J. Phys. A: Math. Gen. 28 (1995) 1325–1334. Printed in the UK

Finite-size effects and optimal test set size in linear perceptrons

David Barber†, David Saad and Peter Sollich

Department of Physics, University of Edinburgh, Edinburgh EH9 3JZ, UK

Received 4 October 1994

Abstract. Fluctuations in the test error are important in the learning theory of finite-dimensional systems as they represent how well the test error matches the average test error. By explicitly finding the variance of the test error due to randomness present in both the data set and algorithm for a linear perceptron of dimension n, we are able to address such questions as the optimal test set size. Where exact results were not tractable, a good approximation is given to the variance. We find that the optimal test set size possesses a phase transition between linear and $\frac{2}{3}$ power-law scaling in the system size n.

1. Introduction

Learning from examples deals with the question of how to find a network which, after being trained on a number of input–output example pairs, i.e. instances of an underlying mapping, is able to generalize well [1]. That is, how to find a network which, given a randomly selected input, accurately predicts the output corresponding to this input. Much theoretical analysis of networks has concentrated on calculating the rate at which the generalization error made by the network decays as the number of training examples increases [1, 2]. Within the physics community, such calculations have typically been achieved using tools from statistical mechanics, for which a corresponding thermodynamic limit ($n \to \infty$) is taken [3]. In this thermodynamic limit, the generalization error is self-averaging, such that fluctuations induced, for example, by the assumed randomly drawn training sets, are neglected. The average case scenario is to be contrasted with other approaches such as the VC formalism [4] which inherently deals with finite variance distributions. The VC formalism deals with the worst-case scenario, i.e. how many examples are required to guarantee, with a certain probability, that the error that the network will make will not be greater than some specified amount. To some extent, the traditional average-case scenario considered in the statistical mechanics literature can be moved closer to this worst-case approach by calculating the variance of the error distribution in addition to the average error.

Such results will enable us to address a question of much interest in the practical field of training neural networks. Namely, given a data set of a finite size, how are we to partition the data optimally into a set on which the network is to be trained, and a set on which the network performance is to be estimated.

† E-mail address: D.Barber@ed.ac.uk

0305-4470/95/051325+10$19.50

2. Learning from examples

We consider the scenario in which the inputs are represented by n-dimensional real vectors, $x \in \Re^n$, and the output is a real variable, $y \in \Re$. A *data set* $\mathcal{L}$ is a set of l input–output pairs, $\mathcal{L} = \{(x^\rho, y^\rho), \rho = 1 \ldots l\}$. The inputs x^ρ are assumed to be drawn independently and identically from a zero mean, unit covariance matrix Gaussian distribution. The outputs are $y^\rho = y^0(x^\rho) + \sigma^\rho$ for some *teacher* function $y^0(\cdot)$, where σ^ρ is additive noise. For the purpose of learning from examples, $\mathcal{L}$ is split into two disjoint sets, the *training set*, $\mathcal{P} = \{(x^\sigma, y^\sigma), \sigma = 1 \ldots p\}$ and the *test set*, $\mathcal{M} = \{(x^\mu, y^\mu), \mu = 1, \ldots m\}$, where $l = p + m$†.

The aim is to find, using the information in $\mathcal{P}$, a *student* function $y(x)$ that matches as closely as possible the output of a randomly chosen input–output pair. That is, we search for student functions that *generalize* well. Clearly, the optimal student is identical to the teacher, and we shall assume that this function is accessible to the student, i.e. that the learning problem is *realizable* [3].

In this paper, we deal with one of the simplest input–output mappings considered in the learning from examples literature, namely the *linear perceptron* [1], for which the output y is related to the input x by

$$y(x) = \frac{1}{\sqrt{n}} w \cdot x$$

where the *weight vector*, $w \in \Re^n$. The data set outputs are generated by a 'noisy' teacher, $y^\sigma = w^0 \cdot x^\sigma / \sqrt{n} + \sigma^\sigma$, where w^0 is the teacher vector, and the noise is drawn from a Gaussian distribution of mean zero, variance σ^2, such that $\langle \sigma^\rho \sigma^\tau \rangle = \sigma^2 \delta_{\rho,\tau}$. In addition, the *spherical constraint* is assumed on the teacher, namely that it lies on the hypersphere $w^0 \cdot w^0 = n$.

