References

[1] V. Vapnik. *Estimation of Dependences Based on Empirical Data.* Springer-Verlag, New York, 1982.

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

[3] Yann Le Cun, J. S. Denker, and S. Solla. Optimal brain damage. In David Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, (Denver, 1989), 1990. Morgan Kaufman.

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

[5] I. Guyon, I. Poujaud, Personnaz L., 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.

where the λ_i are the eigenvalues of H. Weights become negligible for $\gamma \gg \lambda_i$, and remain unchanged for $\gamma \ll \lambda_i$. The capacity is reduced to

$$h' = \sum_i \frac{\lambda_i}{\lambda_i + \gamma} \ . \tag{19}$$

4 Synergistic effects: smoothing, sigma-pi units and regularization

In this section we present an example of synergistic effects obtained by combining several structures. The combination of exponential smoothing (a preprocessing transformation of input space) and regularization (a modification of the learning algorithm) is shown here to improve character recognition. The effect is indeed synergistic: it can be proved that smoothing alone does not lead to better performance. The generalization ability is dramatically improved by the further introduction of second-order units (an architectural modification).

5 Experimental Results and Conclusions

Experiments were performed on the problem of handwritten digit recognition described in reference [5].

Measurements of the classifier capacity were performed according to the method proposed by Vapnik, Le Cun, and Levin, and found to be in good agreement with theoretical predictions. The method gives a measure of the VC-dimension, and accounts for the global capacity of the system, including the effects of input data, architecture, and learning algorithm.

Using the measured capacity, the generalization error can be predicted from the training error. The result is found to be in good agreement with direct measurements of the generalization error on the test set.

As expected, there is an optimal value of the capacity which ensures best generalization. The optimal value is similar in the case of PCA and weight decay. This value is not the one which minimizes the number of errors in the training set; at the price of some errors in the training set, the number of errors in the test set can be decreased by a factor of two. The generalization error can be further decreased by another factor of 1.5 when using smoothing as a preprocessing. Finally, the use of second-order units provides another factor of 4 reduction.

Our results indicate that the VC-dimension must measure the global capacity of the system. It is crucial to incorporate the effects of preprocessing of the input data and modifications of the learning algorithm. Capacities defined solely on the basis of the network architecture give overly pessimistic upper bounds.

Procedure (i) can be understood as a preprocessing, whereas procedure (ii) involves an *a posteriori* modification of the structure of the classifier (network architecture).

In either implementation, the VC-dimension is reduced to

$$h' = dim(\mathbf{w}') = dim(\mathbf{x}') = m \ . \tag{13}$$

The two procedures become identical if the weight elimination in (ii) is based on a 'smallest eigenvalue' criterion.

Procedure (ii) is very reminiscent of *Optimal Brain Damage* (OBD), a weight pruning procedure applied after training. In OBD, the best candidates for pruning are those weights which minimize the increase ΔMSE defined in equation 11. The m weights that are kept do not necessarily correspond to the largest m eigenvalues, due to the extra factor of $(w_i'^*)^2$ in equation 11.

3.5 Weight Decay

Capacity can also be controlled through an additional term in the cost function, to be minimized simultaneously with MSE. Linear classifiers can be ranked according to the norm $\|\mathbf{w}\|^2 = \sum_{j=1}^{n} w_j^2$ of the weight vector. A structure is constructed by allowing within the subset S_r only those classifiers which satisfy

$$\|\mathbf{w}\|^2 \leq c_r \ . \tag{14}$$

The positive bounds c_r form an increasing sequence:

$$c_1 < c_2 < c_3 < ... < c_r < ... \tag{15}$$

This sequence can be matched with a monotonically decreasing sequence of positive Lagrange multipliers

$$\gamma_1 \geq \gamma_2 \geq \gamma_3 \geq ... \geq \gamma_r \geq ... \tag{16}$$

such that our training problem stated as the minimization of a cost function within a specific set S_r is implemented through the minimization of a new cost function:

$$MSE + \gamma_r \|\mathbf{w}\|^2 \tag{17}$$

The minimization of this cost function is equivalent to the procedure referred to as regularization or as weight decay [1]. In a mechanical analogy, the term $\gamma_r \|\mathbf{w}\|^2$ is like the energy of a spring of tension γ_r which pulls the weights to zero. As it is easier to pull in the directions of small curvature of the MSE, weight decay pulls the weights to zero preferably in the principal directions of the Hessian matrix H.

In the principal axis representation, the minimum $\mathbf{w}^\gamma$ of the cost function $MSE + \gamma\|\mathbf{w}\|^2$, is a simple function of the minimum of the MSE, $\mathbf{w}^0$:

$$w_i^\gamma = \frac{\lambda_i}{\lambda_i + \gamma} w_i^0 \ , \tag{18}$$

$$H = \frac{2}{p} \sum_{k=1}^{p} \mathbf{x}^k \mathbf{x}^{kT} \ . \tag{10}$$

The Hessian matrix is symmetric, and can be diagonalized to get rid of cross terms, and 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:

$$\Delta MSE \approx \frac{1}{2} \lambda_i (w_i'^*)^2 \tag{11}$$

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.

