Incorporating Input Information into Learning and Augmented Objective Functions

Thesis by

Zehra Cataltepe

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

1998
(Submitted May 18, 1998)

Acknowledgements

to the loving memory of my father Nurettin Kök

I would like to thank to my advisor Professor Yaser S. Abu-Mostafa for his encouragement, teaching, patience, time and friendship. Also many thanks to professors Alan Barr, Jehoshua Bruck, Mani Chandy and Alain Martin for reviewing this thesis.

The fine people in Learning Systems Group made life at Caltech a joyful "learning" experience. Dr. Amir Atiya, Dr. Malik Magdon-Ismail, Alexander Nicholson, Dr. Joseph Sill and Xubo Song, thanks for everything. Also thanks to Lucinda Acosta for solving all the problems, especially while I was busy writing, and to Robert Freeman for keeping the computers running. The proofs for theorems 2.3.3 and 5.1.1 were enhanced greatly after discussions with Dr. Magdon-Ismail.

Many thanks to my family, Annem and Babam and Ahmet and Rifat, for encouragement, love and support for a life time. Thanks also to my new family members, Anneciğim, Beybabam, Dr. Tayfun Cataltepe, Zeynep, Serap and little Zehra Nur. I was lucky enough to have excellent teachers Jale Gürgen, Varol Akman, Ayhan Esener, Ahmet Hamdi Aykut, Muzaffer Uçaş and Yüksel Kocaoğlu from middle school to college. Also thanks to my friends at Caltech Ayhan İrfanoğlu, Ahmet Kıraç, Murat Meşe, Dr. Gamze Erten, Hülya Peker and Bahadır Erimli, Dr. Eric Bax and Sam Roweis.

Heartful thanks to my beloved husband Dr. Tanju Cataltepe, for his constant love, support and sacrifices, through all these years and over long distances.

Last and most, thanks to God, for all of the above and anything and anybody else I may have forgotten.

Abstract

In many applications, some form of input information, such as test inputs or extra inputs, is available. We incorporate input information into learning by an augmented error function, which is an estimator of the out-of-sample error. The augmented error consists of the training error plus an additional term scaled by the augmentation parameter. For general linear models, we analytically show that the augmented solution has smaller out-of-sample error than the least squares solution. For nonlinear models, we devise an algorithm to minimize the augmented error by gradient descent, determining the augmentation parameter using cross validation.

Augmented objective functions also arise when hints are incorporated into learning. We first show that using the invariance hints to estimate the test error, and early stopping on this estimator, results in better solutions than the minimization of the training error. We also extend our algorithm for incorporating input information to the case of learning from hints.

Input information or hints are additional information about the target function. When the only available information is the training set, all models with the same training error are equally likely to be the target. In that case, we show that early stopping of training at any training error level above the minimum can not decrease the out-of-sample error. Our results are nonasymptotic for general linear models and the bin model, and asymptotic for nonlinear models. When additional information is available, early stopping can help.

Contents

List of Figures

Chapter 1
Introduction

1.1 Learning from Examples

Consider the following task: We have observed a set of input-outputs. In the future, we will observe some test inputs and will need to guess the outputs for these inputs. What should we do to produce the best possible response for the yet unobserved inputs?

The first step in solving this statistical pattern recognition [Duda, 1973, Bishop, 1995, Ripley, 1996] task is to choose a model class. The observed input-outputs are called the **training set**, and how well a model performs on a training set is measured by the **training error**. The goal is to have a model that has the smallest out-of-sample error. We will define the out-of-sample on a specific set of test inputs as the **test error**, and the expected error on a randomly drawn input as the **generalization error**. We assume that the training and test inputs are drawn independently and identically from the same input distribution.[1] The training and test outputs are obtained from the outputs of an unknown target function on the training and test inputs plus additional noise. Under these assumptions, the training error is an unbiased estimator [Ross, 1987] of the test error. Hence, the model minimizing the training error is chosen to be the **training solution**.

In this thesis, we will use general linear models and (artificial) neural networks[2] [Hertz et al., 1991, Ripley, 1996, Bishop, 1995] with one hidden layer of *tanh* hidden units and a linear output as the model class. A general linear model (figure 1.1)

[1] Note that in this case the expected value (with respect to inputs) of the test error is the generalization error.

[2] An artificial neural network is a nonlinear function approximator. Biological or psychological merits or relevance of neural networks are out of the scope of this thesis.

transforms the input $\mathbf{x} \in \mathcal{R}^{d'}$ using fixed transformation functions $\phi_i(\mathbf{x}), i = 0, \ldots, d$ and outputs the weighted sum $g_{\mathbf{w}}(\mathbf{x}) = \sum_{i=0}^{d} w_i \phi_i(\mathbf{x})$. The output $g_{\mathbf{w}}(\mathbf{x})$ is linear in the model parameters $\mathbf{w}$. Depending on the choice of $\phi_i(.)$'s it can be linear or nonlinear in inputs $\mathbf{x}$. The training solution is chosen to be the $\mathbf{w}$ that minimizes the training error of $g_{\mathbf{w}}$. We will compute the training solution for the general linear model using the least squares method [Hocking, 1996].

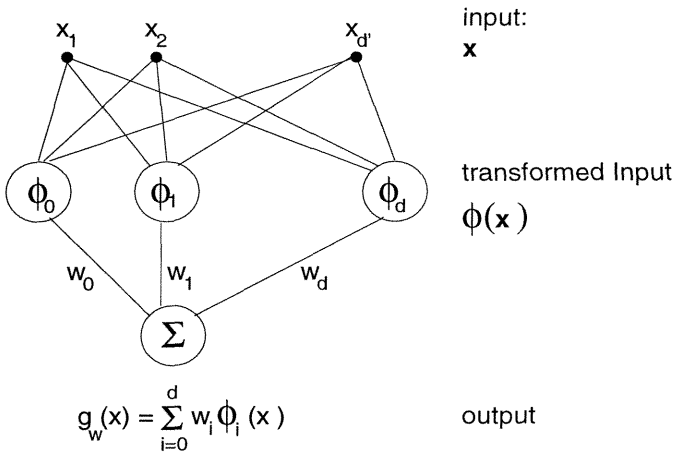

Figure 1.1: General linear model.

The neural network model (figure 1.2) will use a specific type of transformation function, namely *tanh* of weighted sum of inputs. Given enough number of hidden units, this model can approximate any continuous function [Cybenko, 1989] (universal approximation property). Both the input and output weights (parameters) of the neural network are adjustable. The output, $g_{\mathbf{v}}(\mathbf{x}) = v_0 + \sum_{i=1}^{d} v_i \tanh\left(v_{i,0} + \sum_{j=1}^{d'} v_{i,j} x_j\right)$, is nonlinear both in the weights $\mathbf{v}$ and the inputs $\mathbf{x}$. Due to nonlinearity of the model, the training solution for the neural network model has to be searched using an iterative optimization technique. Since it is quite common, we will use gradient descent starting from small initial weights and with adaptive learning rate [Battiti, 1989] as the optimization technique. Backpropagation [Rumellhart et al., 1986] is a method to conveniently compute the gradient of the neural network output with respect to weights. Some other algorithms to

find the training solution are conjugate gradient [Hestenes and Stiefel, 1952] and Levenberg-Marquardt [Levenberg, 1944, Marquardt, 1963] methods. The training error for a neural network has many local minima, hence the experimental results are meaningful only when one speficies the optimization technique and the starting point.

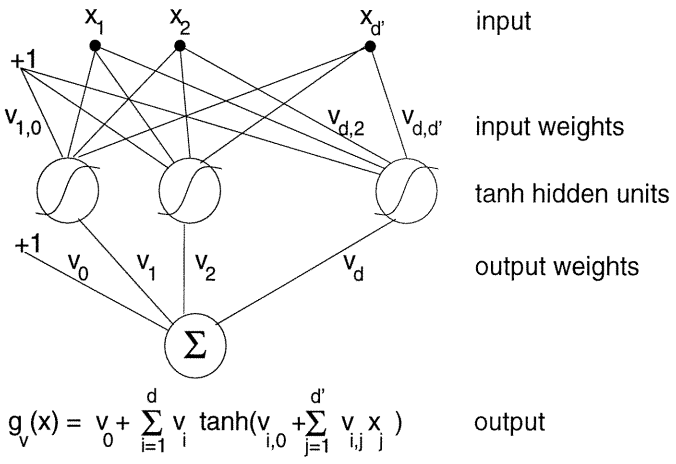

Figure 1.2: Neural network model.

The training error is an unbiased estimator of the test error. In some cases when the variance of this estimator is large, the training solution may overfit the training set. Overfitting can occur when the training data is noisy, the training set size is too small, or the model class is too complex. Overfitting can be described as having a considerably larger out-of-sample error **test error** than the training error. There are studies to bound the maximum difference between the training and test error for a model class, one of the most popular being the Vapnik-Chervonenkis (VC) dimension [Vapnik, 1982, Abu-Mostafa, 1989, Baum and Haussler, 1989] of the model class.

While the training error is being minimized iteratively, after some time, due to overfitting, minimization of the training error may increase the test error. This phenomenon is called **overtraining** (figure 1.3) or overfitting in time. Due to overtraining, the training solution can have a large test error. It is desirable to somehow prevent overtraining.

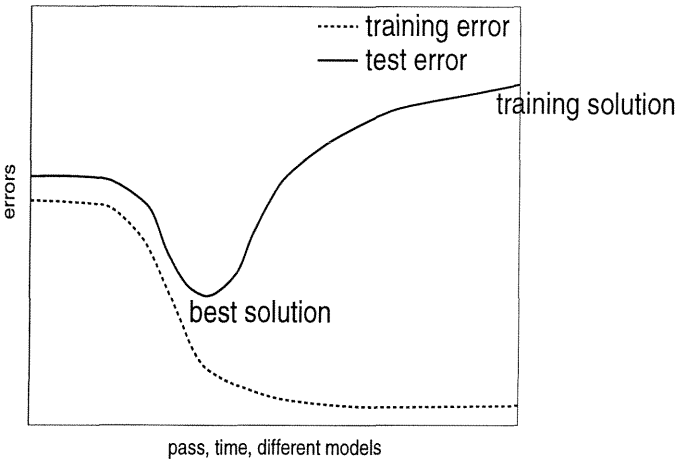

Figure 1.3: Overtraining.

Based on the assumption that the overtrained solution is less smooth (i.e. has larger derivatives with respect to inputs) than the best solution, methods that choose smoother models at the expense of a larger training error have been proposed. Weight decay [Reed, 1993] (also known as ridge regression or shrinkage estimation [Weisberg, 1980]) results in a model with smaller weights.[3] When training starts from small initial weights, early stopping of training is also likely to result in a solution with smaller weights than the training solution.

1.2 Questions

1.2.1 Early Stopping of Training

Both weight decay and early stopping of training using a validation set assume that the target model (the model that has the smallest test error) is a smoother model than the training solution. However, when the only available information about the target is the training set, all models with the same training error should be equally

[3]For the neural network, small weights, correspond to closer to linear and smoother models. When the input weights are small, *tanh* hidden units are close to their linear region. The derivative of the output of the neural network with respect to the input is proportional to the output weights.

likely to be the target. When this is the case, is there any method that could result in a better test error than the training solution? Is there a theoretical limitation?

1.2.2 Input Information

In some applications, we know either the test inputs, or some other input information, such as extra inputs or the input probability distribution.

For example, in the credit card approval task, the training inputs are the information about the past customers and the training outputs are whether they defaulted or not. The information about the current applicants are the test inputs. We are interested in a solution that will predict the default probability best, not for any applicant, but for the specific set of new applicants. Can we use these test inputs to obtain a solution that performs better than the training solution?

Sometimes, we have access to extra inputs that are not necessarily test inputs. For example, in blood cell recognition, many blood cells are drawn from the patient, but only a small subset are labeled by human experts. These labeled cells are used as the training set. Can we use the remaining extra inputs to obtain a solution that performs better than the training solution? What if we know the actual input probability distribution instead of just the extra inputs?

1.2.3 Hints

Now consider another scenario, we know the training set, and in addition we have some additional information, or **hints** [Abu-Mostafa, 1990], about the target function. For example, in character recognition, the characters remain invariant under translation, scaling and slight rotations. In other words we know the translation, scale and rotation invariance hints about the target function. Can we use these hints to obtain a solution better than the training solution?

1.3 Previous Work

1.3.1 Input Information

The availability of test inputs, extra inputs or input probability distribution is a special case of the missing data problem [Little, 1992] in statistics. The most popular approach for the solution of these problems is the EM (expectation maximization) algorithm [Dempster et al., 1977]. [Shahshahani and Landgrebe, 1994] and [Miller and Uyar, 1997] have applied the EM algorithm to the missing output labels problem. The EM algorithm requires density estimation for input-output distribution, however, especially for high dimensional spaces, density estimation is a very difficult problem.

[Vapnik, 1982, page 312] mentions that estimation of the target model for the test inputs only is an easier problem than the estimation of the target model everywhere in the input space. Since the goal is minimization of the test error, one should concentrate on the test error itself. Vapnik suggests "*transduction, deriving the values of the unknown function for points of interest from the given data*" and "*If you are limited to a restricted amount of information, do not solve the particular problem you need by solving a more general problem.*" [Vapnik, 1995, page 169].

There have also been studies on the value of unlabeled examples for classification problems. [Castelli and Cover, 1995] and [Castelli and Cover, 1996] show that labeled examples are exponentially more valuable than the unlabeled examples in reducing the classification error.

1.3.2 Hints

A hint is any additional information about the target [Abu-Mostafa, 1990, Abu-Mostafa, 1993a, Abu-Mostafa, 1993b, Abu-Mostafa, 1994]. Invariance hints [Fyfe, 1992, Cataltepe and Abu-Mostafa, 1993], monotonicity hint [Sill and Abu-Mostafa, 1997], smoothness hint [Ji et al., 1990], minimum Hamming distance between patterns [Al-Mashouq and Reed, 1991] are some of the hints that

have been studied previously.

Hints allow additional information to be expressed in terms of an error function (hint error). Just like the training error, the hint error can also be minimized. Minimization of both the training and the hint error simultaneously to achieve a good model is the goal of learning from hints. Different methods (schedules) have been suggested to learn from hints [Abu-Mostafa, 1994].

1.3.3 Early Stopping of Training

Early stopping has been studied by Wang et. al. [Wang et al., 1994] who analyzed the average optimal stopping time for general linear models (one hidden layer neural networks with a linear output and fixed input weights) and introduced and examined the effective size of the learning machine as training proceeds. Sjoberg and Ljung [Sjoberg and Ljung, 1995] linked early stopping using a validation set to regularization, and showed that emphasizing the validation set too much may result in an unregularized solution. Amari et. al. [Amari et al., 1997] determined the best validation set size in the asymptotic limit and showed that early stopping helps little in this limit even when the best stopping point is known. Dodier [Dodier, 1996] and Baldi and Chauvin [Baldi and Chauvin, 1991] investigated the behavior of validation error curves for linear problems, and the linear auto-association problem respectively.

1.4 Contributions

In this thesis we show that when the training set is the only available information, training solution is the best possible solution. When there is extra information, such as input information or hints available, we propose methods to incorporate them into learning to obtain a solution better than the training solution.

1.4.1 Incorporating Input Information into Learning

We develop an estimator of the test (generalization respectively) error using the test inputs (extra inputs or the input probability distribution respectively). The new estimator, **augmented error**, contains the training error, plus the augmented term scaled by the augmentation parameter.

For general linear models, we prove that the augmented solution is better than the training solution under certain conditions. We show that the optimal augmentation parameter increases as the training set size gets smaller or the signal-to-noise ratio decreases. We also devise the **substitution method** to find the augmentation parameter and prove that it results in a better augmented solution than the training solution.

We experimentally verify that the augmented solution is better than the training solution on real data sets. We compare the augmented solution to early stopping using a validation set and weight decay methods. Our simulations on linear models and linear and noisy targets show that the augmented error is consistently better than the least squares (training) solution. Although early stopping using a validation set solution is the best when the signal-to-noise ratio is very small, for large signal-to-noise ratio it performs very badly. Augmented solution is better or as good as the weight decay solution for all signal-to-noise ratios.

When the model is nonlinear, we propose two different methods of incorporating input information into learning. The first method is applicable to neural networks trained using gradient descent and modifies the output weigths only. The second method descends on the augmented error directly, choosing the augmentation parameter by means of leave-k-out cross validation. Our experiments show that both methods, usually, result in better solutions than the gradient descent solution, especially for small signal-to-noise ratios.

1.4.2 Learning from Hints

We use the invariance hints to estimate the test/generalization error. Early stopping on this estimator results in better solutions than the training solution.

Similar to the augmented error, the error function when learning from hints is also an augmented objective function. We extend the algorithm for descending on the augmented error to learning from hints.

1.4.3 No Free Lunch for Early Stopping

While additional information, such as test inputs or hints, can lead to better solutions, without any additional information, the training solution is the best possible solution.

We show that, when the only available information is the training set, and when the model is general linear model or the bin model, early stopping above the training error minimum increases the out-of-sample error. For nonlinear models the same result holds, but within a small enough neighborhood of the training error minimum and large training set size.

Weight decay and early stopping using a validation set assume that not only the training set, but also some smoothness property of the target are known.

1.5 Outline

The rest of the thesis is organized as follows.

In chapter 2, we derive the augmented error, our mechanism of incorporating test inputs and other type of input information into learning. In this chapter we analyze the augmented error and solution for general linear models. In section 2.3, we prove that when there is noise in the training data, the augmentation parameter that results in the best solution is nonzero. We also show that the best augmentation parameter decreases as the signal-to-noise ratio or the number of training examples increase. We describe the substitution method to find the augmentation parameter and prove that it results in an augmented solution better than the least squares solution in

section 2.4. The experimental verification of results on real and simulated data, and comparisons of the augmented solution to solutions obtained by other methods, such as weight decay, early stopping using a validation set, cross validation, take place in section 2.5.

Chapter 3 is on the incorporation of test inputs into learning for nonlinear models, specifically neural networks. In section 3.3 we use the substitution method to obtain an augmented solution for the output weights, keeping the input weights fixed. Section 3.6 discusses a method of descending on the augmented error instead of the training error alone. The augmentation parameters during the descent are determined by means of leave-k-out cross validation method. Section 3.7 discusses extensions of the augmented error approach to different loss functions, namely, entropic loss, maximum likelihood with input dependent noise variance and p-norm loss.

In chapter 4, we discuss incorporating hints into learning. In section 4.2, we estimate the out-of-sample error using the invariance hints and then early stop training based on this estimator. In section 4.3 we extend the gradient descent on the augmented error method to learning from hints.

Chapter 5 is about early stopping of training when the training set is the only available information. When there is no other information, all models with the same training error are equally likely to be the target. For this case and general linear models, in section 5.1, we prove that early stopping can not decrease the mean generalization error. Section 5.2 proves the same result for nonlinear models but around a training error minimum and for large training set sizes. In section 5.3 we review the bin model and prove that early stopping can not help for this model either. In section 5.4 we experimentally verify the early stopping results for general linear and neural network models. We also show that early stopping can help when additional information is available, for example in the case of weight decay or invariance hints.

1.6 Notation

We denote (column) vectors and matrices using lower and upper case bold letters respectively. $\mathbf{I}_{d\times d}$ is the $d \times d$ identity matrix, $\mathbf{A}^T$ and $\mathbf{A}^{-1}$ denote the transpose and inverse of matrix $\mathbf{A}$. *log* indicates base 10 logarithm.

We summarize the frequently used symbols below:

training set	$\{(\mathbf{x}_1, f_1), \ldots, (\mathbf{x}_N, f_N)\}$
a training input	$\mathbf{x}_n \in \mathcal{R}^{d'}$
a training output	$f_n \in \mathcal{R}$
test set	$\{(\mathbf{y}_1, h_1), \ldots, (\mathbf{y}_M, h_M)\}$
a test input	$\mathbf{y}_m \in \mathcal{R}^{d'}$
a test output	$h_m \in \mathcal{R}$
extra (unlabeled) inputs	$\{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$
learning model	$g_{\mathbf{v}}(\mathbf{x}) : \mathcal{R}^{d'} \rightarrow \mathcal{R}$
adjustable parameters of the learning model	$\mathbf{v}$

Chapter 2
Incorporating Input Information into Learning

In many applications of learning from examples such as disease diagnosis, medical image recognition, financial market forecasting and credit default prediction, some form of input information is available, in addition to the labeled training examples. The additional information can be:

- test inputs without outputs.
- inputs other than test inputs.
- input distribution information.

In this chapter, we provide an analytic solution for incorporating such input information into learning for general linear models. We cover the nonlinear case in the next chapter. The test inputs can provide valuable information not contained in the training error by itself. In figure 2.1 we show a sample training session and the behavior of training, test and augmented errors. The augmented error follows the test error as overtraining occurs, whereas the training error keeps decreasing.

We incorporate input information into learning by estimating the out-of-sample error. The new estimator, augmented error, is obtained by expanding the out-of-sample error. The input information is used to better estimate some of the terms in the expansion. Augmented error, which is computed only by using the training data and the input information, can result in a smaller out-of-sample error than simple training error.

Previous results on the use of input information include Shahshahani and Landgrebe [Shahshahani and Landgrebe, 1994] and Miller and Uyar

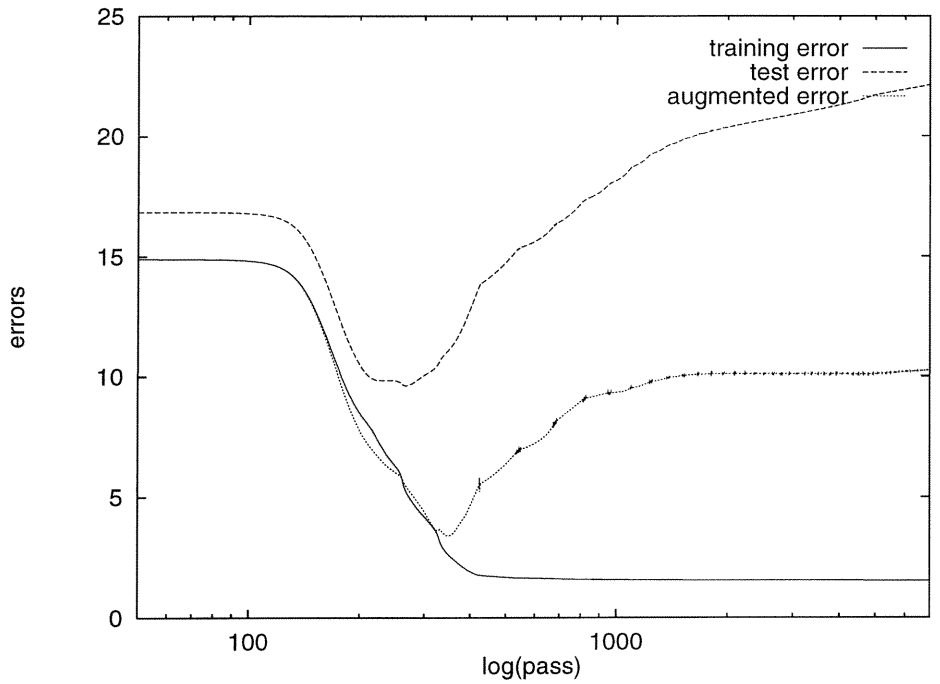

Figure 2.1: The augmented error, computed without looking at the test outputs at all, follows the test error as overtraining occurs.

[Miller and Uyar, 1997] who incorporate unlabeled examples into learning using EM (expectation maximization) [Dempster et al., 1977] algorithm for mixture models and classification problems. They show that unlabeled examples are useful especially when input dimensionality is high and the number of examples is small. [Miller and Uyar, 1997] reports favorable results when the inputs are test inputs. Castelli and Cover [Castelli and Cover, 1995] show that the labeled examples are exponentially more valuable than unlabeled examples in reducing the classification error. Our method is applicable for both classification and regression problems. The EM algorithm and other approaches to the solution of missing data problem in statistics (please see [Little, 1992] for a good review), requires density estimation for input-outputs, however, especially for high dimensional spaces, density estimation is a very difficult problem. The goal in learning-from-examples is to minimize the out-of-sample error [Vapnik, 1982, page 312], hence we concentrate on using the input information to directly estimate the out-of-sample error better. We have presented

some of our results for linear models in [Cataltepe and Magdon-Ismail, 1998].

The rest of the chapter is organized as follows: In section 2.1 we define the training and test errors and derive the augmented error. Section 2.2 describes the general linear models and derives the augmented solution for these models. In section 2.3 we prove that when there is noise in the training data, the augmentation parameter that results in the best solution is nonzero. We also show that the best augmentation parameter decreases as the signal-to-noise ratio or the number of training examples increase. The substitution method to find the augmentation parameter is described in section 2.4. We also prove that the substitution method results in an augmentated solution better than the least squares solution. The experimental results take place in section 2.5. We first show that the augmented solution results in better solution than the least squares solution on two real data sets from UCI Machine Learning Repository. Then we show that knowing the test inputs is more valuable than knowing the input probability distribution, which is more valuable than knowing an extra set of inputs. We compare the augmented solution to early stopping using a validation set and weight decay solutions. We also show that the substitution method results in solutions at least as good as the cross validation method would. Section 2.6 describes extensions of the augmented error. The two parameter augmented error of section 2.6.1 is a generalization of both the one parameter augmented error and the weight decay objective function. It also performs better than one parameter augmented error and hence we concentrate on the two parameter error in the next chapter. Another possible extension to the augmented error takes into account first order difference between test set outputs and training set outputs and is covered in section 2.6.2. Finally, section 2.7 summarizes the results in the chapter.

2.1 Augmented Error

Given a training set and a model class, the goal of learning from examples is to choose a model that performs best on out-of-sample data. Both the training (in-sample) and the test (out-of-sample) data are assumed to come from the same distribution, and

hence the training data is used to guide the search for a good model.

Let the **training set** be $\{(\mathbf{x}_1, f_1), \ldots, (\mathbf{x}_N, f_N)\}$ with inputs $\mathbf{x}_n$ and (possibly noisy) target outputs f_n. Let the model class be G and denote the model by $g_{\mathbf{v}} \in G$, where $\mathbf{v}$ is a parameter vector. When the loss function is quadratic, the **training error** of model $g_{\mathbf{v}}$ is:

$$E_0(g_{\mathbf{v}}) = \frac{1}{N}\sum_{n=1}^{N}(g_{\mathbf{v}}(\mathbf{x}_n) - f_n)^2 \tag{2.1}$$

When the noise in the training outputs is additive, independent, zero mean normal, the training error minimum is also the maximum likelihood solution [Bishop, 1995, pp. 39].

The out-of-sample error is measured by the **test error**. Let $\{(\mathbf{y}_1, h_1), \ldots, (\mathbf{y}_M, h_M)\}$ be an unknown **test set**, where $(\mathbf{y}, h)$ and $(\mathbf{x}, f)$ pairs are drawn from the same distribution. The test error of model $g_{\mathbf{v}}$ is:

$$E(g_{\mathbf{v}}) = \frac{1}{M}\sum_{m=1}^{M}(g_{\mathbf{v}}(\mathbf{y}_m) - h_m)^2$$

Expanding the test error:

$$E(g_{\mathbf{v}}) = \frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}^2(\mathbf{y}_m) - \frac{2}{M}\sum_{m=1}^{M} g_{\mathbf{v}}(\mathbf{y}_m) h_m + \frac{1}{M}\sum_{m=1}^{M} h_m^2 \tag{2.2}$$

The key observation here is that, when we know the test inputs, we know the first term exactly. Therefore we need only to approximate the remaining terms using the training data:

$$\begin{aligned} E(g_{\mathbf{v}}) &\approx \frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}^2(\mathbf{y}_m) - \frac{2}{N}\sum_{n=1}^{N} g_{\mathbf{v}}(\mathbf{x}_n) f_n + \frac{1}{N}\sum_{n=1}^{N} f_n^2 \\ &= E_0(g_{\mathbf{v}}) + \frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}^2(\mathbf{y}_m) - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}^2(\mathbf{x}_n) \end{aligned} \tag{2.3}$$

We scale the addition to the training error by an **augmentation parameter** α

to obtain a more general error function that we call the **augmented error**:

$$E_\alpha\left(g_{\mathbf{v}}\right) = E_0\left(g_{\mathbf{v}}\right) + \alpha\left(\frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}^2\left(\mathbf{y}_m\right) - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}^2\left(\mathbf{x}_n\right)\right) \tag{2.4}$$

where $\alpha = 0$ corresponds to the training error E_0.

When the identity of the test inputs is not known, the **generalization error** (mean square error) becomes the relevant measure of the out-of-sample error. Generalization error of a model is the expected error on an input drawn from the same distribution as the training inputs:

$$E_{gen}\left(g_{\mathbf{v}}\right) = \left\langle \left(g_{\mathbf{v}}(\mathbf{x}) - h(\mathbf{x})\right)^2 \right\rangle_{\mathbf{x}} \tag{2.5}$$

where $h(\mathbf{x})$ denotes a (possibly noisy) realization of the target output for input $\mathbf{x}$ and $\langle . \rangle_{\mathbf{x}}$ denotes expectation with respect to the input distribution $P_{\mathbf{x}}$. Note that the expected value of the test error with respect to the input distribution is the generalization error. In order to keep this correspondence, we have not taken expectation with respect to the noise distribution in the definition of the generalization error.

Expanding the generalization error as we did in equation (2.2), we obtain:

$$E_{gen}\left(g_{\mathbf{v}}\right) = \left\langle g_{\mathbf{v}}^2(\mathbf{x}) \right\rangle_{\mathbf{x}} - 2\left\langle g_{\mathbf{v}}(\mathbf{x}) h(\mathbf{x}) \right\rangle_{\mathbf{x}} + \left\langle h^2(\mathbf{x}) \right\rangle_{\mathbf{x}}$$

When the **input probability distribution** is known, the augmented error becomes:

$$E_\alpha\left(g_{\mathbf{v}}\right) = E_0\left(g_{\mathbf{v}}\right) + \alpha\left(\left\langle g_{\mathbf{v}}^2(\mathbf{x}) \right\rangle_{\mathbf{x}} - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}^2\left(\mathbf{x}_n\right)\right) \tag{2.6}$$

A third case of the augmented error arises when some **extra inputs** that are not necessarily test inputs are available. These can still be used to estimate the $\left\langle g_{\mathbf{v}}^2(\mathbf{x}) \right\rangle_{\mathbf{x}}$ term better. Let $\{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$ be the extra K inputs, then the augmented error for

this case becomes:

$$\begin{aligned} E_\alpha(g_{\mathbf{v}}) &= E_0(g_{\mathbf{v}}) + \alpha\left(\frac{1}{K+N}\left(\sum_{k=1}^{K} g_{\mathbf{v}}^2(\mathbf{z}_k) + \sum_{n=1}^{N} g_{\mathbf{v}}^2(\mathbf{x}_n)\right) - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}^2(\mathbf{x}_n)\right) \\ &= E_0(g_{\mathbf{v}}) + \alpha\frac{K}{K+N}\left(\frac{1}{K}\sum_{k=1}^{K} g_{\mathbf{v}}^2(\mathbf{z}_k) - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}^2(\mathbf{x}_n)\right) \end{aligned} \tag{2.7}$$

The best value of the augmentation parameter depends on a number of factors including the target function, the noise distribution and the model class. In sections 2.3 and 2.4, we will investigate the properties of the augmented solution and the augmentation parameter and devise a method to find the augmentation parameter for general linear models.

2.2 Augmented Solution

We will mostly focus on general linear models for the rest of this chapter. The general linear models are very powerful due to their transformation functions, and they are also very useful since they allow an analytical treatment of the augmented error and the augmented solution.

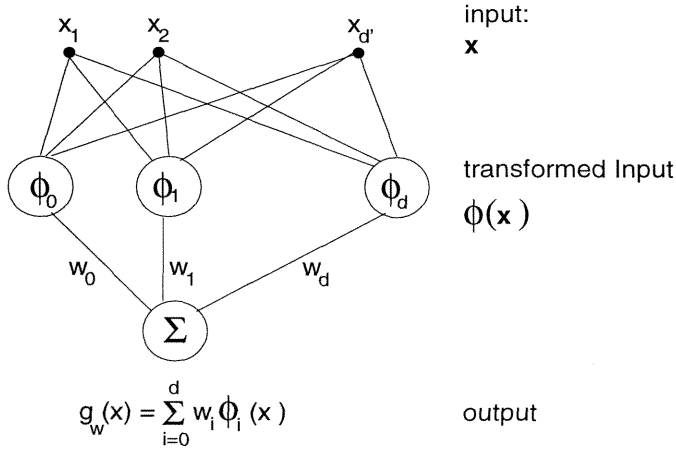

Figure 2.2: General linear model.

Let $\phi_i(\mathbf{x}) : \mathcal{R}^{d'} \rightarrow \mathcal{R}$, $i = 0, \ldots, d$ be fixed transformation (basis) functions

and let $\phi(\mathbf{x}) = [\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), \ldots, \phi_d(\mathbf{x})]^T$. We define a **general linear model** as $g_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})$ with fixed transformation functions $\phi(.)$ and adjustable parameters $\mathbf{w}$ (see figure 2.2). If $\phi_0(\mathbf{x}) = 1$ and $\phi_i(\mathbf{x}) = x_i, 1 \leq i \leq d' = d$ we obtain the usual linear model; if $\phi_i(\mathbf{x}) = \prod_{j=1}^{d'} x_j^{k_j}, k_j \geq 0$ we obtain a polynomial model. When the transformation functions are obvious from context, we will denote a general linear model only by the adjustable parameters $\mathbf{w}$.

Let $\mathbf{X}_{d' \times N}$ be the matrix of training inputs, $\mathbf{Y}_{d' \times M}$ be the matrix of test inputs and $\mathbf{f}_{N \times 1}$ contain the training outputs. Let $\mathbf{\Phi}_{x_{(d+1)\times N}} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]_{(d+1)\times N}$ and $\mathbf{\Phi}_{y_{(d+1)\times M}} = [\phi(\mathbf{y}_1), \ldots, \phi(\mathbf{y}_M)]_{(d+1)\times M}$ be the training and test inputs transformed by the transformation functions $\phi_i(.)$. We will denote $\frac{\mathbf{\Phi}_x \mathbf{\Phi}_x^T}{N}$ by $\mathbf{S}_x$ and $\frac{\mathbf{\Phi}_y \mathbf{\Phi}_y^T}{M}$ by $\mathbf{S}_y$ respectively. When $\mathbf{S}_x$ is full rank [1] the unique training error minimum is given by ([Hocking, 1996] and the section 2.8.1):

$$\mathbf{w}_0 = \mathbf{S}_x^{-1} \frac{\mathbf{\Phi}_x \mathbf{f}}{N} \tag{2.8}$$

The augmented error:

$$E_\alpha(\mathbf{w}) = E_0(\mathbf{w}) + \alpha \mathbf{w}^T (\mathbf{S}_y - \mathbf{S}_x) \mathbf{w}$$

is minimized at the **augmented solution** $\mathbf{w}_\alpha$:

$$\mathbf{w}_\alpha = (\mathbf{I} - \alpha \mathbf{R})^{-1} \mathbf{w}_0 \tag{2.9}$$

where $\mathbf{R} = \mathbf{I} - {\mathbf{S}_x}^{-1} \mathbf{S}_y$. When $\alpha = 0$, the augmented solution $\mathbf{w}_\alpha$ is equal to the simple training solution $\mathbf{w}_0$.

An intuition can be gained about the augmented solution by rewriting the augmented solution as $\mathbf{w}_\alpha = ((1-\alpha)\mathbf{S}_x + \alpha \mathbf{S}_y)^{-1} \frac{\mathbf{\Phi}_x \mathbf{f}}{N}$. The solution that minimizes the test error is $\mathbf{S}_y^{-1} \frac{\mathbf{\Phi}_y \mathbf{h}}{M}$, however we do not have access to the test outputs $\mathbf{h}$. The augmented solution modifies $\mathbf{S}_x^{-1}$ in the simple training solution, to make the solution

[1] Hence we restrict ourselves to problems where $N \geq d + 1$. Since the transformation functions are real valued, for most cases $\frac{\mathbf{\Phi}_x \mathbf{\Phi}_x^T}{N}$ is likely to be full rank.

closer to the test error minimum.

Let $\Sigma_{\phi(x)} = \left\langle \phi(\mathbf{x})\phi(\mathbf{x})^T \right\rangle_{\mathbf{x}}$. When the input probability distribution is known, the augmented solution is the same as equation (2.9) except now $\mathbf{R} = \mathbf{I} - \mathbf{S}_x{}^{-1}\Sigma_{\phi(x)}$. Similarly, when K extra inputs $\{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$ are known, the augmented solution has $\mathbf{R} = \frac{K}{K+N}\left(\mathbf{I} - \mathbf{S}_x{}^{-1}\mathbf{S}_z\right)$ where $\mathbf{S}_z = \frac{\mathbf{\Phi}_z\mathbf{\Phi}_z^T}{K}$ and $\mathbf{\Phi}_{z(d+1)\times K}$ denotes the extra inputs transformed by the transformation functions $\phi_i(.)$.