Student perceptrons that match the outputs of the training set well are found by minimizing the *training energy*

$$E_{\mathrm{tr}} = \sum_{\sigma=1}^{p} \left(y(x^\sigma) - y^0(x^\sigma) \right)^2 = \sum_{\sigma=1}^{p} (\tilde{w} \cdot x^\sigma - \sigma^\sigma)^2$$

where, for convenience, we have defined $\tilde{w} = \left(w - w^0 \right) / \sqrt{n}$.

To prevent the student learning the noise in the training set we add a regularizing term, λw^2, to the training energy to form an energy function, $E = E_{\mathrm{tr}} + \lambda w^2$ [5, 2]. This extra *weight decay* term penalizes large weights and prevents overfitting, improving generalization performance. The gradient descent algorithm 'descends' the energy surface E by updating the student weight components at learning time t according to

$$\frac{\partial w_i}{\partial t} = -\frac{\partial E}{\partial w_i} + F_i(t)$$

where $F_i(t)$ is white noise such that $\langle F_i(t) F_j(t') \rangle = 2T \delta_{ij} \delta(t - t')$ and T is the effective temperature [3]. The equilibrium ($t \to \infty$) distribution of students that this algorithm produces is a *Gibbs* distribution,

$$P(w|\mathcal{P}) = \frac{1}{Z} \exp(-E/T)$$

where Z is a normalization constant.

† A σ index will refer to a training input, and μ to a test input.

The *test error*, defined by

$$\epsilon_{\text{test}}(w|\mathcal{M}, w^0) = \frac{1}{m}\sum_{\mu=1}^{m}\left(y^\mu - y(x^\mu)\right)^2 = \frac{1}{m}\sum_{\mu=1}^{m}(\tilde{w}\cdot x^\mu - \sigma^\mu)^2 \qquad (2.1)$$

measures how well a student performs on examples from the test set. Ideally, one would like to know the *test* or *generalization function*, i.e. the expected error that a student drawn from $P(w|\mathcal{P})$ will make on a random test example, $\epsilon_f(w\|w^0) = \langle\epsilon_{\text{test}}(w|\mathcal{M}, w^0)\rangle_{\mathcal{M}}$†. The generalization function averaged over $P(w|\mathcal{P})$ and all possible training sets $\mathcal{P}$ is termed the *generalization error*, ϵ_g.

The test error forms an m sample estimate of the generalization function. According to the central limit theorem, the generalization function will be distributed in a Gaussian manner around the test function [6]. It is the central aim of this paper to calculate the variance of this distribution. The fluctuations due to random training sets for a particular student generated from the training set $\mathcal{P}$ are quantified by $\left\langle(\epsilon_{\text{test}}(w|\mathcal{M}, w^0) - \epsilon_f(w\|w^0))^2\right\rangle_{\mathcal{M}}$ and the average fluctuation of students generated by the training set can be found by averaging this over $P(w, \mathcal{P}) = P(w|\mathcal{P})P(\mathcal{P})$. We then write the average fluctuation for a p-dimensional training set as‡

$$\Sigma^2 = \left\langle\left(\epsilon_{\text{test}}(w|\mathcal{M}, w^0) - \epsilon_f(w\|w^0)\right)^2\right\rangle_{\mathcal{M},w,\mathcal{P}} = \frac{1}{m}{\Sigma_1}^2 \qquad (2.2)$$

where $\langle\cdots\rangle_{\mathcal{M},w,\mathcal{P}}$ denotes an average over the test set, post training student, and training set distributions, respectively. ${\Sigma_1}^2$ is the variance of the test error calculated for a single test example. If the vast majority of the data examples are assigned to the training set and very few to the test set, the confidence in how well the test error matches the generalization function will generally be small. Indeed, the test error in this case would typically fluctuate wildly over different test sets, i.e. the variance, Σ^2 would be relatively large. Thus, we really want to use the data in a dual manner: to minimize the test error, yet remain confident that it will be representative of the generalization function. That is, given a data set of size l, we aim to know how many examples, m, should constitute the test set, assigning the other $p = l - m$ examples to the training set.