3.3 Principal Component Analysis

One common way of reducing the capacity of a classifier is to reduce the dimension of the input space and thereby reduce the number of necessary free parameters (or weights). *Principal Component Analysis* (PCA) is a feature extraction method based on eigenvalue analysis. Input vectors $\mathbf{x}$ of dimension n are approximated by a linear combination of $m \leq n$ vectors forming an ortho-normal basis. The coefficients of this linear combination form a vector $\mathbf{x}'$ of dimension m. The optimal basis in the least square sense is given by the m eigenvectors corresponding to the m largest eigenvalues of the correlation matrix of the training inputs (this matrix is 1/2 of H). A structure is obtained by ranking the classifiers according to the number m of basis vectors kept in the linear transformation. The VC-dimension of the classifier is reduced to:

$$h' = dim(\mathbf{x}') = m \ . \tag{12}$$

3.4 Optimal Brain Damage

For a linear classifier, pruning can be implemented in two different but equivalent ways:

(i) change input coordinates to a principal axis representation, prune the *components* corresponding to small eigenvalues according to PCA, and then train with the MSE cost function;

(ii) change coordinates to a principal axis representation, train with MSE first, and then prune the *weights*, to get a weight vector $\mathbf{w}'$ of dimension $m \leq n$.

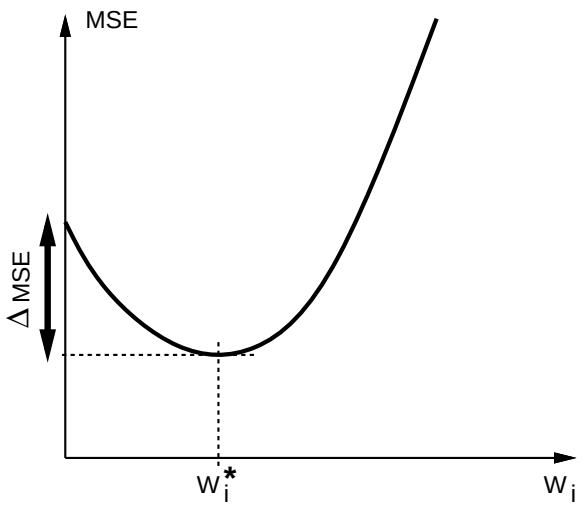

Figure 2: Dependence of MSE on a single parameter w_i.

The problem of minimizing J_{train} as a function of $\mathbf{w}$ can be approached in different ways [4], but it is often replaced by the problem of minimizing a modified function,

$$J_{train} = \frac{1}{p}\sum_{k=1}^{p}(y^k - \mathbf{w}^T\mathbf{x}^k)^2 \ , \tag{8}$$

which differs from the first one in that the nonlinear function θ_0 has been removed. The cost function of equation 8 is referred to as Mean Square Error (MSE).

3.2 Curvature Properties of the MSE Cost Function

The three structures that we investigate rely on curvature properties of the MSE cost function. Consider the dependence of MSE on one of the parameters w_i, as illustrated in figure 2. Training leads to the optimal value w_i^* for the parameter w_i.

One way of reducing the capacity is to set w_i to zero. For the linear classifier, this reduces the VC-dimension by one:

$$h' = dim(\mathbf{w}) - 1 = n - 1 \tag{9}$$

The MSE increase resulting from setting $w_i = 0$ is to lowest order proportional to the curvature of the MSE at w_i^*. Since the decrease in capacity of the classification function 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,

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

Defining a structure is easier than determining the capacity. SRM provides a tool for tuning the capacity to the amount of training data, even when the capacity is not known explicitly. We now examine several ways in which such a structure can be defined.

3 Principal Component Analysis (PCA), Optimal Brain Damage (OBD) and Weight Decay (WD)

The structure required by SRM can be built at either one of the three levels mentioned in the introduction. In this section, we present examples of three equivalent structures. Consider three apparently different methods of improving generalization performance: Principal Component Analysis (a preprocessing transformation of input space) [2], Optimal Brain Damage (an architectural modification through weight pruning) [3], and regularization or Weight Decay (a modification of the learning algorithm) [1]. For the case of a linear classifier, these three approaches are shown here to control the capacity of the learning system 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.

3.1 Linear Classifier and Mean Square Error Training

Consider a linear classifier

$$F(\mathbf{x}^k, \mathbf{w}) = \theta_0(\mathbf{w}^T\mathbf{x}) \ , \tag{5}$$

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:

$$\begin{aligned} \theta_0(a) &= 1 \quad \text{if} \quad a > 0 \\ \theta_0(a) &= 0 \quad \text{if} \quad a \leq 0 \end{aligned}$$

The empirical risk of equation 1 becomes

$$J_{train} = \frac{1}{p}\sum_{k=1}^{p}(y^k - \theta_0(\mathbf{w}^T\mathbf{x}^k))^2 \ . \tag{6}$$

