. In this paper we propose two new ensemble combiners based on the *Mixture of Neural Networks* model. In our experiments, we have applied two different network architectures on the methods based on the *Mixture of Neural Networks*: the *Basic Network (BN)* and the *Multilayer Feedforward Network (MF)*. Moreover, we have used ensembles of *MF* networks previously trained with *Simple Ensemble* to test the performance of the combiners we propose. Finally, we compare the *mixture combiners* proposed with three different mixture models and other traditional combiners. The results show that the mixture combiners proposed are the best way to build Multi-net systems among the methods studied in the paper in general.

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 consists on training some Multilayer Feedforward networks with different weights initialization. Then the mean of the outputs is applied to get the final output. This method, known as *Simple Ensemble* (Fig.1) increases the generalization capability with respect to a single network [17].

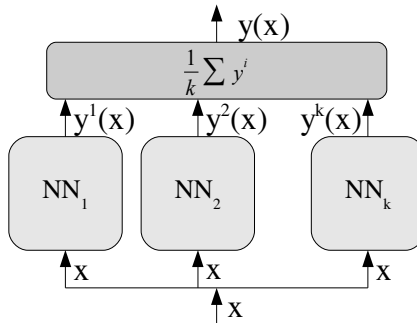


Fig. 1. Simple Ensemble diagram

The *Mixture of Neural Networks* (Fig.2) is a modular model to build a Multi-Net system which consist on training different neural networks, also called expert networks or experts, with a gating network. The method divides the problem into subproblems, each subproblem tends to be solved by one expert. The gating network is used to weight and combine the outputs of the experts to get the final output of the system.

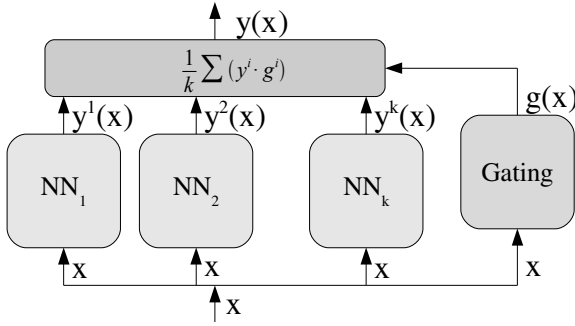


Fig. 2. Mixture of Neural Networks diagram

In a previous paper [16], we analysed the original *Mixture of Neural Networks* model [13] which uses the *Basic Network* as expert and gating networks. Moreover, we successfully proposed a model based on the use of the *Multilayer Feedforward* network.

In this paper we propose two ensemble combiners based on that idea. We think that the the accuracy of an ensemble of *MF* networks can be increased by applying the gating network of the mixture model to combine the ensemble.

In our experiments, we have applied the proposed combiners to ensembles previously trained with *Simple Ensemble (SE)*. We have named them, *Mix-SE-BN* and *Mix-SE-MF*, depending on the method used to train the ensemble (*SE* in this case) and the architecture applied to the gating net (*BN* or *MF*).

Finally, we compare the accuracy of the *mixture models* and the *mixture combiners* on *Simple Ensemble*. Moreover, we also compare the general results of the *mixture combiners* with seven traditional combiners also applied to the ensembles previously trained with *Simple Ensemble*.

This paper is organized as follows. Firstly, some theoretical concepts are briefly reviewed in section 2. Then, the eight databases used and the experimental setup are described in section 3. Finally, the experimental results and their discussion are in section 4.

2 Theory

2.1 Network Architectures

The Basic Network. The *Basic Network* (Figure 3) consists of two layers of neurons that apply the identity function. This network can only solve linear problems [10].

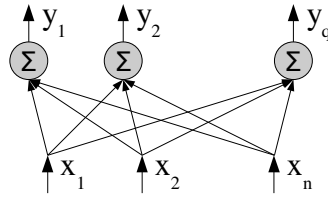


Fig. 3. Basic Network Structure

The Multilayer Feedforward network. The *Multilayer Feedforward* architecture (Figure 4) is the most known network architecture. This kind of networks consists of three layers of computational units. The neurons of the first layer apply the identity function whereas the neurons of the second and third layer apply the sigmoid function. This kind of networks can approximate any function with a specified precision [1,10].