2.3 Properties of the Augmentation Parameter and the Augmented Solution

In the previous sections, we made no assumptions about the relationship between the training input and outputs. In this section, we will derive certain properties of the augmentation parameter and the augmented solution when training and test outputs are generated by a general linear model and then by adding zero mean, independent noise. We will assume that the training outputs are generated according to $\mathbf{f} = \mathbf{\Phi}_x^T\mathbf{w}^* + \epsilon$ where $\left\langle \epsilon\epsilon^T \right\rangle = \sigma_e^2\mathbf{I}_{N\times N}$, and the test outputs are generated according to $\mathbf{h} = \mathbf{\Phi}_y^T\mathbf{w}^* + \delta$ where $\left\langle \delta\delta^T \right\rangle = \sigma_e^2\mathbf{I}_{M\times M}$.

In this section, we will analyze certain properties of the augmentation parameter and the augmented solution. Since the specific realization of the noise in the training and test outputs are not known, conclusions that depend on the identity of the noise would not be useful. Hence, we will average out the noise and instead of the test error of the augmented solution $E\left(\mathbf{w}_\alpha\right)$, we will concentrate on the expected value of the test error of the augmented solution with respect to the training and test noise $\left\langle E\left(\mathbf{w}_\alpha\right)\right\rangle_{\epsilon,\delta}$.

$$\begin{aligned}
\left\langle E\left(\mathbf{w}_\alpha\right)\right\rangle_{\epsilon,\delta} &= \left\langle \frac{1}{M}\sum_{m=1}^{M}\left(\mathbf{w}_\alpha^T\phi(\mathbf{y}_m) - h_m\right)^2 \right\rangle_{\epsilon,\delta} \\
&= \left\langle \frac{1}{M}\sum_{m=1}^{M}\left(\mathbf{w}_\alpha^T\phi(\mathbf{y}_m) - \mathbf{w}^{*T}\phi(\mathbf{y}_m) - \delta_m\right)^2 \right\rangle_{\epsilon,\delta}
\end{aligned}$$

$$= \left\langle \frac{1}{M} \sum_{m=1}^{M} \left(\mathbf{w}_\alpha^T \phi(\mathbf{y}_m) - \mathbf{w}^{*T} \phi(\mathbf{y}_m) \right)^2 \right\rangle_\epsilon + \sigma_e^2$$
$$= \left\langle (\mathbf{w}_\alpha - \mathbf{w}^*)^T \mathbf{S}_y (\mathbf{w}_\alpha - \mathbf{w}^*) \right\rangle_\epsilon + \sigma_e^2$$

Similarly, the expected generalization error is:

$$\langle E_{gen}(\mathbf{w}_\alpha) \rangle_{\epsilon,\delta} = \left\langle (\mathbf{w}_\alpha - \mathbf{w}^*)^T \mathbf{\Sigma}_{\phi(x)} (\mathbf{w}_\alpha - \mathbf{w}^*) \right\rangle_\epsilon + \sigma_e^2$$

2.3.1 Test Inputs

The expected test error of the augmented solution with respect to the noise distribution is:

$$\begin{aligned} \langle E(\mathbf{w}_\alpha) \rangle_{\epsilon,\delta} &= \mathbf{w}^{*T} \left(\left(\mathbf{I} - \alpha \mathbf{R}^T \right)^{-1} - \mathbf{I} \right) \mathbf{S}_y \left((\mathbf{I} - \alpha \mathbf{R})^{-1} - \mathbf{I} \right) \mathbf{w}^* \\ &+ \frac{\sigma_e^2}{N} tr \left(\left(\mathbf{I} - \alpha \mathbf{R}^T \right)^{-1} \mathbf{S}_y (\mathbf{I} - \alpha \mathbf{R})^{-1} \mathbf{S}_x{}^{-1} \right) + \sigma_e^2 \end{aligned} \tag{2.10}$$

where we have used $\left\langle \epsilon^T A \epsilon \right\rangle_\epsilon = \sigma_e^2 tr(A)$, and $tr(A)$ denotes the trace of matrix A. When $\alpha = 0$, we have the simple training solution and:

$$\langle E(\mathbf{w}_0) \rangle_{\epsilon,\delta} = \frac{\sigma_e^2}{N} tr \left(\mathbf{S}_y \mathbf{S}_x^{-1} \right) + \sigma_e^2 \tag{2.11}$$

Now, we identify conditions under which the best augmentation parameter is nonzero, i.e. there exists an augmented solution which has lower expected test error than the simple training solution:

Theorem 2.3.1 *If $\sigma_e^2 > 0$ and $tr(\mathbf{R}\mathbf{S}_x^{-1}\mathbf{S}_y) \neq 0$, then there is an augmentation parameter $\alpha \neq 0$ that minimizes $\langle E(\mathbf{w}_\alpha) \rangle_{\epsilon,\delta}$.*

Proof: The derivative of the expected test error in equation (2.10) with respect to α is:

$$\frac{\partial \langle E(\mathbf{w}_\alpha) \rangle_{\epsilon,\delta}}{\partial \alpha} = 2\mathbf{w}^{*T} \left(\mathbf{I} - \alpha \mathbf{R}^T \right)^{-1} \mathbf{R}^T \left(\mathbf{I} - \alpha \mathbf{R}^T \right)^{-1} \mathbf{S}_y \left((\mathbf{I} - \alpha \mathbf{R})^{-1} - \mathbf{I} \right) \mathbf{w}^*$$

$$+ \quad 2\frac{\sigma_e^2}{N} tr\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{R}^T\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{S}_y(\mathbf{I}-\alpha\mathbf{R})^{-1}\mathbf{S}_x^{-1}\right) \tag{2.12}$$

where we have used the fact that $\frac{\partial \mathbf{B}^{-1}(\alpha)}{\partial \alpha} = -\mathbf{B}^{-1}(\alpha)\frac{\partial \mathbf{B}(\alpha)}{\partial \alpha}\mathbf{B}^{-1}(\alpha)$ for any matrix $\mathbf{B}$ whose elements are scalar functions of α [Hocking, 1996, page 684]. Hence the derivative at $\alpha = 0$ is:

$$\left.\frac{\partial \left\langle E\left(\mathbf{w}_\alpha\right)\right\rangle_{\epsilon,\delta}}{\partial \alpha}\right|_{\alpha=0} = 2\frac{\sigma_e^2}{N} tr\left(\mathbf{R}\mathbf{S}_x^{-1}\mathbf{S}_y\right)$$

If this derivative is nonzero then $\alpha = 0$ is not a local minimum of the expected test error and hence the expected test error is minimized at some nonzero α. □

Theorem 2.3.1 is important and also useful, since the check for the necessity of the nonzero augmentation parameter can be done only by looking at the available data.

The equivalent of theorem 2.3.1 can also be proven when the target and model do not necessarily have the same transformation functions.

Let the training outputs be generated according to $\mathbf{f} = \mathbf{\Psi}_x^T\mathbf{w}^* + \epsilon$ where $\mathbf{\Psi}_{x_{d''\times N}}$ denotes the training inputs transformed by the transformation functions $\psi_i(\mathbf{x}) : \mathcal{R}^{d'} \rightarrow \mathcal{R}$ for $i = 0, \ldots, d''$. Similarly, let the test outputs be $\mathbf{h} = \mathbf{\Psi}_y^T\mathbf{w}^* + \delta$. Let the training and test noise satisfy $\left\langle \epsilon\epsilon^T\right\rangle = \sigma_e^2\mathbf{I}_{N\times N}$ and $\left\langle \delta\delta^T\right\rangle = \sigma_e^2\mathbf{I}_{M\times M}$. In this case, the augmented solution becomes:

$$\begin{aligned}
\mathbf{w}_\alpha &= (\mathbf{I}-\alpha\mathbf{R})^{-1}\mathbf{w}_0 \\
&= (\mathbf{I}-\alpha\mathbf{R})^{-1}S_x^{-1}\frac{\mathbf{\Phi}_x\mathbf{f}}{N} \\
&= (\mathbf{I}-\alpha\mathbf{R})^{-1}S_x^{-1}\frac{\mathbf{\Phi}_x\left(\mathbf{\Psi}_x^T\mathbf{w}^*+\epsilon\right)}{N}
\end{aligned}$$

The expected test error of the augmented solution is:

$$\begin{aligned}
\left\langle E\left(\mathbf{w}_\alpha\right)\right\rangle_{\epsilon,\delta} &= \left\langle \frac{1}{M}\sum_{m=1}^{M}\left(\mathbf{w}_\alpha^T\phi(\mathbf{y}_m) - \mathbf{w}^{*T}\psi(\mathbf{y}_m)\right)^2\right\rangle_\epsilon + \sigma_e^2 \qquad (2.13)\\
&= \mathbf{w}^{*T}\frac{\mathbf{\Psi}_y\mathbf{\Psi}_y^T}{M}\mathbf{w}^* - 2\mathbf{w}^{*T}\frac{\mathbf{\Psi}_x\mathbf{\Phi}_x^T}{N}\mathbf{S}_x^{-1}\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\frac{\mathbf{\Phi}_y\mathbf{\Psi}_y^T}{M}\mathbf{w}^*
\end{aligned}$$

$$+ \quad \mathbf{w}^{*T}\frac{\mathbf{\Psi}_x\mathbf{\Phi}_x^T}{N}\mathbf{S}_x^{-1}\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{S}_y(\mathbf{I}-\alpha\mathbf{R})^{-1}\mathbf{S}_x^{-1}\frac{\mathbf{\Phi}_x\mathbf{\Psi}_x^T}{N}\mathbf{w}^*$$

$$+ \quad \frac{\sigma_e^2}{N}tr\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{S}_y(\mathbf{I}-\alpha\mathbf{R})^{-1}\mathbf{S}_x^{-1}\right)+\sigma_e^2 \tag{2.14}$$

Theorem 2.3.2 *When* $\mathbf{f}=\mathbf{\Psi}_x^T\mathbf{w}^*+\epsilon$ *and* $\mathbf{h}=\mathbf{\Psi}_y^T\mathbf{w}^*+\delta$, *if* $\frac{\sigma_e^2}{N}tr\left(\mathbf{R}\mathbf{S}_x^{-1}\mathbf{S}_y\right) \neq \mathbf{w}^{*T}\frac{\mathbf{\Psi}_x\mathbf{\Phi}_x^T}{N}\mathbf{S}_x^{-1}\mathbf{R}^T\left(\frac{\mathbf{\Phi}_y\mathbf{\Psi}_y^T}{M}-\mathbf{S}_y{\mathbf{S}_x}^{-1}\frac{\mathbf{\Phi}_x\mathbf{\Psi}_x^T}{N}\right)\mathbf{w}^*$ *then there is an augmentation parameter* $\alpha\neq 0$ *that minimizes* $\langle E\left(\mathbf{w}_\alpha\right)\rangle_{\epsilon,\delta}$.

Proof: follows from $\frac{\partial\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial\alpha}$ at $\alpha=0$. □

Unlike the previous theorem, in this case, the check for nonzero α, unfortunately, involves the function that generates the training and test data. An observation is that even if $\sigma_e^2=0$, the best α can be nonzero for this case. Also note that for a fixed $\mathbf{w}^*$ and σ_e^2 the best α is 0 only at the solution of an equation in high dimensional $\mathbf{\Phi}_x,\mathbf{\Phi}_y,\mathbf{\Phi}_x,\mathbf{\Phi}_y$ space.

For the rest of this chapter, unless made explicit otherwise, we will assume that the model and the target have the same transformation functions $\phi_i(.)$.

The following theorem gives an approximate formula for the best α:

Theorem 2.3.3 *If* N *is large,* $M\geq N$, $\phi(\mathbf{x}_n)$ *and* $\phi(\mathbf{y}_m)$ *are independent and identically distributed with* $\langle\phi(\mathbf{x})\rangle_{\mathbf{x}}=\mathbf{0}$, $\left\langle\phi(\mathbf{x})\phi^T(\mathbf{x})\right\rangle_{\mathbf{x}}=\sigma_x^2\mathbf{I}_{(d+1)\times(d+1)}$ *and* $\left\langle\left(\phi(\mathbf{x})\phi^T(\mathbf{x})\right)^2\right\rangle_{\mathbf{x}}=c\sigma_x^4\mathbf{I}_{(d+1)\times(d+1)}$ *for some constant* c *and* $\frac{\sigma_e^2}{N}\ll\sigma_x^2\mathbf{w}^{*T}\mathbf{w}^*$, *then* $\langle E\left(\mathbf{w}_\alpha\right)\rangle_{\epsilon,\delta,\mathbf{x},\mathbf{y}}$ *is minimized at:*

$$\alpha^*\approx\frac{d+1}{N}\frac{\sigma_e^2}{\sigma_x^2\mathbf{w}^{*T}\mathbf{w}^*} \tag{2.15}$$

Proof: is given in the appendix in section 2.8.2. □

This formula determines the behavior of the best α. The best α:

- decreases as the signal-to-noise ratio, $\frac{\sigma_x^2\mathbf{w}^{*T}\mathbf{w}^*}{\sigma_e^2}$, increases.
- decreases as the number of training examples, N, increases.

The goal of augmented error was estimation of the test error. The augmented error contains the training error plus the augmentation term scaled by α. When the signal-to-noise ratio increases or when there are more training examples, the training error becomes a better estimator of the test error and hence the contribution from the augmentation term is reduced by means of smaller α.

The following theorem tells us how much the expected test error of the augmented solution decreases compared to the simple training solution when the α minimizing the expected test error is used.

Theorem 2.3.4 *When the α minimizing the expected test error is used, the expected test error of the augmented solution decreases by $\mathcal{O}\left(\frac{\sigma_e^4}{N^2}\right)$ if N and M are large and $\phi(\mathbf{x}_n)$ and $\phi(\mathbf{y}_m)$ are independent and identically distributed with $\langle\phi(\mathbf{x})\rangle_{\mathbf{x}} = \mathbf{0}$, $\left\langle\phi(\mathbf{x})\phi^T(\mathbf{x})\right\rangle_{\mathbf{x}} = \sigma_x^2\mathbf{I}_{(d+1)\times(d+1)}$ and $\left\langle\left(\phi(\mathbf{x})\phi^T(\mathbf{x})\right)^2\right\rangle_{\mathbf{x}} = c\sigma_x^4\mathbf{I}_{(d+1)\times(d+1)}$ for some constant c. More precisely $\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta} - \langle E(\mathbf{w}_0)\rangle_{\epsilon,\delta} \approx -\frac{\sigma_e^4}{N^2}\frac{tr^2\left(\mathbf{R}{\mathbf{S}_x}^{-1}\mathbf{S}_y\right)}{\mathbf{w}^{*T}\mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{w}^*} \leq 0$.*

Proof: is given in the appendix in section 2.8.3. □

2.3.2 Input Probability Distribution and Extra Inputs

When outputs are generated according to $\mathbf{f} = \mathbf{\Phi}_x^T\mathbf{w}^* + \epsilon$ where $\left\langle\epsilon\epsilon^T\right\rangle_\epsilon = \sigma_e^2\mathbf{I}_{(d+1)\times(d+1)}$, similar to equation (2.10), the expected value of the generalization error of the augmented solution is:

$$\begin{aligned}\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta} &= \mathbf{w}^{*T}\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}-\mathbf{I}\right)\mathbf{\Sigma}_{\phi(x)}\left((\mathbf{I}-\alpha\mathbf{R})^{-1}-\mathbf{I}\right)\mathbf{w}^* \\ &+ \frac{\sigma_e^2}{N}tr\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{\Sigma}_{\phi(x)}(\mathbf{I}-\alpha\mathbf{R})^{-1}{\mathbf{S}_x}^{-1}\right)+\sigma_e^2 \qquad (2.16)\end{aligned}$$

The expected generalization error of the simple training solution is:

$$\langle E_{gen}(\mathbf{w}_0)\rangle_{\epsilon,\delta} = \frac{\sigma_e^2}{N}tr\left(\mathbf{\Sigma}_{\phi(x)}{\mathbf{S}_x}^{-1}\right)+\sigma_e^2 \qquad (2.17)$$

Note that $\mathbf{R} = \mathbf{I} - {\mathbf{S}_x}^{-1}\mathbf{\Sigma}_{\phi(x)}$ when the input probability distribution is known, and $\mathbf{R} = \frac{K}{K+N}\left(\mathbf{I} - {\mathbf{S}_x}^{-1}\mathbf{S}_z\right)$ when extra inputs $\{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$ are known.

Counterparts of theorems 2.3.1 to 2.3.4 for the test inputs are stated below for input probability distribution or extra inputs. The proofs are very similar to the proofs of theorems 2.3.1 to 2.3.4 and are given in the appendix in sections 2.8.4 to 2.8.6.

Theorem 2.3.5 *If $\sigma_e^2 > 0$ and $tr\left(\mathbf{R}{\mathbf{S}_x}^{-1}\mathbf{\Sigma}_{\phi(x)}\right) \neq 0$, then there is an augmentation parameter $\alpha \neq 0$ that minimizes $\langle E_{gen}\left(\mathbf{w}_\alpha\right)\rangle_{\epsilon,\delta}$.*

Theorem 2.3.6 *If N is large, $\phi(\mathbf{x}_n)$ are independent and identically distributed with $\langle\phi(\mathbf{x})\rangle_{\mathbf{x}} = \mathbf{0}$, $\left\langle\phi(\mathbf{x})\phi^T(\mathbf{x})\right\rangle_{\mathbf{x}} = \sigma_x^2\mathbf{I}_{(d+1)\times(d+1)}$ and $\left\langle\left(\phi(\mathbf{x})\phi^T(\mathbf{x})\right)^2\right\rangle_{\mathbf{x}} = c\sigma_x^4\mathbf{I}_{(d+1)\times(d+1)}$ for some constant c and $\frac{\sigma_e^2}{N} \ll \sigma_x^2\mathbf{w}^{*T}\mathbf{w}^*$, then $\langle E_{gen}\left(\mathbf{w}_\alpha\right)\rangle_{\epsilon,\delta,\mathbf{x}}$ is minimized at:*

$$\alpha^* \approx \frac{d+1}{N}\frac{\sigma_e^2}{\sigma_x^2\mathbf{w}^{*T}\mathbf{w}^*} \tag{2.18}$$

Theorem 2.3.7 *When the α minimizing the expected generalization error is used, the expected generalization error of the augmented solution decreases by $\mathcal{O}\left(\frac{\sigma_e^4}{N^2}\right)$, when N is large and $\phi(\mathbf{x}_n)$ are independent and identically distributed with $\langle\phi(\mathbf{x})\rangle_{\mathbf{x}} = \mathbf{0}$, $\left\langle\phi(\mathbf{x})\phi^T(\mathbf{x})\right\rangle_{\mathbf{x}} = \sigma_x^2\mathbf{I}_{(d+1)\times(d+1)}$ and $\left\langle\left(\phi(\mathbf{x})\phi^T(\mathbf{x})\right)^2\right\rangle_{\mathbf{x}} = c\sigma_x^4\mathbf{I}_{(d+1)\times(d+1)}$ for some constant c. More precisely $\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta} - \langle E_{gen}(\mathbf{w}_0)\rangle_{\epsilon,\delta} \approx -\frac{\sigma_e^4}{N^2}\frac{\sigma_x^2 tr^2\left(\mathbf{R}{\mathbf{S}_x}^{-1}\right)}{\mathbf{w}^{*T}\mathbf{R}^T\mathbf{R}\mathbf{w}^*} \leq 0$.*

2.3.3 The Parameter Error of the Augmented Solution

Just like the test error and the generalization error, the parameter error E_{param} measures the goodness of a solution:

$$E_{param}(\hat{\mathbf{w}}) = ||\mathbf{w}^* - \hat{\mathbf{w}}||^2 = (\mathbf{w}^* - \hat{\mathbf{w}})^T(\mathbf{w}^* - \hat{\mathbf{w}})$$

When $\mathbf{\Sigma}_{\phi(x)} = \mathbf{I}_{(d+1)\times(d+1)}$, the parameter error equals the generalization error minus σ_e^2.

The parameter error of the augmented solution $\mathbf{w}_\alpha$ is:

$$\begin{aligned} E_{param}(\mathbf{w}_\alpha) &= \mathbf{w}^{*T}\left(\mathbf{I} - \left(\mathbf{I} - \alpha\mathbf{R}^T\right)^{-1}\right)\left(\mathbf{I} - \left(\mathbf{I} - \alpha\mathbf{R}\right)^{-1}\right)\mathbf{w}^* \\ &\quad + \frac{\sigma_e^2}{N} tr\left(\mathbf{S}_x{}^{-1}\left(\mathbf{I} - \alpha\mathbf{R}^T\right)^{-1}\left(\mathbf{I} - \alpha\mathbf{R}\right)^{-1}\right) \end{aligned}$$

Hence the parameter error of the simple training (least squares) solution $\mathbf{w}_0$ is:

$$E_{param}(\mathbf{w}_0) = \frac{\sigma_e^2}{N} tr\left(\mathbf{S}_x{}^{-1}\right)$$

The following theorem shows that there is an $\alpha \neq 0$ that minimizes the expected parameter error as well:

Theorem 2.3.8 *If* $\sigma_e^2 > 0$ *and* $tr\left(\mathbf{S}_x{}^{-1}\mathbf{R}\right) \neq 0$, *then there is an* $\alpha \neq 0$ *that minimizes* $\langle E_{param}(\mathbf{w}_\alpha)\rangle_\epsilon$.

Proof: is similar to the proof of theorem 2.3.1 and is given in the appendix in section 2.8.7.□

2.3.4 $bias^2 + variance$

We have considered three different types of out-of-sample error in the previous sections: test error, generalization error and parameter error. The expected value with respect to the test output noise for all these errors can be written as:

$$\langle E_{\mathbf{P}}(\mathbf{w}_\alpha)\rangle_\delta = (\mathbf{w}_\alpha - \mathbf{w}^*)^T\mathbf{P}(\mathbf{w}_\alpha - \mathbf{w}^*) + c$$

where $\mathbf{P} = \frac{\mathbf{\Phi}_y\mathbf{\Phi}_y^T}{M}$ for test error, $\mathbf{P} = \mathbf{\Sigma}_{\phi(x)}$ for generalization error and $\mathbf{P} = \mathbf{I}_{(d+1)\times(d+1)}$ for the parameter error, $c = \sigma_e^2$ for the test and generalization errors and $c = 0$ for the parameter error.

The expected out-of-sample error of any estimator $\hat{\mathbf{w}}$ of $\mathbf{w}^*$, can be written as

[Geman et al., 1992, Bishop, 1995]:

$$\begin{aligned}
\langle\langle E_{\mathbf{P}}(\hat{\mathbf{w}})\rangle_{\delta}\rangle_{\epsilon} &= \left\langle (\mathbf{w}^* - \hat{\mathbf{w}})^T \mathbf{P} (\mathbf{w}^* - \hat{\mathbf{w}}) \right\rangle_{\epsilon} + c \\
&= \left\langle (\mathbf{w}^* - \langle\hat{\mathbf{w}}\rangle_{\epsilon} + \langle\hat{\mathbf{w}}\rangle_{\epsilon} - \hat{\mathbf{w}})^T \mathbf{P} (\mathbf{w}^* - \langle\hat{\mathbf{w}}\rangle_{\epsilon} + \langle\hat{\mathbf{w}}\rangle_{\epsilon} - \hat{\mathbf{w}}) \right\rangle_{\epsilon} + c \quad (2.19) \\
&= (\mathbf{w}^* - \langle\hat{\mathbf{w}}\rangle_{\epsilon})^T \mathbf{P} (\mathbf{w}^* - \langle\hat{\mathbf{w}}\rangle_{\epsilon}) + \left\langle (\hat{\mathbf{w}} - \langle\hat{\mathbf{w}}\rangle_{\epsilon})^T \mathbf{P} (\hat{\mathbf{w}} - \langle\hat{\mathbf{w}}\rangle_{\epsilon}) \right\rangle_{\epsilon} + c \\
&= bias^2_{\epsilon,\mathbf{P}}(\hat{\mathbf{w}}) + variance_{\epsilon,\mathbf{P}}(\hat{\mathbf{w}}) + c
\end{aligned}$$

The cross-term $\left\langle (\mathbf{w}^* - \langle\hat{\mathbf{w}}\rangle_{\epsilon})^T \mathbf{P} (\hat{\mathbf{w}} - \langle\hat{\mathbf{w}}\rangle_{\epsilon}) \right\rangle_{\epsilon}$ from equation 2.19 dropped because $\mathbf{w}^*$ is independent of ϵ and $\langle\langle\hat{\mathbf{w}}\rangle_{\epsilon} - \hat{\mathbf{w}}\rangle_{\epsilon} = \langle\hat{\mathbf{w}}\rangle_{\epsilon} - \langle\hat{\mathbf{w}}\rangle_{\epsilon} = 0$.

The simple training (least squares) solution $\mathbf{w}_0$ is and unbiased estimator of the target $\mathbf{w}^*$, because

$$\begin{aligned}
\langle\mathbf{w}_0\rangle_{\epsilon} &= \left\langle \left(\frac{\mathbf{\Phi}_x \mathbf{\Phi}_x^T}{N}\right)^{-1} \frac{\mathbf{\Phi}_x \mathbf{f}}{N} \right\rangle_{\epsilon} \\
&= \left\langle \left(\frac{\mathbf{\Phi}_x \mathbf{\Phi}_x^T}{N}\right)^{-1} \frac{\mathbf{\Phi}_x \left(\mathbf{\Phi}_x^T \mathbf{w}^* + \epsilon\right)}{N} \right\rangle_{\epsilon} \\
&= \mathbf{w}^*
\end{aligned}$$

For zero mean normal and independent noise, the simple training solution $\mathbf{w}_0$ is the minimum variance unbiased linear estimator of $\mathbf{w}^*$ [Montgomery and Peck, 1991, Gauss-Markov Theorem].

On the other hand, the augmented solution is a biased estimator, since $\mathbf{w}_\alpha = (\mathbf{I} - \alpha\mathbf{R})^{-1}\mathbf{w}_0$. Since there is an $\alpha \neq 0$ for which $\langle E_{\mathbf{P}}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta} < \langle E_{\mathbf{P}}(\mathbf{w}_0)\rangle_{\epsilon,\delta}$ (theorems 2.3.1, 2.3.5 and 2.3.8), and the $bias^2$ of the augmented solution is larger than the $bias^2$ of the simple training solution, it follows that the augmented solution has a smaller *variance* than the simple training solution.

As N and M get large, $\mathbf{R} = \mathbf{I} - \mathbf{S}_x{}^{-1}\mathbf{S}_y \to 0$ and $\mathbf{w}_\alpha = (\mathbf{I} - \alpha\mathbf{R})^{-1}\mathbf{w}_0 \to \mathbf{w}_0$. Hence, for large N and M, the *bias* and *variance* of $\mathbf{w}_\alpha$ approach 0, making $\mathbf{w}_\alpha$ an unbiased and consistent estimator of $\mathbf{w}^*$.

2.4 Substitution Method to Find a Good Augmentation Parameter

In this section, we propose a method to find the best α minimizing the test error of $\mathbf{w}_\alpha$, given only the training and test inputs $\mathbf{X}$ and $\mathbf{Y}$, and the training outputs $\mathbf{f}$. The method also covers the case when the input information is the input probability distribution or extra inputs.

Equation (2.10) gives a formula for the expected test error which we want to minimize:

$$\begin{aligned}\langle E(\mathbf{w}_\alpha)\rangle_\epsilon &= \mathbf{w}^{*T}\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}-\mathbf{I}\right)\mathbf{S}_y\left(\left(\mathbf{I}-\alpha\mathbf{R}\right)^{-1}-\mathbf{I}\right)\mathbf{w}^* \\ &+ \frac{\sigma_e^2}{N}tr\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{S}_y(\mathbf{I}-\alpha\mathbf{R})^{-1}\mathbf{S}_x{}^{-1}\right)+\sigma_e^2\end{aligned}$$

However, in practice, we do not know the target $\mathbf{w}^*$ and the noise variance σ_e^2. If we had an estimator $\mathbf{w}$ of the target $\mathbf{w}^*$ then we could estimate σ_e^2 by [2]: $\frac{\left(\mathbf{\Phi}_x^T\mathbf{w}-\mathbf{f}\right)^T\left(\mathbf{\Phi}_x^T\mathbf{w}-\mathbf{f}\right)}{N-d-1}$ [Montgomery and Peck, 1991, page 16]. Then we could replace $\mathbf{w}^*$ by the estimator $\mathbf{w}$ and the σ_e^2 by its estimator and find the α minimizing the resulting approximation to the expected test error.

We will consider the simple training solution $\mathbf{w}_0$ in equation (2.8) as an estimator of the $\mathbf{w}^*$ and estimate the σ_e^2 accordingly. Note that $\frac{\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)^T\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)}{N-d-1}$ is an unbiased estimator of the noise variance σ_e^2. Although $\mathbf{w}_0$ is an unbiased estimator of the target $\mathbf{w}^*$, the first term in equation (2.8) is overestimated. But still, as the following theorem states, the α found by means of this substitution results in a solution better than the simple training solution. While theorem 2.3.1 established a way to understand the existence of a best $\alpha \neq 0$, this theorem shows a method that gives consistently better results than the simple training solution. Only by means of accessing the available data (training input-outputs and the test inputs) and nothing else, the augmented solution found this way is superior to the simple training solution:

[2] If a better estimate of the noise variance is available, that could be used as well.

Theorem 2.4.1 *Let α' minimize the expected test error in equation (2.10) with substitution $\mathbf{w}_0 \rightarrow \mathbf{w}^*$ and $\frac{\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)^T\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)}{N-d-1} \rightarrow \sigma_e^2$. The expected test error of the augmented solution with the augmentation parameter α' is less than that of the simple training solution, i.e. $\langle E(\mathbf{w}_{\alpha'})\rangle_{\epsilon,\delta} \leq \langle E(\mathbf{w}_0)\rangle_{\epsilon,\delta}$ if N and M are large, $\frac{\sigma_e^2}{N} tr\left(2\mathbf{R}^{T^2}\mathbf{S}_y\mathbf{S}_x^{-1} + \mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{S}_x^{-1}\right) > 0$ and $\frac{\sigma_e^2}{N} tr\left(\mathbf{R}\mathbf{S}_x^{-1}\mathbf{S}_y\right) \neq 0$, and $\phi(\mathbf{x}_n)$ and $\phi(\mathbf{y}_m)$ are independent and identically distributed with $\langle\phi(\mathbf{x})\rangle_{\mathbf{x}} = \mathbf{0}$, $\left\langle\phi(\mathbf{x})\phi^T(\mathbf{x})\right\rangle_{\mathbf{x}} = \sigma_x^2\mathbf{I}_{(d+1)\times(d+1)}$ and $\left\langle\left(\phi(\mathbf{x})\phi^T(\mathbf{x})\right)^2\right\rangle_{\mathbf{x}} = c\sigma_x^4\mathbf{I}_{(d+1)\times(d+1)}$ for some constant c.*

Proof: is given in the appendix in section 2.8.8. Figure 2.3 illustrates the theorem. □

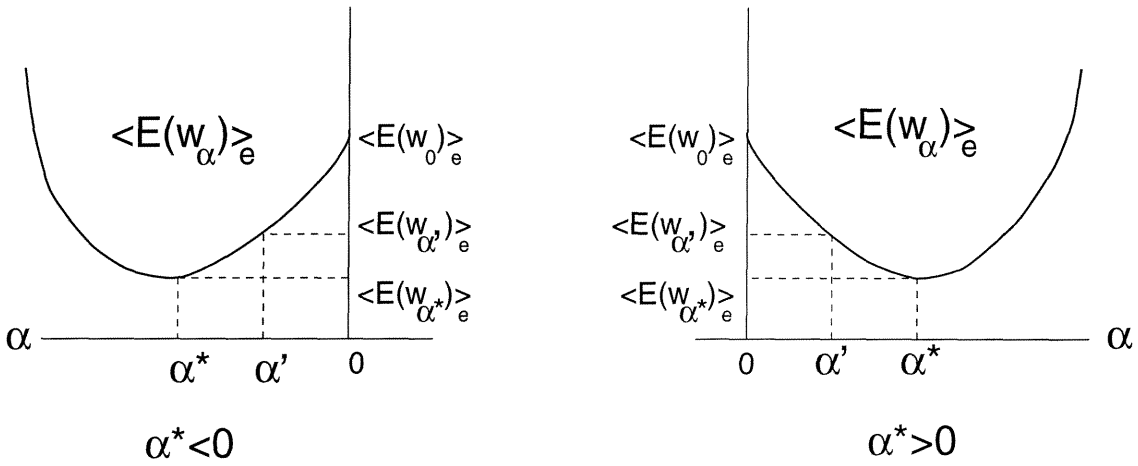

Figure 2.3: When α^* minimizes the expected test error and $|\alpha'| < |\alpha^*|$, the expected test error of $\mathbf{w}_{\alpha'}$ is smaller than the expected test error of the simple training solution $\mathbf{w}_0$.

When input probability distribution or extra inputs are available, the $\mathbf{w}_0$ substitution and minimization with respect to α of the expected generalization (equation (2.16)):

$$\begin{aligned}\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta} &= \mathbf{w}^{*T}\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}-\mathbf{I}\right)\mathbf{\Sigma}_{\phi(x)}\left(\left(\mathbf{I}-\alpha\mathbf{R}\right)^{-1}-\mathbf{I}\right)\mathbf{w}^* \\ &+ \frac{\sigma_e^2}{N} tr\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{\Sigma}_{\phi(x)}(\mathbf{I}-\alpha\mathbf{R})^{-1}{\mathbf{S}_x}^{-1}\right)+\sigma_e^2\end{aligned}$$

instead of the expected test error is performed. Note that, again, $\mathbf{R} = \mathbf{I} - {\mathbf{S}_x}^{-1}\mathbf{\Sigma}_{\phi(x)}$

when the input probability distribution is known, and $\mathbf{R} = \frac{K}{K+N}\left(\mathbf{I} - \mathbf{S}_x{}^{-1}\frac{\mathbf{\Phi_z}\mathbf{\Phi_z}^T}{K}\right)$ when extra inputs $\{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$ are known.

The equivalent of theorem 2.4.1 is valid for the input probability distribution or extra inputs cases as well:

Theorem 2.4.2 *Let α' minimize the expected generalization error in equation (2.16) with substitution $\mathbf{w}_0 \rightarrow \mathbf{w}^*$ and $\frac{\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)^T\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)}{N-d-1} \rightarrow \sigma_e^2$. Then for large N and small α', the expected generalization error of the augmented solution with the augmentation parameter α' is less than that of the simple training solution, i.e. $\langle E_{gen}(\mathbf{w}_{\alpha'})\rangle_{\epsilon,\delta} \leq \langle E_{gen}(\mathbf{w}_0)\rangle_{\epsilon,\delta}$ if N is large, $\frac{\sigma_e^2}{N}tr\left(2\mathbf{R}^{T^2}\mathbf{\Sigma}_{\phi(x)}\mathbf{S}_x^{-1} + \mathbf{R}^T\mathbf{\Sigma}_{\phi(x)}\mathbf{R}\mathbf{S}_x^{-1}\right) > 0$ and $\frac{\sigma_e^2}{N}tr\left(\mathbf{R}\mathbf{S}_x^{-1}\mathbf{\Sigma}_{\phi(x)}\right) \neq 0$, and $\phi(\mathbf{x}_n)$ are independent and identically distributed with $\langle\phi(\mathbf{x})\rangle_{\mathbf{x}} = \mathbf{0}$, $\left\langle\phi(\mathbf{x})\phi^T(\mathbf{x})\right\rangle_{\mathbf{x}} = \sigma_x^2\mathbf{I}_{(d+1)\times(d+1)}$ and $\left\langle\left(\phi(\mathbf{x})\phi^T(\mathbf{x})\right)^2\right\rangle_{\mathbf{x}} = c\sigma_x^4\mathbf{I}_{(d+1)\times(d+1)}$ for some constant c.*

Proof: is given in the appendix in section 2.8.9. □

2.5 Experimental Results

In this section, we compare the performance of the augmented solution to the simple training solution on real and simulated data. We find the augmentation parameter by the substitution method described in the previous section. We also compare the augmented solution to two commonly used regularization methods: early stopping of training and weight decay. All models for these experiments are linear. Finally we compare the substitution method of finding the augmentation or weight decay parameter, to the ordinary cross validation method.

2.5.1 Liver and Bond Data

First we compare the augmented solution to the simple training solution on liver data[3] and bond data[4] . The liver database consists of 345 examples with 6 inputs and 1

[3]ftp://ftp.ics.uci.edu/pub/machine-learning-databases/liver-disorders/bupa.data

[4]We thank Dr. John Moody of OGI for providing the bond data.

output. The inputs are different blood test results and the output is the number of drinks per day. The bond data consists of 196 examples with 11 inputs and 1 output. The inputs are financial properties of the bond and the output is the rating of the bond from AAA to B- or lower.

For both data sets, we repeated the following experiment 1000 times. We randomly selected M examples for the test set, and selected N for the training set among the remaining examples. We found the augmentation parameter by the substitution method. The mean test error (over 1000 different partitionings) of the augmented solution and the simple training solution is shown in figure 2.4. The augmented solution is always better than the simple training solution. The advantage of the augmented solution is more pronounced for small training set size.

2.5.2 Comparison of Different Types of Input Information

We experimentally compared the usefulness of test inputs, extra inputs and the input probability distribution. For $\sigma_e^2 = 1, d = 11, N = 30$ and a certain value of SNR [5] and M, we performed the following experiment (again average of 1000 runs are shown):

select $\mathbf{w}^*$ from a zero mean normal.

scale $\mathbf{w}^*$ to have $\frac{\sigma_x^2 \mathbf{w}^{*T} \mathbf{w}^*}{\sigma_e^2} = SNR$.

select $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_y$ from zero mean unit variance normals.

find the training and test outputs by adding zero mean unit variance normal noise to $\mathbf{\Phi}_x^T \mathbf{w}^*$ and $\mathbf{\Phi}_y^T \mathbf{w}^*$.

compute the simple training solution $\mathbf{w}_0$.

