Capacity Control in Linear Classifiers for Pattern Recognition

I. Guyon, V. Vapnik, B. Boser, L. Bottou, and S. A. Solla
AT&T Bell Laboratories
Holmdel, NJ 07733, USA

Abstract

Achieving good performance in statistical pattern recognition requires matching the capacity of the classifier to the amount of training data. If the classifier has too many adjustable parameters (large capacity), it is likely to learn the training data without difficulty, but will probably not generalize properly to patterns that do not belong to the training set. Conversely, if the capacity of the classifier is not large enough, it might not be able to learn the task at all. In between, there is an optimal classifier capacity which ensures the best expected generalization for a given amount of training data.

The method of Structural Risk Minimization *(SRM) refers to tuning the capacity of the classifier to the available amount of training data. In this paper, we illustrate the method of SRM with several examples of algorithms. We present experiments which confirm theoretical predictions of performance improvement in application to handwritten digit recognition.*

1 Capacity and Structural Risk Minimization

A common way of training a classifier is to adjust the parameters $\mathbf{w}$ in the classification function $F(\mathbf{x}, \mathbf{w})$ to minimize the *training error* E_{train}, i.e. the frequency of errors on a set of p training examples. But the classification function $F(\mathbf{x}, \mathbf{w}^*)$ which minimizes the training error does not necessarily minimize the *generalization error* estimated on a separate test set E_{test}.

Any family of classification functions $\{F(\mathbf{x}, \mathbf{w})\}$ can be characterized by its capacity or Vapnik-Chervonenkis dimension (VC-dimension) [1]. The VC-dimension can be in some cases as simple as the number of free parameters of the classifier, but it is in most practical cases quite difficult to determine analytically.

A typical behavior of training error and generalization error as a function of the capacity is shown in figure 1. For a fixed number p of training examples, as the capacity increases, the training error decreases, while the test error goes through a minimum. Before the minimum, the problem is *overdetermined*, i.e. the capacity is too small for the amount of training data. Beyond the minimum, the problem is *underdetermined*. The key issue is therefore to match the capacity of the classifier to the amount of training data in order to get best generalization performance.

The method of *Structural Risk Minimization* (SRM) [1] provides an efficient way of achieving that goal. On the family of classifiers $\{F(\mathbf{x}, \mathbf{w})\}$, we define a structure consisting in nested subsets of elements of the family:

$$S_1 \subset S_2 \subset S_3 \subset \ldots \subset S_r \subset \ldots .$$

We thus ensure that the capacity h_r of the subset of classifiers S_r is less than h_{r+1} of subset S_{r+1}:

$$h_1 \leq h_2 \leq h_3 \leq \ldots \leq h_r \leq \ldots .$$

The method of SRM amounts to finding the subset S^{opt} for which the classifier $F(\mathbf{x}, \mathbf{w}^*)$, which minimizes the training error within such subset, yields the best overall generalization performance.

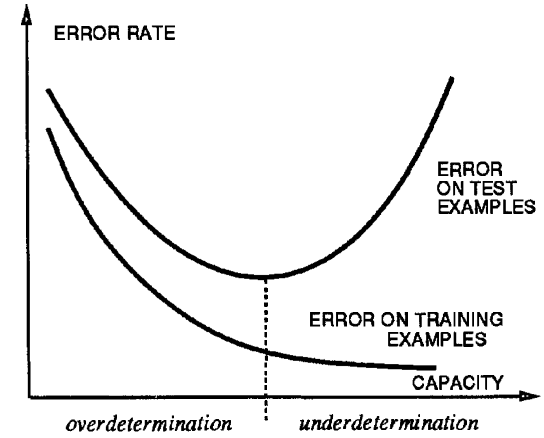

Figure 1: Dependence of the training error and generalization error on the capacity for a fixed size p of the training set.

2 Using Curvature Properties of the MSE Cost Function

Consider three apparently different methods of improving generalization performance: Principal Component Analysis (PCA - a preprocessing transformation of input space) [2], Optimal Brain Damage (OBD - an architectural modification through weight pruning) [3], and a regularization method, Weight Decay (WD - a modification of the learning algorithm) [1]. For the case of a linear classifier, these three approaches control the capacity of the learning system