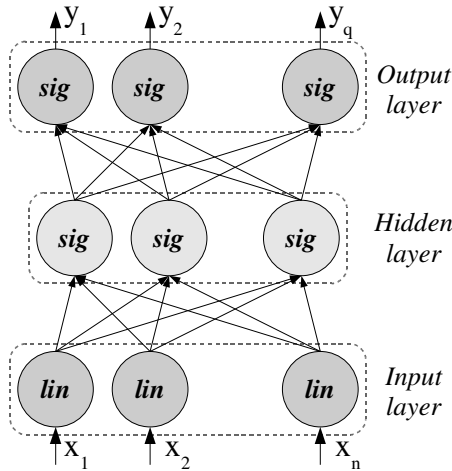


Fig. 4. Multilayer Feedforward Structure

2.2 Simple Ensemble

A simple ensemble can be constructed by training different networks with the same training set, but with different random initialization [3].

Algorithm 1. Simple Ensemble $\{T, V\}$

```

for  $i = 1$  to  $k$  do
    Traditional Network Training  $\{T, V\}$ 
end for
Save Ensemble Configuration
    
```

Algorithm 2. Traditional Network Training $\{T, V\}$

```

Set initial weights randomly
for  $i = 1$  to  $iterations$  do
    Train the network on the training set  $T$ 
    Calculate  $MSE$  over validation set  $V$ 
    Save epoch weights and calculated  $MSE$ 
end for
Select epoch with minimum  $MSE$ 
Assign best epoch configuration to the network
Save network configuration

```

In our experiments we have trained the networks for few iterations. In each iteration, the weights have been adapted with *Back-Propagation* over the training set. At the end of each iteration the Mean Square Error (MSE) has been calculated over the Validation set. When the learning process has finished, we assign the weights of the iteration with minimum MSE to the final network. For this reason the original learning set L is divided into two subsets: The Training set T and the Validation set V .

2.3 Mixture of Neural Networks

The *Mixture of Neural Networks* is a Multi-net system which consists on training k expert networks along with a gating network that weights the output of the experts. In the original version of the method [7] the *Basic Network* was applied as expert and gating networks. In the version *Mixture of Multilayer Feedforward Networks* [16], the *Multilayer feedforward* network was applied as expert and gating networks. In both cases, the objective function L , eq. 1, was applied to adapt the weights of the networks.

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

Where d refers to the desired output or target.

In reference [16] we can find the basic description of the training algorithm carried out when the *Basic Network* or the *Multilayer Feedforward* network are applied.

Finally, in the experiments carried out in the current paper, we have added a new method based in the networks previously described. We have build a modular system with the *Mixture of Neural Networks* where the *Multilayer Feedforward* network has been applied to the experts and the *Basic Network* has been applied to the gating network.

The basic algorithm carried out to build a Multi-net system with the *Mixture of Neural Networks* is described in the following algorithm.

Algorithm 3. Mixture of Neural Networks

```

Random initialization of networks
for  $ite = 1$  to  $iterations$  do
  for each pattern from training set do
    for  $net = 1$  to  $k$  do
      Adapt expert weights
    end for
    Adapt gating weights
  end for
  Calculate  $L_{ite}$  over Validation set
  Save weights
end for
Select iteration with maximum  $L$  (best iteration)
Set best iteration weights to network
Save final configuration

```

Resuming, we have applied three versions of the *Mixture of Neural Networks* model using the *Basic Network* and the *Multilayer Feedforward*. These *Mixture models* are:

- *Mix-BN-BN*: *BN* as experts and gating net.
- *Mix-BN-MF*: *MF* as experts and *BN* as gating net.
- *Mix-MF-MF*: *MF* as experts and gating net.

2.4 Mixture as Combiner

The *Mixture of Neural Networks* approach can be applied as ensemble combiner if the weights of the ensembles are assigned to the experts and they are keep unchanged during the training process. With this simple procedure the *Mixture of Neural Networks* can be applied to combine the networks of ensemble of neural networks.

In this paper, the *mixture model* is modified in order to get the *mixture combiner*, an efficient ensemble combiner. The *mixture combiner* is described in algorithm 4.