Find augmentation parameter α by the substitution method, and compute the augmented solution $\mathbf{w}_\alpha$.

print the test error of $\mathbf{w}_0$ and $\mathbf{w}_\alpha$.

We repeated the same experiments when M extra inputs and the input probability distribution are known. For these cases we printed the generalization errors of $\mathbf{w}_0$

[5] where $SNR = \frac{\sigma_x^2 \mathbf{w}^{*T} \mathbf{w}^*}{\sigma_e^2}$ is the signal-to-noise ratio.

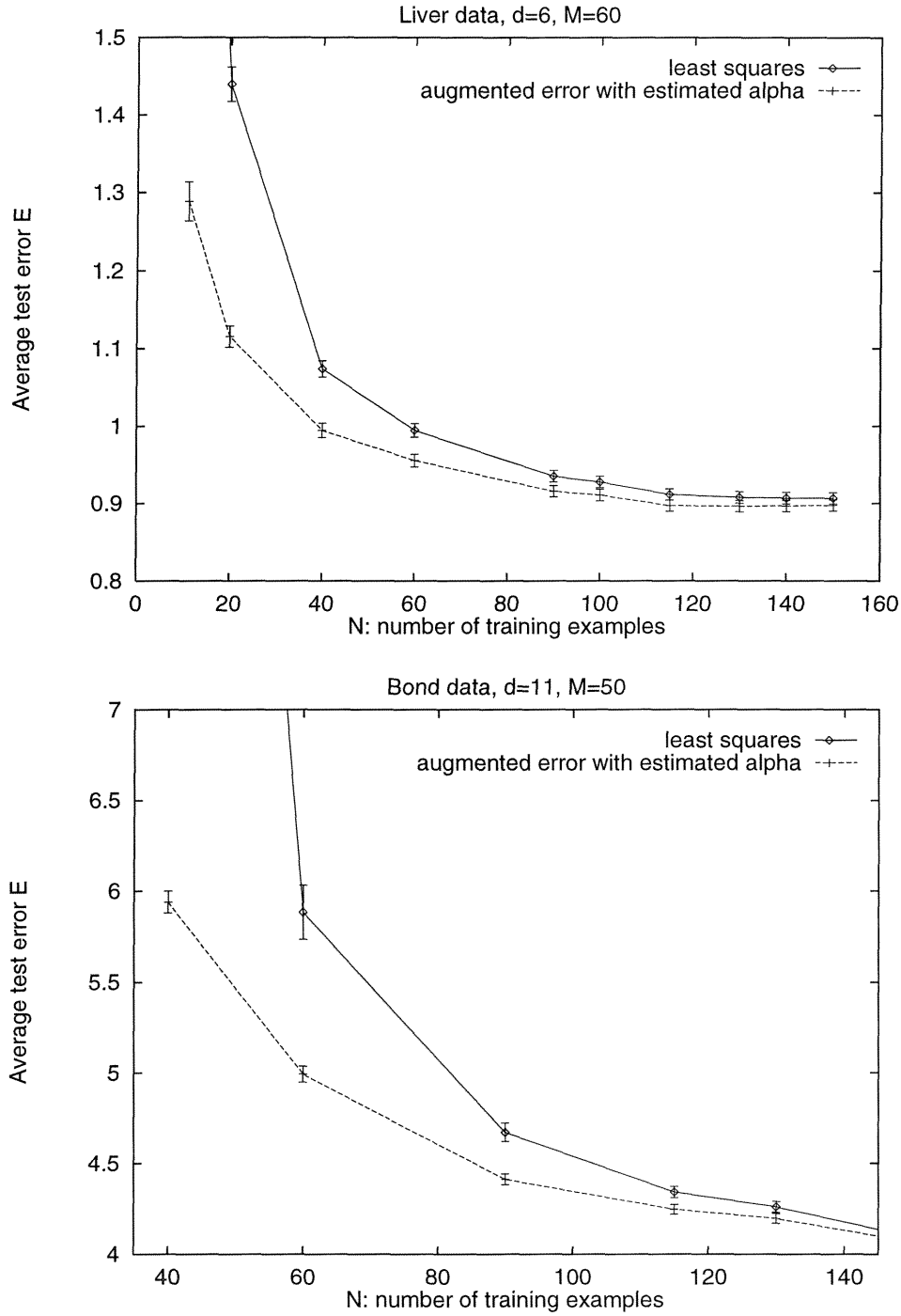

Figure 2.4: Performance of the augmented and simple training solution on liver and bond data.

and $\mathbf{w}_\alpha$.

For $M = 11$ and 100, the $\frac{E(\mathbf{w}_\alpha)}{E(\mathbf{w}_0)}$ for the test inputs and $\frac{E_{gen}(\mathbf{w}_\alpha)}{E_{gen}(\mathbf{w}_0)}$ for extra inputs and input probability distribution information are shown in figure 2.5. For all the SNR values and the type of input information, the ratio is less than 1, hence augmented solution is better than the simple training solution. The M test inputs are more valuable than M extra inputs. Not surprisingly, knowing the input probability distribution is more valuable than knowing M extra inputs. As the number of extra inputs decreases, the augmented solution gets closer to the simple training solution in performance.

2.5.3 Augmented Solution and Early Stopping Solution

When the learning model is more capable than the target model, or the number of training examples is small, while minimizing the training error, after some time, due to overfitting the test/generalization error starts to increase. This phenomenon is called **overtraining**. **Early stopping of training** is a method that aims to prevent overtraining. Early stopping has been shown to be equivalent to regularization under certain conditions [Sjoberg and Ljung, 1995]. We will investigate early stopping further in chapter 5.

Early stopping using a validation set operates as follows: The whole training data is partitioned into two sets: training set of size N_t and validation set of size N_v. Starting from small initial weights $\mathbf{w}$, the learning algorithm minimizes the training error, while the validation error is monitored. The model at the minimum of the validation error is chosen to be the **early stopping solution**.

The validation and training set sizes N_v and N_t play a crucial role for the success of early stopping. Although the best validation set size is known in the asymptotic limit, it is not known for the non asymptotic case [Amari et al., 1997]. For our experiments, we use validation set sizes of $N_v = \frac{N}{3}$ and $N_v = \frac{N}{6}$.

Taking the average of early stopping solutions for different partitionings of the data reduces the variance (please see section 2.3.4) while keeping the bias the

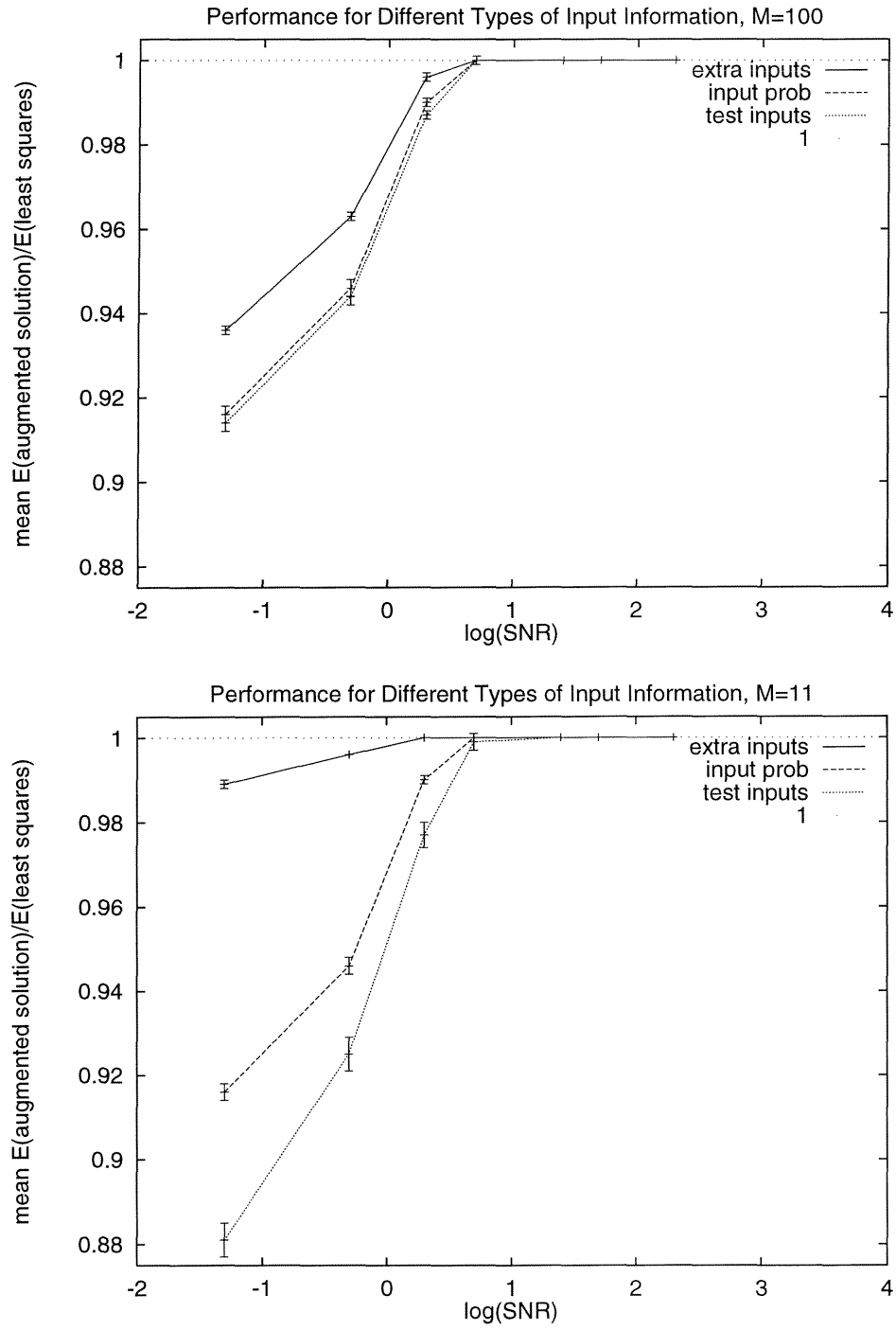

Figure 2.5: Performance of the augmented solution for different types of input information: test inputs, extra inputs and input probability distribution information.

same [Bishop, 1995]. Hence the average of the early stopping solutions has smaller out-of-sample error than the average out-of-sample error of the individual early stopping solutions. In our experiments we investigate the average early stopping solution as well.

In figure 2.6, we show $\frac{E(\mathbf{w}_\alpha)}{E(\mathbf{w}_0)}$, $\frac{E(\mathbf{w}_{e.s})}{E(\mathbf{w}_0)}$ and $\frac{E(mean(\mathbf{w}_{e.s}))}{E(\mathbf{w}_0)}$ for $\sigma_e^2 = 1, d = 11, N = 30$ and $M = 100$ and varying SNR. First of all taking the mean of early stopping solutions always results in better performance. While early stopping solution is better than both the simple training and the augmented solutions for small SNR, it performs worse than the simple training for large SNR. The augmented solution is guaranteed to perform better than the simple training solution, whereas the early stopping solution lacks that property.

2.5.4 Augmented Solution and Weight Decay Solution

When there is additive noise in the training data, the expected magnitude of the simple training solution is larger than the magnitude of the target ($\left\langle \mathbf{w}_0^T \mathbf{w}_0 \right\rangle_\epsilon > \mathbf{w}^{*T}\mathbf{w}^*$). Weight decay aims to shrink the size of the solution by means of adding a term $\mathbf{w}^T\mathbf{P}\mathbf{w}$ to the training error. The matrix $\mathbf{P}$ is positive definite (usually $\mathbf{P} = \mathbf{I}$), hence this term penalizes large weights. For a good introduction to weight decay, please see [Bishop, 1995, Krogh and Hertz, 1992].

For the general linear model, weight decay minimizes the error function:

$$E_\lambda(\mathbf{w}) = E_0(\mathbf{w}) + \lambda \mathbf{w}^T\mathbf{w} \tag{2.20}$$

where λ is the nonnegative weight decay parameter.

The choice of the weight decay parameter is very important for the success of weight decay. We determine the weight decay parameter using the same substitution method we use for the augmentation parameter.

Figure 2.7 shows $\frac{E(\mathbf{w}_\alpha)}{E(\mathbf{w}_0)}$ for the augmented solution $\mathbf{w}_\alpha$ and $\frac{E(\mathbf{w}_\lambda)}{E(\mathbf{w}_0)}$ for the weight decay solution $\mathbf{w}_\lambda$. For these experiments as well we used $\sigma_e^2 = 1, d = 11, N = 30$ and report the average of 1000 runs. Again for large SNR the augmented solution

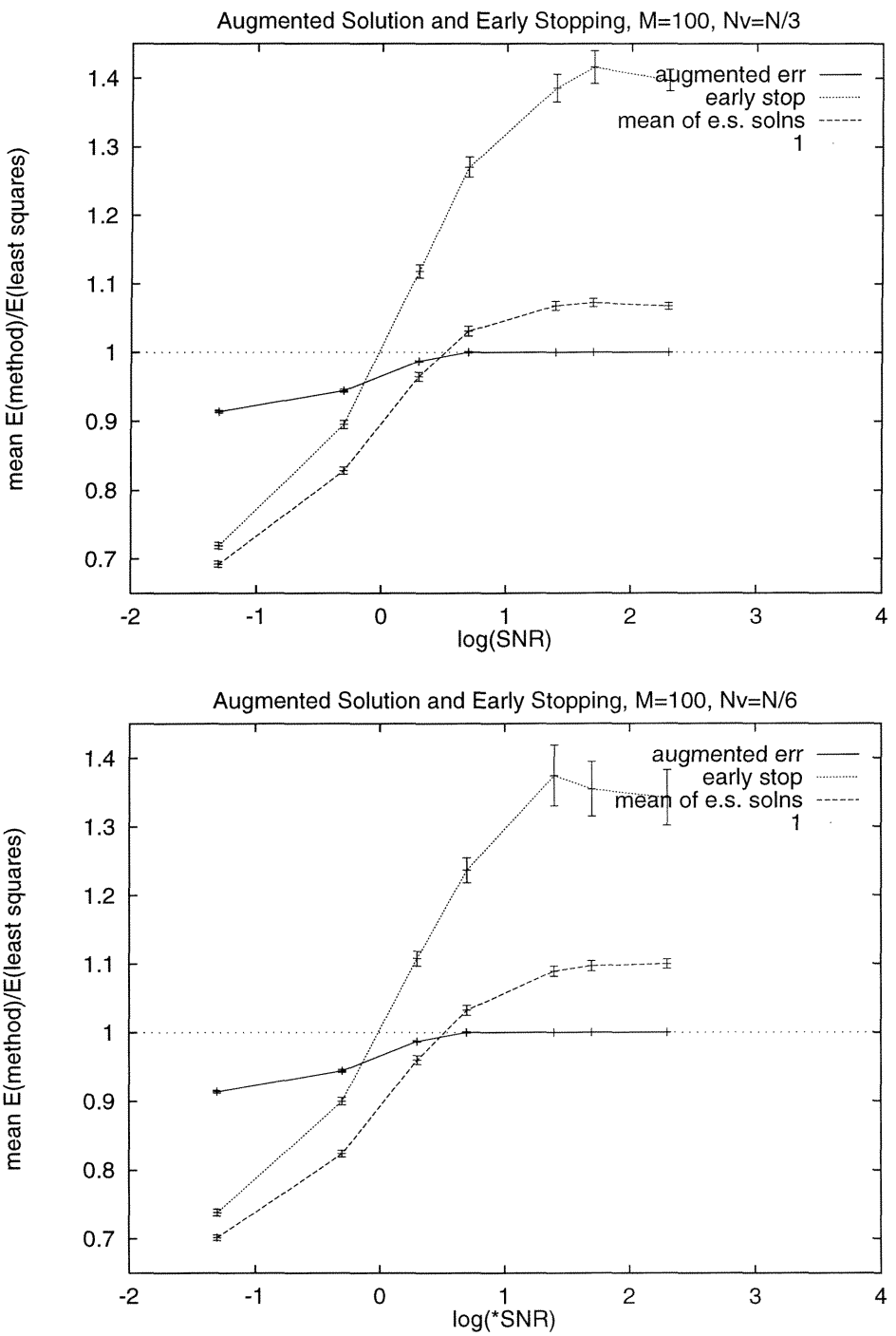

Figure 2.6: Performance of the augmented solution and the early stopping solution.

results in smaller test error than than the weight decay solution. As in the case of early stopping, weight decay may result in solutions worse than the simple training solution for some SNR range, whereas the augmented solution is consistently better than the simple training solution.

2.5.5 Substitution Method and the Ordinary Cross Validation Method

Ordinary cross validation method [Wahba, 1990] is a well-known method for finding a good ridge parameter in statistics. It chooses the parameter that minimizes the average error of the leave-1-out solution on the left out examples. Let $\mathbf{w}_\alpha^k$ minimize:

$$\frac{1}{N}\sum_{n=1,n\neq k}^{N}\left(\mathbf{w}^T\phi(\mathbf{x}_n)-f_n\right)^2+\mathbf{w}^T\mathbf{P}(\alpha)\mathbf{w} \tag{2.21}$$

Then the ordinary cross validation chooses the α that minimizes:

$$\begin{aligned}V_0(\alpha) =&\frac{1}{N}\sum_{k=1}^{N}\left(\mathbf{w}_\alpha^{k\,T}\phi(\mathbf{x}_k)-f_k\right)^2\\ =&\frac{1}{N}\sum_{k=1}^{N}\left(\left(\mathbf{f}^T\mathbf{\Phi}_x^T-f_k\phi^T(\mathbf{x}_k)\right)\left(\mathbf{\Phi}_x\mathbf{\Phi}_x^T-\phi(\mathbf{x}_k)\phi^T(\mathbf{x}_k)+N\mathbf{P}(\alpha)\right)^{-1}\phi(\mathbf{x}_k)-f_k\right)^2\end{aligned}$$

where for weight decay solution $\mathbf{P}(\alpha)=\alpha\mathbf{I}$, and for the augmented solution $\mathbf{P}(\alpha)=\alpha\left(\mathbf{S}_y-\mathbf{S}_x\right)$. We have experimentally compared the ordinary cross validation method to the substitution method for weight decay and augmented solutions. The results are shown in figure 2.8. For the augmented solution, the substitution method always gave better results. For weight decay, although the ordinary cross validation was superior to the substitution method for small SNR, substitution method was better for large SNR.

We have also experimented with the generalized cross validation method [Wahba, 1990], the performance of generalized and ordinary cross validation were very similar for the simulations we performed.

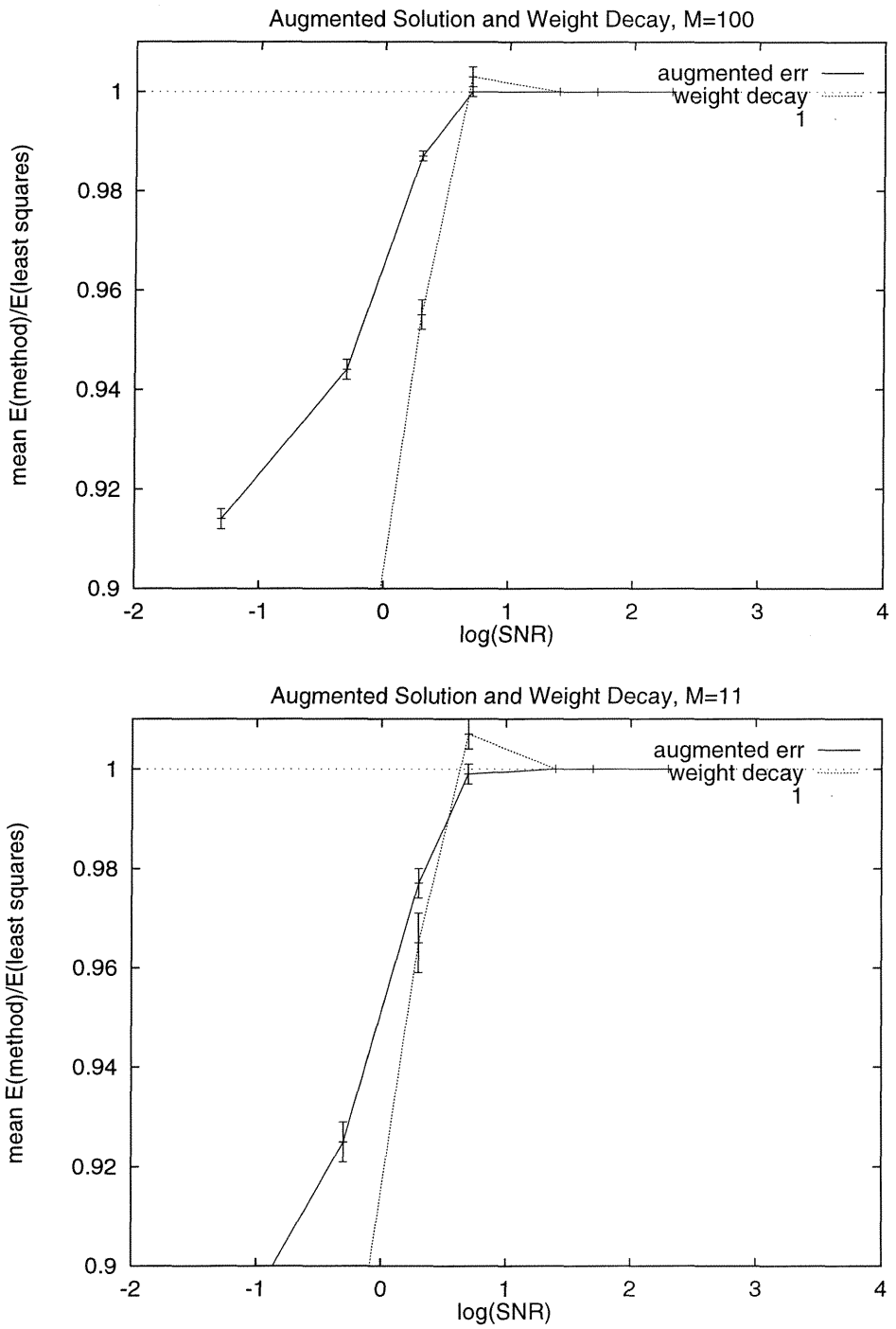

Figure 2.7: Performance of the augmented solution and the weight decay solution.

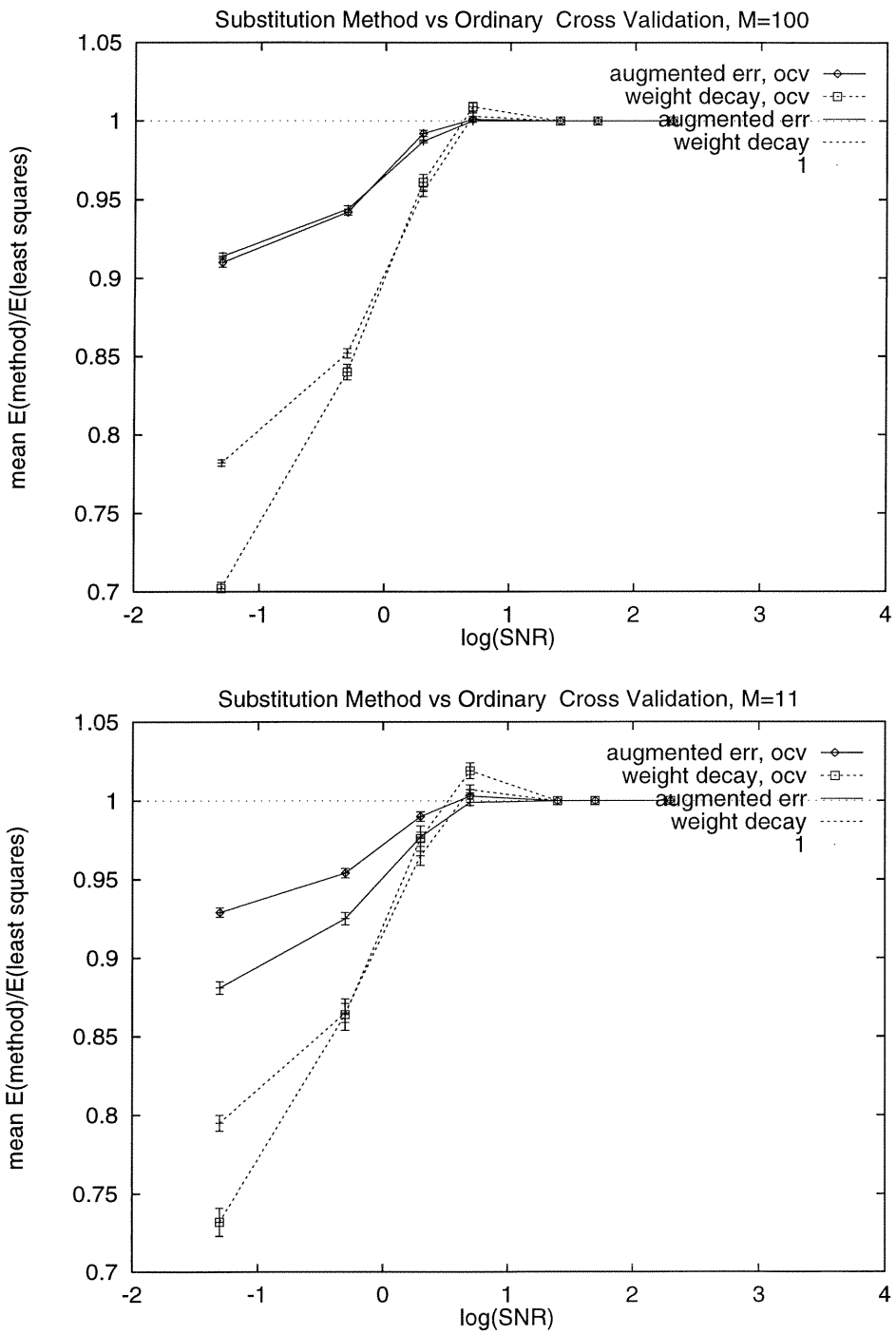

Figure 2.8: Performance comparison of substitution and ordinary cross validation methods to find the augmentation and weight decay parameters.

2.6 Enhanced Forms of Augmented Error

In this section we suggest two other ways of forming the augmented error.

2.6.1 Two Augmentation Parameters

In equation (2.4), when the test inputs are known, we have chosen to have only one augmentation parameter and have the augmented error as:

$$E_\alpha\left(g_{\mathbf{v}}\right) = E_0\left(g_{\mathbf{v}}\right) + \alpha\left(\frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}^2\left(\mathbf{y}_m\right) - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}^2\left(\mathbf{x}_n\right)\right)$$

A more general form of augmented error is formed by using two parameters:

$$E_{\alpha_1,\alpha_2}\left(g_{\mathbf{v}}\right) = E_0\left(g_{\mathbf{v}}\right) + \alpha_1\frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}^2\left(\mathbf{y}_m\right) - \alpha_2\frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}^2\left(\mathbf{x}_n\right) \tag{2.22}$$

For the linear model, the augmented error and the augmented solution become:

$$\begin{aligned} E_{\alpha_1,\alpha_2}\left(\mathbf{w}\right) &= E_0\left(\mathbf{w}\right) + \mathbf{w}^T\left(\alpha_1\mathbf{S}_y - \alpha_2\mathbf{S}_x\right)\mathbf{w} \\ \mathbf{w}_{\alpha_1,\alpha_2} &= \mathbf{Q}_{\alpha_1,\alpha_2}\mathbf{w}_0 \end{aligned} \tag{2.23}$$

where $\mathbf{Q}_{\alpha_1,\alpha_2} = \left(\mathbf{I} - \alpha_2\mathbf{I} + \alpha_1{\mathbf{S}_x}^{-1}\mathbf{S}_y\right)^{-1}$. The expected test error of the augmented solution is then,

$$\begin{aligned} \left\langle E(\mathbf{w}_{\alpha_1,\alpha_2})\right\rangle_\epsilon &= \mathbf{w}^{*T}\left(\mathbf{Q}_{\alpha_1,\alpha_2}^T - \mathbf{I}\right)\mathbf{S}_y\left(\mathbf{Q}_{\alpha_1,\alpha_2} - \mathbf{I}\right)\mathbf{w}^* \\ &+ \frac{\sigma_e^2}{N} tr\left(\mathbf{Q}_{\alpha_1,\alpha_2}^T\mathbf{S}_y\mathbf{Q}_{\alpha_1,\alpha_2}{\mathbf{S}_x}^{-1}\right) + \sigma_e^2 \end{aligned} \tag{2.24}$$

As in the one parameter augmentation case, α_1 and α_2 can be found by the substitution $\mathbf{w}_0 \to \mathbf{w}^*$ and $\frac{\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)^T\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)}{N-d-1} \to \sigma_e^2$ in the expected test error and minimizing the resulting approximation to the expected test error with respect to both α_1 and α_2 simultaneously.

Both weight decay (with decay term $\mathbf{w}^T\mathbf{S}_x\mathbf{w}$) and the one parameter augmented

solution are special cases of the two parameter augmented solution. We compare the two parameter augmented solution to one parameter augmented solution, weight decay solution and the early stopping solution in figure 2.9. The two parameter augmented solution performs better than all the other methods, especially for large SNR. In the experiments shown the augmentation parameters were determined by the substitution method.

2.6.2 First and Second Order Differences

The augmented error, in some sense, forces the sample mean of the model outputs squared on the training and test inputs to be close to each other. How about the closeness of the sample means of the model outputs on the training and test inputs? In equation (2.3) we approximated the second term of the test error in equation (2.2) as: $\frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}(\mathbf{y}_m)\, h_m \approx \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}(\mathbf{x}_n)\, f_n$. Let us reconsider this term [6]:

$$
\begin{aligned}
& \frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}(\mathbf{y}_m)\, h_m \\
= \; & \frac{1}{M}\sum_{m=1}^{M}\left(g_{\mathbf{v}}(\mathbf{y}_m) - \frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}(\mathbf{y}_m)\right) h_m + \frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}(\mathbf{y}_m)\,\frac{1}{M}\sum_{m=1}^{M} h_m \\
\approx \; & \frac{1}{N}\sum_{n=1}^{N}\left(g_{\mathbf{v}}(\mathbf{x}_n) - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}(\mathbf{x}_n)\right) f_n + \frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}(\mathbf{y}_m)\,\frac{1}{N}\sum_{n=1}^{N} f_n \\
= \; & \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}(\mathbf{x}_n)\, f_n + \left(\frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}(\mathbf{y}_m) - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}(\mathbf{x}_n)\right)\frac{1}{N}\sum_{n=1}^{N} f_n
\end{aligned}
$$

Now the new estimator of the test error becomes:

$$
\begin{aligned}
E(g_{\mathbf{v}}) \approx \; & \frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}^2(\mathbf{y}_m) - \frac{2}{N}\sum_{n=1}^{N} g_{\mathbf{v}}(\mathbf{x}_n)\, f_n + \frac{1}{N}\sum_{n=1}^{N} f_n^2 \\
& -2\left(\frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}(\mathbf{y}_m) - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}(\mathbf{x}_n)\right)\frac{1}{N}\sum_{n=1}^{N} f_n \\
= \; & E_0(g_{\mathbf{v}}) + \frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}^2(\mathbf{y}_m) - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}^2(\mathbf{x}_n)
\end{aligned}
$$

[6] Thanks to Dr. Barak Pearlmutter of University of New Mexico for suggesting this approximation

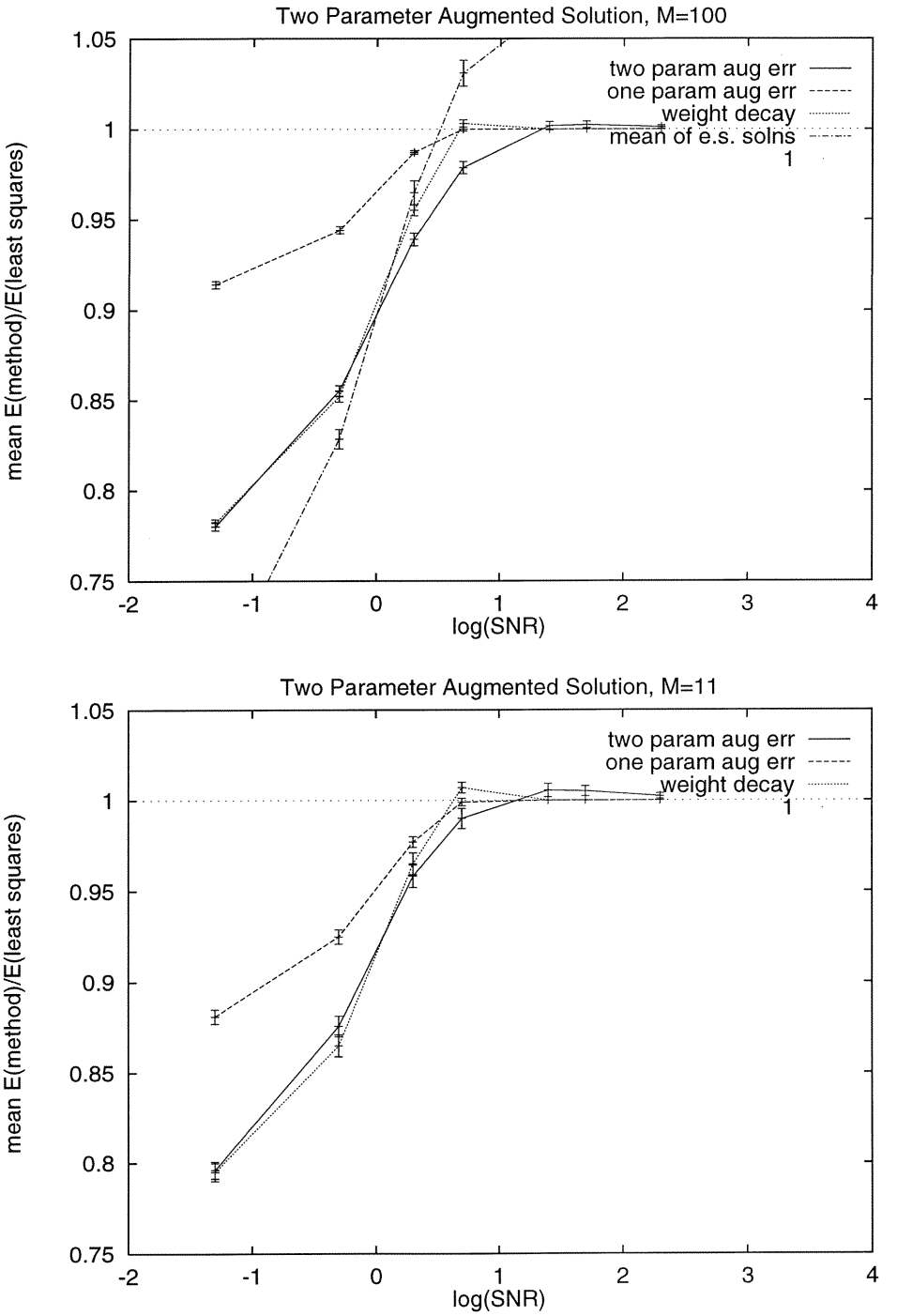

Figure 2.9: Performance of the two parameter augmented solution.

$$-2\left(\frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}\left(\mathbf{y}_m\right) - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}\left(\mathbf{x}_n\right)\right)\frac{1}{N}\sum_{n=1}^{N} f_n$$

Then the augmented error can be formed by means of parameterizing the addition to the training error by one or more parameters.

Let $\overline{f} = \frac{1}{N}\sum_{n=1}^{N} f_n, \overline{\phi_x} = \frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n), \overline{\phi_y} = \frac{1}{M}\sum_{m=1}^{M}\phi(\mathbf{y}_m)$. When only one augmentation parameter α is used, for the linear model, the augmented error becomes:

$$E_\alpha(\mathbf{w}) = E_0(\mathbf{w}) + \alpha\left(\mathbf{w}^T\left(\mathbf{S}_y - \mathbf{S}_x\right)\mathbf{w} - 2\mathbf{w}^T\left(\overline{\phi_y} - \overline{\phi_x}\right)\overline{f}\right)$$

and the augmented solution is:

$$\begin{aligned}\mathbf{w}_\alpha &= (\mathbf{I} - \alpha\mathbf{R})^{-1}\mathbf{S}_x)^{-1}\left(\frac{\mathbf{\Phi}_x\mathbf{f}}{N} + \alpha\left(\overline{\phi_y} - \overline{\phi_x}\right)\overline{f}\right) \\ &= (\mathbf{I} - \alpha\mathbf{R})^{-1}\mathbf{w}_0 + \alpha(\mathbf{I} - \alpha\mathbf{R})^{-1}{\mathbf{S}_x}^{-1}\left(\overline{\phi_y} - \overline{\phi_x}\right)\overline{f}\end{aligned}$$

For this case also, the augmentation parameter can be found by the substitution method.