In order to address this, we form the generalization function *upper bound* $\epsilon_{\text{ub}}(m|l) = \epsilon_g(m) + \tau\Sigma(m)$, where τ is a confidence parameter to be chosen. We view $\epsilon_{\text{ub}}(m|l)$ as an average probabilistic upper bound on the generalization function of students trained on p examples and tested on m examples. In order to calculate the optimal scheme to satisfy the above dual requirement, we minimize $\epsilon_{\text{ub}}(m|l)$ with respect to m to find the optimal test set size, m^*. This requires the calculation of the variance, Σ^2.

In the following section, we calculate the variance exactly for a restricted region of the space of parameters λ, T and σ^2. In section 4, we give results that hold for all parameter values, but are valid only for the large-n regime. Using these results, we present the optimal test set calculations in section 5, concluding with a summary and discussion of our work in section 6.

3. Exact variances

In the following two sections, we present briefly results of calculations that are exact in the sense that they hold for all n. These results represent the continuation of work

† Although $\epsilon_f(w\|w^0)$ is a function of the teacher, due to isotropy of the teacher space, the results of this paper depend only on the length of the teacher vector, which is fixed. To simplify the calculation, however, we include later a teacher average which is implicit in the average over the data set.

‡ An average over the noise is implicit in the average over the test and training sets.

presented elsewhere [7], in which the variance of the noise-free spherical linear perceptron was calculated under exhaustive learning†.

The exact calculations, however, were performed by *not* including a weight decay term in the energy E. We defer presentation of results including weight decay until a later section, as these results rely on a large-n approximation.

3.1. Gibbs learning without weight decay ($\lambda = 0$)

Recently, the generalization error for the finite-n Gibbs learning algorithm, without weight decay, was given [8]. The calculation of the variance employs these results, and we present briefly the line of argument.

The average of the test error, given by (2.1), over the noise distribution, test sets, and student distribution becomes, after straightforward Gaussian integrations,

$$\left\langle \epsilon_{\text{test}}\left(w|\mathcal{M}, w^0\right)\right\rangle_{\mathcal{M},w} = \left\langle \tilde{w}^2\right\rangle_w + \sigma^2 . \tag{3.1}$$

By explicitly evaluating the first term of (3.1), we find

$$\left\langle \epsilon_{\text{test}}\left(w|\mathcal{M}, w^0\right)\right\rangle_{\mathcal{M},w} = \left(\tfrac{1}{2}T + \sigma^2\right) \text{tr}' \mathbf{A}^{-1} + \sigma^2 . \tag{3.2}$$

Here the *covariance matrix* is defined as

$$\mathbf{A} = \frac{1}{n}\sum_{\sigma=1}^{p} x^\sigma (x^\sigma)^{\text{T}} \tag{3.3}$$

and $\text{tr}'(\cdot) = \text{tr}(\cdot)/n$, where $\text{tr}(\cdot)$ is the trace. The generalization error is found by taking an average of $\text{tr}' \mathbf{A}^{-1}$ over the Gaussian inputs of the training set, which we denote by $\langle \cdots \rangle_x$. $\mathbf{A}^{-1}$ is distributed according to an inverse Wishart distribution, $W^{-1}(\mathbf{I}, p)$ [9, 8], where $\mathbf{I}$ is the identity matrix. In order that the average of the inverse is finite‡, we require $p > n + 1$, and have the result, $\left\langle \text{tr}' \mathbf{A}^{-1}\right\rangle_x = n/(p - n - 1)$, which gives

$$\epsilon_g = \left(\tfrac{1}{2}T + \sigma^2\right) \frac{n}{p - n - 1} + \sigma^2 . \tag{3.4}$$

For the variance, we rewrite (2.2) as

$$\Sigma^2 = \left\langle \epsilon_{\text{test}}\left(w|\mathcal{M}, w^0\right)^2\right\rangle_{\mathcal{M},w,\mathcal{P}} - \left\langle \epsilon_{\text{f}}(w\|w^0)^2\right\rangle_{w,\mathcal{P}} \tag{3.5}$$

where, as before, $\epsilon_{\text{f}}(w\|w^0) = \left\langle \epsilon_{\text{test}}\left(w|\mathcal{M}, w^0\right)\right\rangle_{\mathcal{M}}$. After carrying out the average over $\mathcal{M}$, equation (3.5) gives

$$\Sigma_1{}^2 = 2\left\langle \tilde{w}^4 + 2\sigma^2\tilde{w}^2 + \sigma^4\right\rangle_{w,\mathcal{P}} . \tag{3.6}$$