Algorithm 4. Mixture Combiner

```

Assign ensemble weights to experts
Random initialization of gating network weights
for  $ite = 1$  to  $iterations$  do
  for each pattern from training set do
    Adapt gating weights
  end for
  Calculate  $L_{ite}$  over Validation set
  Save weights
end for
Select iteration with maximum  $L$  (best iteration)
Set best iteration weights to network
Save final configuration

```

In our experiments, we have only used ensembles previously trained with *Simple Ensemble (SE)*. Moreover, we have applied to the gating network the two network architectures previously described so two different new *mixture combiners* are proposed in this paper. These proposed combiners are:

- *Mix-SE-BN*: *BN* as gating net to combine a *SE*.
- *Mix-SE-MF*: *MF* as gating net to combine a *SE*.

3 Experimental Testing

In our experiments we have trained Multi-net systems of 3, 9, 20 and 40 MF experts with *Simple Ensemble* and the three different *Mixture models* on the eight problems from the *UCI* repository of machine learning [12]. Moreover, we have applied two new combiners based on the *Mixture of Neural Networks* to ensembles previously trained with *Simple Ensemble*.

Finally, we have repeated ten times the whole learning process in order to get a mean performance of the ensemble for each database and an error in the performance calculated by standard error theory.

3.1 Datasets

We have used the following eight different classification problems from the *UCI repository of machine learning databases* [12] to test the performance of methods: *Balance Scale Database* (*bala*), *Dermatology Database* (*derma*), *Ecoli Database* (*ecoli*), *Glass Identification Database* (*glas*), *The Monk's Problem 1* (*mok1*), *The Monk's Problem 2* (*mok2*), *Haberman's Survival Data* (*survi*) and *Congressional Voting Records Database* (*vote*).

The training parameters of the networks have been set after performing a deep *trial and error* procedure on a validation set.

4 Results and Discussion

4.1 Results

In this section we present the experimental results we have obtained with the ensembles of MF networks trained with *Simple Ensemble*, the *Mixture models* and the *Mixture combiners* proposed.

Table 1 shows the results we got with *Simple Ensemble*.

Table 1. Simple Ensemble results

<i>Database</i>	<i>3 Nets</i>	<i>9 Nets</i>	<i>20 Nets</i>	<i>40 Nets</i>
<i>bala</i>	96 ± 0.5	95.8 ± 0.5	95.8 ± 0.6	95.9 ± 0.5
<i>derma</i>	97.2 ± 0.7	97.5 ± 0.7	97.3 ± 0.7	97.6 ± 0.7
<i>ecoli</i>	86.6 ± 0.8	86.9 ± 0.8	86.9 ± 0.8	86.9 ± 0.7
<i>glas</i>	94 ± 0.8	94 ± 0.7	94 ± 0.7	94.2 ± 0.6
<i>mok1</i>	98.3 ± 0.9	98.8 ± 0.8	98.3 ± 0.9	98.3 ± 0.9
<i>mok2</i>	88 ± 2	90.8 ± 1.8	91.1 ± 1.1	91.1 ± 1.2
<i>survi</i>	74.3 ± 1.3	74.2 ± 1.3	74.3 ± 1.3	74.3 ± 1.3
<i>vote</i>	95.6 ± 0.5	95.6 ± 0.5	95.6 ± 0.5	95.6 ± 0.5

Tables 2-4 show the results of the *Mixture models* reviewed in this paper (*Mix-BN*, *Mix-BN-MF* and *Mix-MF-MF*).