2.6.3 Combination of Different Forms of Input Information

Sometimes the input information available is a combination of test inputs, extra inputs and the input probability distribution. Following the derivation in section 2.1, the augmented error for this case becomes:

$$\begin{aligned}&E_\alpha\left(g_{\mathbf{v}}\right) = E_0\left(g_{\mathbf{v}}\right) + \\ &\alpha\left(\left(\frac{\alpha_1}{K}\sum_{k=1}^{K} g_{\mathbf{v}}^2\left(\mathbf{z}_k\right) + \frac{\alpha_2}{N}\sum_{m=1}^{M} g_{\mathbf{v}}^2\left(\mathbf{y}_m\right) + \alpha_3\left\langle g_{\mathbf{v}}^2(\mathbf{x})\right\rangle_{\mathbf{x}}\right) - \frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{v}}^2\left(\mathbf{x}_n\right)\right)\end{aligned} \quad (2.25)$$

where parameters $\alpha, \alpha_1, \alpha_2, \alpha_3$ can be determined by the substitution method.

2.7 Conclusions

In this chapter we have derived the augmented error for quadratic loss function. For general linear model, we have determined when there is an augmented solution that is better than the simple training solution, and we have devised the substitution method to find a good augmentation parameter. Our experiments on both real and simulated data have shown that the augmented solution is consistently better than the simple training solution and better than the weight decay solution and the early stopping solution for large signal-to-noise ratio. We have shown that the weight decay solution is a special case of the two parameter augmented solution and the two parameter augmented solution is better than the weight decay solution when signal-to-noise ratio is large.

2.8 Appendix

2.8.1 The Least-Squares Solution for the Linear Model

For the general linear model, the training error is:

$$E_0(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\left(\mathbf{w}^T\phi(\mathbf{x}_n) - f_n\right)^2$$

The gradient of the training error is:

$$\frac{dE_0(\mathbf{w})}{d\mathbf{w}} = 2\mathbf{S}_x\mathbf{w} - 2\frac{\mathbf{\Phi}_x\mathbf{f}}{N}$$

Solving for $\frac{dE_0(\mathbf{w})}{d\mathbf{w}} = \mathbf{0}$ we obtain the least squares solution (simple training solution):

$$\mathbf{w}_0 = {\mathbf{S}_x}^{-1}\frac{\mathbf{\Phi}_x\mathbf{f}}{N}$$

2.8.2 Proof of Theorem 2.3.3:

Let $\mathbf{V}_x$ be such that $\mathbf{S}_x = \sigma_x^2\left(\mathbf{I} - \frac{\mathbf{V}_x}{\sqrt{N}}\right)$. Due to the distribution of $\phi(\mathbf{x}_n)$, $\langle \mathbf{S}_x \rangle_{\mathbf{x}} = \sigma_x^2\mathbf{I}$ hence $\left\langle \frac{\mathbf{V}_x}{\sqrt{N}} \right\rangle_{\mathbf{x}} = \mathbf{0}$ and $\langle (\mathbf{S}_x)^2 \rangle_{\mathbf{x}} = c_1\sigma_x^4\mathbf{I}$ for some constant c_1 hence $\left\langle \frac{\mathbf{V}_x^2}{N} \right\rangle_{\mathbf{x}} = c_2\frac{\sigma_x^4}{N}\mathbf{I}$ for some constant c_2. Similarly let $\mathbf{S}_y = \sigma_x^2\left(\mathbf{I} - \frac{\mathbf{V}_y}{\sqrt{M}}\right)$ with $\left\langle \frac{\mathbf{V}_y}{\sqrt{M}} \right\rangle_{\mathbf{y}} = \mathbf{0}$ and $\left\langle \frac{\mathbf{V}_y^2}{M} \right\rangle_{\mathbf{y}} = c_2\frac{\sigma_x^4}{M}\mathbf{I}$. Note also that since the training and test inputs are independent $\left\langle \frac{\mathbf{V}_x^q}{N^{\frac{q}{2}}}\frac{\mathbf{V}_y^p}{M^{\frac{p}{2}}} \right\rangle_{\mathbf{x},\mathbf{y}} = \left\langle \frac{\mathbf{V}_y^q}{M^{\frac{q}{2}}}\frac{\mathbf{V}_x^p}{N^{\frac{p}{2}}} \right\rangle_{\mathbf{x},\mathbf{y}} = \mathbf{0}$ for any q and any odd p.

When the spectral radius (the maximum of the absolute value of the eigenvalues) of $\frac{\mathbf{V}_x}{\sqrt{N}}$ is less than 1, ${\mathbf{S}_x}^{-1} = \left(\sigma_x^2\left(\mathbf{I} - \frac{\mathbf{V}_x}{\sqrt{N}}\right)\right)^{-1} = \frac{1}{\sigma_x^2}\left(\mathbf{I} + \frac{\mathbf{V}_x}{\sqrt{N}} + \frac{\mathbf{V}_x^2}{N}\right) + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)\mathbf{I}$ [Golub and Van Loan, 1993, page 549]. Hence:

$$\begin{aligned}
\mathbf{R} &= \mathbf{I} - {\mathbf{S}_x}^{-1}\mathbf{S}_y \\
&= \mathbf{I} - \left(\mathbf{I} + \frac{\mathbf{V}_x}{\sqrt{N}} + \frac{\mathbf{V}_x^2}{N} + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)\mathbf{I}\right)\left(\mathbf{I} + \frac{\mathbf{V}_y}{\sqrt{M}}\right) \\
&= -\frac{\mathbf{V}_x}{\sqrt{N}} - \frac{\mathbf{V}_y}{\sqrt{M}} - \frac{\mathbf{V}_x^2}{N} - \frac{\mathbf{V}_x\mathbf{V}_y}{\sqrt{MN}} + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)\mathbf{I} \\
\mathbf{R}^2 &= \frac{\mathbf{V}_x^2}{N} + \frac{\mathbf{V}_y^2}{M} + 2\frac{\mathbf{V}_x\mathbf{V}_y}{\sqrt{MN}} + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)\mathbf{I} \\
\mathbf{R}^3 &= \mathcal{O}\left(\frac{1}{N^{1.5}}\right)\mathbf{I}
\end{aligned} \tag{2.26}$$

When the spectral radius of $\alpha\mathbf{R}$ is less than 1:

$$\begin{aligned}
(\mathbf{I} - \alpha\mathbf{R})^{-1} &= \mathbf{I} + \alpha\mathbf{R} + \alpha^2\mathbf{R}^2 + \alpha^3\mathbf{R}^3 + \ldots \\
&= \mathbf{I} + \alpha\mathbf{R} + \alpha^2\mathbf{R}^2 + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)\mathbf{I}
\end{aligned} \tag{2.27}$$

From equation (2.12), the derivative of the expected generalization error with respect to α is:

$$\begin{aligned}
\frac{\partial \langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial\alpha} &= 2\mathbf{w}^{*T}\left(\mathbf{I} - \alpha\mathbf{R}^T\right)^{-1}\mathbf{R}^T\left(\mathbf{I} - \alpha\mathbf{R}^T\right)^{-1}\mathbf{S}_y\left((\mathbf{I} - \alpha\mathbf{R})^{-1} - \mathbf{I}\right)\mathbf{w}^* \\
&+ 2\frac{\sigma_e^2}{N}tr\left(\left(\mathbf{I} - \alpha\mathbf{R}^T\right)^{-1}\mathbf{R}^T\left(\mathbf{I} - \alpha\mathbf{R}^T\right)^{-1}\mathbf{S}_y(\mathbf{I} - \alpha\mathbf{R})^{-1}{\mathbf{S}_x}^{-1}\right)
\end{aligned}$$

Using the approximations for $\mathbf{R}$ and $(\mathbf{I} - \alpha\mathbf{R})^{-1}$ above, we can rewrite this

derivative as:

$$\begin{aligned}\frac{\partial \langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial\alpha} &= 2\alpha \mathbf{w}^{*T}\mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{w}^* + 2\frac{\sigma_e^2}{N} tr\left(\mathbf{R}{\mathbf{S}_x}^{-1}\mathbf{S}_y\right) \\ &+ 2\alpha\frac{\sigma_e^2}{N} tr\left(2{\mathbf{R}^T}^2\mathbf{S}_y{\mathbf{S}_x}^{-1} + \mathbf{R}^T\mathbf{S}_y\mathbf{R}{\mathbf{S}_x}^{-1}\right) \\ &+ \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{1.5}}\right) + \mathcal{O}\left(\frac{\sigma_e^2}{N^{2.5}}\right) \qquad (2.28)\end{aligned}$$

When we rewrite ${\mathbf{S}_x}^{-1}$ and $\mathbf{S}_y$ using $\mathbf{V}_x$ and $\mathbf{V}_y$, and use the approximations for powers of $\mathbf{R}$ we obtain the following:

$$\begin{aligned}\left\langle \frac{\partial \langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial\alpha}\right\rangle_{\mathbf{x},\mathbf{y}} &= 2\left(\alpha\left(\sigma_x^2\mathbf{w}^{*T}\mathbf{w}^* + 3\frac{\sigma_e^2}{N}(d+1)\right) - \frac{\sigma_e^2}{N}(d+1)\right)c_2\sigma_x^4\left(\frac{1}{N}+\frac{1}{M}\right) \\ &+ \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{1.5}}\right) + \mathcal{O}\left(\frac{\sigma_e^2}{N^{2.5}}\right)\end{aligned}$$

Hence the solution of $\left\langle \frac{\partial\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial\alpha}\right\rangle_{\mathbf{x},\mathbf{y}} = 0$ is:

$$\begin{aligned}\alpha &= \frac{\frac{\sigma_e^2}{N}(d+1) + \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{0.5}}\right) + \mathcal{O}\left(\frac{\sigma_e^2}{N^{1.5}}\right)}{\sigma_x^2\mathbf{w}^{*T}\mathbf{w}^* + 3\frac{\sigma_e^2}{N}(d+1)} \\ &= \frac{\frac{\sigma_e^2}{N}(d+1) + \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{0.5}}\right)}{\sigma_x^2\mathbf{w}^{*T}\mathbf{w}^* + \mathcal{O}\left(\frac{\sigma_e^2}{N}\right)}\end{aligned}$$

In the last step we have used $\frac{\sigma_e^2}{N} \ll \sigma_x^2\mathbf{w}^{*T}\mathbf{w}^*$. For large N the best α is:

$$\alpha \approx \frac{d+1}{N}\frac{\sigma_e^2}{\sigma_x^2\mathbf{w}^{*T}\mathbf{w}^*}$$

□

2.8.3 Proof of Theorem 2.3.4:

From equation (2.28), for large N and M and $\Sigma_{\phi(x)} = \sigma_x^2\mathbf{I}$, the α that minimizes $\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}$ is:

$$\alpha^* \approx -\frac{\sigma_e^2}{N}\frac{tr\left(\mathbf{R}\mathbf{S}_x{}^{-1}\mathbf{S}_y\right)}{\mathbf{w}^{*T}\mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{w}^* + \frac{\sigma_e^2}{N}tr\left(2\mathbf{R}^{T^2}\mathbf{S}_y\mathbf{S}_x{}^{-1} + \mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{S}_x{}^{-1}\right)} \tag{2.29}$$

Again for large N and M, the expected test error (equation (2.10)) can be approximated as:

$$\begin{aligned}\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta} &\approx \frac{\sigma_e^2}{N}tr\left(\mathbf{S}_y\mathbf{S}_x{}^{-1}\right) + \sigma_e^2 \\ &+ \alpha^2\mathbf{w}^{*T}\mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{w}^* + 2\alpha\frac{\sigma_e^2}{N}tr\left(\mathbf{R}\mathbf{S}_x{}^{-1}\mathbf{S}_y\right) \\ &+ \alpha^2\frac{\sigma_e^2}{N}tr\left(2\mathbf{R}^{T^2}\mathbf{S}_y\mathbf{S}_x{}^{-1} + \mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{S}_x{}^{-1}\right) \\ &+ \mathcal{O}\left(\frac{\sigma_e^2}{N^{2.5}}\right) + \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{1.5}}\right)\end{aligned} \tag{2.30}$$

From equation (2.11) the first two terms equal $\langle E(\mathbf{w}_0)\rangle_{\epsilon,\delta}$. Substituting for α by the expression for α^* in equation (2.29):

$$\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta} - \langle E(\mathbf{w}_0)\rangle_{\epsilon,\delta} \approx -\frac{\sigma_e^4}{N^2}\frac{tr^2\left(\mathbf{R}\mathbf{S}_x{}^{-1}\mathbf{S}_y\right)}{\mathbf{w}^{*T}\mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{w}^*}$$

□

2.8.4 Proof of Theorem 2.3.5:

The derivative of the expected generalization error in equation (2.16) with respect to α is:

$$\begin{aligned}\frac{\partial\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial\alpha} &= 2\mathbf{w}^{*T}\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{R}^T\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{\Sigma}_{\phi(x)}\left((\mathbf{I}-\alpha\mathbf{R})^{-1}-\mathbf{I}\right)\mathbf{w}^* \\ &+ 2\frac{\sigma_e^2}{N}tr\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{R}^T\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{\Sigma}_{\phi(x)}(\mathbf{I}-\alpha\mathbf{R})^{-1}\mathbf{S}_x{}^{-1}\right)\end{aligned} \tag{2.31}$$

Hence the derivative at $\alpha = 0$ is:

$$\left.\frac{\partial \langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial \alpha}\right|_{\alpha=0} = 2\frac{\sigma_e^2}{N} tr\left(\mathbf{R}\mathbf{S}_x{}^{-1}\mathbf{\Sigma}_{\phi(x)}\right)$$

If this derivative is nonzero then $\alpha = 0$ is not a local minimum of the expected generalization error and hence the expected generalization error is minimized at some nonzero α. □

2.8.5 Proof of Theorem 2.3.6:

Using $\mathbf{V}_x$ and approximations for powers of $\mathbf{R}$ (equation (2.26)) (with $\mathbf{V}_y = \mathbf{0}$) and $(\mathbf{I} - \alpha\mathbf{R})^{-1}$ (equation (2.27)) from the proof of theorem 2.3.3:

$$\begin{aligned}\frac{\partial \langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial \alpha} &= 2\alpha\sigma_x^2 \mathbf{w}^{*T}\mathbf{R}^T\mathbf{R}\mathbf{w}^* + 2\frac{\sigma_e^2}{N}\sigma_x^2 tr\left(\mathbf{R}^T\mathbf{S}_x{}^{-1}\right) \\ &+ 2\alpha\frac{\sigma_e^2}{N}\sigma_x^2 tr\left(2\mathbf{R}^{T^2}\mathbf{S}_x{}^{-1} + \mathbf{R}^T\mathbf{R}\mathbf{S}_x{}^{-1}\right) \\ &+ \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{1.5}}\right) + \mathcal{O}\left(\frac{\sigma_e^2}{N^{2.5}}\right) \qquad (2.32)\end{aligned}$$

When we rewrite $\mathbf{S}_x{}^{-1}$ using $\mathbf{V}_x$, use the approximations for powers of R (equation (2.26)) and $\left\langle \frac{\mathbf{V}_x^p}{N^{\frac{p}{2}}} \right\rangle_{\mathbf{x}} = \mathbf{0}$ for any odd p, we obtain the following:

$$\begin{aligned}\left\langle \frac{\partial \langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial \alpha} \right\rangle_{\mathbf{x}} &= 2\left(\alpha\left(\sigma_x^2\mathbf{w}^{*T}\mathbf{w}^* + 3\frac{\sigma_e^2}{N}(d+1)\right) - \frac{\sigma_e^2}{N}(d+1)\right) c_2\sigma_x^4\frac{1}{N} \\ &+ \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{1.5}}\right) + \mathcal{O}\left(\frac{\sigma_e^2}{N^{2.5}}\right)\end{aligned}$$

Hence the solution of $\left\langle \frac{\partial \langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial \alpha} \right\rangle_{\mathbf{x}} = 0$ is:

$$\begin{aligned}\alpha &= \frac{\frac{\sigma_e^2}{N}(d+1) + \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{0.5}}\right) + \mathcal{O}\left(\frac{\sigma_e^2}{N^{1.5}}\right)}{\sigma_x^2\mathbf{w}^{*T}\mathbf{w}^* + 3\frac{\sigma_e^2}{N}(d+1)} \\ &= \frac{\frac{\sigma_e^2}{N}(d+1) + \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{0.5}}\right)}{\sigma_x^2\mathbf{w}^{*T}\mathbf{w}^* + \mathcal{O}\left(\frac{\sigma_e^2}{N}\right)}\end{aligned}$$

In the last step we have used $\frac{\sigma_e^2}{N} \ll \sigma_x^2 \mathbf{w}^{*T}\mathbf{w}^*$. For large N the best α is:

$$\alpha^* \approx \frac{d+1}{N}\frac{\sigma_e^2}{\sigma_x^2 \mathbf{w}^{*T}\mathbf{w}^*}$$

□

2.8.6 Proof of Theorem 2.3.7:

From equation (2.32), for large N and $\Sigma_{\phi(x)} = \sigma_x^2\mathbf{I}$, the α that minimizes $\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}$ is:

$$\alpha^* \approx -\frac{\sigma_e^2}{N}\frac{tr\left(\mathbf{R}\mathbf{S}_x{}^{-1}\right)}{\mathbf{w}^{*T}\mathbf{R}^T\mathbf{R}\mathbf{w}^* + \frac{\sigma_e^2}{N}\sigma_x^2 tr\left(2\mathbf{R}^{T^2}\mathbf{S}_x{}^{-1} + \mathbf{R}^T\mathbf{R}\mathbf{S}_x{}^{-1}\right)} \tag{2.33}$$

Again for large N, the expected generalization error (equation (2.16)) can be approximated as:

$$\begin{aligned}\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta} \approx\ & \frac{\sigma_e^2}{N}\sigma_x^2 tr\left(\mathbf{S}_x{}^{-1}\right) + \sigma_e^2 \\ +\ & \alpha^2\sigma_x^2\mathbf{w}^{*T}\mathbf{R}^T\mathbf{R}\mathbf{w}^* + 2\alpha\frac{\sigma_e^2}{N}\sigma_x^2 tr\left(\mathbf{R}\mathbf{S}_x{}^{-1}\right) \\ +\ & \alpha^2\frac{\sigma_e^2}{N}\sigma_x^2 tr\left(2\mathbf{R}^{T^2}\mathbf{S}_x{}^{-1} + \mathbf{R}^T\mathbf{R}\mathbf{S}_x{}^{-1}\right) \\ +\ & \mathcal{O}\left(\frac{\sigma_e^2}{N^{2.5}}\right) + \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{1.5}}\right)\end{aligned} \tag{2.34}$$

From equation (2.17) the first two terms equal $\langle E_{gen}(\mathbf{w}_0)\rangle_{\epsilon,\delta}$. Substituting for α by the expression for α^* in equation (2.33):

$$\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta} - \langle E_{gen}(\mathbf{w}_0)\rangle_{\epsilon,\delta} \approx -\frac{\sigma_e^4}{N^2}\frac{\sigma_x^2 tr^2\left(\mathbf{R}\mathbf{S}_x{}^{-1}\right)}{\mathbf{w}^{*T}\mathbf{R}^T\mathbf{R}\mathbf{w}^*}$$

□

2.8.7 Proof of Theorem 2.3.8:

The expected value of the parameter error of the augmented solution with respect to training output noise is:

$$\begin{aligned}\left\langle E_{param}\left(\mathbf{w}_\alpha\right)\right\rangle_\epsilon &= \mathbf{w}^{*T}\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}-\mathbf{I}\right)\left(\left(\mathbf{I}-\alpha\mathbf{R}\right)^{-1}-\mathbf{I}\right)\mathbf{w}^* \\ &+ \frac{\sigma_e^2}{N}tr\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\left(\mathbf{I}-\alpha\mathbf{R}\right)^{-1}{\mathbf{S}_x}^{-1}\right) \qquad (2.35)\end{aligned}$$

The derivative of this expected parameter error with respect to α is:

$$\begin{aligned}&\frac{\partial\left\langle E_{param}\left(\mathbf{w}_\alpha\right)\right\rangle_\epsilon}{\partial\alpha} = 2\mathbf{w}^{*T}\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{R}^T\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\left(\left(\mathbf{I}-\alpha\mathbf{R}\right)^{-1}-\mathbf{I}\right)\mathbf{w}^* \\ &+ 2\frac{\sigma_e^2}{N}tr\left(\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\mathbf{R}^T\left(\mathbf{I}-\alpha\mathbf{R}^T\right)^{-1}\left(\mathbf{I}-\alpha\mathbf{R}\right)^{-1}{\mathbf{S}_x}^{-1}\right) \qquad (2.36)\end{aligned}$$

Hence the derivative at $\alpha = 0$ is:

$$\left.\frac{\partial\left\langle E_{param}\left(\mathbf{w}_\alpha\right)\right\rangle_\epsilon}{\partial\alpha}\right|_{\alpha=0} = 2\frac{\sigma_e^2}{N}tr\left(\mathbf{R}{\mathbf{S}_x}^{-1}\right)$$

If this derivative is nonzero then $\alpha = 0$ is not a local minimum of the expected parameter error and hence the expected parameter error is minimized at some nonzero α.

□

2.8.8 Proof of Theorem 2.4.1:

In order to prove this theorem, we first need the following lemma:

Lemma 2.8.1 *Let $a, b, c, \alpha \in \mathcal{R}$ where $a > 0$ and $b \neq 0$. Let α^* minimize $\alpha^2 a - 2\alpha b + c$. Let α' minimize $\alpha^2 a' - 2\alpha b + c'$ where $0 < a < a'$. Then $|\alpha'| < |\alpha^*|$.*

Proof: The minimum of $\alpha^2 a - 2\alpha b + c$ is at $\alpha^* = \frac{b}{a}$ and at this minimum $\frac{d\alpha^*}{da} = -\frac{b}{a^2}$. If $b < 0$ then $\alpha^* < 0$ and it increases as a increases, if $b > 0$ then $\alpha^* > 0$ and it decreases as a increases. Therefore regardless of b, the magnitude of the α^* decreases as a increases. Therefore, $|\alpha'| < |\alpha^*|$ for $0 < a < a'$. □

Consider the approximation of the $\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}$ in equation (2.30) large N and M:

$$\begin{aligned}\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta} &\approx \frac{\sigma_e^2}{N} tr\left(\mathbf{S}_y {\mathbf{S}_x}^{-1}\right) + \sigma_e^2 \\ &+ \alpha^2 \mathbf{w}^{*T}\mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{w}^* + 2\alpha\frac{\sigma_e^2}{N} tr\left(\mathbf{R}{\mathbf{S}_x}^{-1}\mathbf{S}_y\right) \\ &+ \alpha^2\frac{\sigma_e^2}{N} tr\left(2\mathbf{R}^{T^2}\mathbf{S}_y{\mathbf{S}_x}^{-1} + \mathbf{R}^T\mathbf{S}_y\mathbf{R}{\mathbf{S}_x}^{-1}\right) \\ &+ \mathcal{O}\left(\frac{\sigma_e^2}{N^{2.5}}\right) + \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{1.5}}\right)\end{aligned}$$

The coefficient of α^2 is $\mathbf{w}^{*T}\mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{w}^* + \frac{\sigma_e^2}{N} tr\left(2\mathbf{R}^{T^2}\mathbf{S}_y{\mathbf{S}_x}^{-1} + \mathbf{R}^T\mathbf{S}_y\mathbf{R}{\mathbf{S}_x}^{-1}\right) > 0$. Hence the $\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}$ satisfies the precondition of the lemma.

When $\mathbf{w}_0$ is substituted for $\mathbf{w}^*$, $\left\langle \mathbf{w}_0^T\mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{w}_0\right\rangle_{\epsilon,\delta} = \mathbf{w}^{*T}\mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{w}^* + \left\langle \frac{\epsilon^T\mathbf{\Phi}_x^T}{N}{\mathbf{S}_x}^{-1}\mathbf{R}^T\mathbf{S}_y\mathbf{R}{\mathbf{S}_x}^{-1}\frac{\mathbf{\Phi}_x\epsilon}{N}\right\rangle_\epsilon \geq \mathbf{w}^{*T}\mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{w}^*$. When σ_e^2 is estimated by means of $\mathbf{w}_0$, $\left\langle \frac{\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)^T\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)}{N-d-1}\right\rangle_{\epsilon,\delta} = \sigma_e^2$, hence the coefficient of α in equation (2.30) remains the same. Hence by lemma (2.8.1), the α' that minimizes the expected test error with the $\mathbf{w}_0$ substitutions is smaller in magnitude than α^* that minimizes the actual expected test error.

For large N and M, $\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}$ is convex, because differentiating it twice gives:

$$\begin{aligned}\frac{\partial^2\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial\alpha^2} &= \mathbf{w}^{*T}\mathbf{R}^T\mathbf{S}_y\mathbf{R}\mathbf{w}^* + \frac{\sigma_e^2}{N} tr\left(2\mathbf{R}^{T^2}\mathbf{S}_y{\mathbf{S}_x}^{-1} + \mathbf{R}^T\mathbf{S}_y\mathbf{R}{\mathbf{S}_x}^{-1}\right) \\ &+ \mathcal{O}\left(\frac{\sigma_e^2}{N^{2.5}}\right) + \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{1.5}}\right)\end{aligned}$$

Since α' is in between 0 and α^* and $\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}$ is convex, it follows that $\langle E(\mathbf{w}_{\alpha^*})\rangle_{\epsilon,\delta} \leq \langle E(\mathbf{w}_{\alpha'})\rangle_{\epsilon,\delta} \leq \langle E(\mathbf{w}_0)\rangle_{\epsilon,\delta}$. □

2.8.9 Proof of Theorem 2.4.2:

Consider the approximation of the $\langle E(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}$ in equation (2.34) large N:

$$\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta} \approx \frac{\sigma_e^2}{N}\sigma_x^2 tr\left({\mathbf{S}_x}^{-1}\right) + \sigma_e^2$$

$$
\begin{aligned}
&+ \quad \alpha^2\sigma_x^2\mathbf{w}^{*T}\mathbf{R}^T\mathbf{R}\mathbf{w}^* + 2\alpha\frac{\sigma_e^2}{N}\sigma_x^2 tr\left(\mathbf{R}\mathbf{S}_x{}^{-1}\right) \\
&+ \quad \alpha^2\frac{\sigma_e^2}{N}\sigma_x^2 tr\left(2\mathbf{R}^{T^2}\mathbf{S}_x{}^{-1} + \mathbf{R}^T\mathbf{R}\mathbf{S}_x{}^{-1}\right) \\
&+ \quad \mathcal{O}\left(\frac{\sigma_e^2}{N^{2.5}}\right) + \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{1.5}}\right)
\end{aligned}
$$

The coefficient of α^2 is $\sigma_x^2\mathbf{w}^{*T}\mathbf{R}^T\mathbf{R}\mathbf{w}^* + \frac{\sigma_e^2}{N}\sigma_x^2 tr\left(2\mathbf{R}^{T^2}\mathbf{S}_x{}^{-1} + \mathbf{R}^T\mathbf{R}\mathbf{S}_x{}^{-1}\right) > 0$. Hence the $\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}$ satisfies the precondition of lemma 2.8.1.

When $\mathbf{w}_0$ is substituted for $\mathbf{w}^*$, $\left\langle \sigma_x^2\mathbf{w}_0^T\mathbf{R}^T\mathbf{R}\mathbf{w}_0\right\rangle_{\epsilon,\delta} = \sigma_x^2\mathbf{w}^{*T}\mathbf{R}^T\mathbf{R}\mathbf{w}^* + \sigma_x^2\left\langle \frac{\epsilon^T\mathbf{\Phi}_x^T}{N}\mathbf{S}_x{}^{-1}\mathbf{R}^T\mathbf{R}\mathbf{S}_x{}^{-1}\frac{\mathbf{\Phi}_x\epsilon}{N}\right\rangle_\epsilon \geq \sigma_x^2\mathbf{w}^{*T}\mathbf{R}^T\mathbf{R}\mathbf{w}^*$. When σ_e^2 is estimated by means of $\mathbf{w}_0$, $\left\langle \frac{\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)^T\left(\mathbf{\Phi}_x^T\mathbf{w}_0-\mathbf{f}\right)}{N-d-1}\right\rangle_{\epsilon,\delta} = \sigma_e^2$, hence the coefficient of α in equation (2.34) remains the same. Hence by lemma (2.8.1), the α' that minimizes the expected generalization error with the $\mathbf{w}_0$ substitutions is smaller in magnitude than α^* that minimizes the actual expected generalization error.

For large N, $\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}$ is convex, because differentiating it twice gives:

$$
\begin{aligned}
\frac{\partial^2\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}}{\partial\alpha^2} &= \sigma_x^2\mathbf{w}^{*T}\mathbf{R}^T\mathbf{R}\mathbf{w}^* + \frac{\sigma_e^2}{N}\sigma_x^2 tr\left(2\mathbf{R}^{T^2}\mathbf{S}_x{}^{-1} + \mathbf{R}^T\mathbf{R}\mathbf{S}_x{}^{-1}\right) \\
&+ \quad \mathcal{O}\left(\frac{\sigma_e^2}{N^{2.5}}\right) + \mathcal{O}\left(\frac{\mathbf{w}^{*T}\mathbf{w}^*}{N^{1.5}}\right)
\end{aligned}
$$

Since α' is in between 0 and α^* and $\langle E_{gen}(\mathbf{w}_\alpha)\rangle_{\epsilon,\delta}$ is convex, it follows that $\langle E_{gen}(\mathbf{w}_{\alpha^*})\rangle_{\epsilon,\delta} \leq \langle E_{gen}(\mathbf{w}_{\alpha'})\rangle_{\epsilon,\delta} \leq \langle E_{gen}(\mathbf{w}_0)\rangle_{\epsilon,\delta}$. □

Chapter 3
Test Inputs: Nonlinear Model Case

In the previous chapter, we derived the augmented error for a general model, and then analyzed the augmented solution for the general linear model case. In this chapter, we will concentrate on nonlinear models, specifically neural networks with one hidden layer of tangent hyperbolic units and a linear output. These models can approximate any continuous function given enough number of hidden units [Cybenko, 1989]. They are also a natural extension of the general linear model of the previous chapter.

For the general linear model we were able to find the training and augmented error minimum analytically (section 2.2). For the neural network model (and for most nonlinear models) we have to use an iterative search technique to find the minimum. Due to nonlinearity of the model, there may be many local minima and saddle points. The search technique determines the solution(s) that will be found, and hence performance of different methods. In this chapter, we will use the gradient descent with adaptive learning rate and starting from small random initial weights as the search technique. Since we can not find the solutions analytically, this chapter will be more experimental in nature than the previous one.

When augmentation parameters are chosen to be ∞, only the augmented term is minimized, and when they are chosen to be 0, only the training error is minimized. Determination of an augmentation parameter that can lead the descent algorithm to a good solution, regardless of the starting point, is a difficult task. Actually such a universal augmentation parameter may not even exist. Instead of a globally best augmentation parameter, in this chapter, we will mainly focus on finding a value of the augmentation parameter that can lead from the current solution to a better solution through a perturbation of the current solution.

In section 3.1 we describe the neural network model. Section 3.2 describes how

to modify an existing solution to reduce the test error and the augmented error. In sections 3.3 we use the substitution method to obtain an augmented solution for the output weights, keeping the input weights fixed. Section 3.4 investigates ordinary and leave-$k > 1$-out cross validation as an alternative to the substitution method to find the augmentation parameters. In section 3.5 we present experimental results on using the augmented solution method repetitively and identifying solutions that could be used for augmented solutions during gradient descent on training error. Section 3.6 discusses a method of descending on the augmented error instead of the training error alone. The augmentation parameters during the descent are determined by means of cross validation method. Section 3.7 discusses extensions of the augmented error approach to different loss functions, namely, entropic loss, maximum likelihood with input dependent noise variance and p-norm loss. Finally section 3.8 summarizes the chapter.

3.1 Neural Network Models

Let $\mathbf{x} \in \mathcal{R}^{d'}$ be an input. Let $g_{\mathbf{v}}(\mathbf{x}) : \mathcal{R}^{d'} \rightarrow \mathcal{R}$ denote the output of the neural network with weights (parameters) $\mathbf{v}$ on input $\mathbf{x}$. The weight vector $\mathbf{v}$ consists of weights between inputs and hidden units (input weights) and weights between hidden units and the output (output weights). Let d be the number of hidden units. Then $\mathbf{v} = [v_0\ v_1 \ldots v_d\ v_{1,0}\ v_{1,1} \ldots v_{1,d'} \ldots v_{d,0}\ v_{d,1} \ldots v_{d,d'}]^T$, where v_i is the output weight from ith hidden unit to the output and and $v_{i,j}$ is the input weight from the jth input to the ith hidden unit. Weights $v_0, v_{1,0}, v_{2,0}, \ldots, v_{d,0}$ are the bias weights that are connected to constant $+1$ (figure 3.1). The neural network with weights $\mathbf{v}$, tangent hyperbolic (tanh)[1] hidden unit nonlinearities and a linear output implements the following function:

$$g_{\mathbf{v}}(\mathbf{x}) = v_0 + \sum_{i=1}^{d} v_i \tanh\left(v_{i,0} + \sum_{j=1}^{d'} v_{i,j} x_j\right) \tag{3.1}$$

[1]Both tangent hyperbolic, $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and sigmoid, $sig(x) = \frac{1}{1+e^{-x}}$ units are enough to implement any continuous function.

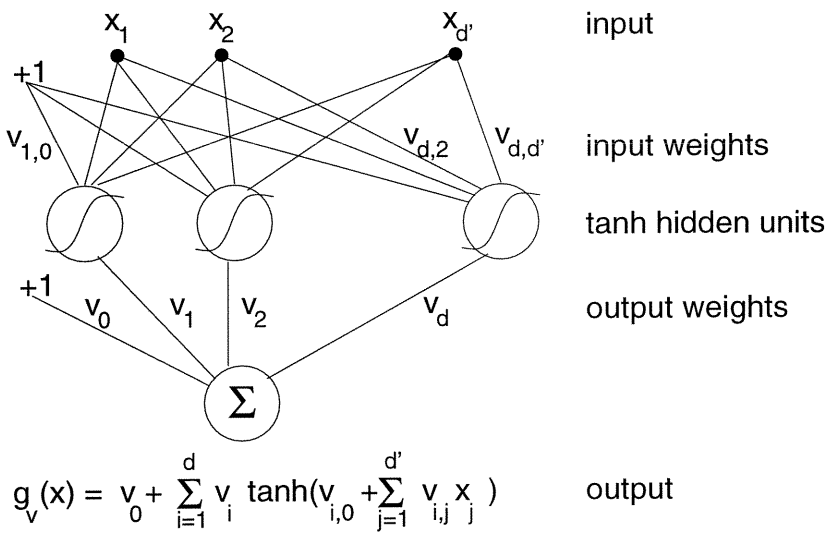

Figure 3.1: One hidden layer neural network.

We will use the notation and definitions from section 2.1 for the training and test sets, training error, test error and the augmented error:

$$\begin{aligned}
\text{training set} \quad & \{(\mathbf{x}_1, f_1), \ldots, (\mathbf{x}_N, f_N)\} \\
\text{test set} \quad & \{(\mathbf{y}_1, h_1), \ldots, (\mathbf{y}_M, h_M)\} \\
\text{training error}, E_0(g_\mathbf{v}) &= \frac{1}{N}\sum_{n=1}^{N}\left(g_\mathbf{v}(\mathbf{x}_n) - f_n\right)^2 \qquad (3.2) \\
\text{test error}, E(g_\mathbf{v}) &= \frac{1}{M}\sum_{m=1}^{M}\left(g_\mathbf{v}(\mathbf{y}_m) - h_m\right)^2
\end{aligned}$$

two parameter augmented error,

$$\begin{aligned}
E_{\alpha_1,\alpha_2}(g_\mathbf{v}) &= E_0(g_\mathbf{v}) + \alpha_1 \frac{1}{M}\sum_{m=1}^{M} g_\mathbf{v}^2(\mathbf{y}_m) - \alpha_2 \frac{1}{N}\sum_{n=1}^{N} g_\mathbf{v}^2(\mathbf{x}_n) \\
&= E_0(g_\mathbf{v}) + A_{\alpha_1,\alpha_2}(g_\mathbf{v})
\end{aligned}$$

where we have renamed the augmented term $\alpha_1 \frac{1}{M}\sum_{m=1}^{M} g_\mathbf{v}^2(\mathbf{y}_m) - \alpha_2 \frac{1}{N}\sum_{n=1}^{N} g_\mathbf{v}^2(\mathbf{x}_n)$ as $A_{\alpha_1,\alpha_2}(g_\mathbf{v})$. In this chapter we will use the augmented error for the test inputs and with two augmentation parameters.

3.2 Perturbing a Solution to Minimize the Test and Augmented Errors

The arbitrary complexity of the neural network model and the iterative search procedure requires redefinition of the "best augmentation parameters". Instead of trying to find globally best augmentation parameters, i.e. regardless of where the search starts, the minimization of the augmented error results in the minimization of the test error, we will focus on finding augmentation parameters that perform good locally, and obtaining a solution around an existing solution by means of these augmentation parameters.

