# Global and Local Neural Network Ensembles.

A. Sierra, C. Santa Cruz

*Escuela Técnica Superior de Informática. Universidad Autónoma de Madrid.*
*28049 Madrid. Spain.*

Surprisingly simple local learning algorithms are known to outperform many other global non-linear machines. Unfortunately, these algorithms are computationally costly. A means of assembling both learning approaches is proposed in this letter and shown to enhance performance.

*Key words:* Neural Network Ensemble, Global Neural Network, Local Neural Network, Handwritten Digit Recognition.

## 1 Introduction

Human beings are capable of learning from examples. Compelling evidence can be found in a recent paper by Saffran, Aslin and Newport (1996) where 8-month-old infants are found to qualify as excellent statistical learners. The mathematical analysis of this ability is scarcely a 40-year-old discipline that dates back to Rosemblatt's Perceptron (Rosenblatt (1960)). Although much work has been done since then, learning machines are still poor classifiers as compared to human beings. There are a number of reasons which compromise performance and a corresponding number of ways to enhance it.

First of all, feature design is more of an engineering art than a science. Somewhat discouragingly, its relevance in pattern recognition may be dramatic as Simard, Le Cun and Denker (1993) demonstrate with a smart concept of distance that takes invariances into account and turns classification into simple template matching.

Secondly, in a high-dimensional feature space it is hardly possible to have enough examples so as to cover the space properly. As a consequence the boundary among classes may end up being unnecessarily ambiguous. There is an obvious way to address this problem: fill in the gaps. Here the symmetries of the feature space can be used to great advantage: we can either generate extra patterns through transformations of the available ones (Drucker, Schapire and Simard (1993)) or alter the very structure of the algorithm to incorporate invariances. The first way is always feasible but makes training slower. The latter is more subtle and has to be worked out for each algorithm. There are ways in between these two schemes. For example, Schölkopf, Burges and Vapnik (1996) try to enhance Support Vector Machine networks by applying transformations only to certain patterns, the so-called support vectors, instead of transforming the whole training set.

In the third place, every single learning algorithm suffers from limitations. For example, most global optimization methods can not guarantee to reach a global minimum. Different local minima may be viewed as corresponding to different ways of learning the training set. Mainly for this reason, Hansen and Salamon (1990) propose letting an ensemble of different networks decide. This battery of machines can be trained over the same data base or different sets of training examples. Record performance has been attained by a special boosting ensemble of networks trained over an enlarged training set (Drucker, Schapire and Simard (1993)). Another way to escape the global optimization problem consists in training local machines (Vapnik and Bottou (1993)). These devices are very suggestive but also time consuming since local training involves searching for nearest neighbors. Branch and bound methods (Jiang and Zhang (1993)) can help to alleviate this kind of calculation.

In this letter a new step to enhance performance is taken. Guided by the success of local learning machines, and in order to alleviate the computational burden, ensembles of global and local networks have been constructed (Gürgen et al. (1994)). Interestingly enough, the drastically different training nature of these machines makes their combination surprisingly profitable: not only the speed problem is addressed but the joint performance is found to surpass the original ones. The structure of the letter is as follows. Global and local approaches to learning are reviewed in section 2. The procedure that combines both learning methods is introduced in section 3. The efficiency of this construction is exemplified in the last section of the letter.

## 2 Global and Local Learning Machines

Standard practice dictates training learning machines by minimizing the empirical risk, i. e., the mean square error incurred by a trial function. The global minimization of this risk is a very ambitious task. It implies the estimation of a function over the whole feature space. In general, a non-linear function is required for this purpose, as in the back-propagation trained multi-layer perceptron (MLP) and the support vector machine network (SVM). However, the minimization of the risk can also be accomplished locally, in the neighborhood of each test pattern (Vapnik and Bottou (1993)). This problem is much easier to solve and a simple linear function is now enough. Although this approach circumvents non-linearity it is not free from limitations. Training is slow and takes place while performing, as in the regularized local linear regression (RLR) used in this letter. Later a procedure will be introduced to take advantage of both learning approaches simultaneously. We now show the results of a set of experiments conducted on a NIST data base of handwritten digits containing 40,000 training and 10,000 test examples. The resolution of the images is 20x20 pixels. Throughout this article a fast Karhunen Loève expansion (Oja (1983)) in a 40 dimensional feature space is used.