A straightforward Gaussian average over $P(w, \mathcal{P})$ gives

$$\Sigma_1{}^2 = 2\left\langle \frac{1}{2n}\text{tr}' \mathbf{A}^{-2}\left(T + 2\sigma^2\right)^2 + \left[\text{tr}' \mathbf{A}^{-1}\left(\tfrac{1}{2}T + \sigma^2\right) + \sigma^2\right]^2\right\rangle_x . \tag{3.7}$$

This can be explicitly evaluated for $p > n + 3$ by employing [9]

$$\left\langle \left(\text{tr}' \mathbf{A}^{-1}\right)^2\right\rangle_x = \frac{(pn + 2 - n^2 - 2n)n}{(p - n)(p - n - 1)(p - n - 3)} \tag{3.8}$$

† In the exhaustive learning scenario considered in [7], $P(w|\mathcal{P})$ is given by the distribution that is uniform over those student weights that reproduce the training set exactly and that satisfy the constraint $w \cdot w = n$.

‡ For $p < n$, there are unconstrained directions for the student, which lead to a divergent integral in the average.

and

$$\left\langle \mathrm{tr}' \mathbf{A}^{-2} \right\rangle_x = \frac{n^2(p-1)}{(p-n)(p-n-1)(p-n-3)}. \tag{3.9}$$

The full expression for $\Sigma_1{}^2$ is somewhat cumbersome, but simplifies in the large-n limit to

$$\Sigma_1{}^2 = \frac{1}{2}\left(\frac{2\sigma^2\alpha + T}{\alpha - 1}\right)^2 + \mathrm{O}\left(\frac{1}{n}\right) \tag{3.10}$$

where $\alpha = p/n > 1$. Thus both the generalization error and variance diverge for $\alpha \to 1$. As α increases beyond 1, $\Sigma_1{}^2$ decreases to its asymptotic value $2\sigma^4$.

3.2. Pseudo-inverse

The pseudo-inverse algorithm is a limiting case of the general Gibbs algorithm in which the temperature and weight decay both tend to zero such that $T/\lambda \ll 1$ [5, 2]. The benefit from the point of view of the analysis here is that we are able to calculate exactly the variance for both $p < n$ and $p > n$, rather than being restricted to essentially $p > n$ in section 3.1.

The generalization error for the pseudo-inverse algorithm for $p > n + 1$ is given by employing the $T = 0$ limit of (3.4) [8]. Similarly, the results for the variance for $p > n + 3$ can readily be obtained from (3.7) by setting $T = 0$. For $p < n$, the pseudo-inverse algorithm is given by $w = \mathbf{P}w^0$, where $\mathbf{P}$ is the projection onto the subspace spanned by the training inputs [1]. Thus $P(w|\mathcal{P})$ is zero except for the single point, $w = \mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}})^{-1}\mathbf{Y}$, where $\mathbf{Y}^{\mathrm{T}} = (y^1, \ldots, y^p)$ and $\mathbf{X}^{\mathrm{T}} = (x^1, \ldots, x^p)$. This gives

$$\epsilon_{\mathrm{g}} = 1 - \frac{p}{n} + \sigma^2\left(1 + \left\langle \mathrm{tr}' \mathbf{B}^{-1} \right\rangle_{\mathbf{x}}\right)$$

where $\mathbf{B} = \mathbf{X}\mathbf{X}^{\mathrm{T}}/n$. Comparing $\mathbf{B}$ with the $n \times n$ correlation matrix for p patterns, $\mathbf{A} = \mathbf{X}^{\mathrm{T}}\mathbf{X}/n$ (cf equation (3.3)), we remark that $\mathbf{B}$ is also a correlation matrix, distributed identically to $\mathbf{A}$, but with the roles of p and n reversed. The results from section 3.1 concerning the averages of the correlation matrix can then be employed directly by interchanging p and n. For $p < n - 1$, we obtain

$$\epsilon_{\mathrm{g}} = 1 - \frac{p}{n} + \sigma^2\frac{n-1}{n-p-1}$$

in agreement with known results for $n \to \infty$, $\alpha = p/n =$ constant [5]†.

