

The main point of the method is the procedure for generating new training data X_t' called *neural network reduction of the class separation* (NN_RCS) [4]. We generate X_t' which implies normal class-conditional densities for the responses $y_j'=y(\mathbf{x}_j';\mathbf{w}^*)$ for $\mathbf{x}_j'\in X_t'$, $j=1, 2, \dots, N_t$. For this purpose, we use an auto-associative network having non-linear activation functions in the hidden units (Fig.1).

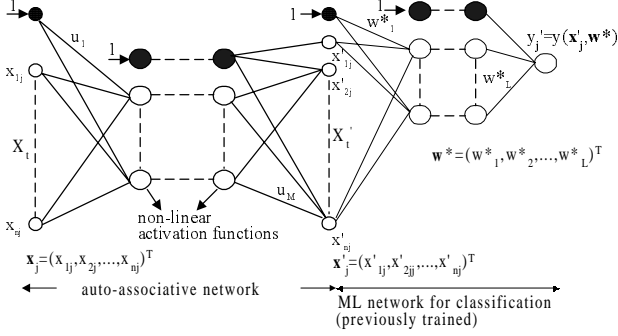


Fig.1. Auto-associative network. It is trained to map input vectors into themselves in such a way that the non-linear discriminant function $y(\mathbf{x}',\mathbf{w}^*)$ has normal class-conditional densities.

The targets used to train the auto-associative network are the input vectors themselves, so that the network is attempting to map each input vector onto itself. We train the network by minimizing an error function of the form

$$E(\mathbf{u}) = (1-\nu)E^{AA}(\mathbf{u}) + \nu\Omega(\mathbf{u}). \quad (2)$$

Here,

$$E^{AA}(\mathbf{u}) = \frac{1}{N_t} \sum_{j=1}^{N_t} [\mathbf{r}(\mathbf{x}_j; \mathbf{u}) - \mathbf{x}_j]^T [\mathbf{r}(\mathbf{x}_j; \mathbf{u}) - \mathbf{x}_j] \quad (3)$$

is the MS error of the auto-associative network, $\Omega(\mathbf{u})$ is the penalty function and ν is the parameter controlling the extent to which the penalty term $\Omega(\mathbf{u})$ influences the form of the solution. In (3) $\mathbf{r}(\mathbf{x}_j; \mathbf{u})$ represents the output vector $\mathbf{x}_j' = \mathbf{r}(\mathbf{x}_j; \mathbf{u})$ of the auto-associative network (see Fig.1) as a function of the input training vectors \mathbf{x}_j , $j=1, 2, \dots, N_t$ and vector \mathbf{u} comprising the adjustable weights of the network. The penalty term $\Omega(\mathbf{u})$ measures the departure of the responses of the classification network $y_j' = y(\mathbf{x}_j'; \mathbf{w}^*)$ from the numbers q_j , $j=1, 2, \dots, N_t$ having normal class-conditional densities

$$\Omega(\mathbf{u}) = \frac{1}{N_t} \sum_{j=1}^{N_t} [y_j' - q_j]^2. \quad (4)$$

The auto-associative network is trained by minimizing the total error function $E(\mathbf{u})$ (2) with respect to \mathbf{u} . A function $\mathbf{r}(\mathbf{x}_j; \mathbf{u})$ which provides a good fit to the training

data \mathbf{x}_j , $j=1, 2, \dots, N_t$ will give a small value for $E^{AA}(\mathbf{u})$ (3), while one which produces data with the normal densities for $y_j' = y(\mathbf{x}_j'; \mathbf{w}^*)$ will give a small value for $\Omega(\mathbf{u})$ (4). Minimizing $E(\mathbf{u})$ (2) we obtain the network mapping $\mathbf{r}(\mathbf{x}_j; \mathbf{u})$ which is a compromise between fitting the training data \mathbf{x}_j and reducing the class separation for $y(\mathbf{x}_j'; \mathbf{w}^*)$ (deflating $E^{ML}(\mathbf{w})$ (1) at \mathbf{w}^*) for suitable normal densities of q_j , $j=1, 2, \dots, N_t$. We obtain the appropriate q_j in the following way. We propagate $\mathbf{x}_j \in X_t$ (current training data) through the previously trained classification network and obtain the corresponding outputs $y_j = y(\mathbf{x}_j; \mathbf{w}^*)$, $j=1, 2, \dots, N_t$. Then we estimate the class-conditional means, variances and (cumulative) distribution functions of the network output using the sample y_j , $j=1, 2, \dots, N_t$. We denote the estimates by \hat{m}_i , $\hat{\sigma}_i^2$ and $\hat{F}(y|\omega_i)$ for ω_i , $i=1, 2$. We compute N_t numbers q_j by the percentile transformation method

$$q_j = \begin{cases} \text{for } c_{1j} = 1 \\ [\Phi^{-1}(\hat{F}(y_j|\omega_1))](\hat{\sigma}_1^2 \pm \nabla\sigma^2)^{\frac{1}{2}} + (\hat{m}_1 - \nabla m_1) \\ \text{for } c_{2j} = 1 \\ [\Phi^{-1}(\hat{F}(y_j|\omega_2))](\hat{\sigma}_2^2 \pm \nabla\sigma^2)^{\frac{1}{2}} + (\hat{m}_2 - \nabla m_2) \end{cases} \quad (5)$$

for y_j , $j=1, 2, \dots, N_t$. Here $\Delta\sigma^2$, Δm_1 , Δm_2 are user-supplied parameters. The numbers q_j have approximately the distribution $N(\hat{m}_1 - \Delta m_1, \hat{\sigma}_1^2 \pm \Delta\sigma^2)$ for $c_{1j}=1$ and $N(\hat{m}_2 - \Delta m_2, \hat{\sigma}_2^2 \pm \Delta\sigma^2)$ for $c_{2j}=1$.