Table 2. Mix-BN-BN results

<i>Database</i>	<i>3 Nets</i>	<i>9 Nets</i>	<i>20 Nets</i>	<i>40 Nets</i>
<i>bala</i>	90.5 ± 0.9	90.2 ± 1	91 ± 0.7	89.8 ± 0.8
<i>derma</i>	96.8 ± 0.7	97 ± 0.6	97.3 ± 0.6	97.2 ± 0.8
<i>ecoli</i>	52.5 ± 1.1	48 ± 2	69.9 ± 1.5	74.7 ± 1.4
<i>glas</i>	89.4 ± 1	91.2 ± 1.1	90.2 ± 1.3	91 ± 1.1
<i>mok1</i>	88 ± 2	94 ± 3	94 ± 2	94 ± 3
<i>mok2</i>	62.1 ± 1.7	67.5 ± 2	66.8 ± 1.6	68 ± 2
<i>survi</i>	72.3 ± 1.2	72.6 ± 0.9	73.8 ± 0.9	73.6 ± 1.2
<i>vote</i>	95 ± 1.2	96.1 ± 0.6	96.1 ± 0.6	96.5 ± 0.7

Table 3. Mix-MF-BN results

<i>Database</i>	<i>3 Nets</i>	<i>9 Nets</i>	<i>20 Nets</i>	<i>40 Nets</i>
<i>bala</i>	94.1 ± 1	93.9 ± 1.2	94.6 ± 1.1	95.2 ± 0.7
<i>derma</i>	97 ± 0.8	97.2 ± 0.8	97 ± 0.7	96.9 ± 0.8
<i>ecoli</i>	85 ± 0.8	86.5 ± 1	85.9 ± 0.7	84.6 ± 1.3
<i>glas</i>	94.6 ± 1	94.6 ± 1.2	94.2 ± 1.3	95 ± 1.2
<i>mok1</i>	99.3 ± 0.8	99.3 ± 0.8	98.8 ± 0.9	100 ± 0
<i>mok2</i>	77 ± 3	77 ± 2	84 ± 1.8	80.3 ± 1.8
<i>survi</i>	74.6 ± 1.3	74.9 ± 1.2	74.6 ± 1.1	75.1 ± 1.2
<i>vote</i>	96.1 ± 0.6	96.1 ± 0.6	96.1 ± 0.6	95.8 ± 0.6

Table 4. Mix-MF-MF results

<i>Database</i>	<i>3 Nets</i>	<i>9 Nets</i>	<i>20 Nets</i>	<i>40 Nets</i>
<i>bala</i>	95.1 ± 0.6	95 ± 1	94.2 ± 0.9	94.9 ± 0.6
<i>derma</i>	97.2 ± 0.7	96.9 ± 0.9	97 ± 0.8	96.3 ± 1.1
<i>ecoli</i>	85.4 ± 0.6	84.3 ± 0.8	86.3 ± 1	86.5 ± 0.8
<i>glas</i>	95.2 ± 0.7	94.6 ± 1	95.2 ± 1	93.8 ± 1
<i>mok1</i>	98.6 ± 0.9	98.3 ± 0.9	99.5 ± 0.5	98.4 ± 0.8
<i>mok2</i>	90.3 ± 1.2	87.3 ± 1.5	88.5 ± 1.5	90.8 ± 1.6
<i>survi</i>	74.6 ± 1.2	74.8 ± 1.4	74.4 ± 1.2	74.3 ± 1.3
<i>vote</i>	96 ± 0.6	96 ± 0.6	96 ± 0.6	96.1 ± 0.6

Table 5. Mix-SE-BN results

<i>Database</i>	<i>3 Nets</i>	<i>9 Nets</i>	<i>20 Nets</i>	<i>40 Nets</i>
<i>bala</i>	96.2 ± 0.6	96.1 ± 0.7	96.5 ± 0.5	96.6 ± 0.5
<i>derma</i>	97 ± 0.7	97.3 ± 0.7	97.3 ± 0.7	97.3 ± 0.7
<i>ecoli</i>	86.2 ± 0.9	87.4 ± 0.8	87.2 ± 0.6	86.9 ± 0.6
<i>glas</i>	93.8 ± 0.6	94.4 ± 0.6	94.4 ± 0.6	94.8 ± 0.7
<i>mok1</i>	98.5 ± 0.8	99.8 ± 0.3	100 ± 0	100 ± 0
<i>mok2</i>	87 ± 2	91.1 ± 1.5	91.8 ± 1.4	91.1 ± 0.9
<i>survi</i>	74.1 ± 1.6	74.8 ± 1.6	74.3 ± 1.2	74.3 ± 1.2
<i>vote</i>	95.6 ± 0.5	95.6 ± 0.5	95.6 ± 0.5	95.6 ± 0.5