The VC-dimension of such a linear classifier is equal to the dimension of input space (or the number of weights):

$$h = dim(\mathbf{w}) = dim(\mathbf{x}) = n \tag{7}$$

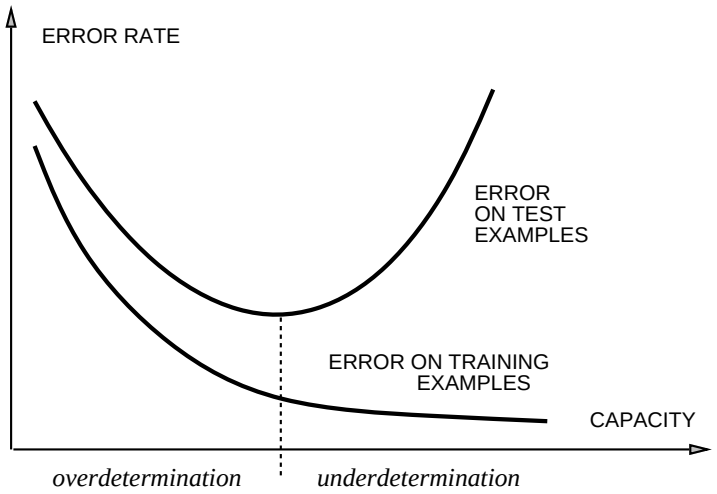

Figure 1: Dependence of the training error and generalization error on the capacity for a fixed size p of the training set.

as the number of free parameters of the classifier, but it is in most practical cases quite difficult to determine analytically.

The behavior of training error and generalization error as a function of the capacity is sketched in figure 1 for a fixed number p of training examples. As the capacity increases, the training error keeps decreasing monotonically, 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) provides an efficient way of achieving that goal.

2.3 Structural Risk Minimization

Let us choose a family of classifiers $\{F(\mathbf{x}, \mathbf{w})\}$, and define a structure consisting in nested subsets of elements of the family:

$$S_1 \subset S_2 \subset S_3 \subset ... \subset S_r \subset ... \tag{3}$$

Let the structure be such that the capacity h_r of the subset of classifiers S_r is less than h_{r+1} of subset S_{r+1}:

$$h_1 \leq h_2 \leq h_3 \leq ... \leq h_r \leq ... \tag{4}$$

2 Risk Minimization and Capacity

2.1 Empirical Risk Minimization

Pattern recognition problems can be restricted to a binary classification problem: given the input vector $\mathbf{x}$ and a parameter vector $\mathbf{w}$, the classification function $F(\mathbf{x}, \mathbf{w})$ is an indicator function, taking the value *one* for one class and *zero* for the other.

A common way of training a given classifier is to adjust the parameters $\mathbf{w}$ in the classification function $F(\mathbf{x}, \mathbf{w})$ so as to minimize the number of errors on a set of p training examples:

$$J_{train} = \frac{1}{p} \sum_{k=1}^{p} (y^k - F(\mathbf{x}^k, \mathbf{w}))^2 \ , \tag{1}$$

where $\mathbf{x}^k$ is the k^{th} example of the training set and y^k is the corresponding desired output of the classifier. The training error J_{train} provides an estimate of the expected risk based on the empirical data provided by the p available examples. The method is thus called *Empirical Risk Minimization.*

The limitations of this method can be easily understood: the classification function $F(\mathbf{x}, \mathbf{w}^*)$ which minimizes the empirical risk does not necessarily minimize also the expected value of the risk over the full distribution of possible inputs and their corresponding outputs. Such expected risk J_{expect} cannot in general be computed, but it can be estimated. The error rate on the training set gives a poor estimate, and an independent test set can be used to obtain:

$$J_{expect} \hat{=} J_{test} = \frac{1}{q-p} \sum_{k=p+1}^{q} (y^k - F(\mathbf{x}^k, \mathbf{w}))^2 \ . \tag{2}$$

The total number q of available examples has thus been divided into p training examples and $q - p$ testing examples. The generalization error J_{test} is used here as an estimate of J_{expect}. Other ways of estimating J_{expect} include the *leave-one-out* or *moving control* method [1].

2.2 Capacity

Any family of classification functions $\{F(\mathbf{x}, \mathbf{w})\}$ can be characterized by its capacity. The generalization properties of a given classifier can be predicted given the error J_{train} on the training set and its capacity, with no need for a test set. The Vapnik-Chervonenkis dimension (or VC-dimension) [1] is such a measure of capacity, defined as the maximum number h of training examples which can be learnt without error, *for all possible class labelings.* The VC-dimension can be in some cases as simple

Capacity Control in Linear Classifiers for Pattern Recognition

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

EXTENDED ABSTRACT

1 Introduction

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. This capacity is influenced by several factors, including: (1) properties of the input space, (2) nature and structure of the classifier, and (3) learning algorithm. Actions based on these three factors can be combined to control capacity and achieve optimal generalization. These three factors are not independent. In some cases they provide three different ways of achieving equivalent effects; in other cases they produce different effects that can be combined synergistically.

In this paper, we briefly review the method of SRM. We present several examples of algorithms and show the influence of the three fundamental factors. Finally, we present experiments which confirm theoretical predictions of performance improvement in application to handwritten digit recognition.

Category: Pattern Recognition Methodology and Systems.
Copyright (c) 1991 AT&T (unpublished). For confidential review only.