Consider a neural network with weights $\hat{\mathbf{v}}$. In this chapter $\hat{\mathbf{v}}$ will be obtained after a certain number of gradient descent steps on the training error. In general $\hat{\mathbf{v}}$ can be any solution that is believed to be close to the test error minimum. Assume that the test error minimum is a small $\boldsymbol{\Delta}\mathbf{v}$ away from $\hat{\mathbf{v}}$. Then the gradient of the test error at $\hat{\mathbf{v}} + \boldsymbol{\Delta}\mathbf{v}$ with respect to $\boldsymbol{\Delta}\mathbf{v}$ is given by:

$$\begin{aligned} \frac{\partial E(g_{\hat{\mathbf{v}}+\boldsymbol{\Delta}\mathbf{v}})}{\partial \boldsymbol{\Delta}\mathbf{v}} &\approx \frac{\partial}{\partial \boldsymbol{\Delta}\mathbf{v}} \left(E(g_{\hat{\mathbf{v}}}) + \boldsymbol{\Delta}\mathbf{v}^T \nabla E(g_{\hat{\mathbf{v}}}) + \frac{1}{2} \boldsymbol{\Delta}\mathbf{v}^T HE(g_{\hat{\mathbf{v}}}) \boldsymbol{\Delta}\mathbf{v} \right) \\ &= \nabla E(g_{\hat{\mathbf{v}}}) + HE(g_{\hat{\mathbf{v}}}) \boldsymbol{\Delta}\mathbf{v} \end{aligned} \tag{3.3}$$

where $\nabla E(g_{\hat{\mathbf{v}}}) = \frac{dE(g_{\hat{\mathbf{v}}})}{\partial \boldsymbol{\Delta}\mathbf{v}}$ is the gradient and $HE(g_{\hat{\mathbf{v}}})_{i,j} = \frac{\partial^2 E(g_{\hat{\mathbf{v}}})}{\partial \Delta v_i \Delta v_j}$ is the Hessian of the test error with respect to $\boldsymbol{\Delta}\mathbf{v}$. At the test error minimum the gradient will be $\mathbf{0}$. Equating the gradient in (3.3) to $\mathbf{0}$ and solving for $\boldsymbol{\Delta}\mathbf{v}$, we find the $\boldsymbol{\Delta}\mathbf{v}$ that results in the minimum test error:

$$\boldsymbol{\Delta}\mathbf{v} = -(HE(g_{\hat{\mathbf{v}}}))^{-1} \nabla E(g_{\hat{\mathbf{v}}}) \tag{3.4}$$

Now comes the connection with the augmented error. Let the augmentation parameters α_1, α_2 be such that the gradient and the Hessian of the augmented error $E_0(g_{\hat{\mathbf{v}}}) + A_{\alpha_1,\alpha_2}(g_{\hat{\mathbf{v}}})$ is very close to the gradient and Hessian of the test error $E(g_{\hat{\mathbf{v}}})$

around $\hat{\mathbf{v}}$ (see figure 3.2). Then the perturbation $\mathbf{\Delta v}$ can also be written as:

$$\mathbf{\Delta v} = -(HE_0(g_{\hat{\mathbf{v}}}) + HA_{\alpha_1,\alpha_2}(g_{\hat{\mathbf{v}}}))^{-1} (\nabla E_0(g_{\hat{\mathbf{v}}}) + \nabla A_{\alpha_1,\alpha_2}(g_{\hat{\mathbf{v}}})) \quad (3.5)$$

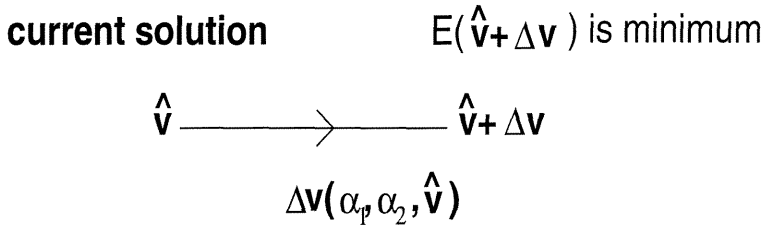

Figure 3.2: Perturbing the current solution to get to the test error minimum.

Once a good value of augmentation parameters is determined, $\mathbf{\Delta v}$ and the new solution $\mathbf{v} + \mathbf{\Delta v}$ can be computed. A good value of the augmentation parameter is the one that results in a $\mathbf{\Delta v}$ such that $E(\mathbf{v} + \mathbf{\Delta v}) < E(\mathbf{v})$.

Now we extend the substitution method into this setting to obtain a good augmentation parameter.

3.3 Augmented Solution Around the Gradient Descent Solution

In this section, we will keep the input weights of the solution $\hat{\mathbf{v}}$ as it is and concentrate on obtaining a better solution for the output weights. This allows a direct extension of the substitution method from section 2.4 to the nonlinear case. Due to the tanh nonlinearities of the neural network model and the gradient descent method we use to minimize the training error, modification of the output weights only is an acceptable solution.

During gradient descent on the training error, an input weight $v_{i,j}$ is modified according to the gradient of the training error. From equations (3.1) and (3.2), the gradient of the training error with respect to an input weight $v_{i,j}$ is: $\frac{\partial E_0(g_{\mathbf{v}})}{\partial v_{i,j}} =$

$\frac{2}{N}\sum_{n=1}^{N}(g_{\mathbf{v}}(\mathbf{x}_n)-f_n)\frac{\partial g_{\mathbf{v}}(\mathbf{x}_n)}{\partial v_{i,j}} = \frac{2}{N}\sum_{n=1}^{N}(g_{\mathbf{v}}(\mathbf{x}_n)-f_n)\,v_i\frac{\partial \tanh(u_i)}{\partial u_i(\mathbf{x}_n)}\frac{\partial u_i(\mathbf{x}_n)}{\partial v_{i,j}}$ where $u_i(\mathbf{x}) = v_{i,0}+\sum_{j=1}^{d'} v_{i,j}x_j$. Training starts from small initial weights and hence the tanh hidden unit outputs are linear in the inputs initially. After training for a while the input weights get larger and the tanh hidden units enter their nonlinear region (figure 3.3). As the nonlinearity increases the gradient $\frac{\partial \tanh(u_i)}{\partial u_i}$ gets smaller, and hence the changes to the input weight $v_{i,j}$ get smaller.

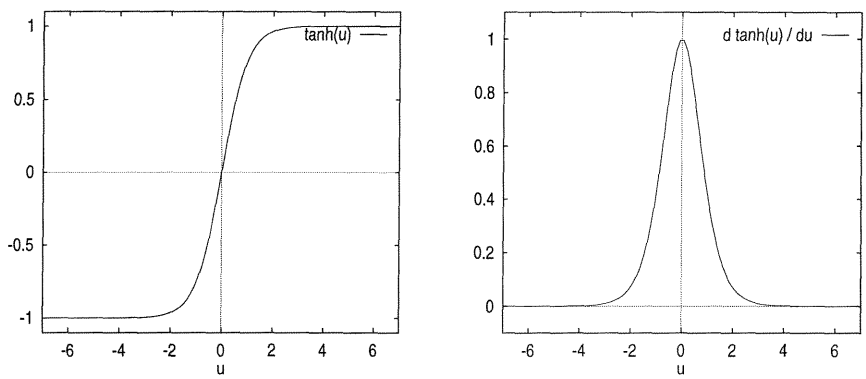

Figure 3.3: The tanh nonlinearity and its derivative with respect to its argument.

In figure 3.4 we show training and test errors and the input and output weights while descending on the training error by means of the gradient descent with adaptive learning rate. Notice that the input weights remain almost constant after some time (pass 250) of training, while the output weights still keep changing. On the other hand, the test error [2] starts increasing (i.e. overtraining starts) after the input weights have settled. Given the solution at the end of the training session, keeping the input weights as they are and changing the output weights to decrease the test error is our goal in this section.

When the input weights are kept fixed and the output weights are allowed to change, we obtain exactly the general linear model (section 2.2). Let $\hat{\mathbf{v}}$ be the solution after some number of gradient descent steps on the training error. Denote the output weights of $\hat{\mathbf{v}}$ by $\hat{\mathbf{w}} = [\hat{v}_0\ \hat{v}_1\ \ldots\ \hat{v}_d]^T$, and let $\Phi_x = [\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_N)]_{(d+1)\times N}$ and

[2] The jumps in the training and test errors are due to the adaptive learning rate we use for the gradient descent algorithm. At the jumps, the descent rate has become too high and it is reduced to a right value after a number of passes.

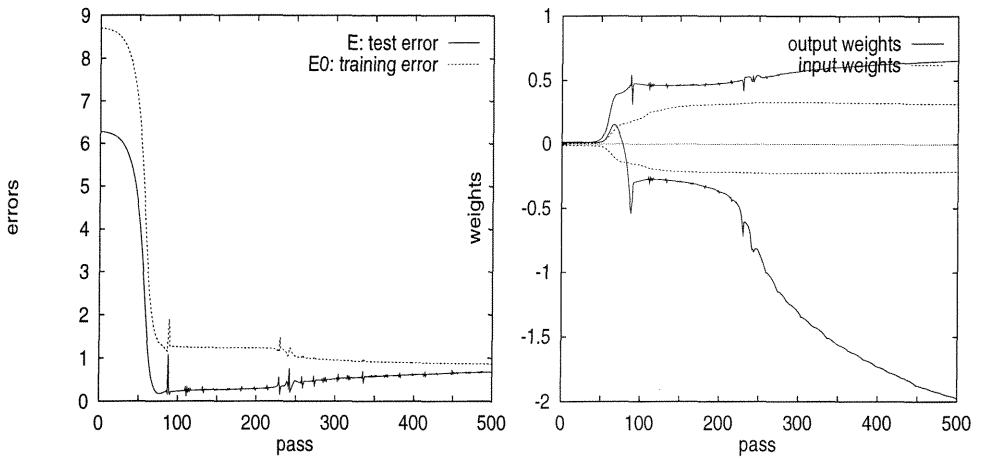

Figure 3.4: Training and test errors and the input and output weigths of a neural network while overtraining occurs.

$\Phi_y = [\phi(\mathbf{y}_1), \ldots, \phi(\mathbf{y}_M)]_{(d+1)\times M}$ be the training and test inputs transformed by the tanh nonlinearities ϕ_i. Let $\frac{\Phi_x \Phi_x^T}{N} = \mathbf{S}_x$ and $\frac{\Phi_y \Phi_y^T}{M} = \mathbf{S}_y$. The gradient and Hessians of the training error E_0 and the augmented term A_{α_1,α_2} become:

$$\begin{aligned}
\nabla E_0(\mathbf{w}) &= 2\mathbf{S}_x\mathbf{w} - 2\frac{\Phi_x \mathbf{f}}{N} \\
\nabla A_{\alpha_1,\alpha_2}(\mathbf{w}) &= 2\left(\alpha_1\mathbf{S}_y - \alpha_2\mathbf{S}_x\right)\mathbf{w} \\
HE_0(\mathbf{w}) &= 2\mathbf{S}_x \\
HA_{\alpha_1,\alpha_2}(\mathbf{w}) &= 2\left(\alpha_1\mathbf{S}_y - \alpha_2\mathbf{S}_x\right)
\end{aligned}$$

3.3.1 Augmented Solution Around Least Squares Solution for Output Weights

Let $\mathbf{\Delta w}_{(d+1)\times 1}$ denote the change to the output weights so that $\hat{\mathbf{w}} + \mathbf{\Delta w}$ is the minimum of the test error. From equation (3.5) we obtain:

$$\begin{aligned}
\mathbf{\Delta w} &= -\left(HE_0(\hat{\mathbf{w}}) + HA_{\alpha_1,\alpha_2}(\hat{\mathbf{w}})\right)^{-1}\left(\nabla E_0(\hat{\mathbf{w}}) + \nabla A_{\alpha_1,\alpha_2}(\hat{\mathbf{w}})\right) \\
&= -\left(\mathbf{S}_x + \alpha_1\mathbf{S}_y - \alpha_2\mathbf{S}_x\right)^{-1}\left(\mathbf{S}_x\hat{\mathbf{w}} - \frac{\Phi_x\mathbf{f}}{N} + \left(\alpha_1\mathbf{S}_y - \alpha_2\mathbf{S}_x\right)\hat{\mathbf{w}}\right) \\
&= -\left(\mathbf{S}_x + \alpha_1\mathbf{S}_y - \alpha_2\mathbf{S}_x\right)^{-1}\left(\mathbf{S}_x + \alpha_1\mathbf{S}_y - \alpha_2\mathbf{S}_x\right)\hat{\mathbf{w}}
\end{aligned}$$

$$
\begin{aligned}
& +(\mathbf{S}_x + \alpha_1 \mathbf{S}_y - \alpha_2 \mathbf{S}_x)^{-1} \frac{\Phi_x \mathbf{f}}{N} \\
= \; & -\hat{\mathbf{w}} + (\mathbf{S}_x + \alpha_1 \mathbf{S}_y - \alpha_2 \mathbf{S}_x)^{-1} \frac{\Phi_x \mathbf{f}}{N} \\
= \; & -\hat{\mathbf{w}} + \left(\mathbf{I} - \left(\alpha_2 \mathbf{I} - \alpha_1 \mathbf{S}_x^{-1} \mathbf{S}_y\right)\right)^{-1} \mathbf{S}_x^{-1} \frac{\Phi_x \mathbf{f}}{N} \\
= \; & -\hat{\mathbf{w}} + \mathbf{Q}_{\alpha_1,\alpha_2} \mathbf{w}_0 \\
\hat{\mathbf{w}} + \Delta\mathbf{w} = \; & \mathbf{Q}_{\alpha_1,\alpha_2} \mathbf{w}_0 = \mathbf{w}_{\alpha_1,\alpha_2} = \mathbf{w}_{\alpha_1,\alpha_2,\mathbf{w}_0}
\end{aligned}
$$

where $\mathbf{Q}_{\alpha_1,\alpha_2} = \left(\mathbf{I} - \left(\alpha_2 \mathbf{I} - \alpha_1 \mathbf{S}_x^{-1} \mathbf{S}_y\right)\right)^{-1}$ and $\mathbf{w}_0 = \mathbf{S}_x^{-1} \frac{\Phi_x \mathbf{f}}{N}$ is the simple training (least squares) solution for the output layer weights given fixed input weights. Note that $\mathbf{w}_{\alpha_1,\alpha_2} = \mathbf{w}_{\alpha_1,\alpha_2,\mathbf{w}_0}$ is exactly the two parameter [3] augmented solution in equation (2.23). Since the input weights will be equal to the input weights of $\hat{\mathbf{v}}$, we will denote the different neural network models by their output weights.

It is possible to use the substitution method (section 2.4) as we did for the general linear model, to find good augmentation parameters. Then we can compute the augmented solution $\mathbf{w}_{\alpha_1,\alpha_2,\mathbf{w}_0}$ according to $\mathbf{Q}_{\alpha_1,\alpha_2} \mathbf{w}_0$, and replace the output weights $\hat{\mathbf{w}}$ by the augmented solution $\mathbf{w}_{\alpha_1,\alpha_2,\mathbf{w}_0}$.

We performed experiments to analyze this method of incorporating test inputs into learning. We obtained the solution $\hat{\mathbf{v}}$ after 1000 passes of training. Then we computed the least squares solution $\mathbf{w}_0$ given the input weights of $\hat{\mathbf{v}}$. Using $\mathbf{w}_0$ and the substitution method, we found augmentation parameters and the augmented solution $\mathbf{w}_{\alpha_1,\alpha_2,\mathbf{w}_0}$ for the output weights. We show the average (over 100 experiments [4]) test error ratios $\frac{E(\mathbf{w}_{\alpha_1,\alpha_2,\mathbf{w}_0})}{E(\mathbf{w}_0)}$ of the augmented and the least squares solution in

[3] For one parameter augmented error we would get $\hat{\mathbf{w}} + \Delta\mathbf{w} = (\mathbf{I} - \alpha\mathbf{R})^{-1} \mathbf{w}_0$ where $\mathbf{R} = \mathbf{I} - \mathbf{S}_x^{-1} \mathbf{S}_y$.

[4] Since other experiments in this chapter will be performed similarly, here are the full details about the experiments shown in the figure: There were 30 noisy training input-outputs and 50 noiseless test input-outputs for each experiment. The inputs were chosen equally spaced from $[-10 : 10]$ range. A teacher neural network with 1 input, 10 hidden and 1 output units was generated by randomly choosing weights from a unit normal. The training inputs were fed to the teacher network. The training outputs were obtained by summing teacher outputs with zero mean normal noise with variance according to the signal-to-noise ratio. The (student) network used for learning was of the same architecture as the teacher. Training started from random weights drawn from zero mean 0.0001 variance normals. The initial learning rate was 0.0001. While descending on the training error, the learning rate was multiplied by 1.1 if the training error decreased and the learning rate was halved when the training error increased. The training was continued for 1000 passes (descent

figure 3.5. The test error ratio is less than 1, therefore the augmented solution has better test error than the solution with least squares output weights.

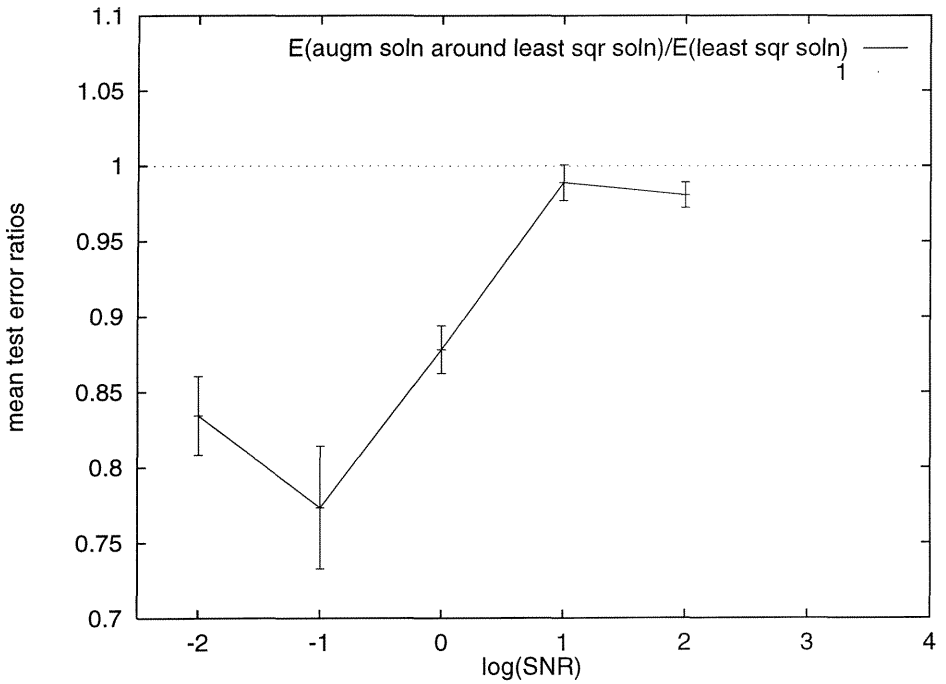

Figure 3.5: The augmented solution $\mathbf{w}_{\alpha_1,\alpha_2,\mathbf{w}_0}$ obtained from the least squares solution $\mathbf{w}_0$, results in smaller test error than the least squares solution $\mathbf{w}_0$.

3.3.2 Augmented Solution Around a Given Solution for Output Weights

When α_1, α_2 are small enough $\mathbf{Q}_{\alpha_1,\alpha_2} = (\mathbf{I} - (\alpha_2\mathbf{I} - \alpha_1\mathbf{S}_x^{-1}\mathbf{S}_y))^{-1} = \mathbf{I} + (\alpha_2\mathbf{I} - \alpha_1\mathbf{S}_x^{-1}\mathbf{S}_y) + (\alpha_2\mathbf{I} - \alpha_1\mathbf{S}_x^{-1}\mathbf{S}_y)^2, \ldots$. Therefore the augmented solution $\mathbf{Q}_{\alpha_1,\alpha_2}\mathbf{w}_0$ is close to the least squares solution $\mathbf{w}_0$, not to the solution $\hat{\mathbf{w}}$ that we started with. The least squares solution is guaranteed to have smaller training error than $\hat{\mathbf{w}}$. However, the test error of the least squares solution is not necessarily smaller than $\hat{\mathbf{w}}$, especially for high noise problems where overfitting to the training data is

steps). In order for the approximation in equation (3.5) to hold, the augmented solution should not to be too far from the least squares solution $\mathbf{w}_0$. Therefore we searched for the augmented parameters starting from 0 and in the region where the spectral radius $(\alpha_2\mathbf{I} - \alpha_1\mathbf{S}_x^{-1}\mathbf{S}_y) < 1$.

possible. In figure 3.6 average (over 100 experiments) test error ratios $\frac{E(\hat{\mathbf{w}})}{E(\mathbf{w}_0)}$ and training error ratios $\frac{E_0(\hat{\mathbf{w}})}{E_0(\mathbf{w}_0)}$ are shown. As expected, the training error of the least squares solution $\mathbf{w}_0$ is smaller than the gradient descent solution $\hat{\mathbf{w}}$. However, the test error of the least squares solution is larger.

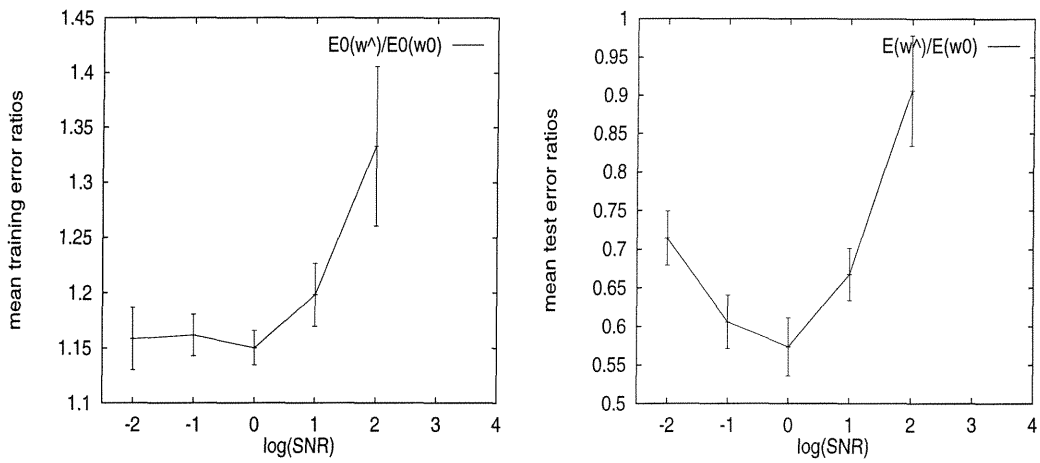

Figure 3.6: Least squares solution $\mathbf{w}_0$ to the output weights of an existing solution $\hat{\mathbf{w}}$ decreases the training error, however it increases the test error.

Similar to the case of the gradient descent solution, an existing solution can be better than the least squares solution. Therefore it is desirable to obtain an augmented solution not necessarily around the least squares solution, but around some given $\hat{\mathbf{w}}$. Now we will discuss extension of the augmented error to obtain an augmented solution around any given $\hat{\mathbf{w}}$.

Let the input weights of the neural network be fixed and consider the output weights as the only variables again. The usual augmented error was:

$$E_{\alpha_1,\alpha_2}(\mathbf{w}) = E_0(\mathbf{w}) + A_{\alpha_1,\alpha_2}(\mathbf{w})$$

For small α_1, α_2 the augmented solution to this equation is in the neighborhood of the minimum of the training error $E_0(\mathbf{w})$. In order to extend the augmented solution

to a neighborhood of a specific $\hat{\mathbf{w}}$ we minimize:

$$E_{\alpha_1,\alpha_2,\hat{\mathbf{w}}}(\mathbf{w}) = (\mathbf{w}-\hat{\mathbf{w}})^T\mathbf{S}_x(\mathbf{w}-\hat{\mathbf{w}}) + A_{\alpha_1,\alpha_2}(\mathbf{w})$$

where $\mathbf{S}_x = \frac{\Phi_x\Phi_x^T}{N}$. The gradient of the new augmented error is:

$$\begin{aligned}\frac{\partial E_{\alpha_1,\alpha_2,\hat{\mathbf{w}}}(\mathbf{w})}{\partial \mathbf{w}} &= 2\mathbf{S}_x\mathbf{w} - 2\mathbf{S}_x\hat{\mathbf{w}} + 2\left(\alpha_1\mathbf{S}_y - \alpha_2\mathbf{S}_x\right)\mathbf{w} \\ &= 2\mathbf{S}_x\left(\mathbf{I} - \left(\alpha_2\mathbf{I} - \alpha_2\mathbf{S}_x^{-1}\mathbf{S}_y\right)\right) - 2\mathbf{S}_x\hat{\mathbf{w}}\end{aligned}$$

At the minimum, the augmented error gradient is $\mathbf{0}$. Therefore the new augmented solution is: $\mathbf{w}_{\alpha_1,\alpha_2,\hat{\mathbf{w}}} = \mathbf{Q}_{\alpha_1,\alpha_2}\hat{\mathbf{w}}$. When α_1, α_2 are such that the spectral radius of $\alpha_1\mathbf{I} - \alpha_2\mathbf{S}_x^{-1}\mathbf{S}_y$ is less than 1, $\mathbf{Q}_{\alpha_1,\alpha_2} = \left(\mathbf{I} - \left(\alpha_2\mathbf{I} - \alpha_1\mathbf{S}_x^{-1}\mathbf{S}_y\right)\right)^{-1} = \mathbf{I} + \left(\alpha_2\mathbf{I} - \alpha_1\mathbf{S}_x^{-1}\mathbf{S}_y\right) + \left(\alpha_2\mathbf{I} - \alpha_1\mathbf{S}_x^{-1}\mathbf{S}_y\right)^2, \ldots$. Therefore the new augmented solution is around $\hat{\mathbf{w}}$.

The substitution method can again be used with this solution. Let us assume that the current solution $\hat{\mathbf{w}}$ is a noisy version of the target the same way the least squares solution $\mathbf{w}_0$ is. Then $\hat{\mathbf{w}}$ can be used for the estimation of the target and the noise variance in the expected test error. Once the α_1, α_2 that minimizes the approximation to the expected test error is found, it can be used to find the new solution $\mathbf{w}_{\alpha_1,\alpha_2,\hat{\mathbf{w}}}$. Replacing the output weights $\hat{\mathbf{w}}$ by $\mathbf{w}_{\alpha_1,\alpha_2,\hat{\mathbf{w}}}$, we obtain the new augmented solution. Note that if any specific properties of $\hat{\mathbf{w}}$ or the noise in it, is known, it can be incorporated in the substitution method, while computing the expected test error of $\mathbf{w}_{\alpha_1,\alpha_2,\hat{\mathbf{w}}}$.

We show the experimental performance of the augmented solution around the solution $\hat{\mathbf{w}}$ in figure 3.7. In this experiment also, the weights $\hat{\mathbf{v}}$ of the neural network was obtained after 1000 passes of gradient descent on the training data. Especially for small signal-to-noise ratio, the test error of the augmented solution $\mathbf{w}_{\alpha_1,\alpha_2,\hat{\mathbf{w}}}$ is smaller than the solution $\hat{\mathbf{w}}$. The augmented solution again is better than the solution that was used to obtain it.

In figure 3.8 we show the functions implemented by different methods for a single

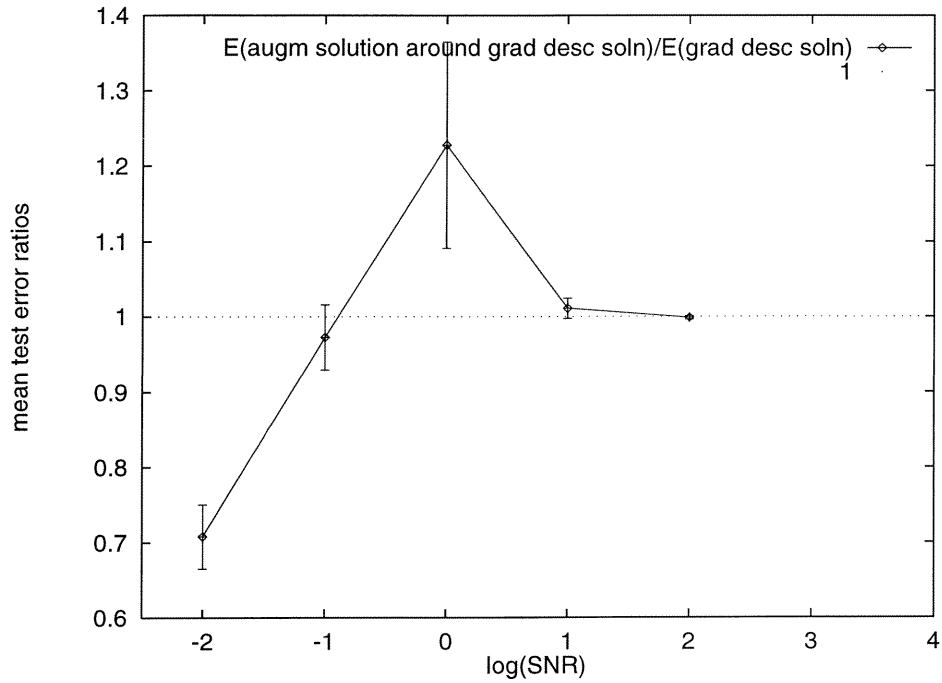

Figure 3.7: The augmented solution $\mathbf{w}_{\alpha_1,\alpha_2,\hat{\mathbf{w}}}$ obtained from the gradient descent solution $\hat{\mathbf{w}}$, results in smaller test error than the gradient descent solution $\hat{\mathbf{w}}$.

experiment. The top plot shows the outputs of the gradient descent solution and the augmented solution around the gradient descent solution. Notice that the augmented solution is fitting the noise less than the gradient descent solution does. The bottom plot shows the least squares solution and the augmented solution around the least squares solution for the same experiment. The least squares solution is closer to the training data points, and it fits the noise more than the gradient descent solution does. The augmented solution, again, is fitting the noise less than the least squares solution.

3.4 Cross Validation to Find the Augmentation Parameters

As an alternative to the substitution method, the ordinary cross validation method can also be used to obtain the augmentation parameters.

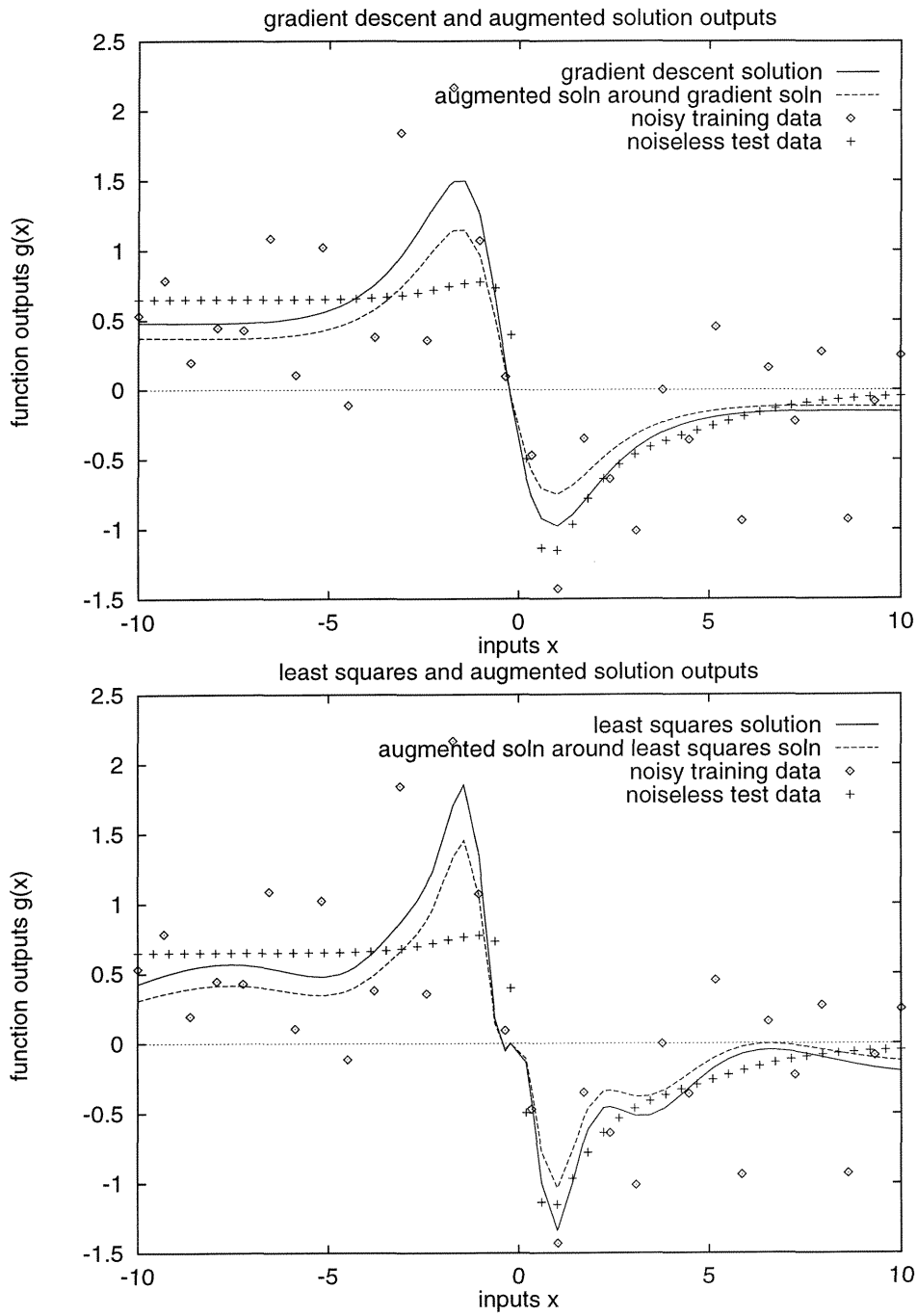

Figure 3.8: The gradient descent (top) and least squares solutions (bottom) and the augmented solutions around these solutions.

Let $\hat{\mathbf{v}}$ be a given solution that is close to the test error minimum. Let $\hat{\mathbf{w}}$ be the output weights of $\hat{\mathbf{v}}$. The ordinary cross validation method measures goodness of the augmentation parameters α_1, α_2 around $\hat{\mathbf{w}}$ according to:

$$V(\alpha_1, \alpha_2, \hat{\mathbf{w}}) = \frac{1}{N} \sum_{n=1}^{N} \left(g_{\mathbf{w}_{\alpha_1,\alpha_2,\hat{\mathbf{w}}^n}}(\mathbf{x}_n) - f_n \right)^2 \tag{3.6}$$

where $\mathbf{w}_{\alpha_1,\alpha_2,\hat{\mathbf{w}}^n}$ is the augmented solution around $\hat{\mathbf{w}}$ with parameters α_1, α_2 and using all training examples except the nth one.

Since ordinary cross validation solutions may fluctuate due to noisy examples in the training set, instead of leaving only 1 example out, leaving $k > 1$ examples out at a time may make the solutions more stable. However, since there are $\binom{N}{k}$ such cross validation set choices, the computational overhead is too much. Instead of all $\binom{N}{k}$ possible cross validation sets, we experimented with partitioning the training set into 10 sets of size $k = \frac{N}{10}$ and choosing the parameters that minimize the mean cross validation error on the left out cross validation set.

Beginning from $\alpha_1 = \alpha_2 = 0$ we search for the minimum of $V(\alpha_1, \alpha_2, \hat{\mathbf{w}})$ within the region where the spectral radius of $\alpha_1 \mathbf{I} - \alpha_2 \mathbf{S}_x^{-1} \mathbf{S}_y$ is less than 1. We use the gradient descent with adaptive learning rate as the search algorithm.

In figure 3.9 we show the experimental results of using ordinary cross validation and leave-$\frac{N}{10}$-out methods to determine the augmentation parameters. The mean (over 100 experiments) of augmented solution test error divided by the gradient descent solution test error is shown in the figure. For comparison we also show the mean test error ratios for the augmented solution with augmentation parameters determined by means of substitution method. Substitution method results in smaller test error than both cross validation methods. Although the leave-$\frac{N}{10}$-out cross validation performs poorly for this case, in section 3.6 we will see its better use.

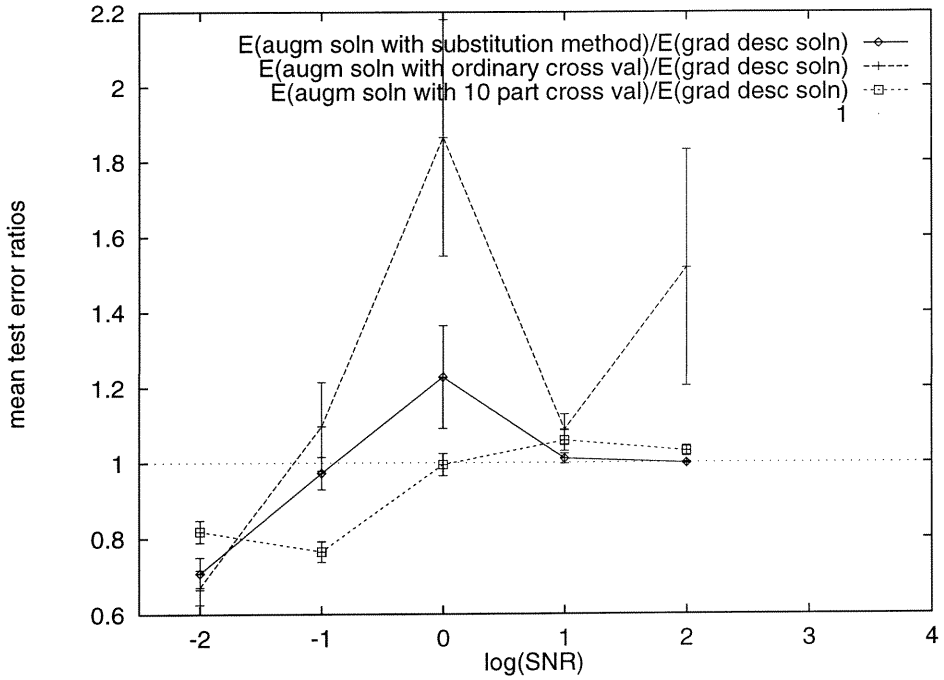

Figure 3.9: Augmentation parameters determined using substitution method result in smaller test error than augmented parameters determined using ordinary and leave-$\frac{N}{10}$-out cross validation methods.