Following Rumelhart, Hinton and Williams (1986) a multi-layer perceptron with sigmoid maps can be easily trained. In this letter a two hidden layer

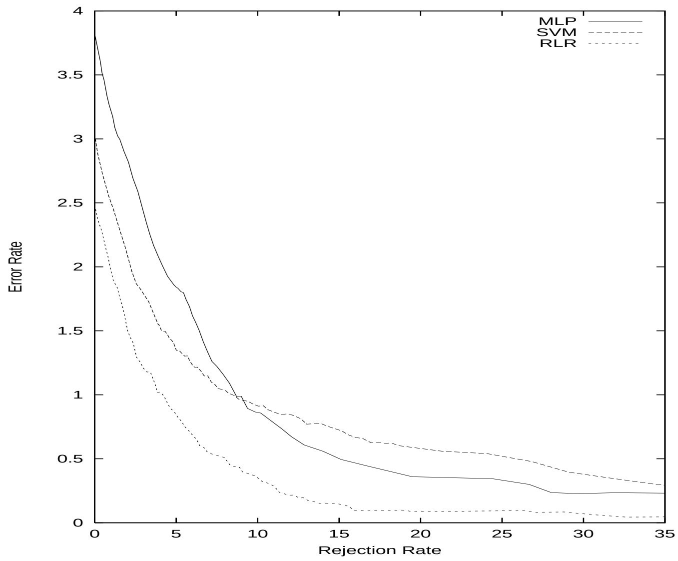


Fig. 1.

architecture 40-80-60-10 is used. The raw error reached by this machine on the test set is 3.82 %.

Support Vector Machine networks (Vapnik (1995)) perform a mapping from the feature space into a high dimensional Hilbert space where a separating hyper-plane is searched for. In this letter 10 different binary machines of this type have been constructed based on degree 2 polynomial scalar products. For non-separable training sets like the one used in this letter, Cortes and Vapnik (1995) propose to constraint the values of the coefficients in terms of which the hyper-plane is written. A restriction equal to 0.005 has been found to be the optimal one. The raw error for this machine on the test set is 3.04 %.

A regularized local linear regression with $k$ nearest neighbors has also been constructed (Bottou and Vapnik (1992)). Adding a regularization factor $\gamma$ to the empirical risk avoids the singularity that appears when too few nearest neighbors are considered, particularly in a high dimensional feature space. Alternatively, a singular value decomposition could have been applied. Training yields the following optimal values: $k = 16$ and $\gamma = 8.6$. The raw error attained on the test set with these parameters is 2.48 %. This figure is 35 % better than the MLP's one and 18% better than the SVM's one.

Figure 1 shows the error-rejection curves on the test set for all of the learning algorithms described above. The local regression behaves clearly better than both global algorithms. The SVM machine is outperformed by the perceptron for rejection rates above 10 %, although it has a lower raw error. This observation is important since it is the global method's rejection ability that will be used to construct ensembles as described in the next section.

## 3 A general procedure to construct a semi-global ensemble

We have seen that local classifiers are surprisingly accurate but slow. This is due to the fact that training takes place during classification and involves searching for nearest neighbors. Global machines are considerably faster but most of them do not guarantee global solutions. The question addressed in this section is: can both types of machines be assembled so as to make them benefit from each other? Interestingly enough, the answer is yes and the procedure involved very simple (see figure 2):

(i) Apply a global machine with a tuned error rejection rate.
(ii) Classify by means of a local machine those patterns rejected by the global learning algorithm.