A straightforward calculation of the variance for $p < n - 3$ leads to

$$\begin{aligned}\tfrac{1}{2}n^2\Sigma_1{}^2 = \sigma^4\left[2\left\langle \mathrm{tr}' \mathbf{B}^{-2} \right\rangle_x + \left\langle (\mathrm{tr}' \mathbf{B}^{-1})^2 \right\rangle_x + 2\left\langle \mathrm{tr}' \mathbf{B}^{-1} \right\rangle_{\mathbf{x}} + 1\right] \\ + (n-p)\left[2\sigma^2\left(\left\langle \mathrm{tr}' \mathbf{B}^{-1} \right\rangle_{\mathbf{x}} + 1\right) + \frac{n}{n+2}(2 + n - p)\right].\end{aligned}$$

The results in (3.8) and (3.9) can then be employed to find the variance explicitly. In figure 1, we plot the generalization error and $\Sigma_1/\sqrt{2}$ against α. We remark that the two curves are very similar, a result which we show in the next section is not coincidental. Note that both curves possess the characteristic divergence as the training set size p approaches the system size n.

† Note that in [5] the generalization error is calculated for uncorrupted test sets.

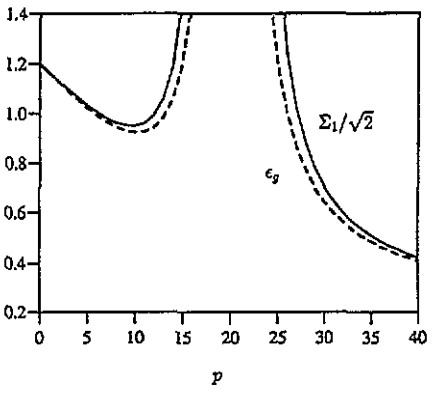

Figure 1. Pseudo-inverse algorithm, $T = 0$ and $\lambda = 0$: *broken curve, generalization error; full curve, scaled standard* deviation $\Sigma_1/\sqrt{2}$. The noise is $\sigma^2 = 0.2$, and $n = 20$.

4. Weight decay

In this section we present results for the general Gibbs learning algorithm for arbitrary temperature, weight decay, and noise. We proceed to calculate the variance as before, but with the inclusion of a weight decay term.

After carrying out the Gaussian integrations over the noise and test set inputs, the resulting generalization error and variance are necessarily of the same form as (3.1) and (3.6), respectively. The only difference is the distribution $P(w|\mathcal{P})$ which now includes a weight decay term. By continuing the Gaussian integrations required for the average over $P(w|\mathcal{P})$, we obtain,

$$\epsilon_g = \sigma^2 + \left(\tfrac{1}{2}T + \sigma^2\right)\left\langle \mathrm{tr}' \mathbf{M}^{-1}\right\rangle_x + \lambda\left(\lambda - \sigma^2\right)\left\langle \mathrm{tr}' \mathbf{M}^{-2}\right\rangle_x \tag{4.1}$$

where

$$\mathbf{M} = \mathbf{A} + \lambda \mathbf{I}.$$

Here $\mathbf{A}$ is the correlation matrix defined earlier in (3.3). The difficulty arises in the calculation of the averages of inverse powers of the matrix $\mathbf{M}$. $\mathrm{tr}' \mathbf{M}^{-1}$ is termed the *response function*, $\mathcal{G}$, which can be shown to be self-averaging in the thermodynamic limit, with $\langle(\mathcal{G} - G)^2\rangle_x = \mathrm{O}(1/n^2)$, where $G = \langle\mathcal{G}\rangle_x$ [10]. Moreover, Sollich [10] obtained the first-order corrections to the average of the finite-n response function. These results give explicitly that

$$G = G_0 + G_1/n + \mathrm{O}\left(1/n^2\right)$$

where G_0 is the averaged response function in the thermodynamic limit, and has the value

$$G_0 = \frac{1}{2\lambda}\left(1 - \alpha - \lambda + \sqrt{(1 - \alpha - \lambda)^2 + 4\lambda}\right).$$

G_1 is related to G_0 by the equation, $G_1 = G_0^2\left(1 - \lambda G_0\right) / \left(1 + \lambda G_0^2\right)^2$. Using these results, the first-order approximation to the average $\langle \mathrm{tr}' \mathbf{M}^{-1}\rangle_x$ can readily be found. Similarly, $\langle \mathrm{tr}' \mathbf{M}^{-2}\rangle_x$ can be found by using $\langle \mathrm{tr}' \mathbf{M}^{-2}\rangle_x = -\left(\partial/\partial\lambda\right)\langle \mathrm{tr}' \mathbf{M}^{-1}\rangle_x M$.