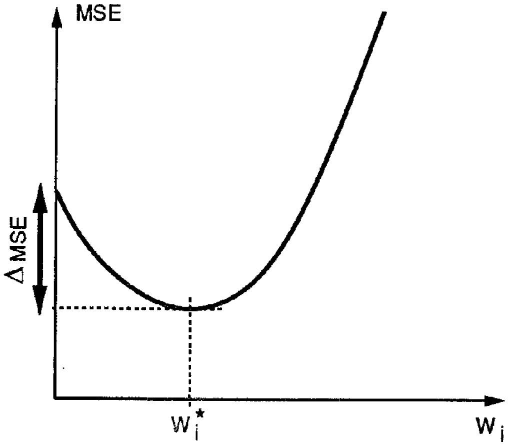

Figure 2: Dependence of MSE on a single parameter w_i.

through the same underlying mechanism: a reduction of the *effective dimension* of weight space, based on the curvature properties of the Mean Squared Error (MSE) cost function used for training.

The classification function of a linear classifiers is $F(\mathbf{x}, \mathbf{w}) = \theta_0(\mathbf{w}^T\mathbf{x})$, where $\mathbf{w}^T$ is the transpose of $\mathbf{w}$ and the function θ_0 takes two values 0 and 1 indicating to which class $\mathbf{x}$ belongs. The VC-dimension of such classifier is equal to the dimension of input space [1] (or the number of weights): $h = dim(\mathbf{w}) = dim(\mathbf{x}) = n$.

Our training algorithm consists in minimizing the Mean Square Error cost function (MSE) [4]:

$$MSE = (1/p)\sum_{k=1}^{p}(y^k - \mathbf{w}^T\mathbf{x}^k)^2 ,$$

where $\mathbf{x}^k$ is the k^{th} example, and y^k is the corresponding desired output and p the number of training examples. Consider the dependence of MSE on one of the parameters w_i (figure 2). One way of reducing the capacity is to set w_i to zero. For the linear classifier, this reduces the VC-dimension by one: $h' = dim(\mathbf{w}) - 1 = n - 1$. At the optimum $\mathbf{w}*$, the MSE increase resulting from setting $w_i = 0$ is to lowest order proportional to the curvature of the MSE at $\mathbf{w}*$. Since the decrease in capacity should be achieved at the smallest possible expense in MSE increase, directions in weight space corresponding to small MSE curvature are good candidates for elimination.

The curvature of the MSE is specified by the Hessian matrix H of second derivatives of the MSE with respect to the weights. For a linear classifier, the Hessian matrix is given by twice the correlation matrix of the training inputs,

$$H = (2/p)\sum_{k=1}^{p}\mathbf{x}^k\mathbf{x}^{kT} .$$

The Hessian matrix is symmetric, and can be diagonalized to get rid of cross terms, to facilitate decisions about the simultaneous elimination of several directions in weight space. The elements of the Hessian matrix after diagonalization are the eigenvalues λ_i; the corresponding eigenvectors give the principal directions w'_i of the MSE. In the rotated axis, the increase ΔMSE due to setting $w'_i = 0$ takes a simple form

$$\Delta MSE \approx \frac{1}{2}\lambda_i(w_i'^*)^2 .$$

The quadratic approximation becomes an exact equality for the linear classifier. Principal directions w'_i corresponding to small eigenvalues λ_i of H are good candidates for elimination.

In [5], we show that PCA, OBD and WD are three similar ways of implementing SRM and reducing the VC-dimension to $h' < h = n$:

1. PCA ranks the classifiers according to the number $m < n$ of largest eigenvalues λ_i kept in the transformation, $h' = m$.

2. OBD ranks the classifiers according to the number $m < n$ of weights corresponding to the largest ΔMSE that survived pruning, $h' = m$.

3. WD ranks the classifiers according to the norm of the weight vector $\mathbf{w}$. This can be shown to be equivalent to minimizing the new cost function

$$MSE + \gamma||\mathbf{w}||^2 .$$

As a function of the structural parameter γ (Lagrange multiplier), an effective capacity [5] can be defined as:

$$h' = \sum_{i=1}^{n}\lambda_i/(\lambda_i + \gamma) .$$

Weights become negligible for $\gamma \gg \lambda_i$, and remain unchanged for $\gamma \ll \lambda_i$.

3 Smoothing, Polynomial Classifiers and Weight Decay

Combining several different structures achieves further performance improvements. The combination of exponential smoothing (a preprocessing transformation of input space) and WD (a modification of the learning algorithm) is shown here to improve character recognition. The generalization ability is dramatically improved by the further introduction of second-order polynomial classifiers (an architectural modification).

Smoothing is a preprocessing which aims at reducing the effective dimension of input space by degrading

[1] We assume, for simplicity, that the first component of vector $\mathbf{x}$ is constant and set to 1, so that the corresponding weight introduces the bias value.

