

Finding Committee Solutions by Clustering Models in Function Space

Thomas Ragg

Institut für Logik, Komplexität und Deduktionssysteme

Universität Karlsruhe, Germany

email: ragg@ira.uka.de

http://www.ira.uka.de

phone: ++49 721 608 4338

fax: ++49 721 608 4211

September 28, 2001

Abstract

Forming a committee is an approach for integrating several opinions or functions instead of favouring a single one. Selecting and weighting the committee members is done in several ways by different algorithms. Possible solutions to this problem is still the topic of current research. Our starting point is the decomposition of the committee error into a bias- and variance-like term. Two requests can be derived from this equation: Models should on the one hand be regularized properly to reduce the average error. On the other hand they should be as independent as possible (in the mathematical sense) to decrease the committee error.

The first request of regularization can be handled by a Bayesian learning framework. For the second request I want to suggest a new selection method for committee members based on the pairwise stochastic dependence of their output functions, which maximizes the overall independence.

Given these pairwise similarity values the models can be separated in classes by a hierarchical clustering algorithm. From the error decomposition of committees I derive a criterion that allows to find the optimal number of classes, i.e. the optimal stop criteria for the clustering algorithm.

The benefits of the approach are demonstrated for committees of neural networks on a noisy benchmark problem as well as on some problems from the UCI repository.

1 Introduction

If one wants to learn a functional relationship from empirical data, then the goal of training a model is to recognize a structure in the data or an underlying process and to generalize this knowledge to former unknown data points. When estimating a functional relationship we face three basic problems.

1.1 Noisy and finite-sized data sets:

Firstly we have to deal with noisy and finite-sized data sets which is usually done by regularization techniques and/or bootstrapping [Vapnik, 1995, Bishop, 1995]. Vapnik states that the problem of density estimation based on empirical data is ill posed, i.e., small changes in the learning situation can result in a totally different

model; for example little distortions in the target data. The theory of regularization shows that instead of minimizing the difference between the target data and the output of the network, a regularized error function

$$E = E_D + \lambda E_R \tag{1}$$

should be minimized where E_D is the error on the data and λ is a weighting factor. E_R is an additional term that measures the complexity of the model; for example the often used weight decay regularizer in neural network training [Bishop, 1995]. Thus, regularization is not an optional possibility, but a fundamental technique [Ramsay & Silverman, 1997]. One crucial problem is to determine the weighting factor λ . In case of neural networks, its optimal value depends on the size of the network, i.e., the number of weights, the weight initialization, as well as the patterns used for training, and the noise in the data. Often this value is determined by cross-validation which is clearly suboptimal for the reasons just given. Adjusting this value properly has been solved for neural networks by using a Bayesian learning algorithm. It was introduced by MacKay and provides an elegant theory to prevent neural networks from overfitting by determining λ during the training process without the necessity of additional validation data [MacKay, 1992, Bishop, 1995]. Furthermore, the Bayesian framework provides an analytical criterion to compare different models, the so-called model evidence. A discussion of Bayesian learning for neural networks as used in this work can be found in [Ragg & Gutjahr, 1998] or more detailed in [Gutjahr, 1999, Ragg, 2000].

1.2 Feature selection:

Secondly, for many applications we need to encode the problem by features and have to decide which and how many of them to use. Bearing in mind the empty space phenomenon, it is often an advantage to select few features and estimate a non-linear function in a low-dimensional space [Silverman, 1986, Bishop, 1995]. In practical applications only a limited amount of data is available for determining the parameters of a model. The dimensionality of the input vector must be in a sensible relation to the number of data points. By adding more features the information content of the input about the target increases, but at the same time the number of data points per dimension decreases in order to determine the parameters. Silverman gives some values for the number of data points needed to estimate a multivariate normal distribution up to a an error of 10% in the origin, which grow approximately like 4^d , when d is the number of dimensions [Silverman, 1986]. That means that the class of functions in which the solution is searched increases greatly with every additional component. This problem is called the *curse of dimensionality* [White, 1989, Bishop, 1995] or the *empty space phenomenon* [Scott & Thompson, 1983, Silverman, 1986]. A consequence is that renunciation of supplementary information leads to a more precise approximation of the underlying process in a low-dimensional input space. If one's problem solution is designed from the beginning as a committee, it is not necessary to discard input features and therefore all the information can be used for the overall solution. In this way one can train models which use different features, i.e., they have different viewpoints of the problem. Therefore, every individual model has a small input space, i.e., the problem caused by using many input features is avoided. However, the committee can utilize the information of the entire input space through the combination of its members.