3.5 Different Uses of the Augmented Solution

In the previous sections we discussed how to obtain an augmented solution from an already existing solution. In this section, we discuss other possible uses of the augmented solution.

3.5.1 Using the Augmented Solution Method Repetitively

Since the augmented solution found by the substitution method is better than the gradient descent solution, can we find a better augmented solution using this augmented solution and the substitution method again? Our experiments show that the answer depends on the signal-to-noise ratio. We experimented with using an augmented solution (1st) to obtain another augmented solution (2nd) and then using that augmented solution to obtain another augmented solution (3rd). We

performed the same experiment both for augmented solution around the gradient descent solution and the augmented solution around the least squares solution. The mean test errors of the augmented solutions (1,2,3) divided by the test error of the gradient descent (least squares respectively) solution are shown in the top (bottom respectively) plot in figure 3.10. For small signal-to-noise ratio using the augmented solution repetitively helps, but it hurts when the signal-to-noise ratio is large. When the augmented solution is used repetitively, in some sense "overtraining" on the augmented solution starts.

3.5.2 Which Solution is Good Enough to Obtain an Augmented Solution Around

We have obtained augmented solutions around the gradient descent solution obtained after some fixed number of passes of gradient descent. After how many number of passes is the augmented solution better than the gradient solution? It turns out that, except some first passes of training, especially for small signal-to-noise ratio, the augmented solution is better than the gradient descent solution. In figure 3.11 we show the test error of the gradient descent solution and the augmented solution obtained around that gradient descent solution while descending on the training error. The augmented solution consistently has smaller test error after the first 100 passes. The bottom plot shows, the augmented term for $\alpha_1 = \alpha_2 = 1$. The augmented term for the gradient solution keeps increasing as overtraining occurs, whereas the augmented term for the augmented solution does not show such an increase.

3.5.3 Augmented Solution Around Early Stopping Using a Validation Set Solution

Early stopping using a validation set works as follows: A subset of all training data is spared as the validation set, while minimizing the error on the remaining data, the error on the validation set is monitored. Among the solutions visited during the descent, the one with the smallest validation error is chosen to be the early stopping

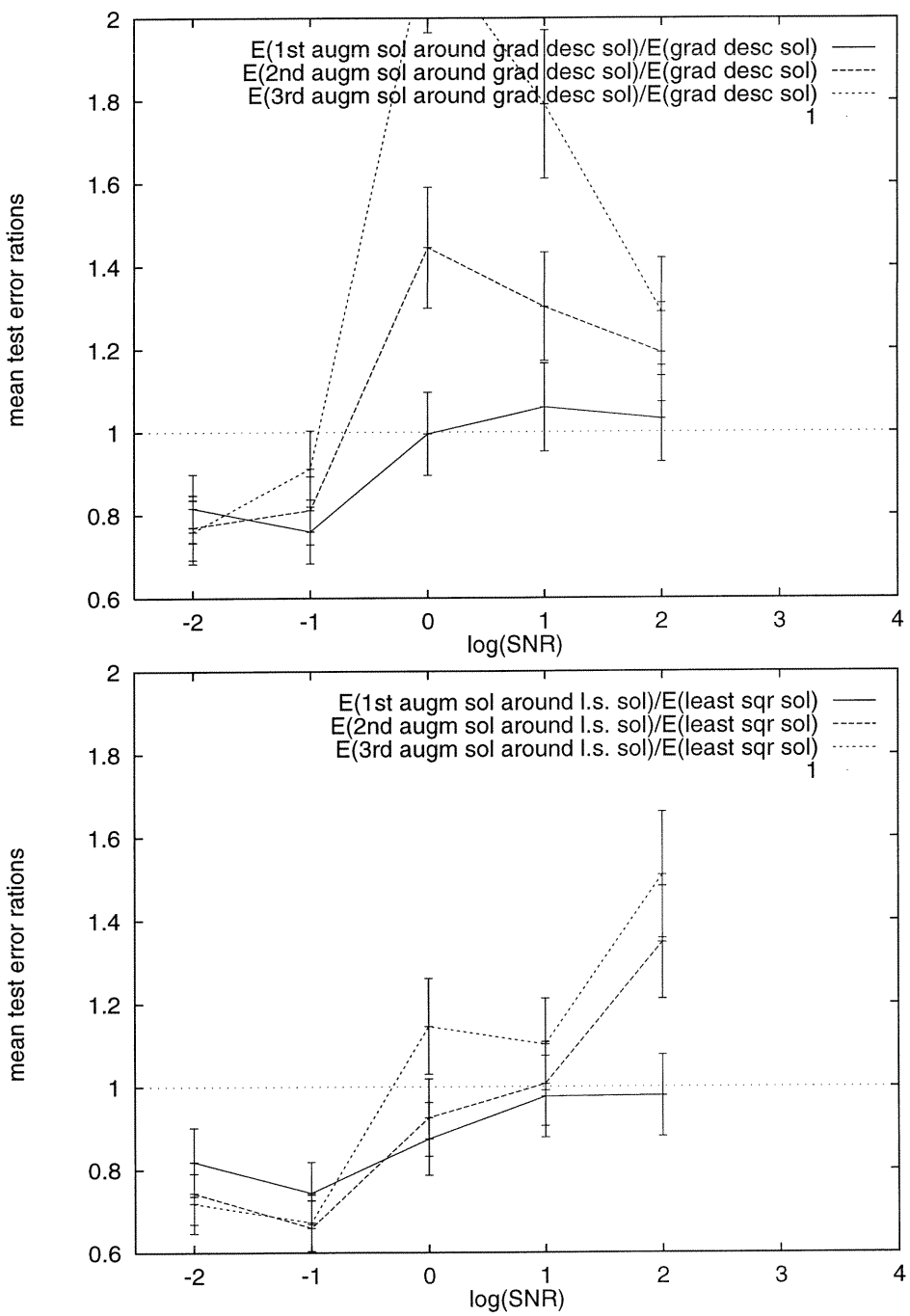

Figure 3.10: Repetition of the augmented solution finding process around the newly found solutions may result in worse test error.

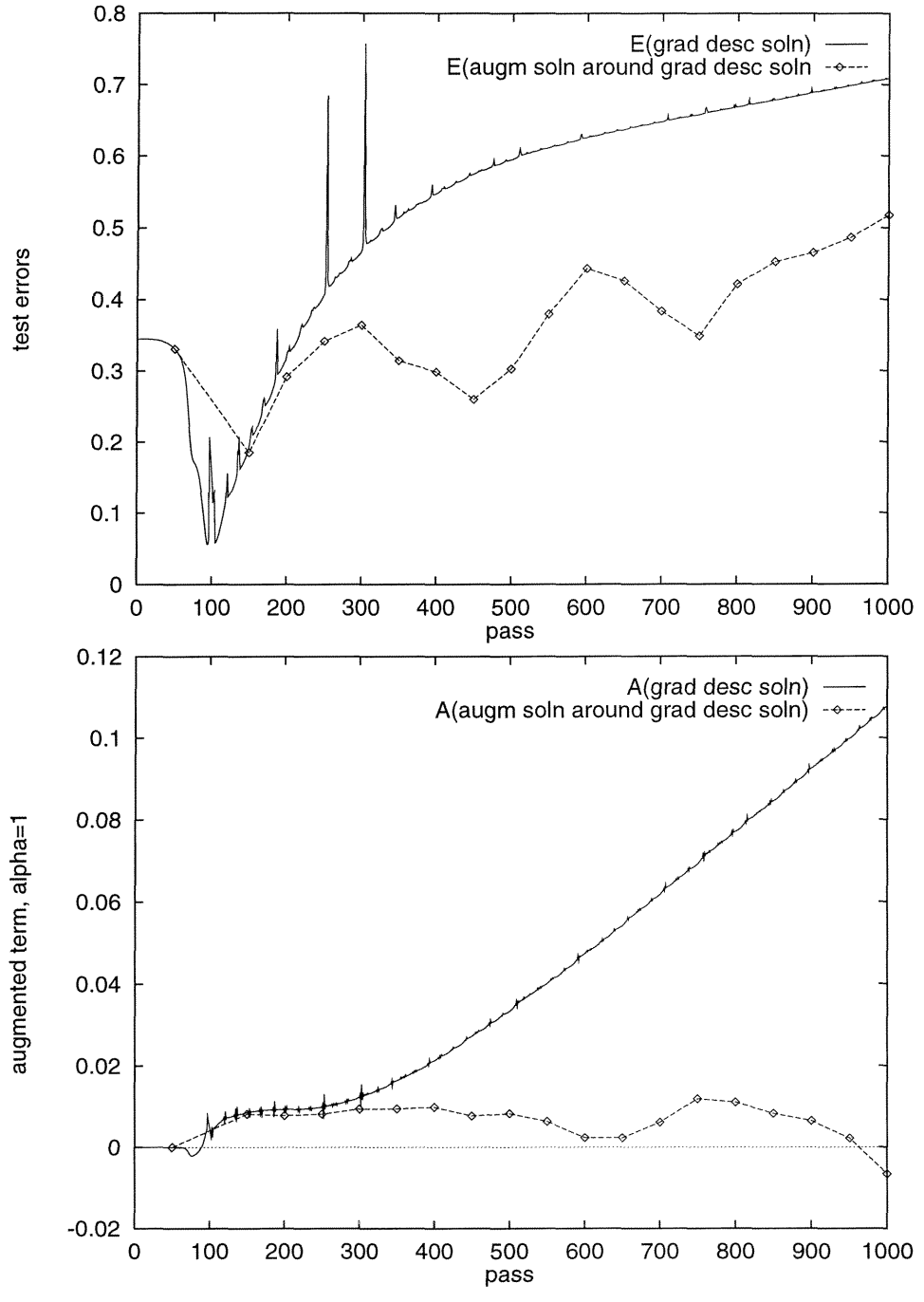

Figure 3.11: When the signal-to-noise ratio is small, except the solutions at the first passes, the augmented solution is better than the gradient descent solution.

using a validation set solution.

In sections 3.2 and 3.3.2 we computed the augmented solution around the gradient descent solution. The early stopping using a validation set also results in a solution and augmented solution can be computed around this solution as well. As can be seen from figure 3.12 early stopping using a validation set results in much smaller test error than the gradient descent algorithm when the signal-to-noise ratio is small. However, it performs equally badly when the signal-to-noise ratio is high. The augmented solution around the early stopping using a validation set solution performs about as good (or bad) as the early stopping using a validation set solution. Early stopping using a validation set is a good idea only when the signal-to-noise ratio is small. In the experiments shown, the validation set size was $\frac{N}{3}$ where N is the training set size. The validation set size plays a very important role in the success of early stopping using a validation set.

3.5.4 Early Stopping Based on the Augmented Term

The augmented term $A_{\alpha_1,\alpha_2}(g_{\mathbf{v}}) = \alpha_1 \frac{1}{M} \sum_{m=1}^{M} g_{\mathbf{v}}^2(\mathbf{y}_m) - \alpha_2 \frac{1}{N} \sum_{n=1}^{N} g_{\mathbf{v}}^2(\mathbf{x}_n)$ for $\alpha_1 = \alpha_2 = 1$ is very small at the beginning of training. Because the function $g_{\mathbf{v}}$ implemented by the neural network is initially linear in $\mathbf{x}$ and takes small values. As training proceeds $g_{\mathbf{v}}$ starts to get closer to the training data and A starts to increase in absolute value. When overtraining occurs, especially for small signal-to-noise ratio, the augmented term tends to increase (or decrease) constantly and takes values much larger in magnitude then its values at the beginning of training (for example, see figure 3.11 the bottom plot).

Although, for small signal-to-noise ratio, early stopping of training based on the constant increase/decrease in the augmented term results in better test error than the gradient descent solution, for large signal-to-noise ratio early stopping based on the same criterion results in worse test error than the gradient descent solution. If the augmented term remains small in magnitude, it usually means that overtraining is not taking place. However, if it is getting large, depending on the signal-to-noise

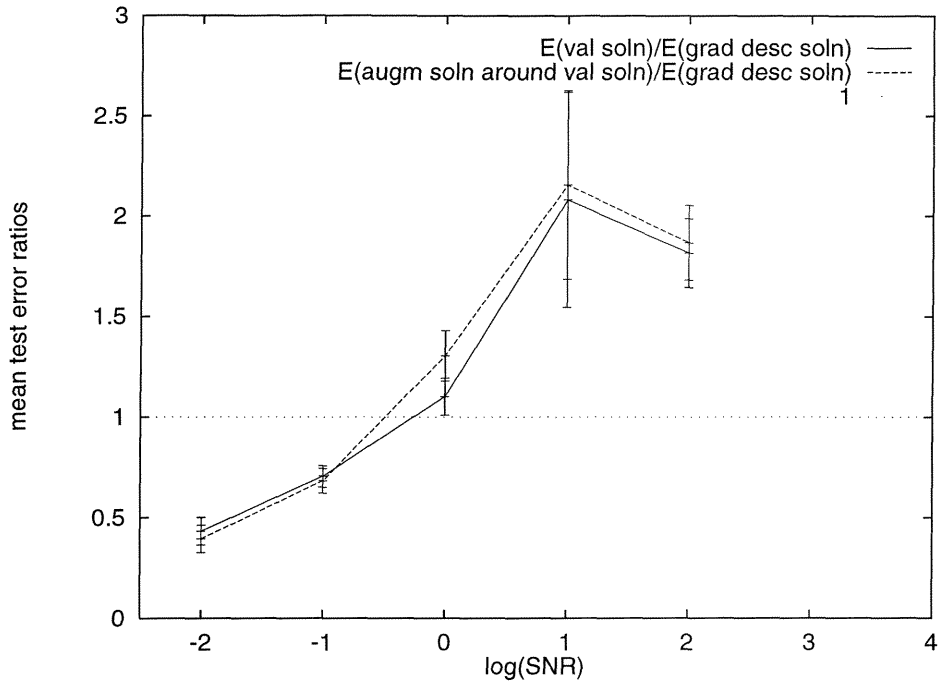

Figure 3.12: When the signal-to-noise ratio is small early stopping using a validation set solution has smaller test error than gradient descent solution. When the signal-to-noise ratio is large the opposite happens. In both cases, the augmented solution around the early stopping using a validation set performs as good (or bad) as the early stopping using a validation set solution.

ratio, there may or may not be overtraining.

3.6 Gradient Descent on the Augmented Error

Previous sections concentrated on gradient descent on the training error only. The augmented error was used on the solutions obtained after gradient descent on the training error. In this section, we will consider descending along the gradient of the augmented error, computing the augmentation parameters adaptively during the descent. Let $g_{\hat{\mathbf{v}}}$ be the current function being implemented by the neural network. The best gradient descent direction at $\hat{\mathbf{v}}$ is $-\nabla E(g_{\hat{\mathbf{v}}}) = -\left.\frac{dE(g_{\mathbf{v}})}{d\mathbf{v}}\right|_{\hat{\mathbf{v}}}$. Let augmentation

parameters α_1, α_2 be such that in the neighborhood of $\hat{\mathbf{v}}$:

$$\begin{aligned} \nabla E(g_{\hat{\mathbf{v}}}) &\approx \nabla E_{\alpha_1,\alpha_2}(g_{\hat{\mathbf{v}}}) \\ &= \nabla E_0(g_{\hat{\mathbf{v}}}) + \nabla A_{\alpha_1,\alpha_2}(g_{\hat{\mathbf{v}}}) \end{aligned}$$

where E_{α_1,α_2} is the augmented error, E_0 is the training error and A_{α_1,α_2} is the augmented term as we defined in section 3.1. If we had not known the test inputs, we would approximate the test error gradient ∇E by the training error gradient ∇E_0 only.

Now we need to find the parameters α_1, α_2 at each step of training, for any $\hat{\mathbf{v}}$. We can not use the substitution method in this case, since the hidden unit tanh functions may still be in their linear region and the input weights may need to be changed. Instead we will use cross validation, leaving out k examples at a time, to find the augmentation parameters.

Let the training set $(\mathbf{X}, \mathbf{f})$ be partitioned into k disjoint parts: $(\mathbf{X}_1, \mathbf{f}_1), \ldots, (\mathbf{X}_k, \mathbf{f}_k)$. Denote the training error of $g_{\mathbf{v}}$ on all 9 parts except the ith one by $E_{t,i}(g_{\mathbf{v}})$, denote the validation error on the ith part by $E_{v,i}(g_{\mathbf{v}})$. Denote the augmented term using all training inputs except the ones in part i by $A_{\alpha_1,\alpha_2,i}(g_{\mathbf{v}}) = \alpha_1 A_y(g_{\mathbf{v}}) - \alpha_2 A_{x,i}(g_{\mathbf{v}})$ where $A_y(g_{\mathbf{v}}) = \frac{1}{M}\sum_{m=1}^{M} g_{\mathbf{v}}^2(\mathbf{y}_m)$ and $A_{x,i}(g_{\mathbf{v}}) = \frac{1}{N-|\mathbf{X}_i|}\sum_{n=1,\mathbf{x}_n \notin \mathbf{X}_i}^{N} g_{\mathbf{v}}^2(\mathbf{x}_n)$. We need to find α_1, α_2 such that:

$$\begin{aligned} \nabla E_{v,1}(g_{\mathbf{v}}) &= \nabla E_{t,1}(g_{\mathbf{v}}) + \alpha_1 \nabla A_y(g_{\mathbf{v}}) - \alpha_2 \nabla A_{x,1}(g_{\mathbf{v}}) \\ \vdots &= \vdots \\ \nabla E_{v,k}(g_{\mathbf{v}}) &= \nabla E_{t,k}(g_{\mathbf{v}}) + \alpha_1 \nabla A_y(g_{\mathbf{v}}) - \alpha_2 \nabla A_{x,k}(g_{\mathbf{v}}) \end{aligned}$$

We obtain α_1, α_2 by simultaneously solving all $kdim(\mathbf{v})$ equations. We shuffle the training set and obtain different partitionings $(\mathbf{X}_1, \mathbf{f}_1), \ldots, (\mathbf{X}_k, \mathbf{f}_k)$ at each pass of descent. There are $\binom{N}{k}$ cross validation choices and shuffling allows different partitionings to be used to find the augmentation parameter.

In figure 3.13 we show the results of using leave-1-out (ordinary) and leave-$\frac{N}{10}$-out

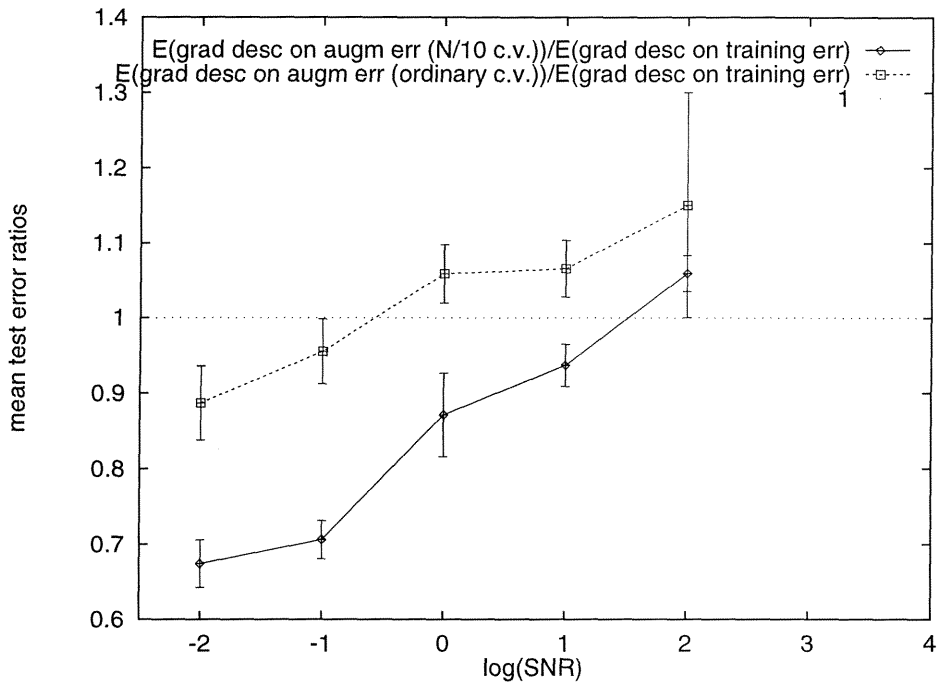

Figure 3.13: Obtaining the augmentation parameters via leave-$\frac{N}{10}$-out cross validation and then gradient descent on the augmented error results in better test error than gradient descent on the training error alone.

cross validation to find the augmentation parameters and then descending on the augmented error with adaptive learning rate. Cross validation leaving $\frac{N}{10}$ examples at a time results in smaller test error than both gradient descent along the training error E_0 and the ordinary cross validation methods. Moreover, comparison with figure 3.7 shows that using leave-$\frac{N}{10}$-out cross validation to determine the augmentation parameters and descending along the gradient of the augmented error than using the substitution method to find the augmented solution for the output weights only.

3.7 Loss Functions other than Quadratic Loss

In section 2.1 we derived the augmented error for the quadratic loss function. The same method of better estimating the test error by means of input information can be used for other loss functions as well. We will derive the augmented error for different

loss functions in this section.

3.7.1 Entropic Loss

The entropic loss function is appropriate especially for binary decision problems. Entropic loss is the maximum likelihood solution when the data is noisy and the noise has binomial distribution [Bishop, 1995, page 231]. For some problems gradient descent has been shown to find the solution for entropic loss, whereas it could not succeed for the quadratic loss [Hertz et al., 1991, page 109].

When the loss function is entropic, the test error is:

$$E\left(g_{\mathbf{v}}\right) = \frac{1}{M}\sum_{m=1}^{M}\left(1+h_m\right)\log\frac{1+h_m}{1+g_{\mathbf{v}}(\mathbf{y}_m)} + \left(1-h_m\right)\log\frac{1-h_m}{1-g_{\mathbf{v}}(\mathbf{y}_m)}$$

Similarly, the training error is:

$$E_0\left(g_{\mathbf{v}}\right) = \frac{1}{N}\sum_{n=1}^{N}\left(1+f_n\right)\log\frac{1+f_n}{1+g_{\mathbf{v}}(\mathbf{x}_n)} + \left(1-f_n\right)\log\frac{1-f_n}{1-g_{\mathbf{v}}(\mathbf{x}_n)}$$

Expanding log's in the test error and taking the terms that solely depend on the test inputs as they are, we obtain the augmented error:

$$E_\alpha\left(g_{\mathbf{v}}\right) = E_0\left(g_{\mathbf{v}}\right) + \alpha\left(\frac{1}{N}\sum_{n=1}^{N}\log\left(\left(1+g_{\mathbf{v}}(\mathbf{x}_n)\right)\left(1-g_{\mathbf{v}}(\mathbf{x}_n)\right)\right) - \frac{1}{M}\sum_{m=1}^{M}\log\left(\left(1+g_{\mathbf{v}}(\mathbf{y}_m)\right)\left(1-g_{\mathbf{v}}(\mathbf{y}_m)\right)\right)\right)$$

3.7.2 Maximum Likelihood with Input Dependent Noise Variance

Assuming that the noise added to the outputs are normal, minimizing the simple training error in equation (2.1) maximizes the probability of the training outputs (with input-independent noise) given the model. When the noise variance σ_e^2 is input dependent, and model $s_{\mathbf{u}}(\mathbf{x}) \geq 0$ is used to predict the $\sigma_e(\mathbf{x})$, the negative

log-likelihood of the data given $g_{\mathbf{v}}$ and $s_{\mathbf{u}}$ is:

$$\begin{aligned} -\log \mathcal{L}\left(g_{\mathbf{v}}, s_{\mathbf{u}}\right) &= -\log \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi} s_{\mathbf{u}}} e^{-\frac{\left(g_{\mathbf{v}}(\mathbf{x}_n)-f_n\right)^2}{2 s_{\mathbf{u}}^2(\mathbf{x}_n)}} \\ &\propto \frac{1}{N} \sum_{n=1}^{N} 2 \log s_{\mathbf{u}}(\mathbf{x}_n) + \frac{\left(g_{\mathbf{v}}\left(\mathbf{x}_n\right)-f_n\right)^2}{s_{\mathbf{u}}^2(\mathbf{x}_n)} = E_0\left(g_{\mathbf{v}}, s_{\mathbf{u}}\right) \end{aligned}$$

Similarly the test error is:

$$E\left(g_{\mathbf{v}}, s_{\mathbf{u}}\right) = \frac{1}{M} \sum_{m=1}^{M} 2 \log s_{\mathbf{u}}(\mathbf{y}_m) + \frac{\left(g_{\mathbf{v}}\left(\mathbf{y}_m\right)-h_m\right)^2}{s_{\mathbf{u}}^2(\mathbf{y}_m)}$$

Again collecting the terms in the test error that solely depend on the test inputs, and estimating the remaining terms using the training data, the augmented error becomes:

$$\begin{aligned} &E_\alpha\left(g_{\mathbf{v}}, s_{\mathbf{u}}\right) = E_0\left(g_{\mathbf{v}}, s_{\mathbf{u}}\right) + \\ &\alpha\left(\frac{1}{M} \sum_{m=1}^{M} 2 \log s_{\mathbf{u}}(\mathbf{y}_m) + \frac{g_{\mathbf{v}}^2\left(\mathbf{y}_m\right)}{s_{\mathbf{u}}^2(\mathbf{y}_m)} - \frac{1}{N} \sum_{n=1}^{N} 2 \log s_{\mathbf{u}}(\mathbf{x}_n) + \frac{g_{\mathbf{v}}^2\left(\mathbf{x}_n\right)}{s_{\mathbf{u}}^2(\mathbf{x}_n)}\right) \end{aligned}$$

3.7.3 p-norm Loss

Let the test and training errors be:

$$E\left(g_{\mathbf{v}}\right) = \frac{1}{M} \sum_{m=1}^{M} \left|g_{\mathbf{v}}\left(\mathbf{y}_m\right) - h_m\right|^p \qquad E_0\left(g_{\mathbf{v}}\right) = \frac{1}{N} \sum_{n=1}^{N} \left|g_{\mathbf{v}}\left(\mathbf{x}_n\right) - f_n\right|^p$$

where $p \in R$. When quadratic error is used, the augmented error made the 2nd power of model outputs on training and test inputs close to each other. For the p-norm error we suggest the following augmented error [5]:

$$E_\alpha(g_{\mathbf{v}}) = E_0(g_{\mathbf{v}}) + \alpha\left(\frac{1}{M} \sum_{m=1}^{M} \left|g_{\mathbf{v}}\left(\mathbf{y}_m\right)\right|^p - \frac{1}{N} \sum_{n=1}^{N} \left|g_{\mathbf{v}}\left(\mathbf{x}_n\right)\right|^p\right)$$

[5] Thanks to Hans-Georg Zimmermann of Siemens for bringing this error function to my attention at NIPS 97.

3.8 Conclusions

In this chapter we have analyzed two different methods of incorporating test inputs into learning for the neural network models. In general, both methods result in smaller test error than gradient descent on the training error. If there is a solution that is believed to be close to the test error minimum, then an augmented solution using substitution method can be obtained by means of changing only the output weights of the neural network. Descending on the augmented error, obtaining the augmented parameters by means leave-k-out cross validation, also results in better solutions than gradient descent on the training error only. We have also shown that for loss functions other than the quadratic loss, different forms of augmented error can be derived.

Chapter 4
Learning from Hints

In learning-from-examples we are given a set of input-output examples (training set) and our goal is to find the model that performs best out-of-sample. Sometimes, in addition to the training set, some additional information or hint [Abu-Mostafa, 1990] about the underlying mapping is also available. For example, in character recognition, in addition to characters and their labels, it is also known that when characters are translated, scaled or rotated slightly, the label should remain the same (invariance hints). Another example (monotonicity hint) is from credit card approval: if person A and B have the same specifications except that A earns more money, then A is less likely to default than B.

Hints have been shown to be helpful in learning [Abu-Mostafa, 1990, Abu-Mostafa, 1993a, Abu-Mostafa, 1993b, Abu-Mostafa, 1994]. Invariance hints [Fyfe, 1992, Cataltepe and Abu-Mostafa, 1993], monotonicity hint [Sill and Abu-Mostafa, 1997], smoothness hint [Ji et al., 1990], minimum Hamming distance between patterns [Al-Mashouq and Reed, 1991] are some of the hints that have been studied previously.

We first review hints and learning-from-hints in section 4.1. In this section we also define the hint objective function that allows teaching both the training examples and the hints. In section 4.2, a method of estimating the out-of-sample error using the invariance hints and then early stopping on this estimate of test error is analyzed. The hint objective function that needs to be minimized for learning-from-hints is very similar to the augmented error of chapters 2 and 3. Similar to section 3.6, in section 4.3 we show a method of descending on the hint objective function, obtaining the hint parameters by means of cross validation method. Finally section 4.4 summarizes the chapter.

4.1 Definitions and Notation

Notation in this chapter is similar to notation of section 2.1. Let the training set be $\{(\mathbf{x}_1, f_1), \ldots, (\mathbf{x}_N, f_N)\}$ with inputs $\mathbf{x}_n$ and (possibly noisy) target outputs f_n. Let the model class be G and denote the model by $g_{\mathbf{v}} \in G$, where $\mathbf{v}$ is a parameter vector. Usually G will contain all functions that can be implemented by different settings of the weights of a neural network with a specific architecture. We will assume that the performance of the model will be measured on an unknown test set $\{(\mathbf{y}_1, h_1), \ldots, (\mathbf{y}_M, h_M)\}$ where $(\mathbf{y}, h)$ and $(\mathbf{x}, f)$ pairs are drawn from the same distribution. The training error E_0, test error E and the generalization error E_{gen} of model $g_{\mathbf{v}}$ is defined as:

$$E_0\left(g_{\mathbf{v}}\right) = \frac{1}{N}\sum_{n=1}^{N}\left(g_{\mathbf{v}}\left(\mathbf{x}_n\right) - f_n\right)^2 \tag{4.1}$$

$$E\left(g_{\mathbf{v}}\right) = \frac{1}{M}\sum_{m=1}^{M}\left(g_{\mathbf{v}}\left(\mathbf{y}_m\right) - h_m\right)^2 \tag{4.2}$$

$$E_{gen}\left(g_{\mathbf{v}}\right) = \left\langle\left(g_{\mathbf{v}}(\mathbf{x}) - h(\mathbf{x})\right)^2\right\rangle_{\mathbf{x}} \tag{4.3}$$

where $\langle . \rangle_{\mathbf{x}}$ denotes expectation with respect to the (unknown) input distribution $P_{\mathbf{x}}$.

A **hint** is any piece of information known about mapping f that generated the training outputs $\mathbf{f}$. We will define a hint H_h by an error function $e_h(g_{\mathbf{v}}, \mathbf{x})$ associated with the input $\mathbf{x}$. $e_h(g_{\mathbf{v}}, \mathbf{x})$ will measure how much $g_{\mathbf{v}}$ does not agree with the hint on input $\mathbf{x}$. For example:

- H_0, examples hint given by the training set:
  $e_0(g_{\mathbf{v}}, \mathbf{x}_n) = \left(f_n - g_{\mathbf{v}}(\mathbf{x}_n)\right)^2$
- invariance hints: $f(\mathbf{x}) = f(\mathbf{x}')$ where $\mathbf{x}'$ is obtained by the invariant transformation of $\mathbf{x}$,
  $e_h(g_{\mathbf{v}}, \mathbf{x}) = \left(g_{\mathbf{v}}(\mathbf{x}) - g_{\mathbf{v}}(\mathbf{x}')\right)^2$, where, for example:
  - evenness: $\mathbf{x}' = -\mathbf{x}$
  - scale invariance: $\mathbf{x}' = a\mathbf{x}$ for a constant a;

– cyclic shift invariance: if $\mathbf{x} = [x_1\ x_2\ \ldots\ x_d]^T$, then $\mathbf{x}' = [x_2\ \ldots\ x_d\ x_1]^T$

In this thesis we will concentrate on the evenness invariance hint, which is basically the scale invariance with $a = -1$.

- binary hint: f is a binary function,
  $e_h(g_\mathbf{v}, \mathbf{x}) = g_\mathbf{v}(\mathbf{x}) * (1 - g_\mathbf{v}(\mathbf{x}))$

- monotonicity hint: $f(\mathbf{x})$ is an increasing function of $\mathbf{x}$ according to some ordering $\prec$ of $\mathbf{x}$:
  $$e_h(g_\mathbf{v}, \mathbf{x}) = \begin{cases} (g_\mathbf{v}(\mathbf{x}) - g_\mathbf{v}(\mathbf{x}'))^2 & \text{if } \mathbf{x}' \prec \mathbf{x} \text{ and } g_\mathbf{v}(\mathbf{x}) > g_\mathbf{v}(\mathbf{x}') \\ & \text{or } \mathbf{x} \prec \mathbf{x}' \text{ and } g_\mathbf{v}(\mathbf{x}) < g_\mathbf{v}(\mathbf{x}') \ ; \\ 0 & \text{otherwise.} \end{cases}$$

- approximation hint: $f(\mathbf{x}) \in [c_{min}, c_{max}]$:
  $$e_h(g_\mathbf{v}, \mathbf{x}) = \begin{cases} (c_{min} - g_\mathbf{v}(\mathbf{x}))^2 & \text{if } g_\mathbf{v}(\mathbf{x}) < c_{min}; \\ (g_\mathbf{v}(\mathbf{x}) - c_{max})^2 & \text{if } g_\mathbf{v}(\mathbf{x}) > c_{max}; \\ 0 & \text{otherwise.} \end{cases}$$

- smoothness hint: magnitude of the kth derivative of f with respect to the input is always less than some c_k:
  $e_h(g_\mathbf{v}, \mathbf{x}) = \left|\left|\frac{\partial^{(k)} g_\mathbf{v}(\mathbf{x})}{\partial \mathbf{x}^{(k)}}\right|\right|$, if $\left|\left|\frac{\partial^{(k)} g_\mathbf{v}(\mathbf{x})}{\partial \mathbf{x}^{(k)}}\right|\right| > c_k$.

We will define the hint error for hint H_h by:

$$E_h(g_\mathbf{v}) = \langle e_h(g_\mathbf{v}, \mathbf{x}_n) \rangle_\mathbf{x} \tag{4.4}$$

where $\langle . \rangle_\mathbf{x}$ denotes expectation with respect to input $\mathbf{x}$.

The examples hint H_0 is defined on the training inputs. With a uniform distribution on the training inputs, the hint error for the examples hint is:

$$E_0(g_\mathbf{v}) = \frac{1}{N}\sum_{n=1}^{N} e_0(g_\mathbf{v}, \mathbf{x}_n) = \frac{1}{N}\sum_{n=1}^{N} (g_\mathbf{v}(\mathbf{x}_n) - f_n)^2$$

which is the training error itself.

The other hints can be defined on one or more of the following input sets:

- On training inputs:

$$E_h(g_{\mathbf{v}}) = \frac{1}{N}\sum_{n=1}^{N} e_h(g_{\mathbf{v}}, \mathbf{x}_n)$$

- If the test inputs are available, on test inputs:

$$E_h(g_{\mathbf{v}}) = \frac{1}{M}\sum_{m=1}^{M} e_h(g_{\mathbf{v}}, \mathbf{y}_m)$$

- If the hint is known to hold everywhere in the input space, on a set of inputs $\mathbf{z}_1, \ldots, \mathbf{z}_{N_h}$ randomly drawn from a distribution centered around the training (and test, if they are available) inputs:

$$E_h(g_{\mathbf{v}}) = \frac{1}{N_h}\sum_{i=1}^{N_h} (g_{\mathbf{v}}, \mathbf{z}_i)$$

  N_h depends on the resources available. If N_h is too large, too much time may be spent on computations for the hint error.

When we know both the examples hint H_0 and hints $H_1, \ldots, H_H$, we would like to obtain a model that implements both the training set and the other hints available. If a hint is known to hold certainly, it may be possible to implement the hint directly on the model (see, for example, [Giles and Maxwell, 1987] for direct implementation of invariance hints). However, direct implementation is not always possible, and it may not even be desirable since the hint is not known to hold for sure. Similar to the case of noisy training examples, we may not want the model to fit the hint exactly if the hint is "noisy". For example, in the case of credit card approval, although high salary makes a person less likely to default, due to other factors, another person earning less may be less likely to default.

It is desirable to simultaneously minimize the training error together with the other hint errors to obtain a good solution. Simultaneous minimization of training error E_0 and hint errors $E_1, \ldots, E_H$ can be achieved by minimization of the hint

objective function:

$$E_{\gamma_1,\ldots,\gamma_H}(g_{\mathbf{v}}) \;=\; E_0(g_{\mathbf{v}}) + \sum_{h=1}^{H} \gamma_h E_h(g_{\mathbf{v}}) \tag{4.5}$$

where $\gamma_1, \ldots, \gamma_H \geq 0$.