At this point, however, we note that for the linear perceptron under consideration, we can rewrite the equation for the variance as

$$\tfrac{1}{2}{\Sigma_1}^2 = \epsilon_g^2 + \mathrm{var}(\tilde{w}^2)_{w,\mathcal{P}} \tag{4.2}$$

where $\mathrm{var}(\tilde{w}^2)_{w,\mathcal{P}}$ is the variance of $\tilde{w} \cdot \tilde{w}$ over the distribution $P(w, \mathcal{P})$ and $\tilde{w} = (w - w^0)/\sqrt{n}$. By straightforward Gaussian integration, one finds that this variance

is a function of the average of terms involving $\mathrm{tr}'\mathbf{M}^{-i}$, $i = 1 \ldots 4$. Furthermore, the resulting expression is $O(1/n)$, such that any finite-size corrections to $\mathrm{tr}'\mathbf{M}^{-i}$ will be $O(1/n^2)$ corrections to Σ_1^2. Whilst these corrections are straightforward to obtain, the resulting lengthy expressions do not merit inclusion here. To a very good approximation, therefore, the standard deviation of the test error scales linearly with the generalization error. Indeed, looking back at (3.10), we note that the large-n expansion of the variance satisfies $\Sigma_1{}^2 = 2\epsilon_g^2 + O(1/n)$.

Evaluating (4.1) and expanding for small λ, T gives

$$\epsilon_g = \frac{1}{2}\frac{2\sigma^2\alpha + T}{\alpha - 1} - \frac{\alpha\lambda}{2}\frac{T + 4\sigma^2}{(\alpha - 1)^3} + O\left(\frac{1}{n}\right) \tag{4.3}$$

where $\alpha > 1$ and $T \ll \lambda \ll 1$ (a similar expansion holds for $\alpha < 1$). We see here that a weight decay is advantageous in reducing the generalization error and hence also the variance.

So far we have considered an isotropic input distribution. More general input distributions can be considered in which the inputs are in some way correlated (see, for example, [10, 11]). For the case in which the inputs are 'spatially' correlated, $P(x) \propto \exp(-x^{\mathrm{T}}\Gamma^{-1}x/2)$, equation (4.2) still holds for the modified input distribution on replacing $\tilde{w}$ with $\Gamma^{1/2}\tilde{w}$. The variance of a single test example can then be well approximated as before by twice the square of the generalization error under the new input distribution.

5. Optimal test set size

Now that the variance has been calculated, we can proceed to establish the optimal test set size.

A data set $\mathcal{L}$, consisting of l elements, is split into the two disjoint subsets, $\mathcal{P}$ and $\mathcal{M}$. As before, $\mathcal{P}$ is the training set consisting of p examples, and $\mathcal{M}$ is the test set of m examples, such that $\mathcal{L} = \mathcal{P} \cup \mathcal{M}$, and $l = p + m$. Given a data set of l elements, we can then set $p = l - m$ in the equations for the variance and generalization error, and let m vary between 1 and $l - 1$.

For small m, the standard deviation is relatively large and the generalization error is small, as the perceptron has been trained on a relatively large number of examples and tested on only a few. This situation reverses as m is increased. The resulting competition between the generalization error and standard deviation leads to the following definition:

- The probabilistic *upper bound* on the generalization function is defined by $\epsilon_{ub}(m|l) = \epsilon_g + \tau\Sigma$, where τ is a confidence parameter.

From the central limit theorem, the generalization function will be distributed in a Gaussian manner around the test error [6]. On average, the generalization function will be distributed similarly around the generalization error. Setting $\tau = 1$, we will be 84% confident that the generalization function will lie below $\epsilon_{ub}(m|l)$. Similarly, for $\tau = 2$, we will be 98% confident†. For convenience, we set $\tau = 1$ throughout.