1.3 Model selection:

Thirdly, if we have trained several models, we are left with the problem of model selection. It is common practice to train several networks and then select the best one according to the performance on an independent validation set. This procedure has the disadvantage of introducing an additional dependency on a fixed sample of data bearing the danger of overfitting if it is used iteratively. Forming a committee of networks is a promising approach to overcome these drawbacks and to avoid favouring one model while discarding all others. This is sensible if one would like to minimize the insecurities during the learning process. This can be seen directly from the inequation

$$\frac{1}{L}E_{AV} \leq E_{COM} \leq E_{AV} \quad (2)$$

which says, that the committee error E_{COM} is always smaller than the average error E_{AV} of several networks y_i , when y_{COM} is defined as $y_{COM}(x) := \frac{1}{L} \sum_{i=1}^L y_i(x)$. The right inequation follows directly by applying Jensen's inequality [Bishop, 1995, Henze, 1997] to a convex error function. The left inequation, i.e., the rather dramatic possibility of error reduction by a factor L follows under the assumption that the deviations $\epsilon_i(x)$ of the committee members $y_i(x)$ from the target function $h(x)$ are uncorrelated and have zero mean, and justifies the efforts invested in these kind of algorithms. Thus, several methods of forming committees were suggested in recent years. An overview is given in the book *Combining Artificial Neural Nets* [Sharkey, 1999]. These algorithms differ in the way how the committee members are selected or weighted within the committee.

Up to now, there is no algorithm which tries to minimize both the average model error and the independence of committee members systematically without resorting to cross-validation techniques or by neglecting the importance of regularization. It's obvious that averaging reduces the generalization error, when arbitrary models can be in the committee, but does it also reduce the generalization error compared to a proper regularized model?

In the following I want to show that this is possible, and derive a method for selecting committee members to achieve a maximal performance under certain constraints, e.g. all models receive the same weight in the committee. For clarity I will not write the dependency on the input vector \mathbf{x} in the following.

2 Committee error decomposition

The error of a committee can be decomposed into the sum of two terms similar to the bias-variance decomposition, which gives us a further insight into the reasons for better generalization capabilities. The equation provides a connection between the average error of the network and the error of the single networks through the expectation value of the committee error [Krogh & Vedelsby, 1995, Bishop, 1995]:

$$\begin{aligned} \mathcal{E} [(y_{COM} - h)^2] &= \frac{1}{L} \sum_{i=1}^L \mathcal{E} [(y_i - h)^2] - \\ &\quad \frac{1}{L} \sum_{i=1}^L \mathcal{E} [(y_i - y_{COM})^2]. \end{aligned} \quad (3)$$

Since the first term depends on the generalization errors of the single networks we can conclude that the average value should be as small as possible, which can be

ensured if the models are regularized. The second term measures the spread of predictions of the single networks to the committee prediction itself. If we have a set of trained networks we should prefer that networks to form a committee, which maximize the second term.

Bagging for example averages over all models and does not try to maximize the variance of the models [Breiman, 1996]. *Boosting* adapts the training set for each model depending on the error of the previously trained model [Freund & Schapire, 1996] and tends to overfit for noisy data [Rätsch *et al.*, 1998]. Furthermore, neither boosting nor bagging expect the models to be regularized, which might cause the first term to be larger than necessary. Other methods make use of validation sets to select the committee members and determine the weights [Hashem, 1999] or optimize solely the second term of eq. 3 which allows causes the single models to overfit strongly [Rosen, 1996].

That is, our goal must be to find a subset of networks, such that the first term does not increase while at the same time the second term is maximal large. We cannot measure the average generalization error without an additional validation set, but we can expect the value to stabilize if the number L of networks in our subset increases, and even more, if the models are regularized properly.