When the hint errors E_h are differentiable with respect to model parameters $\mathbf{v}$, $E_{\gamma_1,\ldots,\gamma_H}$ can be minimized by gradient descent along:

$$\partial E_{\gamma_1,\ldots,\gamma_H}(g_{\mathbf{v}}) \;=\; \partial E_0(g_{\mathbf{v}}) + \sum_{h=1}^{H} \gamma_h \partial E_h(g_{\mathbf{v}})$$

where ∂ denotes the derivative with respect to $\mathbf{v}$.

If a hint H_h is "noiseless" the hint parameter γ_h should be chosen as large as possible, but not too large to make the optimization algorithm unable to implement anything other than the hint. For example, in the case of invariance hints, setting the hint parameter too large and minimizing the hint objective function, starting from small initial weights, results in function $g_{\mathbf{v}}(\mathbf{x}) = 0$. Of course this constant function obeys all possible invariances!

Since the goal of learning is the minimization of the out-of-sample error, we would like to obtain hint parameters $\gamma_1, \ldots, \gamma_H$ so that minimization of, or early stopping on, the hint objective function $E_{\gamma_1,\ldots,\gamma_H}(g_{\mathbf{v}})$ results in a small out-of-sample error. As we will see in the next section, in some cases, it is possible to have an estimator of the out-of-sample error in terms of hint errors.

4.2 Estimation of the Out-of-Sample Error Using Invariance Hints

In this section we will demonstrate a method of estimating the generalization and test error in terms of invariance hint errors. In [Cataltepe and Abu-Mostafa, 1993] we had analyzed the same estimation method for binary targets and models with

outputs in $[0:1]$. In this section we will assume that the training input-outputs were generated by a function $f : \mathcal{R}^d \to \mathcal{R}$. The model will be denoted by $g_{\mathbf{v}} : \mathcal{R}^d \to \mathcal{R}$ with adjustable parameters $\mathbf{v}$.

We will approximate the error that the model $g_{\mathbf{v}}$ makes on the target f at input $\mathbf{x}$ by a noise function $n(\mathbf{x})$ (not to be confused with the possible noise on the training outputs):

$$n(\mathbf{x}) = g_{\mathbf{v}}(\mathbf{x}) - f(\mathbf{x}) \tag{4.6}$$

Assume that the noise function n has mean μ and variance σ^2.

$$\begin{aligned} \langle n(\mathbf{x}) \rangle_{\mathbf{x}} &= \mu & (4.7) \\ \langle n^2(\mathbf{x}) \rangle_{\mathbf{x}} - \mu^2 &= \sigma^2 & (4.8) \end{aligned}$$

Then the generalization error of model $g_{\mathbf{v}}$ is:

$$E_{gen}(g_{\mathbf{v}}) = \left\langle (g_{\mathbf{v}}(x) - f(x))^2 \right\rangle_{\mathbf{x}} = \left\langle n^2(\mathbf{x}) \right\rangle_{\mathbf{x}} = \mu^2 + \sigma^2 \tag{4.9}$$

The hint error for the invariance hint H_1 is:

$$\begin{aligned} E_1(g_{\mathbf{v}}) &= \left\langle (g_{\mathbf{v}}(x) - g_{\mathbf{v}}(x'))^2 \right\rangle_{\mathbf{x}} \\ &\quad \text{Insert } -f(x) + f(x') = 0 \\ E_1(g_{\mathbf{v}}) &= \left\langle (n(\mathbf{x}) - n(\mathbf{x}'))^2 \right\rangle_{\mathbf{x}} \\ &= 2 \left\langle n^2(\mathbf{x}) \right\rangle_{\mathbf{x}} - 2 \left\langle n(\mathbf{x}) n(\mathbf{x}') \right\rangle_{\mathbf{x}} \\ &\quad \text{Assume } n(\mathbf{x}) \text{ and } n(\mathbf{x}') \text{ indep.} \\ &= 2(\sigma^2 + \mu^2) - 2\mu^2 \\ &= 2\sigma^2 \end{aligned} \tag{4.10}$$

We will use the training set to estimate the mean μ:

$$\hat{\mu} = \frac{1}{N} \sum_{i=n}^{N} n(\mathbf{x}_n) = \frac{1}{N} \sum_{i=n}^{N} f(\mathbf{x}_n) - g_{\mathbf{v}}(\mathbf{x}_n) \tag{4.11}$$

The training set can also be used to estimate the mean and variance in 4.9:

$$E_{gen}(g_{\mathbf{v}}) = \mu^2 + \sigma^2 \approx E_0(g_{\mathbf{v}}) \tag{4.12}$$

Hence, $\sigma^2 \approx E_0 - \hat{\mu}^2$. Combining this estimator of variance with equation 4.10, and giving each estimator equal weight, we obtain a combined estimator of the variance:

$$\hat{\sigma^2} = \frac{2(E_0 - \hat{\mu}^2) + E_1}{4} \tag{4.13}$$

and finally, an estimate of E_{gen}, using both training and hint errors and $\hat{\mu}$ becomes:

$$\begin{aligned} \hat{E}_{gen} &= \hat{\sigma^2} + \hat{\mu}^2 \\ &= \frac{2(E_0 - \hat{\mu}^2) + E_1}{4} + \hat{\mu}^2 \end{aligned} \tag{4.14}$$

When the test inputs are known, similar to the generalization error estimate, an estimate of the test error can be obtained using the test inputs. For this case, instead of the mean μ and the variance σ^2, sample mean and variance on the test inputs, μ_Y, σ_Y^2 are used. It is also assumed that the sample mean and variance are the same on the test input set, and the transformed test input set according to the invariance. When the hint error is measured on the test inputs, $E_1(g_{\mathbf{v}}) = \frac{1}{M}\sum_{m=1}^{M}\left(g_{\mathbf{v}}(\mathbf{y}_m) - g_{\mathbf{v}}(\mathbf{y}'_m)\right)^2$, we obtain the estimate of the test error:

$$\begin{aligned} \hat{E} &= \hat{\sigma^2} + \hat{\mu}^2 \\ &= \frac{2(E_0 - \hat{\mu}^2) + E_1}{4} + \hat{\mu}^2 \end{aligned} \tag{4.15}$$

The estimator of the test error in equation 4.14 can either be used as an early stopping criterion, or to descend on it. For an example run, we show the training and test errors and the test error estimate [1] in figure 4.1. Three different approximations of the hint error, on training ($\mathbf{X}$), test ($\mathbf{Y}$) and random ($\mathbf{Z}$) inputs are used in the figure. As the overtraining occurs, the estimate using the hint error follows the test

[1] We have taken $\mu = 0$ for these experiments.

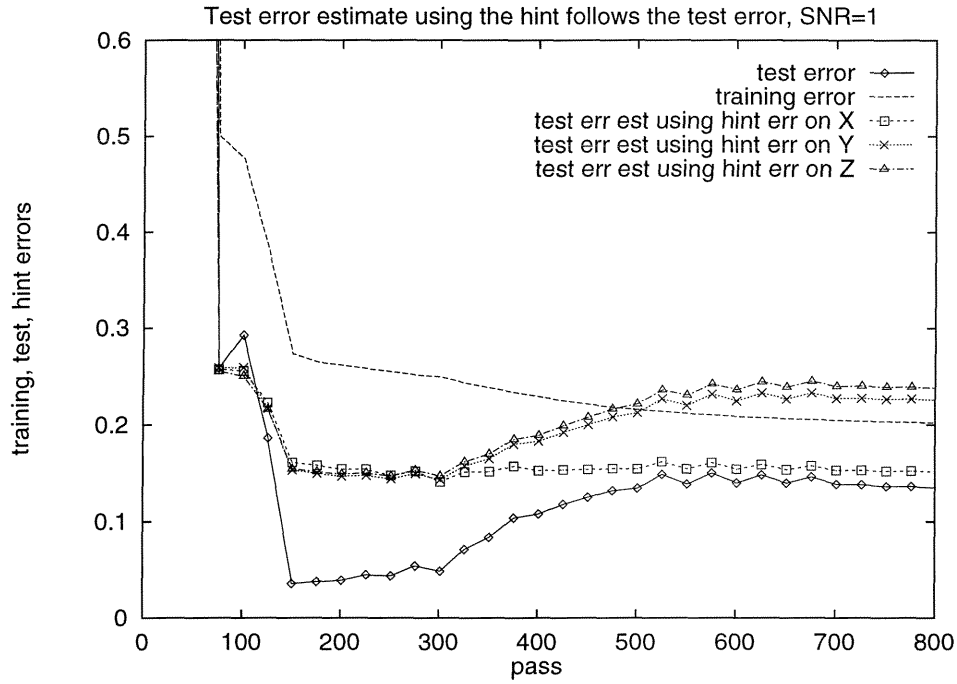

Figure 4.1: When overtraining occurs, the estimate of the test error using the hint error (on training inputs $\mathbf{X}$, or on test inputs $\mathbf{Y}$, or on random inputs $\mathbf{Z}$) follows the test error.

error, whereas the training error keeps decreasing.

In figure 4.2 we show the performance of this estimator as an early stopping criterion. In this experiment [2] the target function was an even function generated by a neural network. The mean (over 100 experiments) test error ratios $\frac{E(\text{early stopping on test error estimate})}{E(\text{minimum training error solution})}$ is shown in the figure. Early stopping on the test error estimate results in smaller test error than the test error of the training

[2]The targets were generated by (teacher) neural networks whose weights were drawn from unit normal. First a neural network with 5 hidden units was generated. Then the function was made even by adding five more hidden units with exactly the same connections, except negative of the input weights of the first five hidden units. The training, test and random inputs were drawn from a zero mean and variance 10 normal. The training outputs were obtained by adding zero mean noise to the teacher outputs on the training inputs. The noise variance was determined according to the specific signal-to-noise (SNR) ratio for the experiment. The test outputs were not noisy. There were $N = 30$ training and $M = 50$ test examples. The number of random inputs was also 50. The student (model) neural network had 10 hidden units, and its weights were drawn from a zero mean 0.001 variance normal. The training method was gradient descent. The learning rate was initially 0.0001, during training, it was multiplied by 1.1 when the training error decreased and halved otherwise. Training continued for 1000 passes.

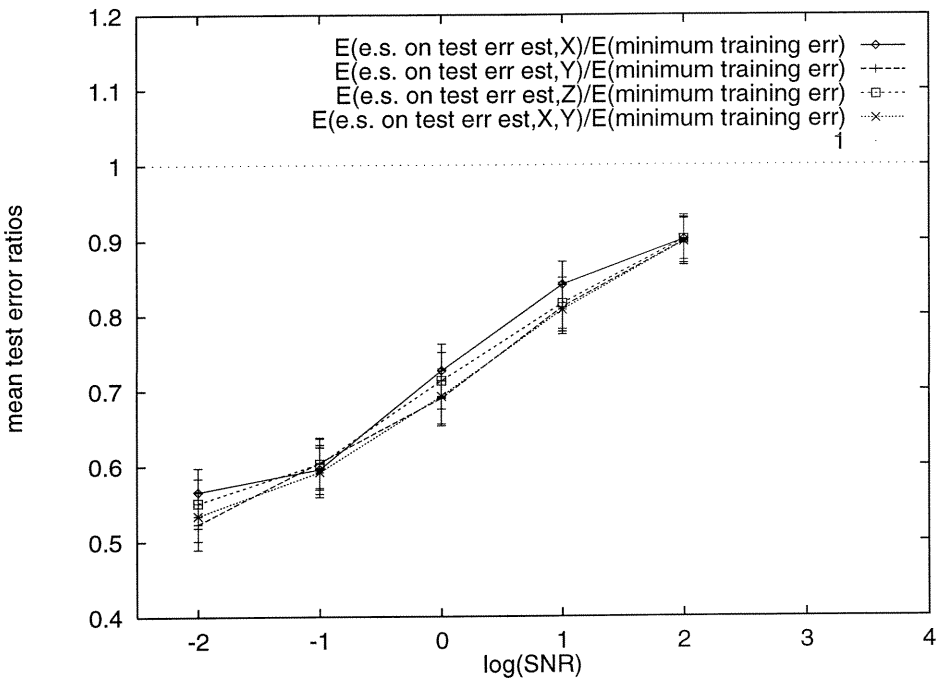

Figure 4.2: Early stopping on the test error estimate using the hint error results in smaller test error than stopping at the minimum training error. When the hint error is estimated using training inputs ($\mathbf{X}$), test inputs ($\mathbf{Y}$), random inputs ($\mathbf{Z}$), or training and test inputs ($\mathbf{X}, \mathbf{Y}$), the same performance increase is obtained.

error minimum. The performance does not differ much among different estimations of the hint error E_1 (on training inputs ($\mathbf{X}$), test inputs ($\mathbf{Y}$), random inputs ($\mathbf{Z}$), training and test inputs ($\mathbf{X}, \mathbf{Y}$)).

We also experimented with descending on the test error estimate $\hat{E}$ directly. When we descended on the estimate directly, most experiments were stuck in a local minima, especially when $SNR \approx 1$. We think the training error and the hint were conflicting and hence it was not possible to settle on descent either one of them at the beginning of training. We also experimented with descending on the training error first, till a certain training error was reached [3] and then descending on the test error estimate. For this experiment, the performance was comparable to figure 4.2.

[3] We determined this training error level by means of early stopping using a validation set runs.

4.3 Gradient Descent on the Hint Objective Function, Estimating the Hint Parameters Using Cross Validation

In section 3.6 we descended on the augmented error, using the gradient of the augmented error and determining the augmentation parameters by means of cross validation method. Similarly, in this section we will descend on the hint objective function $E_{\gamma_1,\ldots,\gamma_H}$, determining the hint parameters $\gamma_1,\ldots,\gamma_H$ by means of cross validation at each pass. Since leave-$\frac{N}{10}$-out cross validation with shuffling of the training set at each pass resulted in better performance than ordinary (leave-1-out) cross validation for augmented error, we will use leave-$\frac{N}{10}$-out cross validation in this section.

Let the training set $(\mathbf{X},\mathbf{f})$ be partitioned into 10 disjoint parts: $(\mathbf{X}_1,\mathbf{f}_1),\ldots,(\mathbf{X}_{10},\mathbf{f}_{10})$. Denote the (training) error of $g_{\mathbf{v}}$ on all 9 parts except the ith one by $E_{t,i}(g_{\mathbf{v}})$, denote the (validation) error on the ith part by $E_{v,i}(g_{\mathbf{v}})$. Denote the hint error using all training inputs except the ones in part i by $E_{h,i}(g_{\mathbf{v}}) = \frac{1}{N-|\mathbf{X}_i|}\sum_{n=1,\mathbf{x}_n\notin\mathbf{X}_i}^{N} e_h(g_{\mathbf{v}},\mathbf{x}_n)$. We need to find the hint parameters $\gamma_1,\ldots,\gamma_H$ such that:

$$\begin{aligned}
\partial E_{v,1}(g_{\mathbf{v}}) &= \partial E_{t,1}(g_{\mathbf{v}}) + \gamma_1\partial E_{1,1}(g_{\mathbf{v}}) + \gamma_H\partial E_{H,1}(g_{\mathbf{v}}) \\
\vdots &= \vdots \\
\partial E_{v,10}(g_{\mathbf{v}}) &= \partial E_{t,10}(g_{\mathbf{v}}) + \gamma_1\partial E_{1,10}(g_{\mathbf{v}}) + \gamma_H\partial E_{H,10}(g_{\mathbf{v}})
\end{aligned}$$

We obtain $\gamma_1,\ldots,\gamma_H$ by simultaneously solving all $10dim(\mathbf{v})$ equations. We shuffle the training set and obtain different partitionings $(\mathbf{X}_1,\mathbf{f}_1),\ldots,(\mathbf{X}_{10},\mathbf{f}_{10})$ each time we use cross validation to find $\gamma_1,\ldots,\gamma_H$.

In figure 4.3 [4] we show the mean (over 100 experiments) test error ratios

[4] Olympic score is obtained as follows: The test error ratios are sorted, and the largest 10% and the smallest 10% are not considered. This eliminates the effect of outliers. Note also that for all the plots, only when $\frac{E_0(\text{gradient descent on hint objective function})}{E_0(\text{gradient descent on training error})} < 2$, the test error ratio is taken

$\frac{E(\text{gradient descent on hint objective function})}{E(\text{gradient descent on training error})}$. For these experiments we determined the hint error on the training inputs, $\mathbf{X}$. As it can be seen from the figure, gradient descent on the hint objective function, determining the hint parameters by means of leave 10 out cross validation at each descent step, results in smaller test error than gradient descent on the training error only. Comparison of figure 4.2 and figure 4.3 shows that the estimation of the test error by means of the hint error, and then early stopping on that estimate results in smaller test error than the method of this section. When the hint error estimate is available, it should be used. However, when it is not available, gradient descent and cross validation to find the hint parameters is a better method than gradient descent on the training error only. Furthermore, enforcing the hint on the additional test inputs resulted in larger test error. This could be due to the fact that we are enforcing the hint from the very beginning of training, and the hint on the test inputs take effect before the training data is fitted well enough.

4.4 Conclusions

In this chapter, we reviewed the learning-from-hints. We demonstrated a method of estimation of the test error using invariance hint errors and then early stopping on this estimate of the test error. We also extended the gradient descent combined with cross validation to estimate the augmentation parameters from the previous chapter, to learning-from-hints. This descent method is a general enough algorithm that can be used not only for invariance hints, but other types of hints as well.

into the average. When training with the hint, training may be stuck at a local minima, we try to avoid counting the local minimum effects by this.

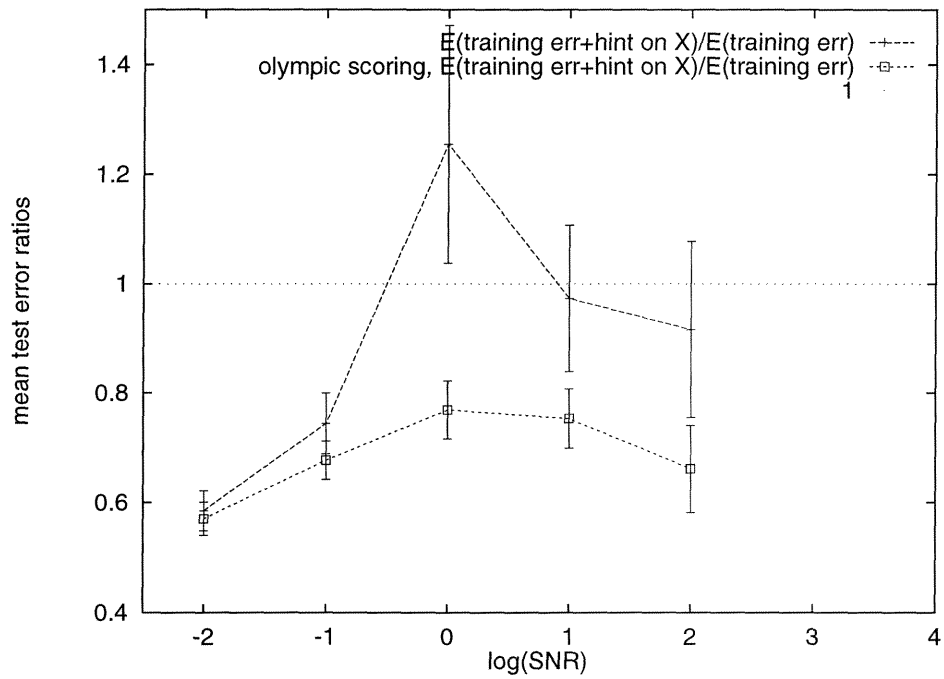

Figure 4.3: Gradient descent on the hint objective function, determining the hint parameters by means of leave-$\frac{N}{10}$-out cross validation, usually, results in smaller test error than gradient descent on the training error only.

Chapter 5
No Free Lunch for Early Stopping

Early stopping of training is one of the methods that aim to prevent overtraining due to too powerful model class, noisy training examples or small training set. In this chapter, we study early stopping at a predetermined training error level. If there is no prior information, other than the training examples, all models with the same training error should be equally likely to be chosen as the early stopping solution. When this is the case, we show that, for general linear models, early stopping at any training error level above the training error minimum increases the expected generalization error. Moreover, we also show that the generalization error is an increasing function of the training error. Our results are nonasymptotic and independent of the training data noise, and they hold when instead of generalization error, iid test error or off training set error [1] are used as the performance criterion. For general nonlinear models, around a small enough neighborhood of a training error minimum, the mean generalization error again increases, when all models with the same training error are equally likely. For classification problems and the bin model [Abu-Mostafa and Song, 1996], the expected generalization error increases regardless of the probability of selection of models. Regularization methods such as weight decay and early stopping using a validation set, or early stopping of training using a hint error are equivalent to early stopping at a fixed training error level but with a nonuniform probability of selection over models with the same training error. If this nonuniform probability agrees with the target function, early stopping may help. One should be aware of what nonuniform probability of selection is implied by an early stopping procedure.

Early stopping has been studied by Wang et. al. [Wang et al., 1994] who

[1] Off training set error does not assume that the training and test inputs come from the same distribution.

analyzed the average optimal stopping time for general linear models (one hidden layer neural networks with a linear output and fixed input weights) and introduced and examined the effective size of the learning machine as training proceeds. Sjoberg and Ljung [Sjoberg and Ljung, 1995] linked early stopping using a validation set to regularization, and showed that emphasizing the validation set too much may result in an unregularized solution. Amari et. al. [Amari et al., 1997] determined the best validation set size in the asymptotic limit and showed that early stopping helps little in this limit even when the best stopping point is known. Dodier [Dodier, 1996] and Baldi and Chauvin [Baldi and Chauvin, 1991] investigated the behavior of validation error curves for linear problems, and the linear auto-association problem respectively. Our results in this section will also appear in [Cataltepe et al., 1998].

We borrow the term "no free lunch" from [Wolpert, 1996b, Wolpert, 1996a]. Wolpert shows that when the prior distribution over the target functions is uniform, and the off training set error is taken to be the performance criterion, there is no difference between learning algorithms. In other words, if a learning algorithm results in good off training set error for one target function, it results in equally worse off training set error for another target function. Like [Zhu and Rohwer, 1996, Goutte, 1997] who put no free lunch theorems into the framework of cross validation, our work puts the no free lunch into the framework of early stopping.

Our method of early stopping, choosing a model uniformly among the models with the same training error, is similar to the Gibbs algorithm [Wolpert, 1995]. Although the uniform probability of selection around the training error minimum is equivalent to the isotropic distributions of [Amari et al., 1997], they assume certain noise (zero mean normal) characteristics, a training minimum close to the generalization error minimum and large number of training examples.

Notation in this chapter is similar to the notation of chapter 2. We are given a fixed training set $\{(\mathbf{x}_1, f_1), \ldots, (\mathbf{x}_N, f_N)\}$ with inputs $\mathbf{x}_n \in \mathcal{R}^{d'}$ and outputs $f_n \in \mathcal{R}$. The model to fit the training data will be denoted by $g_{\mathbf{v}}(\mathbf{x})$, with adjustable parameters $\mathbf{v}$. We will refer to models by their adjustable parameters $\mathbf{v}$, unless indicated otherwise. We assume that the training outputs were generated from the

training inputs according to some unknown and fixed distribution $f(\mathbf{x}_n)$, hence $f_n = f(\mathbf{x}_n)$. For example, if the outputs were generated by a teacher model with parameters $\mathbf{v}^*$ and additive zero mean normal noise, we would have $f(\mathbf{x}_n) = g_{\mathbf{v}^*}(\mathbf{x}_n) + e_n$ where $e_n \sim \mathcal{N}(0, \sigma_e^2)$ for $\sigma_e^2 \geq 0$.

We define the quadratic training error E_0 and the generalization error E at $\mathbf{v}$ as:

$$E_0(\mathbf{v}) = \frac{1}{N}\sum_{n=1}^{N}(g_{\mathbf{v}}(\mathbf{x}_n) - f_n)^2 \qquad E(\mathbf{v}) = \left\langle (g_{\mathbf{v}}(\mathbf{x}) - f(\mathbf{x}))^2 \right\rangle_{\mathbf{x}}$$

Let $\mathbf{v}_0$ be a local minimum of the training error E_0. Let $\delta \geq 0$ and $E_\delta = E_0(\mathbf{v}_0) + \delta$. Let $\mathbf{W}_\delta = \{\Delta\mathbf{v} : E_0(\mathbf{v}_0 + \Delta\mathbf{v}) = E_\delta\}$. The set of models $\mathbf{v}_0 + \mathbf{W}_\delta$ form the **early stopping set**. We define **early stopping at training error** E_δ as choosing a model from the early stopping set according to a probability distribution on the models in the early stopping set. We denote the probability of selecting $\mathbf{v}_0 + \Delta\mathbf{v}$ as the early stopping solution by $P_{\mathbf{W}_\delta}(\Delta\mathbf{v})$. This probability is zero if $\Delta\mathbf{v} \notin \mathbf{W}_\delta$. The **mean generalization error at training error level** E_δ is:

$$E_{mean}(E_\delta) = \int_{\Delta\mathbf{v}\in\mathbf{W}_\delta} P_{\mathbf{W}_\delta}(\Delta\mathbf{v})E(\mathbf{v}_0 + \Delta\mathbf{v})d\Delta\mathbf{v}$$

$P_{\mathbf{W}_\delta}$ is said to be **uniform** if $\forall\Delta\mathbf{v}, \Delta\mathbf{v}' \in \mathbf{W}_\delta, \quad P_{\mathbf{W}_\delta}(\Delta\mathbf{v}) = P_{\mathbf{W}_\delta}(\Delta\mathbf{v}')$, i.e. if models with the same training error are equally likely to be chosen as the early stopping solution. (See figure 5.1)

The rest of the chapter is organized as follows: In section 5.1, we prove that early stopping can not decrease the mean generalization error for general linear models when all models with the same training error are equally likely to be the target. Section 5.2 proves the same result for nonlinear models but around a training error minimum only. In section 5.3 we review the bin model and prove that early stopping can not help for this model either. In all these cases, we assume that there is no prior information about the target that generated the training data. In section 5.4 we experimentally verify the early stopping results for general linear and neural network models. We also show that early stopping can help when additional information is

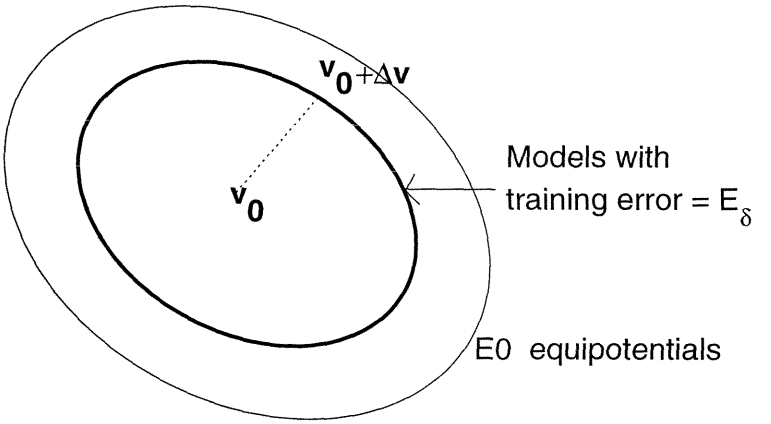

Figure 5.1: The models with training error $E_\delta = E_0(\mathbf{v}_0) + \delta$ form the early stopping set at training error level E_δ.

available, for example in the case of weight decay or invariance hints. Section 5.5 summarizes the chapter.

5.1 Early Stopping for a General Linear Model

In this section we will use the general linear models as described in section 2.2 and figure 2.2. Since the transformation functions $\phi_i(.)$ are fixed and only the output weights are adjustable, we will denote a general linear model only by its output weights $\mathbf{w}$. Again, $\Phi_{x(d+1)\times N} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]$ will denote the training inputs transformed by the fixed transformation functions $\phi_i(\mathbf{x}) : \mathcal{R}^{d'} \rightarrow \mathcal{R}$, $i = 0, \ldots, d$, and $\mathbf{f}_{N\times 1} = [f_1, \ldots, f_N]^T$ is the training outputs. Define $\mathbf{S}_x = \frac{\mathbf{\Phi}_x \mathbf{\Phi}_x^T}{N}$ and $\Sigma_{\phi(x)} = \left\langle \phi(\mathbf{x})\phi(\mathbf{x})^T \right\rangle_{\mathbf{x}}$. When $\Phi_x\Phi_x^T$ is full rank [2], the unique training error minimum is given by (see section 2.8.1):

$$\mathbf{w}_0 = (\Phi_x\Phi_x^T)^{-1}\Phi_x\mathbf{f} = \mathbf{S}_x^{-1}\frac{\Phi_x\mathbf{f}}{N}$$

The Hessians of training and generalization errors are constant positive

[2]Hence we restrict ourselves to problems where $N \geq d + 1$. Since the transformation functions are real valued, for most cases $\mathbf{\Phi}_x\mathbf{\Phi}_x^T$ is likely to be full rank.

semi-definite[3] matrices at all $\mathbf{w}$:

$$HE_0(\mathbf{w}) = 2\mathbf{S}_x \qquad HE(\mathbf{w}) = 2\mathbf{\Sigma}_{\phi(x)}$$

Any higher derivatives of E and E_0 are zero everywhere. Hence, for any $\mathbf{\Delta w}$, the generalization and training errors of $\mathbf{w}_0 \pm \mathbf{\Delta w}$ can be written as:

$$E(\mathbf{w}_0 \pm \mathbf{\Delta w}) = E(\mathbf{w}_0) \pm \mathbf{\Delta w}^T \nabla E(\mathbf{w}_0) + \mathbf{\Delta w}^T \mathbf{\Sigma}_{\phi(x)} \mathbf{\Delta w} \tag{5.1}$$

$$E_0(\mathbf{w}_0 \pm \mathbf{\Delta w}) = E_0(\mathbf{w}_0) + \mathbf{\Delta w}^T \mathbf{S}_x \mathbf{\Delta w} \tag{5.2}$$

The following lemma proves that when all models with the training error $E_0(\mathbf{w}_0)+\delta, \delta \geq 0$ are equally likely to be chosen as the solution, the mean generalization error at training error level $E_0(\mathbf{w}_0) + \delta$ can not be smaller than the generalization error of $\mathbf{w}_0$.

Lemma 5.1.1 *When all models with training error $E_\delta = E_0(\mathbf{w}_0) + \delta \geq E_0(\mathbf{w}_0)$ are equally likely to be chosen as the early stopping solution, the mean generalization error at training error level $E_0(\mathbf{w}_0)+\delta$ is at least as much as the generalization error of the training error minimum. More specifically, for any $\delta \geq 0$, $E_{mean}(E_\delta) = E(\mathbf{w}_0)+\beta(\delta)$, for some $\beta(\delta) \geq 0$.*

Proof: is given in section 5.6.1. Please see figure 5.2 for an illustration of the lemma.

Note that this result does not depend on the noise level, number of training examples or the target function versus model complexity. Even if the target function is a constant and the model is a 100th degree polynomial, lemma 5.1.1 tells us that we should stop only at the training error minimum.

If the error criterion is the test error on iid or non-iid inputs $\{\mathbf{y}_1, \ldots, \mathbf{y}_M\}$, the lemma still holds. Because $\mathbf{S}_y = \frac{\Phi_y \Phi_y^T}{M}$ is positive semi-definite.

[3] Any matrix of the form AA^T is positive semi-definite, because for any $\mathbf{w}$ of proper dimensions, $\mathbf{w}^T AA^T \mathbf{w} = ||A^T \mathbf{w}||^2 \geq 0$, hence $\mathbf{S}_x = \frac{\Phi_x \Phi_x^T}{N}$ is positive semi-definite. $\mathbf{\Sigma}_{\phi(x)} = \left\langle \phi(\mathbf{x})\phi(\mathbf{x})^T \right\rangle_{\mathbf{x}}$ is also positive semi-definite since $\frac{\Phi_x \Phi_x^T}{N} \longrightarrow_{N\to\infty} \left\langle \phi(\mathbf{x})\phi(\mathbf{x})^T \right\rangle_{\mathbf{x}}$

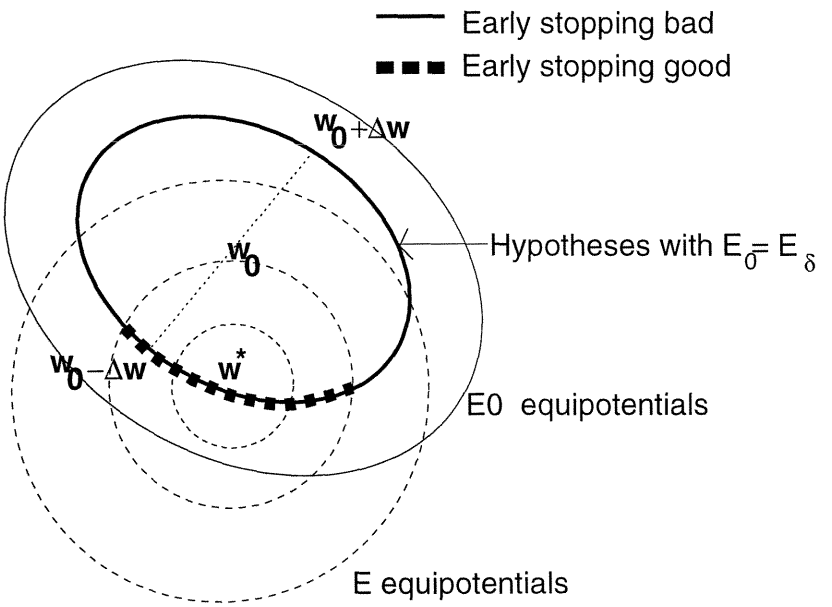

Figure 5.2: Early stopping at a training error δ above $E_0(\mathbf{w}_0)$ results in higher generalization error when all models having the same training error are equally likely to be chosen as the early stopping solution.

Furthermore, lemma 5.1.1 holds not only for quadratic loss, but for any loss function which has a positive semi-definite test error Hessian and zero third and higher derivatives at the training error minimum.

The following theorem compares the mean generalization error between any two training error levels:

Theorem 5.1.1 *When all models with the same training error are equally likely to be chosen as the early stopping solution, the mean generalization error is an increasing function of the early stopping training error. In other words, for* $0 < \delta_1 < \delta_2$, $E_{mean}(E_{\delta_1}) < E_{mean}(E_{\delta_2})$.

Proof: is given in section 5.6.2.

Therefore, when the model is general linear, the best strategy is to minimize the training error as much as possible.

5.2 Early Stopping for a Nonlinear Model

When the model is general linear we are able to prove lemma 5.6.1 without any assumptions about the location of the generalization error minimum with respect to the training error minimum. Also, our results were valid for all models with the same training error, regardless of how far they are from the training error minimum. For the nonlinear model we will assume that the distance between the training error minimum and the generalization error minimum is $\mathcal{O}\left(\frac{1}{N}\right)$, which is asymptotically the case if the output noise is additive zero mean normal, see for example [Amari et al., 1997]. Also we will prove the increase in the mean generalization error only around the training error minimum.

Let the model $g_{\mathbf{v}}$ be a nonlinear (continuous and differentiable) model with adjustable parameters $\mathbf{v}$. Let $\mathbf{v}_0$ be a minimum of the training error, and let $\nabla E(\mathbf{v}_0), \nabla E_0(\mathbf{v}_0), HE(\mathbf{v}_0), HE_0(\mathbf{v}_0)$ denote the gradient and Hessians of the generalization error and the training error at $\mathbf{v}_0$. Let $\mathbf{v}^*$ be a minimum of the generalization error. Let $\mathbf{\Delta v}$ be such that $E_0(\mathbf{v}_0 + \mathbf{\Delta v}) = E_0(\mathbf{v}_0) + \delta$, for $\delta \geq 0$.

Now we assert the counterpart of lemma 5.6.1 for the nonlinear models:

Theorem 5.2.1 *Let $\delta \geq 0$ and let $E_\delta = E_0(\mathbf{v}_0) + \delta$. Let $\mathbf{\Delta v} = \mathcal{O}\left(\frac{1}{N^{0.5}}\right)$, and $E_0(\mathbf{v}_0 + \mathbf{\Delta v}) = E_\delta + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)$. Let $\mathbf{v}_0 - \mathbf{v}^* = \mathcal{O}\left(\frac{1}{N^{0.5}}\right)$. When all models with training error $E_\delta + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)$ are equally likely to be chosen as the early stopping solution, the mean generalization error at training error level $E_0(\mathbf{v}_0) + \delta + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)$ is $E_{mean}(E_\delta) = E(\mathbf{v}_0) + \beta(\delta) + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)$, for some $\beta(\delta) \geq 0$.*

Proof: is given in section 5.6.3.

5.3 Early Stopping for Classification Problems and the Bin Model

For classification problems, bin model [Abu-Mostafa and Song, 1996] can be utilized to prove that mean generalization error increases as the training error increases. Since

the proof does not have any assumptions about the probability distribution on the models with the same training error, it is worth mentioning here.

We will use the following version of the bin model: Consider $M < \infty$ learning models [4] with generalization errors $\pi_1, \ldots, \pi_M$. Determine the training errors of models $\nu_1, \ldots, \nu_M$ by picking N i.i.d. inputs and finding the errors on these samples for each bin. π_m corresponds to the generalization error E of a model m, and ν_m corresponds to the training error E_0. (please see figure 5.3).