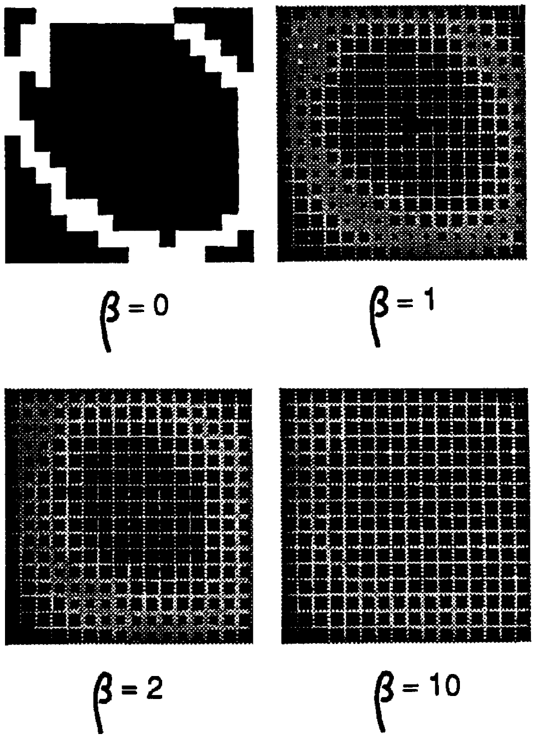

Figure 3: Examples of various level of smoothing performed with an exponential convolutional kernel: $exp[-sqrt(k^2 + l^2)/\beta]$.

Table 1: E_{test} for Smoothing, WD, and Higher-Order Combined.

β	γ	1^{st} order	2^{nd} order
0	γ^*	6.3	1.5
1	γ^*	5.0	0.8
2	γ^*	4.5	1.2
10	γ^*	4.3	1.3
any	0^+	12.7	3.3

the resolution: after smoothing, decimation of the inputs could be performed without further image degradation. Smoothing is achieved here through convolution with an exponential kernel [5], which smoothing parameter β determines the structure. Examples of handwritten digits for various levels of smoothing are shown in figure 3.

Polynomial classifiers can be substituted to linear classifiers: $F(\mathbf{x}, \mathbf{w}) = \theta_0(\mathbf{w}^T \xi(\mathbf{x}))$, where $\xi(\mathbf{x})$ is an m-dimensional vector $(m \geq n)$ with components: $x_1, x_2, ..., x_n$, $(x_1 x_1)$, $(x_1 x_2)$, ... , $(x_n x_n)$, ..., $(x_1 x_2 ... x_n)$. The structure is geared towards increasing the capacity, and is controlled by the order of the polynomial: S_1 contains all the linear terms, S_2 linear plus quadratic, etc. Computations are kept tractable with the method proposed in reference [6].

4 Experimental results

Experiments were performed on the benchmark problem of handwritten digit recognition described in reference [7]. The database consists of 1200 (16 x 16) binary pixel images, divided into 600 training examples and 600 test examples. Ten classifiers were trained, each one separating one class from all others.

In figure 4, we present the results obtained with Weight Decay alone. Effective capacity h' and structural parameter γ vary in opposite direction. For the value $\gamma*$ yielding the smallest error on the test set, the capacity is only 1/3 of the nominal capacity, in the absence of Weight Decay. At the price of some error on the training set, the error rate on the test set is reduced by half. Very similar curves are obtained with PCA and OBD.

In table 1 we report results obtained when several structures are combined. Weight Decay with $\gamma = \gamma^*$ reduces E_{test} by a factor of 2. Input space smoothing used in conjunction with WD results in an additional reduction by a factor of 1.5. The best performance is achieved for the highest level of smoothing, $\beta = 10$, for which the blurring is considerable (figure 3). Smoothing has no effect in the absence of WD. This is a property of exponential smoothing which is a linear invertible operation.

The use of a second-order polynomial classifier provides an additional factor of 5 reduction in E_{test}. For the second order, the number of weights scales like the square of the number of inputs $n^2 = 66049$. But the effective capacity $h' = \sum_{i=1}^{n} \lambda_i/(\lambda_i + \gamma)$ is found to be only 196, for the optimum values of γ and β.