2.1 Committee member selection:

To find a criterion that maximizes the second term of equation (3), we can resolve the quadratic form to

$$\frac{1}{L} \sum_{i=1}^L \mathcal{E}[y_i^2] - \frac{2}{L} \sum_{i=1}^L \mathcal{E}[y_{COM} y_i] + \frac{1}{L} \sum_{i=1}^L \mathcal{E}[y_{COM}^2] \quad (4)$$

and then further evaluate the second and third term. The second term becomes

$$\begin{aligned} -\frac{2}{L} \sum_{i=1}^L \mathcal{E}[y_{COM} y_i] &= -\frac{2}{L} \sum_{i=1}^L \mathcal{E} \left[\frac{1}{L} \sum_{j=1}^L y_j y_i \right] \\ &= -\frac{2}{L^2} \sum_{i=1}^L \sum_{j=1}^L \mathcal{E}[y_j y_i] \\ &= -\frac{2}{L^2} \left(2 \sum_{i=1}^{L-1} \sum_{j=i+1}^L \mathcal{E}[y_j y_i] + \sum_{i=1}^L \mathcal{E}[y_i^2] \right) \\ &= -\frac{4}{L^2} \sum_{i=1}^{L-1} \sum_{j=i+1}^L \mathcal{E}[y_j y_i] - \frac{2}{L^2} \sum_{i=1}^L \mathcal{E}[y_i^2] \end{aligned} \quad (5)$$

where we summarized over the diagonal elements and used $y_j y_i = y_i y_j$. Together with the definition of y_{COM} the third term can be transformed to

$$\begin{aligned} \mathcal{E}[y_{COM}^2] &= \mathcal{E} \left[\left(\frac{1}{L} \sum_{i=1}^L y_i \right) \left(\frac{1}{L} \sum_{j=1}^L y_j \right) \right] \\ &= \mathcal{E} \left[\frac{1}{L^2} \left(\sum_{i=1}^L y_i^2 + 2 \sum_{i=1}^{L-1} \sum_{j=i+1}^L y_i y_j \right) \right] \end{aligned}$$

$$= \frac{1}{L^2} \sum_{i=1}^L \mathcal{E}[y_i^2] + \frac{2}{L^2} \sum_{i=1}^{L-1} \sum_{j=i+1}^L \mathcal{E}[y_i y_j] \quad (6)$$

Substituting these two expressions into equation (4) and taking $\mathcal{E}[XY] = \mathcal{E}[X]\mathcal{E}[Y] + Cov(X, Y)$ into account, leaves us with

$$\begin{aligned} & \frac{1}{L} \sum_{i=1}^L \mathcal{E}[y_i^2] - \frac{2}{L^2} \sum_{i=1}^{L-1} \sum_{j=i+1}^L \mathcal{E}[y_j y_i] - \frac{1}{L^2} \sum_{i=1}^L \mathcal{E}[y_i^2] \\ &= \frac{L-1}{L^2} \sum_{i=1}^L \mathcal{E}[y_i^2] - \frac{2}{L^2} \sum_{i=1}^{L-1} \sum_{j=i+1}^L \mathcal{E}[y_j y_i] \\ &= \frac{L-1}{L^2} \sum_{i=1}^L (\mathcal{E}[y_i])^2 + \frac{L-1}{L^2} \sum_{i=1}^L Cov(y_i, y_i) - \frac{2}{L^2} \cdot \\ & \quad \sum_{i=1}^{L-1} \sum_{j=i+1}^L \mathcal{E}[y_j] \mathcal{E}[y_i] - \frac{2}{L^2} \sum_{i=1}^{L-1} \sum_{j=i+1}^L Cov(y_i, y_j) \end{aligned} \quad (7)$$

The expectation values of y_i depend upon the data which was used to train the specific model, since the mean value \bar{y}_i after training should be equal to the mean value of the target data [Bishop, 1995]. Thus, the first and third term will be nearly constant. Since $Cov(X, X) = V(X)$ the variance of the network output functions should be maximal, while the last term should be as small as possible. The value of $Cov(y_j, y_i)$ depends on the stochastic dependence of y_j and y_i . The covariance of two random variables, $Cov(X, Y)$, is 0 when they are independent and it is maximal if $X = Y$ [Berger, 1980, Henze, 1997]. Thus it is possible to optimize equation (7) by minimizing a sum over stochastic dependencies. A measure for the stochastic dependence between two variables is their mutual information $I(X, Y)$ [Cover & Thomas, 1991]. An estimation procedure for the mutual information will be defined below. Our goal is then to find a subset of networks such that

$$\frac{L-1}{L^2} \sum_{i=1}^L I(y_i, y_i) - \frac{2}{L^2} \sum_{i=1}^{L-1} \sum_{j=i+1}^L I(y_i, y_j) \longrightarrow \max \quad (8)$$

It is already for small L impossible to compute the value of the criterion for all $\binom{L}{k}$ combinations for all k . A straightforward possibility is to use a cluster algorithm to group the networks and choose from each cluster a representative, i.e., the network with the highest model evidence.