Table 6. Mix-SE-MF results

Database	3 Nets	9 Nets	20 Nets	40 Nets
<i>bala</i>	96.2 ± 0.5	96.2 ± 0.5	95.9 ± 0.8	96.2 ± 0.7
<i>derma</i>	97.2 ± 0.7	97.2 ± 0.7	97.3 ± 0.7	97.3 ± 0.7
<i>ecoli</i>	86.5 ± 0.9	87.5 ± 0.7	86.8 ± 0.8	86.9 ± 0.6
<i>glas</i>	94 ± 0.8	94 ± 0.7	94.4 ± 0.7	94.2 ± 0.6
<i>mok1</i>	98.5 ± 0.8	99.8 ± 0.3	100 ± 0	100 ± 0
<i>mok2</i>	88 ± 2	91.5 ± 1.3	91.8 ± 0.9	91.6 ± 1.4
<i>survi</i>	74.3 ± 1.5	74.4 ± 1.5	74.3 ± 1.3	74.3 ± 1.3
<i>vote</i>	95.6 ± 0.5	95.6 ± 0.5	95.6 ± 0.5	95.6 ± 0.5

The results of the *Mixture combiners* proposed (*Mix-SE-BN* and *Mix-SE-MF*) are in tables 5-6.

4.2 General Measurements

We have also calculated the Increase of Performance (*IoP* eq.2) and the Percentage of Error Reduction (*PER* eq.3) of the results with respect to a single network in order to get a general value for the comparison among the studied methods.

The *IoP* value is an absolute measurement that denotes the increase of performance of the ensemble with respect to a single network.

$$IoP = Error_{SingleNet} - Error_{Ensemble} \quad (2)$$

The *PER* value is a relative measurement which ranges from 0%, where there is no improvement by the use of an ensemble method with respect to a single network, to 100%.

$$PER = 100 \cdot \frac{Error_{SingleNet} - Error_{Ensemble}}{Error_{SingleNet}} \quad (3)$$

Table 7. Global Results - Mean Increase of Performance

method	3 nets	9 nets	20 nets	40 nets
<i>simple ensemble</i>	9.17	9.61	9.59	9.67
<i>mix-bn-bn</i>	-1.29	-0.1	2.76	3.56
<i>mix-mf-bn</i>	7.61	7.84	8.58	8.27
<i>mix-mf-mf</i>	9.48	8.81	9.32	9.3
<i>mix-se-bn</i>	9.01	9.98	10.06	10
<i>mix-se-mf</i>	9.14	9.94	9.93	9.94
<i>bayesian</i>	9.2	8.91	8.51	8.37
<i>borda</i>	9.12	9.33	9.34	9.53
<i>choquet</i>	8.99	9.61	—	—
<i>dan</i>	8.44	8.55	8.47	8.52
<i>nash</i>	9.18	9.76	9.73	9.81
<i>w.ave</i>	9.08	9.86	9.84	9.25
<i>zimm</i>	9.18	9.71	8.81	7.34

Table 8. Global Results - Mean Percentage of Error Reduction

<i>method</i>	<i>3 nets</i>	<i>9 nets</i>	<i>20 nets</i>	<i>40 nets</i>
<i>simple ensemble</i>	42.51	44.85	44.11	45.45
<i>mix-bn-bn</i>	-11.85	-6.27	13.29	17.48
<i>mix-mf-bn</i>	36.79	38.54	40.24	38.2
<i>mix-mf-mf</i>	43.39	39.75	42.3	40.2
<i>mix-se-bn</i>	41.36	45.97	46.38	46.22
<i>mix-se-mf</i>	42.41	45.39	45.45	45.65
<i>bayesian</i>	41.18	37.14	32.21	29.2
<i>borda</i>	41.21	42.9	42.39	44.26
<i>choquet</i>	41.53	44.18	–	–
<i>dan</i>	37.44	38.81	37.41	39.27
<i>nash</i>	42.7	45.01	45.34	45.45
<i>w.ave</i>	39.63	44.72	44.75	38.69
<i>zimm</i>	42.92	45.11	38.94	25.92

There can also be negative values in both measurements, *IoP* and *PER*, which means that the performance of the ensemble is worse than the performance of a single net.