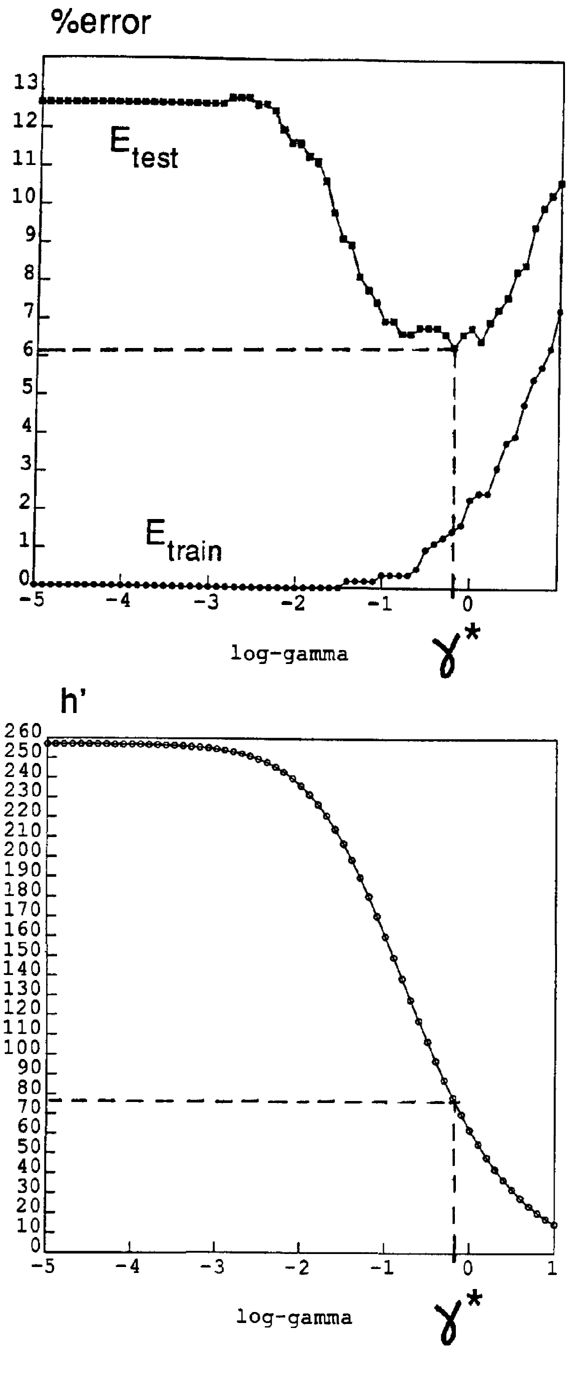

Figure 4: Weight Decay (linear classifier, no smoothing). Percent error (top) and capacity h' (bottom) as a function of $\log \gamma$.

5 Conclusions

The method of SRM provides a powerful tool for tuning the capacity. We have shown that structures acting at different levels (preprocessing, architecture, learning mechanism) can produce similar effects. We have then combined three different structures to improve generalization. These structures have interesting complementary properties. The introduction of higher-order polynomial increases the capacity. Smoothing and Weight Decay act in conjunction to decrease it.

Acknowledgments

We wish to thank L. Jackel's group at Bell Labs Holmdel and T. Hastie at Bell Labs Murray Hill for useful discussions, and are particularly grateful to E. Levin and Y. Le Cun for communicating the method of computing the effective VC-dimension prior publication.

References

[1] V. Vapnik. *Estimation of dependences based on empirical data.* Springer, New York, 1982.

[2] C. W. Therrien. *Decision, Estimation and Classification: An Introduction to Pattern Recognition and Related Topics.* Wiley, 1989.

[3] Y. Le Cun, J. S. Denker, and S. Solla. Optimal brain damage. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2 (NIPS 89)*, pages 598–605, San Mateo CA, 1990. IEEE, Morgan Kaufmann.

[4] R.O. Duda and P.E. Hart. *Pattern Classification And Scene Analysis.* Wiley and Son, 1973.

[5] I. Guyon, V. Vapnik, B. Boser, L. Bottou, and S.A. Solla. Structural risk minimization for character recognition. In J. Moody and et al., editors, *NIPS-91*, San Mateo CA, 1992 (to appear). IEEE, Morgan Kaufmann.

[6] T. Poggio. On optimal nonlinear associative recall. *Biol. Cybern.*, 19:201, 1975.

[7] I. Guyon, I. Poujaud, L. Personnaz, G. Dreyfus, J. Denker, and Y. Le Cun. Comparing different neural network architectures for classifying handwritten digits. In *Proceedings of the International Joint Conference on Neural Networks*, volume II, pages 127–132, Washington DC, 1989. IEEE.