2.2 Clustering neural networks:

The mutual information $I(X; Y)$ of two random variables X, Y is defined as the Kullback-Leibler distance between the joint distribution $p(x, y)$ and the product $p(x)p(y)$.

$$I(X; Y) = \int \int p(x, y) \cdot \log \frac{p(x, y)}{p(x)p(y)} \quad (9)$$

and measures the degree of stochastic independence of X and Y . In order to calculate (9), we approximate the three probability distributions $p(x, y)$, $p(x)$ and $p(y)$, where we use a nonparametric density function approximation with a Epanechnikov kernel function K [Silverman, 1986]. If \mathbf{z} is a d -dimensional vector, then

$$K(\mathbf{z}) = (3/4)^d (1 - (z_1^2 + z_2^2 + \dots + z_d^2)) \quad (10)$$

if $\|\mathbf{z}\|_2 < 1$ and 0 otherwise. The normalizing constant $(3/4)^d$ guarantees that $\int K(\mathbf{z})d\mathbf{z} = 1$. Finally, the probability density for a given data point k is estimated as

$$p(\mathbf{z}(k)) = \frac{1}{N \cdot h^d} \sum_{j=1}^N K\left(\frac{1}{h}(\mathbf{z}(k) - \mathbf{z}(j))\right) \quad (11)$$

where N is the number of training patterns and h is the spread of the kernel function. In case of the Epanechnikov kernel a sensible value for h can be determined

$$h_{opt} = \left(\frac{8}{c_d}(d+4)(2\sqrt{\pi})^d\right)^{-(d+4)} \frac{1}{N^{1/d+4}} \quad (12)$$

where N is the number of data points, d the dimension of the vector and c_d is the volume of the unit d -dimensional sphere [Silverman, 1986]. By summarizing equation (11) over all data points we get the desired density estimations.

If we compute the mutual information between pairs of network output functions we can derive a matrix of pairwise similarity values. It is not possible to define a metric based on the Kullback-Leibler divergence, since the triangle property does not hold [Cover & Thomas, 1991]. Clustering neural networks based on the stochastic dependence can be done on basis of this matrix with a hierarchical agglomerative cluster method, e.g. the complete linkage algorithm, which generates homogenous classes [Kaufmann & Pape, 1996]. This is important, if we select a member as representative.

The algorithm starts by putting each network in an own class. In each step two classes are merged. They are determined by computing the minimal similarity between two elements for each pair of classes

$$Sim(G_1, G_2) := \min_{i,j} I(y_i, y_j) \quad \text{with } y_i \in G_1; y_j \in G_2$$

and then choose that pair of classes $\{G_1, G_2\}$ that has maximal similarity. This is done until only one cluster is left. In every step the heterogeneity criterion (8) is computed. After termination of the clustering procedure the number of classes is determined by the maximal value of the heterogeneity criterion as shown in figure ?? . Figure 2 plots the clusters for the benchmark problem from figure 1 in the test error/evidence space. From each of the corresponding clusters one network is selected as a committee member. If a Bayesian framework is used for training then it is sensible to pick the network with the highest evidence from each class [Ragg, 2000].

3 Results

At first, a noisy artificial regression problem serves as benchmark to explore the algorithm presented here in detail. The data for this benchmark was generated by adding Gaussian noise to a sinus function (Figure 1). The training data contains 40 points, which do not cover the complete domain. Note that because of the noise there are several sensible models for the data except the underlying process. Figure 1 shows 4 models, two of them are networks trained with Bayesian learning and give plausible explanations for the data. Thus, even proper regularization gives still a variety of different models.