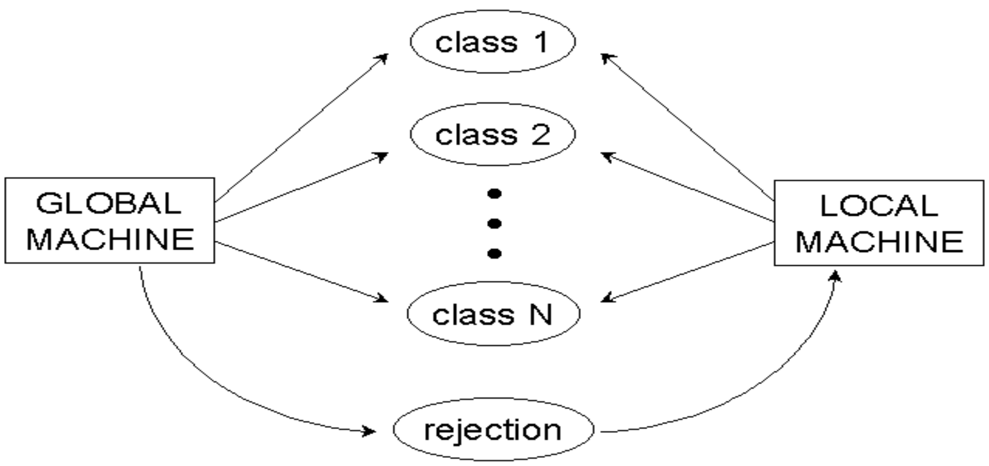


Fig. 2.

The reason why this construction works is the following. In order to generalize, global learning methods tend to overlook local singularities. Local learning methods, for obvious reasons, do not suffer from this limitation and are better suited for classification of patterns close to class boundaries, where local structure becomes most important. A properly trained global method is expected to reject patterns in these ambiguous areas if asked to. We only have to learn the proper rejection rate to be applied and throw rejected patterns to a local classifier. Applying a local classifier everywhere throughout the feature space would be a waste of time. Most of the patterns can be accurately classified by a fast global method that can also spot those patterns that need further local treatment.

## 4 Examples

In this section two different ensembles are constructed. Figure 3 shows the ensemble's error rate versus the global method's rejection rate for both of them on the NIST training (left) and test sets (right).

– MLP+RLR: Multi-layer perceptron followed by Local Linear Regression. The back-propagation trained multi-layer perceptron of section 2 is used as global method in this first example. The test curve starts at the MLP raw error (3.82 %) when no rejection is present and tends to the RLR raw error (2.48 %) for 100 % MLP rejection rate. There is a value in between these two for which the ensemble performs better than both individual learning methods. The training curve yields a value around 5 % while the test one suggests a significantly higher MLP rejection rate: 15 %. This is due to the fact that the perceptron has adjusted particularly well to training patterns during the training process. The local algorithm, however, behaves similarly with both training and test sets. The raw error for the ensemble with 15 % MLP rejection rate is 2.39 % on the test set. This is close to a 40 % improvement with respect to the perceptron's performance and a 4 % enhancement with respect to the original local machine (2.48 % raw error rate).

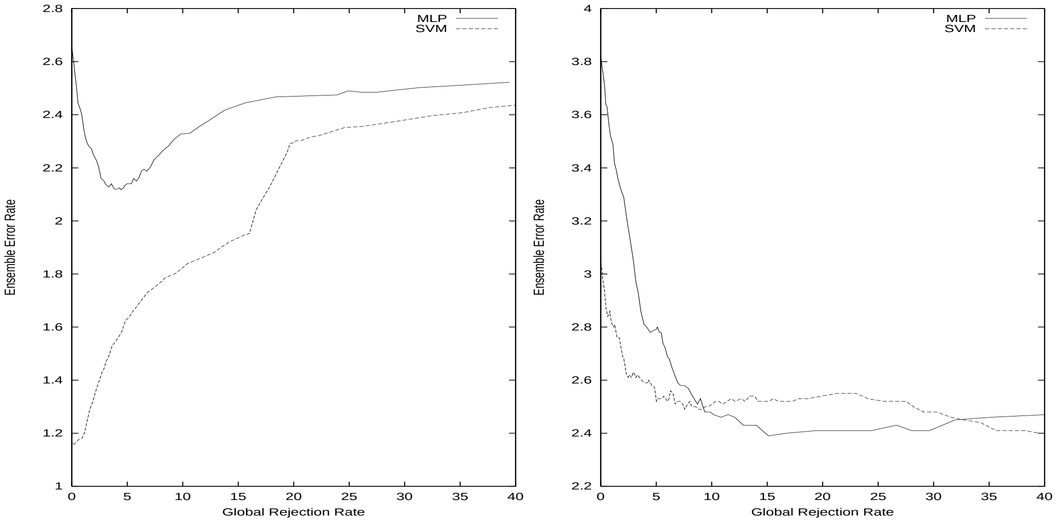


Fig. 3.