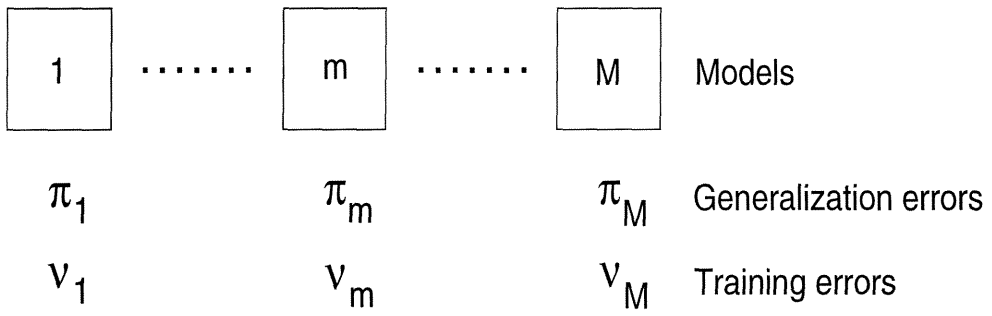

Figure 5.3: The bin model.

Let $Pr[.]$ denote the probability of the occurrence of an event. The mean generalization error for training error level ν is:

$$E_{mean}(\nu) = \mathcal{E}[\pi|\nu] = \sum_{m=1}^{M} \pi_m Pr[\pi_m|\nu_m = \nu] \tag{5.3}$$

The following theorem is a generalization of theorem 5.1.1 for the bin model:

Theorem 5.3.1 *For classification problems and models that can be formalized using the bin model, the mean generalization error is an increasing function of the training error.*

Proof: is given in section 5.6.4.

[4] Each learning model corresponds to a unique function. For example, each model could be the function implemented by a neural network for a specific setting of the weights of the neural network.

5.4 Experimental Verification of Results

We experimented with linear and nonlinear models to verify the theorems and lemma for these models. We also investigated the weight decay solution for the linear model and effect of evenness hint on the mean generalization error.

5.4.1 Linear Model

We computed the minimum training error (least squares) solution $\mathbf{w}_0$, then we computed the average generalization error of solutions $\mathbf{w}$ with training error $E_0(\mathbf{w}_0)+\delta$. For comparison, we also computed the generalization error of the weight decay solution with training error $E_0(\mathbf{w}_0)+\delta$. In figure [5] 5.4 we show the behavior of the mean generalization error as the training error increases. When all models with the same training error are chosen with the same probability, in agreement with lemma 5.1.1, the mean generalization error increases as the training error increases. On the other hand, the weight decay solution has smaller generalization error for small enough weight decay parameter. Note that choosing the weight decay solution with probability 1 corresponds to a nonuniform (delta function) probability distribution on models with the same training error, therefore lemma 5.1.1 does not apply. Note also that, for this experiment both the target and the model are linear and the training points have zero mean normal noise, therefore, the weight decay provably results in better generalization error when the weight decay parameter is small enough [Bishop, 1995].

[5] For this experiment, both the target and the model were linear. Input dimensionality was $d = 5$, plus constant bias 1. Training inputs were chosen from a zero mean unit normal. There were $N = 20$ training input-outputs. The target (teacher) model was also linear with weights chosen from zero mean 9 variance normal. Zero mean normal noise was added to the training outputs. Noise variance was determined according to 0.1 signal-to-noise ratio. The mean generalization error for the uniform P was computed on 500 different models with the same training error. The generalization error was computed as the squared distance between the target and the model.

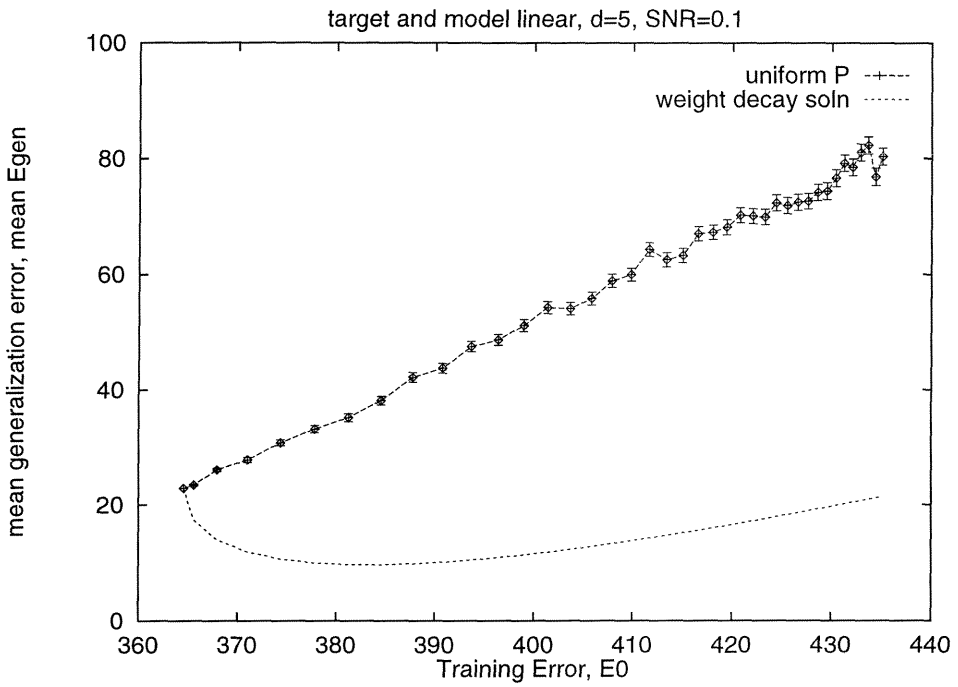

Figure 5.4: The mean generalization/test error versus training error of a linear model for a given target and training set. The mean generalization error increases as the training error increases when all models with the same training error are given equal probability of selection. When the weight decay parameter is small enough, choosing the weight decay solution with probability 1 and all other models with the same training error with probability 0 improves the generalization error.

5.4.2 Nonlinear Model

We experimented with a neural network model, and a noisy and even target function, also generated by a (teacher) neural network model. We first found a training error minimum using the gradient descent with adaptive learning rate. Then we chose random weights $\mathbf{\Delta v}$ [6] such that $E_0(\mathbf{v}_0 + \mathbf{\Delta v}) \approx E_0(\mathbf{v}_0) + \delta$. In figure 5.5 we show the mean test error versus the training error for a specific target, training set and model [7] $g_{\mathbf{v}_0}$. When the mean test error for a certain training error level is computed

[6] Since the gradient at the minimum $\mathbf{v}_0$ is very small but not exactly zero, we scaled $\mathbf{\Delta v}$ as $k\mathbf{\Delta v}$ where k is the best possible solution for $k\mathbf{\Delta v}^T \nabla E_0(\mathbf{v}_0) + k^2 \frac{1}{2} \mathbf{\Delta v}^T HE0(\mathbf{v}_0) \mathbf{\Delta v} = \delta$. Hence $k = \frac{-b \pm \sqrt{b^2 + 4a\delta}}{2a}$ where $a = \frac{1}{2} \mathbf{\Delta v}^T HE0(\mathbf{v}_0) \mathbf{\Delta v}$ and $b = \mathbf{\Delta v}^T \nabla E_0(\mathbf{v}_0)$.

[7] The training outputs were generated by (teacher) neural network whose weights were drawn from unit normal. First a neural network with 5 hidden units was generated. Then the function was made even by adding five more hidden units with exactly the same connections, except negative

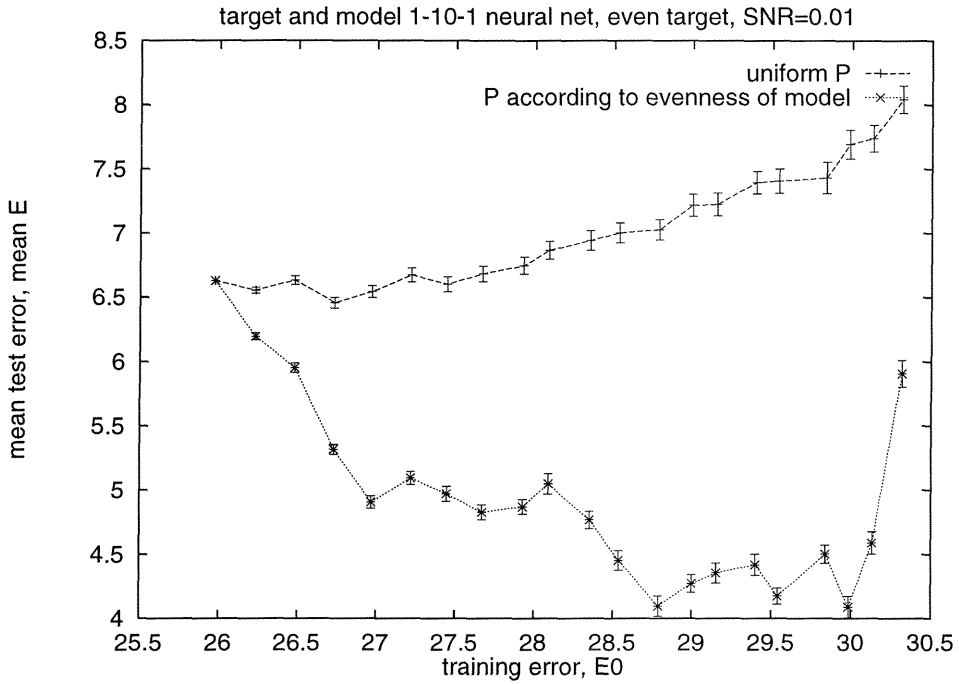

Figure 5.5: The mean test error versus training error of a nonlinear model for a given even target and the training set. The mean test error increases as the training error increases when all models with the same training error are given equal probability of selection. Choosing the models with the smaller evenness error with higher probability reduces the mean test error.

by giving each model with the same training error equal probability, the mean test error increases. On the other hand, when the models with smaller evenness hint error $E_1(\mathbf{v}_0 + \mathbf{\Delta v})$ are given more weight, the mean test error seems to decrease for sometime and then increase. In other words, early stopping, choosing models with smaller hint errors with higher probability can decrease the mean test error.

Note that, as shown in figure 5.6, the decrease in the mean test error using

of the input weights of the first five hidden units. The training and test inputs were drawn from a zero mean and variance 10 normal. The training outputs were obtained by adding zero mean noise to the teacher outputs on the training inputs. The noise variance was determined according to the signal-to-noise ratio. The test outputs were not noisy. There were $N = 30$ training and $M = 50$ test examples. The student (model) neural network had 10 hidden units, and its weights were drawn from a zero mean 0.001 variance normal. The training method was gradient descent. The learning rate was initially 0.0001, during training, it was multiplied by 1.1 when the training error decreased and halved otherwise. Training continued for 1000 passes and the model with the smallest training error was taken to be $g_{\mathbf{v}_0}$. When computing the mean test error using the evenness hint, we weighed the model $g_{\mathbf{v}_0+\mathbf{\Delta v}^i}$ according to: $\frac{\exp -E_1(\mathbf{v}_0+\mathbf{\Delta v}^i)}{\sum_{i=1}^{1000} \exp -E_1(\mathbf{v}_0+\mathbf{\Delta v}^i)}$ for $i = 1, \ldots, 1000$.

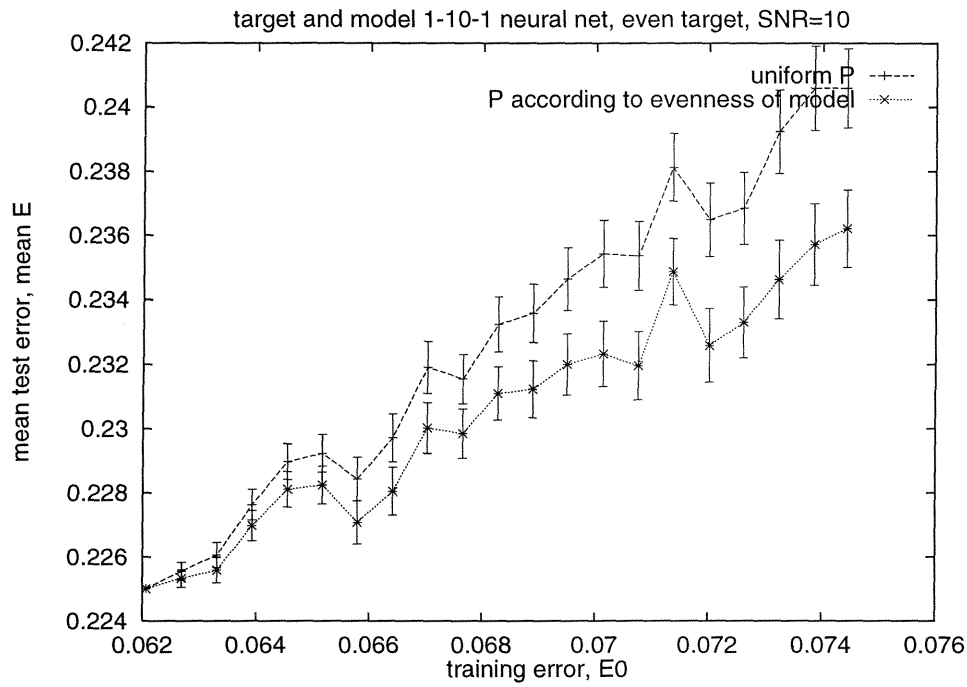

Figure 5.6: When the signal-to-noise ratio is high and the target is even, the mean test error around the training error minimum may increase, even if the models with the same training error are weighed according to their hint error.

the hint is dependent on not only the number of training examples N, but also the signal-to-noise ratio. For the same N, but now for $SNR = 10$, selecting the models according to the evenness hint error, in the same way we did for the previous experiment that had $SNR = 0.01$, does not decrease the mean test error. It is possible that the probability of selection of a model should depend not only on the hint error E_1, but also the level of training error and the signal-to-noise ratio.

5.5 Conclusions

In this chapter we analyzed early stopping at a certain training error minimum, and showed that one should minimize the training error as much as possible when all the information available about the target is the training set. We demonstrated that using the additional information about the target to choose a model with higher training

error can improve the generalization error.

5.6 Appendix

5.6.1 Proof of Lemma 5.1.1:

Let the early stopping training error level be $E_\delta = E_0(\mathbf{w}_0) + \delta$ for some $\delta \geq 0$. Then, from equation (5.2), the early stopping set consists of $\mathbf{w}_0 + \mathbf{W}_\delta = \mathbf{w}_0 + \{\boldsymbol{\Delta w} : \boldsymbol{\Delta w}^T \mathbf{S}_x \boldsymbol{\Delta w} = \delta\}$. The mean generalization error is:

$$E_{mean}(E_\delta) = \int_{\boldsymbol{\Delta w} \in \mathbf{W}_\delta} P_{\mathbf{W}_\delta}(\boldsymbol{\Delta w}) E(\mathbf{w}_0 + \boldsymbol{\Delta w}) d\boldsymbol{\Delta w}$$

For any $\boldsymbol{\Delta w} \in \mathbf{W}_\delta$, hence satisfying $\boldsymbol{\Delta w}^T \mathbf{S}_x \boldsymbol{\Delta w} = \delta$, there exists a $-\boldsymbol{\Delta w} \in \mathbf{W}_\delta$, therefore we can rewrite the mean generalization error as:

$$E_{mean}(E_\delta) =$$
$$0.5 \int_{\boldsymbol{\Delta w} \in \mathbf{W}_\delta} \left(P_{\mathbf{W}_\delta}(\boldsymbol{\Delta w}) E(\mathbf{w}_0 + \boldsymbol{\Delta w}) + P_{\mathbf{W}_\delta}(-\boldsymbol{\Delta w}) E(\mathbf{w}_0 - \boldsymbol{\Delta w}) \right) d\boldsymbol{\Delta w}$$

Now, since $P_{\mathbf{W}_\delta}$ is uniform, it is also symmetric, i.e. $P_{\mathbf{W}_\delta}(\boldsymbol{\Delta w}) = P_{\mathbf{W}_\delta}(-\boldsymbol{\Delta w})$. For the proof of this theorem symmetry is the only restriction we need on $P_{\mathbf{W}_\delta}$. Using symmetry of $P_{\mathbf{W}_\delta}$, equation (5.1), and the fact that $\int_{\boldsymbol{\Delta w} \in \mathbf{W}_\delta} P_{\mathbf{W}_\delta}(\boldsymbol{\Delta w}) d\boldsymbol{\Delta w} = 1$:

$$\begin{aligned} E_{mean}(E_\delta) &= E(\mathbf{w}_0) + \int_{\boldsymbol{\Delta w} \in \mathbf{W}_\delta} P_{\mathbf{W}_\delta}(\boldsymbol{\Delta w}) \boldsymbol{\Delta w}^T \boldsymbol{\Sigma}_{\phi(x)} \boldsymbol{\Delta w} d\boldsymbol{\Delta w} \\ &= E(\mathbf{w}_0) + \beta(\delta) \end{aligned}$$

Since $\boldsymbol{\Sigma}_{\phi(x)} = \left\langle \phi(\mathbf{x}) \phi(\mathbf{x})^T \right\rangle_{\mathbf{x}}$ is positive semi-definite and $P_{\mathbf{W}_\delta}(\boldsymbol{\Delta w}) \geq 0$,

$$\beta(\delta) = \int_{\boldsymbol{\Delta w} \in \mathbf{W}_\delta} P_{\mathbf{W}_\delta}(\boldsymbol{\Delta w}) \boldsymbol{\Delta w}^T \boldsymbol{\Sigma}_{\phi(x)} \boldsymbol{\Delta w} d\boldsymbol{\Delta w} \geq 0 \tag{5.4}$$

□

5.6.2 Proof of Theorem 5.1.1:

By lemma 5.1.1, $E_{mean}(E_{\delta_1}) = E(\mathbf{w}_0) + \beta(\delta_1)$ and $E_{mean}(E_{\delta_2}) = E(\mathbf{w}_0) + \beta(\delta_2)$ for $\beta(\delta_1), \beta(\delta_2) > 0$. Let $0 < \delta_1 < \delta_2$. We need to prove $\beta(\delta_1) < \beta(\delta_2)$.

Let $V(\delta) = \int\limits_{\mathbf{\Delta w} \in \mathbf{W}_\delta} \mathbf{\Delta w}^T \mathbf{\Sigma}_{\phi(x)} \mathbf{\Delta w} d\mathbf{\Delta w}$, and let $\frac{1}{P_\delta}$ be the surface area of the d dimensional ellipsoid $\mathbf{\Delta w}^T \mathbf{S}_x \mathbf{\Delta w} = \delta$. Since $P_{\mathbf{W}_\delta}$ is uniform, from equation 5.4:

$$\frac{\beta(\delta_2)}{\beta(\delta_1)} = \frac{P_{\delta_2}}{P_{\delta_1}} \frac{V(\delta_2)}{V(\delta_1)}$$

Define $k^2 = \frac{\delta_2}{\delta_1} > 1$. Let $\mathbf{W}_{\delta_1} = \{\mathbf{\Delta w} : \mathbf{\Delta w}^T \mathbf{S}_x \mathbf{\Delta w} = \delta_1\}$. Then $\mathbf{W}_{\delta_2} = \{k\mathbf{\Delta w} : \mathbf{\Delta w} \in \mathbf{W}_{\delta_1}\}$. By means of change of variables $\mathbf{\Delta u} = k\mathbf{\Delta w}$ in $V(\delta_2)$ we have $\frac{V(\delta_2)}{V(\delta_1)} = k^{d+1}$.

We can define the surface area as the derivative of the volume:

$$\begin{aligned}
\frac{1}{P_\delta} &= \lim_{l \to 0} \frac{\int\limits_{\mathbf{\Delta w}^T \mathbf{S}_x \mathbf{\Delta w} \leq \delta + l} d\mathbf{\Delta w} - \int\limits_{\mathbf{\Delta w}^T \mathbf{S}_x \mathbf{\Delta w} \leq \delta} d\mathbf{\Delta w}}{l} \\
&= \lim_{l \to 0} \frac{\left(\frac{\delta + l}{\delta}\right)^{\frac{h+1}{2}} - 1}{l} \int\limits_{\mathbf{\Delta w}^T \mathbf{S}_x \mathbf{\Delta w} \leq \delta} d\mathbf{\Delta w} \\
&= \frac{h+1}{2\delta} \int\limits_{\mathbf{\Delta w}^T \mathbf{S}_x \mathbf{\Delta w} \leq \delta} d\mathbf{\Delta w}
\end{aligned}$$

Hence $\frac{1}{P_{\delta_1}} = \frac{h+1}{2\delta_1} \int\limits_{\mathbf{\Delta w}^T \mathbf{S}_x \mathbf{\Delta w} \leq \delta_1} d\mathbf{\Delta w}$. By means of change of variables $\mathbf{\Delta u} = \frac{\mathbf{\Delta w}}{k}$ we have $\frac{1}{P_{\delta_2}} = k^{d-1} \frac{1}{P_{\delta_1}}$. Therefore, $\frac{P_{\delta_2}}{P_{\delta_1}} = k^{-d+1}$.

Hence, $\frac{\beta(\delta_2)}{\beta(\delta_1)} = k^{-d+1} k^{d+1} = k^2 > 1$. □

5.6.3 Proof of Theorem 5.2.1:

Similar to equations 5.1 and 5.2, the training and generalization errors at $\mathbf{v}_0 + \mathbf{\Delta v}$ are:

$$E(\mathbf{v}_0 \pm \mathbf{\Delta v}) = E(\mathbf{v}_0) \pm \mathbf{\Delta v}^T \nabla E(\mathbf{v}_0) + \frac{1}{2}\mathbf{\Delta v}^T HE(\mathbf{v}_0)\mathbf{\Delta v} + \mathcal{O}\left(\frac{1}{N^{1.5}}\right) \quad (5.5)$$

$$E_0(\mathbf{v}_0 \pm \mathbf{\Delta v}) = E_0(\mathbf{v}_0) + \frac{1}{2}\mathbf{\Delta v}^T HE_0(\mathbf{v}_0)\mathbf{\Delta v} + \mathcal{O}\left(\frac{1}{N^{1.5}}\right) \quad (5.6)$$

Since $\mathbf{v}_0 = \mathbf{v}^* + \mathcal{O}\left(\frac{1}{N^{0.5}}\right)$:

$$HE(\mathbf{v}_0) = HE\left(\mathbf{v}^* + \mathcal{O}\left(\frac{1}{N^{0.5}}\right)\right) = HE(\mathbf{v}^*) + \mathcal{O}\left(\frac{1}{N^{0.5}}\right)$$

Therefore, using the fact that $\mathbf{\Delta v} = \mathcal{O}\left(\frac{1}{N^{0.5}}\right)$, and equation (5.5), we can write the average generalization error among $\mathbf{v}_0 + \mathbf{\Delta v}$ and $\mathbf{v}_0 - \mathbf{\Delta v}$ as:

$$\frac{E(\mathbf{v}_0 + \mathbf{\Delta v}) + E(\mathbf{v}_0 - \mathbf{\Delta v})}{2} = E(\mathbf{v}_0) + \frac{1}{2}\mathbf{\Delta v}^T HE(\mathbf{v}^*)\mathbf{\Delta v} + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)$$

Define $\mathbf{W}_\delta = \{\mathbf{\Delta v} : E_0(\mathbf{v}_0 + \mathbf{\Delta v}) = E_0(\mathbf{v}_0) + \delta + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)\}$. Therefore for each $\mathbf{\Delta v} \in \mathbf{W}_\delta$, there is a $-\mathbf{\Delta v} \in \mathbf{W}_\delta$. As we did for the proof of lemma 5.6.1, using the uniform probability of selection $P_{\mathbf{W}_\delta}$, we can compute the mean generalization error as:

$$\begin{aligned}
E_{mean}(E_\delta) &= \int_{\mathbf{\Delta v} \in \mathbf{W}_\delta} P_{\mathbf{W}_\delta}(\mathbf{\Delta v}) E(\mathbf{v}_0 + \mathbf{\Delta v}) d\mathbf{\Delta v} \\
&= 0.5 \int_{\mathbf{\Delta v} \in \mathbf{W}_\delta} \left(P_{\mathbf{W}_\delta}(\mathbf{\Delta v}) E(\mathbf{v}_0 + \mathbf{\Delta v}) + P_{\mathbf{W}_\delta}(-\mathbf{\Delta v}) E(\mathbf{v}_0 - \mathbf{\Delta v})\right) d\mathbf{\Delta v} \\
&= E(\mathbf{v}_0) + 0.5 \int_{\mathbf{\Delta v} \in \mathbf{W}_\delta} P_{\mathbf{W}_\delta}(\mathbf{\Delta v}) \mathbf{\Delta v}^T HE(\mathbf{v}^*) \mathbf{\Delta v} d\mathbf{\Delta v} + \mathcal{O}\left(\frac{1}{N^{1.5}}\right) \\
&= E(\mathbf{v}_0) + \beta(\delta) + \mathcal{O}\left(\frac{1}{N^{1.5}}\right)
\end{aligned}$$

Since $\mathbf{v}^*$ is the generalization error minimum, $HE(\mathbf{v}^*)$ is positive semi-definite. Hence,

$$\beta(\delta) = 0.5 \int_{\mathbf{\Delta v} \in \mathbf{W}_\delta} P_{\mathbf{W}_\delta}(\mathbf{\Delta v}) \mathbf{\Delta v}^T H E(\mathbf{v}^*) \mathbf{\Delta v} d\mathbf{\Delta v} \geq 0. \ \square$$

5.6.4 Proof of Theorem 5.3.1:

Expanding the mean generalization error from equation (5.3):

$$\begin{aligned} E_{mean}(\nu) &= \mathcal{E}[\pi|\nu] = \sum_{m=1}^{M} \pi_m Pr[\pi_m|\nu_m = \nu] \\ &= \frac{\sum_{m=1}^{M} \pi_m Pr[\nu_m = \nu|\pi_m] Pr[\pi_m]}{\sum_{m=1}^{M} Pr[\nu_m = \nu|\pi_m] Pr[\pi_m]} \\ &= \frac{\sum_{m=1}^{M} \pi_m Pr[\pi_m] {\pi_m}^{N\nu} (1-\pi_m)^{N(1-\nu)}}{\sum_{m=1}^{M} Pr[\pi_m] {\pi_m}^{N\nu} (1-\pi_m)^{N(1-\nu)}} \end{aligned}$$

Taking the derivative of $\mathcal{E}[\pi|\nu]$ w.r.to ν:

$$\frac{d\mathcal{E}[\pi|\nu]}{d\nu} = Q_0 \sum_{\pi_m < \pi_k} Q_{m,k} (\pi_m - \pi_k) ln \left(\frac{\pi_m}{1-\pi_m} \frac{1-\pi_k}{\pi_k} \right)$$

where $Q_0 = \frac{1}{N\left(\sum_{m=1}^{M} Pr[\pi_m] {\pi_m}^{N\nu} (1-\pi_m)^{N(1-\nu)}\right)^2} > 0$ and
$Q_{m,k} = {\pi_m}^{N\nu} (1-\pi_m)^{N(1-\nu)} {\pi_k}^{N\nu} (1-\pi_k)^{N(1-\nu)} > 0$. When $\pi_m < \pi_k$ both $(\pi_m - \pi_k)$ and $ln \left(\frac{\pi_m}{1-\pi_m} \frac{1-\pi_k}{\pi_k} \right)$ are negative hence the derivative is positive. Therefore the mean generalization error is an increasing function of the training error. $\square$

Chapter 6
Conclusion

6.1 Summary of Results

In this thesis, we studied a method of incorporating input information into learning. We suggested obtaining an estimator of the out-of-sample error using the input information. The solution of this estimator, augmented solution is superior to the least squares solution for general linear models. We also provided an algorithm to descend on the augmented error, determining the augmentation parameters using cross validation method. This algorithm also performed better than the minimization of only the training error.

The descent on the augmented objective function by means of finding the parameters using cross validation is a general method that can be applied to any augmented objective function, and in particular to objective functions when learning from hints. For invariance hints, we have shown that this algorithm results in better performance than gradient descent on the training error, however direct estimation of the test error seems to result in even better performance. When direct estimation of the test error using a hint is not an option, the gradient descent on the augmented objective function can serve as a technique to descend on the training error and the hint errors at the same time.

Additional information, such as test inputs or hints, result in better performance than minimizing the training error only. In the last chapter of this thesis, we prove that unless there is additional information, the training error minimum is the best possible solution. If a method is choosing a solution other than the training error minimum, then there is an assumption of prior information. One should be aware of the prior assumptions implied by an algorithm.

6.2 Further Study

The following research directions could be investigated for further study:

- Investigation of the best k for leave-k-out cross validation when gradient descending on an augmented error (for input information or hints).
- The augmented error for classification problems and neural networks with *tanh* output units.
- The augmented error for different loss functions, such as entropic, p-norm loss and loss for input dependent noise.
- Performance of the augmented solution for linear and nonlinear models when the test inputs come from a different distribution than the training inputs.
- Performance of the augmented solution when instead of gradient descent, other learning algorithms, such as conjugate gradient or Levenberg-Marquardt optimization method are used.
- The effect of the test input information on VC bounds.
- Comparison of the augmented solution to EM (expectation maximization) results.

Bibliography

[Abu-Mostafa, 1989] Abu-Mostafa, Y. (1989). The vanik-chervonenkis dimension: Information versus complexity in learning. *Neural Computation*, 1:312–317.

[Abu-Mostafa, 1990] Abu-Mostafa, Y. (1990). Learning from hints in neural networks. *Journal of Complexity*, 6:192–198.

[Abu-Mostafa, 1993a] Abu-Mostafa, Y. (1993a). Hints and the vc dimension. *Neural Computation*, 4:278–288.

[Abu-Mostafa, 1993b] Abu-Mostafa, Y. (1993b). A method for learning from hints. In Hanson, S. J., Cowan, J. D., and Giles, C. L., editors, *Advances in Neural Information Processing Systems*, volume 5, pages 73–80. Morgan Kaufmann, San Mateo.

[Abu-Mostafa, 1994] Abu-Mostafa, Y. (1994). Learning from hints. *Journal of Complexity*, 10:165–178.

[Abu-Mostafa and Song, 1996] Abu-Mostafa, Y. and Song, X. (1996). Bin model for neural networks. In *Proceedings of the International Conference on Neural Information Processing, Hong Kong, 1996*, pages 169–173.

[Al-Mashouq and Reed, 1991] Al-Mashouq, K. A. and Reed, I. S. (1991). Including hints in training neural nets. *Neural Computation*, 3:418–427.

[Amari et al., 1997] Amari, S., , Murata, N., Mullet, K., Finke, M., and Yang, H. H. (1997). Asymptotic statistical theory of overtraining and cross-validation. *IEEE Transactions on Neural Networks*, 8(5):985–996.

[Baldi and Chauvin, 1991] Baldi, P. and Chauvin, Y. (1991). Temporal evolution of generalization during learning in linear networks. *Neural Computation*, 3:589–603.

[Battiti, 1989] Battiti, R. (1989). Accelerated backpropagation learning: Two optimization methods. *Complex Systems*, 3:331–342.

[Baum and Haussler, 1989] Baum, E. B. and Haussler, D. (1989). What size net gives valid generalization. *Neural Computation*, 1(1):151–160.

[Bishop, 1995] Bishop, C. (1995). *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford.

[Castelli and Cover, 1995] Castelli, V. and Cover, T. (1995). On the exponential value of labeled samples. *Pattern Recognition Letters*, 16:105–111.

[Castelli and Cover, 1996] Castelli, V. and Cover, T. (1996). The relative value of labeled and unlabeled samples. *IEEE Transactions on Information Theory*, 42(6):2102–2117.

[Cataltepe and Abu-Mostafa, 1993] Cataltepe, Z. and Abu-Mostafa, Y. (1993). Estimating learning performance using hints. In Mozer, M. et al., editors, *Proceedings of the 1993 Connectionist Models Summer School*, pages 380–386. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ.

[Cataltepe et al., 1998] Cataltepe, Z., Abu-Mostafa, Y. S., and Magdon-Ismail, M. (1998). No free lunch for early stopping. *to appear in Neural Computation.*

[Cataltepe and Magdon-Ismail, 1998] Cataltepe, Z. and Magdon-Ismail, M. (1998). Incorporating test inputs into learning. In Mozer, M., Jordan, M., and Petsche, T., editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, Cambridge.

[Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B*, 39:1–22.

[Dodier, 1996] Dodier, R. (1996). Geometry of early stopping in linear networks. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 365–371. The MIT Press, Cambridge.

[Duda, 1973] Duda, R. O. (1973). *Pattern Classification and Scene Analysis.* John Wiley & Sons, Inc., NY.

[Fyfe, 1992] Fyfe, A. (1992). *Invariance Hints and the VC Dimension.* PhD thesis, California Institute of Technology, Pasadena, CA 91125.

[Geman et al., 1992] Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58.

[Giles and Maxwell, 1987] Giles, C. L. and Maxwell, T. (1987). Learning, invariance and generalization in high-order neural networks. *Applied Optics*, 26(23):4972 – 4978.

[Golub and Van Loan, 1993] Golub, G. H. and Van Loan, C. F. (1993). *Matrix Computations.* The Johns-Hopkins University Press, Baltimore, MD.

[Goutte, 1997] Goutte, C. (1997). Note on free lunches and cross-validation. *Neural Computation*, 9(6):1053–1059.

[Hertz et al., 1991] Hertz, K., Krough, A., and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation, Lecture Notes, vol. 1, Santa Fe Institute Studies in The Sciences of Complexity.* Santa Fe Institute.

[Hestenes and Stiefel, 1952] Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436.

[Hocking, 1996] Hocking, R. R. (1996). *Methods and Applications of Linear Models.* John Wiley & Sons, NY.

[Ji et al., 1990] Ji, C., Snapp, R. R., and Psaltis, D. (1990). Generalizing smoothness constraints from discrete samples. *Neural Computation*, 2:188–197.

[Krogh and Hertz, 1992] Krogh, A. and Hertz, J. A. (1992). A simple weight decay can improve generalization. In Moody, J. E., Hanson, S. J., and Lippmann, R. P., editors, *Advances in Neural Information Processing Systems*, volume 4, pages 950–957. Morgan Kaufmann, San Mateo.

[Levenberg, 1944] Levenberg, K. (1944). A method for the solution of certain nonlinear problems in least squares. *Quarterly Journal of Applied Mathematics, II (2)*, pages 164–168.

[Little, 1992] Little, R. J. (1992). Regression with missing x's: A review. *Journal of the American Statistical Association*, 87(420):1227–1237.

[Marquardt, 1963] Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society of Industrial and Applied Mathematics*, 11(2):431–441.

[Miller and Uyar, 1997] Miller, D. J. and Uyar, H. S. (1997). A mixture of experts classifier with learning based on both labeled and unlabeled data. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural and Information Processing Systems*, volume 9, pages 571–577, Cambridge, MA. MIT Press.

[Montgomery and Peck, 1991] Montgomery, D. C. and Peck, E. A. (1991). *Introduction to Linear Regression Analysis.* John Wiley & Sons Inc., NY.

[Reed, 1993] Reed, R. (1993). Pruning algorithms – a survey. *IEEE Transactions on Neural Networks*, 4(5):740–747.

[Ripley, 1996] Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press.

[Ross, 1987] Ross, S. M. (1987). *Introduction to Probability and Statistics for Engineers and Scientists.* John Wiley & Sons Inc., NY.

[Rumellhart et al., 1986] Rumellhart, D. E., McClelland, J. L., and Williams, R. J. (1986). *Learning Internal Representations by Error Propagation*, volume 1, pages 318–362. MIT Press, Cambridge, MA.

[Shahshahani and Landgrebe, 1994] Shahshahani, B. M. and Landgrebe, D. A. (1994). The effect of unlabeled samples in reducing small sample size problem and mitigating the hughes phonemenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5):1087–1095.

[Sill and Abu-Mostafa, 1997] Sill, J. and Abu-Mostafa, Y. S. (1997). Monotonicity hints. In Mozer, M., Jordan, M., and Petsche, T., editors, *Advances in Neural Information Processing Systems*, volume 9, pages 634–640. The MIT Press, Cambridge.

[Sjoberg and Ljung, 1995] Sjoberg, J. and Ljung, L. (1995). Overtraining, regularization, and searching for a minimum, with application to neural networks. *International Journal of Control*, 62(6):1391–1407.

[Vapnik, 1982] Vapnik, V. (1982). *Estimation of Dependencies Based on Empirical Data.* Springer-Verlag New York Inc., NY.

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

[Wahba, 1990] Wahba, G. (1990). *Spline Models for Observational Data.* SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA.

[Wang et al., 1994] Wang, C., , Venkatesh, S. S., and Judd, J. S. (1994). Optimal stopping and effective machine complexity in learning. In Cowan, J. D., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems*, volume 6, pages 303–310. Morgan Kaufmann, San Francisco.

[Weisberg, 1980] Weisberg, S. (1980). *Applied Linear Regression.* John Wiley & Sons, NY.

[Wolpert, 1995] Wolpert, D. H. (1995). *The Mathematics of Generalization, the Proceedings of the SFI/CNLS Workshop on Formal Approaches to Supervised Learning.* Addison Wesley, Reading, MA.

[Wolpert, 1996a] Wolpert, D. H. (1996a). The existence of A priori distinctions between learning algorithms. *Neural Computation*, 8(7):1391–1420.

[Wolpert, 1996b] Wolpert, D. H. (1996b). The lack of A priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390.

[Zhu and Rohwer, 1996] Zhu, H. and Rohwer, R. (1996). No free lunch for cross-validation. *Neural Computation*, 8(7):1421–1426.