Our next goal is to demonstrate that forming a committee can further reduce this variance in performance, thus lowering the probability of choosing the 'wrong'

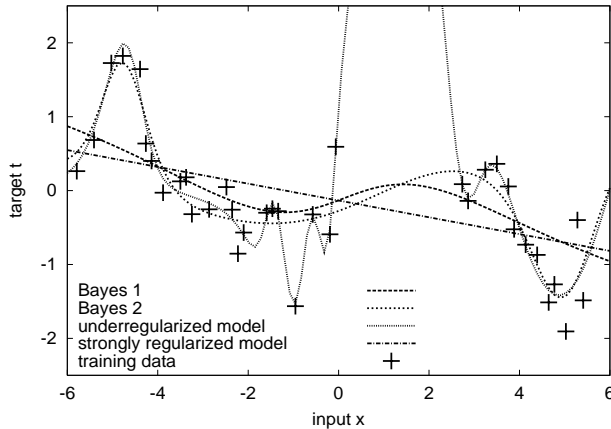


Figure 1: The figure shows the data for the benchmark regression problem. The underlying process is just the sinus function $\sin(x)$. The target data was generated by adding Gaussian noise with variance 0.4 and zero mean. Note that there is an interval, where no data is available. Output functions of 4 models are plotted: An overregularized network, an underregularized network and two proper regularized networks trained with Bayesian learning.

model in a practical application. If we train 50 networks and apply then the cluster algorithm as described above we get 8 classes which are maximal independent. The result of this process is shown in figure 2. Visualization of the functions shows that the classes are sensibly chosen [Ragg, 2000]. Note that it is possible, that some networks with lower generalization performance will be part of the committee, as long as they are independent from others. This is sensible, since they were optimized with Bayesian learning and thus reflect the probability of this parameter vector to be a model for the given data. Committees integrate over the various posterior probabilities to form an overall solution [Bishop, 1995]. The committee error is smaller than the error of almost all of the single networks (dashed line). The test error was computed on 100 data points of the underlying process (without noise).

Figure 3 shows the value of the heterogeneity criterion (8) and the committee error during the clustering process. For each step the members of the committee are determined and the corresponding error is plotted over the number of classes. The committee error reaches its minimum when the heterogeneity criterion is maximal as expected.

Table 1 gives the result of the suggested method for three benchmark problems of the UCI repository, which show that forming a committee of independent networks, which were trained using Bayesian learning is a sensible design strategy.

More results on time series prediction tasks are given in [Ragg, 2000]. They also show, that in case of small training data sets and noisy data the method provides robust and well performing problem solutions.

The proposed method was also tested against other machine learning methods, especially Support Vector Machines. The following table (Table 2) shows the miss-classification rates for all benchmarks of the IDA-repository [Müller *et al.*, 2001]. We first want to note that about half of these public benchmarks are best solved with a linear model. A learning algorithm can in these cases be judged, concerning the fact if the linear solution is found or not. All in all, there are three benchmarks where committees of neural networks constructed by the clustering algorithm perform

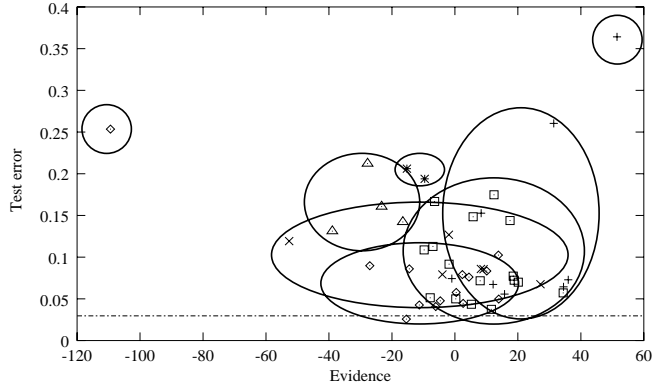


Figure 2: The figure shows the result of the clustering process for 50 networks which were trained with the data from figure 1. The cluster, indicated by ellipses, are shown in the test error-evidence space. Note that this is always a slightly distorted illustration. Networks with similar output function will have a similar test error, but the evidence might vary because its value depends on the overall network complexity. Conversely, different network functions can still have a similar test error and a similar evidence. The dashed line at the bottom shows the error (for 100 test points) of the resulting committee, which is below the error of most of the single networks.