Finally, we have calculated the mean *IoP* and the mean *PER* across all databases to get a global measurement to compare the methods presented in the paper. Table 7 shows the results of the mean *IoP* whereas table 8 shows the results of the mean *PER*. In these tables we have also included the general results of seven traditional combiners: *Bayesian Combination* (bayesian) [20,11,6], *Borda Count* (borda) [5], *Choquet Integral* (choquet) [2,4,18], *Dinamically Averaged Networks* (dan) [8], *Nash Vote* (nash) [18,19], *Weighted Average* (w.ave) [9] and *Zimmermann's Operator* (zimm) [18,21].

The results of these seven traditional combiners on ensembles trained with *Simple Ensemble* were published in [14,15]. Moreover, we could not apply the *choquet integral* to 20 and 40 network ensembles due to its complexity.

4.3 Discussion

Although some important conclusions can be derived by analysing the results, this discussion will be focused on the comparison of the combiners proposed with the mixture models and with the seven traditional combiners.

In a previous paper we concluded that the idea of applying a sophisticated weighted average based on the gating network of the mixture model should be seriously considered. As we thought, the results show that the mixture combiners we propose in this paper provides the best overall global results. In fact the best overall results is got by the mixture model composed by a simple ensemble as experts and a basic network as gating network *mix-se-bn*.

We can also see that the accuracy of the combiners proposed increases as the number of expert networks increases even if the performance of the ensemble (expert networks) do not increase. The performance of the 20-network ensemble trained with simple ensemble is lower than the 9-network one.

The combiners proposed are more robust than the traditional combiners. For instance, the *Zimmermann's operator* provides the best mean *PER* for the case of 3-network ensembles but it also provides the worst mean *PER* for the case of 40-network ensembles. The *Nash vote* also provides good results for the case of 3-network but the accuracy of the combiner in the other cases are better than *Simple Ensemble* but worse than the *Mixture Combiners* proposed.

5 Conclusions

In this paper, we have proposed two combiners based on the *Mixture of Neural Networks*. We have applied them to ensembles of *Multilayer Feedforward* networks previously trained with *Simple ensemble*. In the first combiner, *Mix-SE-BN*, we have applied the *Basic Network* as gating network to weight and combine the outputs provided by the networks of the ensemble previously trained with *Simple Ensemble*. In the second one, *Mix-SE-MF*, we have applied the *Multilayer Feedforward* network as gating network to combine the ensemble previously trained with *Simple ensemble*.

In our experiments we have compared the two new *Mixture combiners* with three different *Mixture models* and seven traditional combiners. In the first mixture model, *Mix-BN-BN*, the *Basic Network* is used as expert and gating networks. In the second, *Mix-MF-BN*, the *Multilayer Feedforward* network is used as expert networks whereas the *Basic Network* is used as gating network. In the last one, *Mix-MF-MF*, the *Multilayer Feedforward* network is used as expert and gating networks.

To compare the combiners proposed with the seven traditional combiners, we have used ensembles of 3, 9, 20 and 40 networks previously trained with *Simple Ensemble*. Then, to compare the *Mixture Combiners* with the *Mixture Models*, we have built the *Mixture models* with 3, 9, 20 and 40 experts and a single gating network.

Finally, we have calculated the mean *Increase of Performance* and the mean *Percentage of Error Reduction* with respect to a single *MF* network to compare all the methods.

According the general measurements, the *mixture combiners* on *Simple Ensemble* are the best way to build Multi-Net systems among the models and combiners studied in this paper. In fact, the best results are provided by ensembles of 20 networks trained with *Simple Ensemble* and combined with a *Basic network* as gating network, *Mix-SE-BN*. Moreover, the combiners proposed are more robust than the traditional ones. The *Zimmermann's operator* provides the best and the worst mean *PER* whereas the *Nash vote* provides, in the majority of cases, better results than *Simple Ensemble* but worse than the *mixture combiners* proposed.