In figure 2, we plot the generalization error and upper bound function for two values of the weight decay for $n = 100$, $l = 200$, $\sigma^2 = 0.2$ and $T = 0$. We note that the two graphs

† Here we have quoted the percentage of the normal curve less than a certain number of standard deviations from the mean [6].

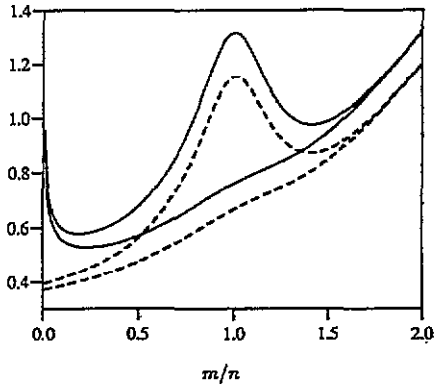

Figure 2. Full curves, upper bounds for $\lambda = 0.01$ (upper curve), and 0.05. The broken curve is the generalization error. The noise is $\sigma^2 = 0.2$, $n = 100$, $l = 200$, $T = 0$. The global minimum in each upper bound represents the optimal test set size.

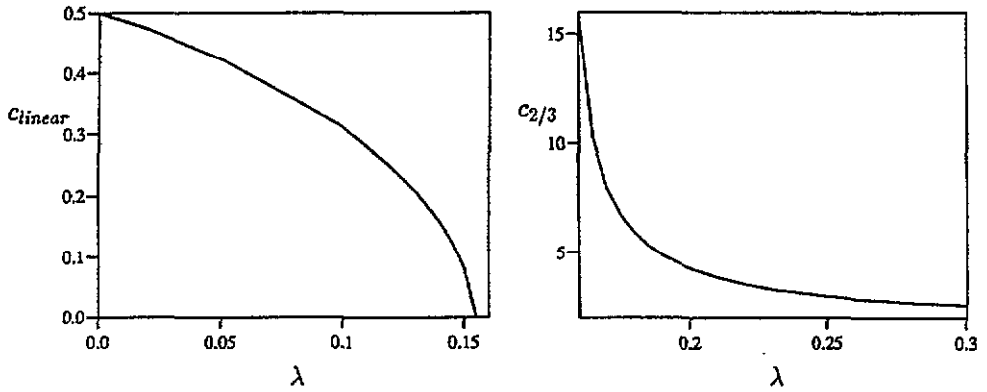

Figure 3. Scaling law prefactors for the optimal test set size for a data set of size $l = 0.6n$, $\sigma^2 = 0.8$, and $T = 0$.

are qualitatively similar, differing maximally for small m. This can be explained by using the approximation to the variance, and writing the upper bound as

$$\epsilon_{ub}(m|l) = \left(1 + \sqrt{\frac{2}{m}}\right)\epsilon_g + O\left(\frac{1}{\sqrt{mn}}\right). \tag{5.1}$$

We see from figure 2 that the optimal test set size, m^*, for both weight decays is $m^* \approx 24$‡. If we then increase the system size, n, we find that m^* scales like $n^{2/3}$. Further observations lead to the conclusion that, in general, there exist two scaling laws for m^*. One is the aforementioned $\frac{2}{3}$ scaling, and the other is linear. These scaling phases occur due to the existence of two competing local minima in the upper bound function. $\frac{2}{3}$ scaling implies a relatively small test set compared with linear scaling. We would expect that, for small noise levels, or large weight decay, the optimal test set size, m^*, would be minimal, and that as we increase the noise, m^* grows. This conjecture is borne out in figure 3, where we plot the prefactors of the linear and $\frac{2}{3}$ scaling laws for $l = 0.6n$, $\sigma^2 = 0.8$, $T = 0$. For $\lambda < 0.15$, the scaling is linear (m^* large), and the prefactor reduces quickly as λ tends to 0.15. There is then a transition to $\frac{2}{3}$ scaling (m^* small) as λ increases beyond this transition point. Initially, the prefactor for the $\frac{2}{3}$ scaling is large, reducing as λ increases.

‡ Equation (5.1) also holds for (spatially) correlated inputs on replacing ϵ_g with the generalization error calculated for correlated inputs, from which the modified optimal test set size can be calculated accordingly.