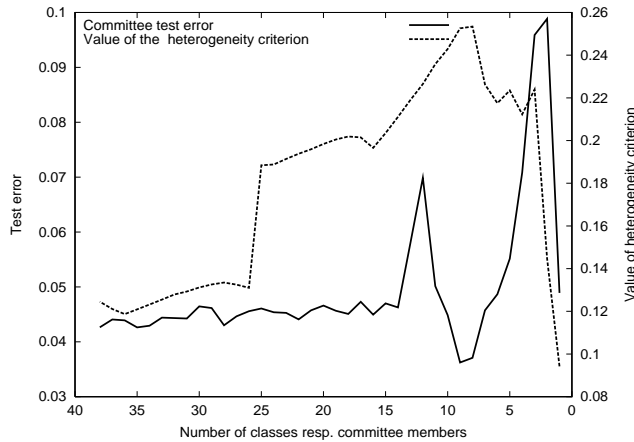


Figure 3: The figure shows how the number of classes is determined for a given set of networks for the benchmark problem from figure 1. The heterogeneity criterion is a curve with a clear maximum, which is reached here for 8 classes. The committee error is relatively constant for a long time and decreases then strongly. Greater changes in one step indicate that two heterogenous classes were merged together.

clearly better, one benchmark where results are worse (ringnorm). The latter might be due to the fact that the data has a radial nature which is difficult to learn with sigmoidal functions but easier with gaussians.

4 Conclusions

In this paper an approach was presented that combines several important steps of neural network design into an optimization procedure. This method was primarily

Problem	Bagging	Boosting	Highest evidence	Clustered networks
Diabetes	79.06 ± 1.9	73.23 ± 4.3	79.14 ± 0.88	79.39 ± 1.0
Cancer	98.73 ± 0.23	96.8 ± 1.77	98.3 ± 0.69	98.72 ± 0.42
Thyroid	97.4 ± 0.1	–	98.1 ± 0.28	98.3 ± 0.32

Table 1: The table shows the correct classification rates for three benchmarks of the UCI-repository. 50 runs were made for each method, with different initializations. Selecting a model with high evidence or forming a committee of independent networks are both sensible strategies for designing a classification system. Thyroid has a 3 class output coding. The Boosting algorithm was not applied to that problem. For thyroid classification 720 patterns were used for training and 6480 patterns for testing. Otherwise the original training and test data sets were used.

Benchmark	Literature	SVM Literature	SVM own results	Committee (Clustering)	linear model
Banana	10.8 - 12.3%	11.5%	11.5%	11.1%	39.5%
B.Cancer	25.8 - 30.4%	26.0%	28.6%	27.27%	26%
Diabetis	23.2 - 26.5%	23.5%	24.3%	24%	24%
+ German	23.6 - 27.5%	23.6%	21%	20.3%	21.3%
Heart	16.0 - 20.3%	16%	17%	18%	20%
+ Image	2.7 - 3.3%	3%	4.1%	1.5%	17.5%
- Ringnorm	1.5 - 1.9%	1.7%	1.8%	5.2%	26.3%
Flare S.	32.4 - 35.7%	32.4%	34.3%	34.7%	34.25%
+ Splice	9.5 - 10.9%	10.9%	12.3%	6%	14.6%
Thyroid	4.2 - 4.8%	4.8%	4%	4%	12%
Titanic	22.4 - 23.3%	22.4%	19.3%	22.0%	21.8%
Twonorm	2.6 - 3.0%	3.0%	2.7%	2.9%	3.6%
Waveform	9.8 - 10.8%	9.9%	10.7%	10.6%	14.3%

Table 2: The table shows the miss rates for all benchmarks of the IDA-repository. The first column gives the values reported in (Müller et. al, 2000), the second column refers to the results for Support Vector Machines. The third column gives our own results for SVMs, where we used a gaussian kernel and determined the width γ and the complexity parameter C by 5-fold cross-validation over a large range of values. The fourth column gives the result for the committee approach described in this paper, and the last column gives as reference value the miss rates of a linear model. The seven benchmarks with names in boldface have a underlying process which is significantly non-linear. The + signs are used for benchmarks where the committee of neural networks was clearly better, while the -- sign shows the benchmark with worse results.

aimed at developing committees of neural networks for tasks where the data is noisy and limited, e.g. as for time series prediction [Ragg *et al.*, 2000], because these task bear the danger of overfitting easily.

The averaging process of forming a committee is always sensible, since it can be shown that the committee error is never larger than the average error of set of trained networks. On the other hand, the committee error can be drastically reduced if the network functions are uncorrelated. Based on the committee error decomposition we derived a criterion that allows us to select a subset of networks which form the best committee. This criterion measures the overall independence of the network functions. By applying a hierarchical cluster algorithm we can group the networks according to their similarities and determine the number of classes such that the criterion (8) reaches its maximal value.