Moreover, the mixture models (*Mix-BN-BN*, *Mix-MF-BN* *Mix-MF-MF*) perform worse than *Simple Ensemble* in general. In fact, *Mix-BN-BN* works worse than a single *MF* network in two cases, 3-network and 9-network ensembles. The complexity of the *mixture model* is its main problem, all the networks are trained at the same time and the experts tend to be more cooperative and less competitive. Some authors defend the idea that competitive systems provides good results and are less complex than cooperative systems. The *mixture combiners* are less complex, a previously trained ensemble is assigned to the expert networks and their weights are keep unchanged, only the gating

network is trained so adding a new expert only requires the training of the new expert and the gating net.

We can conclude by remarking that the accuracy of an ensemble of *Multilayer feed-forward* networks can be improved by applying the gating network of the *Mixture of Neural Networks* as ensemble combiner.

References

1. Bishop, C.M.: *Neural Networks for Pattern Recognition*, New York, NY, USA. Oxford University Press, Inc., Oxford (1995)
2. Cho, S.-B., Kim, J.H.: Combining multiple neural networks by fuzzy integral for robust classification. *IEEE Transactions on System, Man, and Cybernetics* 25(2), 380–384 (1995)
3. Fernández-Redondo, M., Hernández-Espinosa, C., Torres-Sospedra, J.: Multilayer feedforward ensembles for classification problems. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) *ICONIP 2004*. LNCS, vol. 3316, pp. 744–749. Springer, Heidelberg (2004)
4. Gader, P.D., Mohamed, M.A., Keller, J.M.: Fusion of handwritten word classifiers. *Pattern Recogn. Lett.* 17(6), 577–584 (1996)
5. Ho, T.K., Hull, J.J., Srihari, S.N.: Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(1), 66–75 (1994)
6. Jacobs, R.A.: Methods for combining experts' probability assessments. *Neural Comput.* 7(5), 867–888 (1995)
7. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. *Neural Computation* 3, 79–87 (1991)
8. Jimenez, D.: Dynamically weighted ensemble neural networks for classification. In: *Proceedings of the 1998 International Joint Conference on Neural Networks, IJCNN 1998*, pp. 753–756 (1998)
9. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. In: Tesauro, G., Touretzky, D., Leen, T. (eds.) *Advances in Neural Information Processing Systems*, vol. 7, pp. 231–238. The MIT Press, Cambridge (1995)
10. Kuncheva, L.I.: *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, Chichester (2004)
11. Lam, L., Suen, C.Y.: Optimal combinations of pattern classifiers. *Pattern Recogn. Lett.* 16(9), 945–954 (1995)
12. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
13. Sharkey, A.J. (ed.): *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*. Springer, Heidelberg (1999)
14. Torres-Sospedra, J., Fernández-Redondo, M., Hernández-Espinosa, C.: A research on combination methods for ensembles of multilayer feedforward. In: *IJCNN 2005 Proceedings*, pp. 1125–1130 (2005)
15. Torres-Sospedra, J., Hernández-Espinosa, C., Fernández-Redondo, M.: Combinación de conjuntos de redes MF. In: *SICO 2005 Proceedings*, pp. 11–18. Thomson (2005)
16. Torres-Sospedra, J., Hernández-Espinosa, C., Fernández-Redondo, M.: Designing a new multilayer feedforward modular network for classification problems. In: *WCCI 2006 proceedings*, pp. 2263–2268 (2006)
17. Tumer, K., Ghosh, J.: Error correlation and error reduction in ensemble classifiers. *Connection Science* 8(3-4), 385–403 (1996)
18. Verikas, A., Lipnickas, A., Malmqvist, K., Bacauskiene, M., Gelzinis, A.: Soft combination of neural classifiers: A comparative study. *Pattern Recognition Letters* 20(4), 429–444 (1999)

19. Wanas, N.M., Kamel, M.S.: Decision fusion in neural network ensembles. In: Proceedings of the 2001 International Joint Conference on Neural Networks, IJCNN 2001, vol. 4, pp. 2952–2957 (2001)
20. Xu, L., Krzyzak, A., Suen, C.Y.: Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics* 22(3), 418–435 (1992)
21. Zimmermann, H.-J., Zysno, P.: Decision and evaluations by hierarchical aggregation of information. *Fuzzy Sets and Systems* 10(3), 243–260 (1984)