In [2],[3] we proposed a procedure for defining the values of the control parameters ν in (2); $\Delta\sigma^2$, Δm_1 , Δm_2 and the sign (+ or -) of the change $\pm\Delta\sigma^2$ in (5). It directs the local optimizer to a new minimum of $E^{ML}(\mathbf{w})$, and keeps the new training data X_t' as close to the original data X_t as is possible.

3. Experiment

We carried out an experiment with data set containing 150 upper-case handwritten letters "M" and 450 lower-case handwritten letters "m". Nine numerical results from a feature extraction procedure (developed in a company in Israel) were recorded for each case. We divided the data into a training set and a test set. The training set contained 75 M-cases and 225 m-cases and the rest of the data was used for validation.

We used two layer networks with sigmoid activation functions in the hidden units. We set the number of the hidden units to be 3 and 6 for the classification and for the

auto-associative networks, respectively. We set $\hat{m}_i - \Delta m_i = 0$ and $\hat{\sigma}_i^2 \pm \Delta \sigma^2 = 1$ for $i=1,2$ in (5), and $v=0.5$ in (2). This setting implies that the class-conditional densities of q_i (5) are $N(0,1)$ which results in a modified training data X'_i with an approximately full overlap of the classes for the previously defined discriminant function $y(\mathbf{x}, \mathbf{w}^*)$. This certainly eliminates the local minimum of $E^{ML}(\mathbf{w})$ (1) at the previous solution but it causes large destructuring of X'_i which is highly unfavorable to our procedure. We carried out three successive runs of the "reduction of the class separation" (see [3]).

We compared the discrimination qualities of the classification networks trained by our recursive method (Section 2) and those trained by a *conventional training* with a random initialization of the weights [5, pp.260-69]. We evaluated the classification quality of the networks by the resulting *test error rates*. We calculated them by the percentage of the wrongly allocated test observations by a plug-in Bayes allocation rule [7]. We replicated the training, starting from 100 different random initializations of the weights of the network for classification (Fig.1).

Fig.2 shows the obtained result. Our recursive training (solid path) outperforms the conventional training (dashed path) for the most of the initializations. The dots in the bottom of Fig.2 indicate the sequential number of the reduction of the class separation with the smallest test error rate (. - first, .. - second and ... - third reduction). The dots which are missing indicate the random initializations for which the conventional training was better than our procedure.

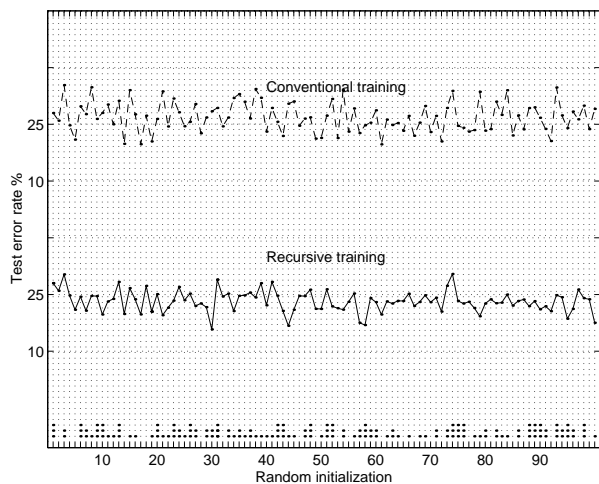


Fig.2. OCR data. 100 different random initializations of the training.

We calculated the averaged difference of the test error rates of the networks trained by the conventional training

and our recursive procedure, which was 3%. We evaluated the significance of this difference by the paired t-test. The 99 percent confidence interval for the population mean difference was $3 \pm 1.4\%$ which confirms a decrease of the test error rate of the network trained by our recursive procedure.

4. Conclusion

We have proposed a method for training an ML network for classification. It was more successful than the conventional training with random initialization of the weights for an OCR application considered in Section 3.

We were stimulated in our research by an idea of Friedman, called "structure removal" [6], which has great potential, like simulated annealing in the field of optimization. We extended Friedman's "structure removal" to non-linear classification functions. The price that we pay for the non-linear extension is the high computational complexity of an intensive non-linear optimization technique for training the auto-associative network, while in the linear case [1],[2],[6] a simple rotation of the data is carried out.

References

- [1] M.E.Aladjem, "Linear discriminant analysis for two-classes via removal of classification structure", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.19, pp.187-192, 1997.
- [2] M.E.Aladjem, "Nonparametric discriminant analysis via recursive optimization of Patrick-Fisher distance", *IEEE Trans. on Syst., Man, Cybern.*, vol.28B, pp.292-299, 1998.
- [3] M.E.Aladjem, "Supervised learning of a neural network for classification via successive modification of the training data - an experimental study" in A.P. del Pobil, J.Mira and M.Ali (eds.), *Lecture Notes in Artificial Intelligence- 11th Int. Conf. on Industrial & Engineering Applications of the Artificial Intelligence & Expert Systems (IEA/AIE-98)*, Benicassim, Castellon (Spain), June 1-4, 1998.
- [4] M.E.Aladjem, "Linear discriminant analysis for two classes via recursive neural network reduction of the class separation", in A.Amin and P.Pudil (eds.), *Lecture Notes in Computer Science- 2nd Int. Workshop on Statistical Techniques in Pattern Recognition*, Sydney, Australia, August 11-13, 1998.
- [5] C.M.Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press Inc., New York, 1995.
- [6] J.H. Friedman, "Exploratory projection pursuit", *Journal of the American Statistical Association*, vol. 82, pp. 249-266, 1987.
- [7] G.J.McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley & Sons, Inc., New York, 1992.