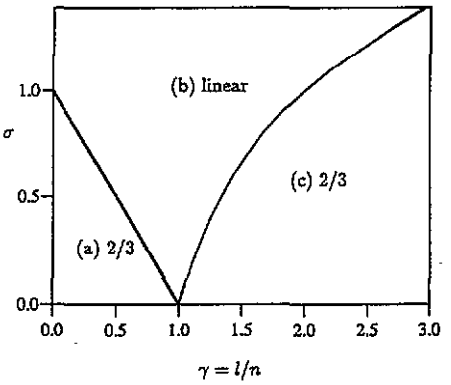

Figure 4. Phase diagram for the pseudo-inverse algorithm. In each region, the optimal test set size scales either linearly with n, or like $n^{2/3}$.

In general, isolating the phase boundaries involves the solution of a rather complicated expression and, as such, the boundary needs to be found numerically. For the pseudo-inverse algorithm, however, analytical expressions for the the large-n limit are readily found. In figure 4 we plot the phase diagram for the pseudo-inverse rule ($n \gg 1$). The values of the prefactors in the regions (a), (b), and (c) are, respectively,

$$\frac{1}{2^{1/3}}\left[\frac{\lambda(\alpha_{\rm tot}-1)(\sigma^2+(\alpha_{\rm tot}-1)^2)}{\sigma^2-(\alpha_{\rm tot}-1)^2}\right]^{2/3} \qquad \sigma+\alpha_{\rm tot}-1 \qquad \frac{1}{2^{1/3}}\left[\alpha_{\rm tot}(\alpha_{\rm tot}-1)\right]^{2/3}$$

where $\alpha_{\rm tot} = l/n$.

For large n, the variance is essentially zero, and the transition regions are simply given by consideration of the generalization error. If this is a monotonically decreasing function of α, such phase transitions will not exist as the 'optimal' scheme in this sense is to simply take the smallest test set. For a large enough value of λ, the generalization error will necessarily be monotonic, and we will have $\frac{2}{3}$ scaling. Thus, small test sets are reasonable for a large weight decay or small noise levels, in that the test error will be a good estimate of the generalization function.

6. Summary and outlook

We have calculated the variance in the test error of the linear perceptron due to randomness present in both the data set and algorithm. Where an exact calculation was not tractable, we showed that the variance can be very well approximated by a simple scaling of the square of the generalization error. We applied these results to address the question of the best assignment of a data set into a test and training set. We found that essentially there exist two different regions for the scaling of the optimal test set size with the system dimension: one linear, which operates for example for relatively large noise, and one $\frac{2}{3}$ scaling. That the variance is essentially trivial to approximate for the linear perceptron is undoubtably due to the simple mapping that it performs. Of future interest is the determination of the variance for nonlinear systems.

References

[1] Hertz A, Krogh J and Palmer G 1991 *Introduction to the Theory of Neural Computation* (Redwood City, CA: Addsion-Wesley)

[2] Dunmur A P and Wallace D J 1993 Learning and generalization in a linear perceptron stochastically trained with noisy data *J. Phys. A: Math. Gen.* **26** 5767–79

[3] Watkin T H L, Rau A and Biehl M 1993 The statistical mechanics of learning a rule *Rev. Mod. Phys* **65** 499–556
[4] Vapnik V N and Chervonenkis A Y 1971 On the uniform convergence of relative frequencies of events to their probabilities *Theor. Prob. Appl.* **16** 264–80
[5] Krogh A and Hertz J A 1992 Generalization in a linear perceptron in the presence of noise *J. Phys. A: Math. Gen.* **25** 1135–47
[6] Feller W 1970 *Introduction to Probability Theory and Its Applications* vol 1 (New York: Wiley) (1st edn 1950)
[7] Barber D, Saad D and Sollich P 1995 Test error fluctuations in finite linear perceptrons *Neural Comput.* to appear
[8] Hansen L K 1993 Stochastic linear learning: exact test and training error averages *Neural Networks* **6** 393–6
[9] Eaton M 1983 *Multivariate Statistics—A Vector Space Approach* (New York: Wiley)
[10] Sollich P 1994 Finite size effects in learning and generalization in linear perceptrons *J. Phys. A: Math. Gen.* **27** 7771–84
[11] Tarkowski W and Lewenstein M 1993 Learning from correlated examples in a perceptron *J. Phys. A: Math. Gen.* **26** 3669–79