The algorithm presented here can also be used with other function approximators, e.g., Support Vector Machines or Gaussian Processes, as long as they provide

an analytical quality measure like the model evidence in the Bayesian framework.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft (DFG), ME 672/10-1, 'Integrierte Entwicklung von Komitees neuronaler Netze'.

References

- [Berger, 1980] Berger, J. O. *Statistical decision theory and Bayesian analysis*. Springer Verlag, 1980.
- [Bishop, 1995] Bishop, C. M. *Neural Networks for Pattern Recognition*. Oxford Press, 1995.
- [Breiman, 1996] Breiman, L. Bagging predictors. *Machine Learning*, 24:123–140, 1996.
- [Cover & Thomas, 1991] Cover, T. and Thomas, J. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, 1991.
- [Freund & Schapire, 1996] Freund, Y. and Schapire, R. E. Experiments with a new boosting algorithm. In *Proceedings of 13th International Conference on Machine Learning*, pages 148–156, 1996.
- [Gutjahr, 1999] Gutjahr, S. *Optimierung Neuronaler Netze mit der Bayes'schen Methode*. Dissertation, Universität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme, 1999.
- [Hashem, 1999] Hashem, S. Treating Harmful Collinearity in Neural Network Ensembles. In Sharkey, A. J., editor, *Combining Artificial Neural Nets*, pages 101–125. Springer, 1999.
- [Henze, 1997] Henze, N. *Stochastik für Einsteiger*. Vieweg, 1997.
- [Kaufmann & Pape, 1996] Kaufmann, H. and Pape, H. Clusteranalyse. In Fahrmeir, L., Hamerle, A., and Tutz, G., editors, *Multivariate Statistische Verfahren*. de Gruyter, 1996.
- [Krogh & Vedelsby, 1995] Krogh, A. and Vedelsby, J. Neural Network Ensembles, Cross Validation and Active Learning. In D.S. Touretzky, G. Tesauro, T. L., editor, *Advances in Neural Information Processing*, volume 7. MIT press, 1995.
- [MacKay, 1992] MacKay, D. J. C. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.
- [Müller *et al.*, 2001] Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K., and Schölkopf, B. An Introduction to Kernel-Based Learning Algorithms. *IEEE Transactions on Neural Networks*, page to appear, 2001.
- [Ragg & Gutjahr, 1998] Ragg, T. and Gutjahr, S. Optimizing the Evidence – with an application to Time Series Prediction. In *Proceedings of the International Conference on Artificial Neural Networks 1998, Sweden*, Perspectives in Neural Computing, pages 275–280. Springer, 1998.

- [Ragg *et al.*, 2000] Ragg, T., Menzel, W., Baum, W., and Wigbers, M. Predicting Sales Rates for Thousands of Retail Traders. In Tsaprasinos, D., editor, *Proceedings of the International Conference on Engineering Applications of Neural Networks, Kingston, England*, pages 199–206, 2000.
- [Ragg, 2000] Ragg, T. *Problemlösung durch Komitees neuronaler Netze*. Dissertation, Universität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme, 2000.
- [Ramsay & Silverman, 1997] Ramsay, J. O. and Silverman, B. *Functional data analysis*. Springer, 1997.
- [Rätsch *et al.*, 1998] Rätsch, G., Onoda, T., and Müller, K. Soft margins for adaboost. Technical Report NC-TR-1998-021, GMD, Berlin, 1998.
- [Rosen, 1996] Rosen, B. E. Ensemble Learning Using Decorrelated Neural Networks. *Connection Science*, 8:373–384, 1996.
- [Scott & Thompson, 1983] Scott, D. and Thompson, J. Probability density estimation in higher dimensions. In Gentle, J., editor, *Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface*, pages 173–179. 1983.
- [Sharkey, 1999] Sharkey, A. J. Multi-Net Systems. In Sharkey, A. J., editor, *Combining Artificial Neural Nets*, pages 1–30. Springer, 1999.
- [Silverman, 1986] Silverman, B. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.
- [Vapnik, 1995] Vapnik, V. *The Nature of Statistical Learning Theory*. Springer, 1995.
- [White, 1989] White, H. Learning in artificial neural networks: a statistical perspective. *Neural Computation*, 1:425–464, 1989.