– SVM+RLR: Support Vector Machine Network followed by Local Linear Regression. The global method of this ensemble is composed of 10 different SVM networks, one for each class of digits. Due to the fact that SVM keeps the training risk equal to zero, the ensemble's training error increases monotonically for increasing SVM rejection rates. However, on unseen patterns, a 9 % SVM rejection rate gives rise to an ensemble that significantly outperforms SVM, lowering the raw error from 3.04 % down to 2.49 %. This means an 18 % improvement. In this ensemble, the original dimension of the feature space is globally increased by the SVM projection and locally decreased by the regularization procedure.

We have succeeded in engineering neural network ensembles with learning machines as building blocks. These blocks are joined together by adjusting their error-rejection rates. To be more specific, the performance of two different state-of-the-art global methods has been enhanced by their combination with a surprisingly simple local learning machine. The global machine resorts to the local one only where needed, i. e., wherever it feels that local structure turns relevant. This is an interesting way to spot locality and can be considered as an alternative to local acceleration methods. Furthermore, the local machine performance also benefits from the association. Not merely we find patterns that only the local classifier is able to resolve, but there also exist digits that only the global device is capable of classifying. As an example, a back-propagation trained perceptron, performing on a handwritten digit database, achieves a 40 % improvement by means of its association with a regularized local linear regression. Cross-validation is used in order to tune the perceptron's threshold. The overall classification speed does not suffer too much since the local classifier is used for only 15 % of the patterns. Several global methods can be mixed together in order to further reduce the need of a local algorithm. This will be part of future research.

## References


[1] Saffran, J. R., Aslin, R. N. and Newport, E. L. (1996). Statistical Learning by 8-Month-Old Infants. Science **274** , 1926-1929.

[2] Rosenblatt, F. (1960). On the Convergence of Reinforcement Procedures in Simple Perceptrons. Cornel Aeronautical Laboratory Report VG-1916-G-4, Buffalo, New York.

[3] Simard, P. Y., Le Cun, Y. and Denker, J. (1993). Efficient pattern recognition using a new transformation distance. Neural Information Processing Systems, **5**, 50-58.

[4] Drucker, H., Schapire, R. and Simard, P. (1993). Boosting performance in neural networks. International Journal in Pattern Recognition and Artificial Intelligence, **7**(4), 705-719.

[5] Schölkopf, B., Burges, C. and Vapnik, V. (1996). Incorporating Invariances in Support Vector Learning Machines. Proceedings of the International Conference on Artificial Neural Networks, ICANN 96, LNCS 1112, 47-52.

[6] Hansen, L. Kai and Salamon, P. (1990). Neural Network Ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**, 993-1001.

[7] Vapnik, V. and Bottou, L. (1993). Local Algorithms for Pattern Recognition and Dependencies Estimation. Neural Comp. **5**, 893-909.

[8] Jiang, Q. and Zhang, W. (1993). An improved method for finding nearest neighbors. Pattern Recognition Letters **14**, 531-535.

[9] Gürgen, F., Alpaydin, R., Ünlünkin, U. and Alpaydin, E. (1994). Distributed and Local Neural Classifiers for Phoneme Recognition. Pattern Recognition Letters **15**, 1111-1118.

[10] Bottou, L. and Vapnik, V. (1992). Local Learning Algorithms. Neural Comp. **4**(6), 888-900.

[11] Oja, E. (1983). *Subspace Methods of Pattern Recognition.* Research Studies Press.

[12] Rumelhart, K. E., Hinton, G. E. and Williams, R. J. (1986), Learning internal representations by error propagation, *Parallel distributed processing: Explorations in macrostructure of cognition*, Vol. I, Badford Books, Cambridge, MS., pp. 318-362.

[13] Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* Springer Verlag, New York.

[14] Cortes, C. and Vapnik, V. (1995). Support Vector Networks. Machine Learning **20**, 1-25.