Linear Unlearning for Cross-Validation*

Lars Kai Hansen and Jan Larsen
CONNECT, Electronics Institute B349
Technical University of Denmark
DK-2800 Lyngby, Denmark
Phones: (+45) 45253889, (+45) 45253923
Fax: (+45) 45880117
Emails: lkhansen,jlarsen@ei.dtu.dk

February 1, 1996

Abstract

The leave-one-out cross-validation scheme for generalization assessment of neural network models is computationally expensive due to replicated training sessions. In this paper we suggest linear unlearning of examples as an approach to approximative cross-validation. Further, we discuss the possibility of exploiting the ensemble of networks offered by leave-one-out for performing ensemble predictions. We show that the generalization performance of the equally weighted ensemble predictor is identical to that of the network trained on the whole training set.

Numerical experiments on the sunspot time series prediction benchmark demonstrates the potential of the linear unlearning technique.

1 Introduction

Consider nonlinear regression in which the output y is regressed nonlinearly on the input vector $\boldsymbol{x}$. In this paper we focus on a neural network implementation, in which the output is predicted by $\widehat{y} = F(\boldsymbol{x}; \boldsymbol{w})$ where $F(\cdot)$ denotes the nonlinear mapping of the neural net and $\boldsymbol{w}$ is the vector of network parameters.

The *conditional input-output distribution*, i.e., the probability distribution of the output conditioned on a test input, is a basic objective for neural net modeling. A main source of uncertainty, when estimating the parameters of the conditional distribution, is the random selection of training data. The associated risk of overfitting is of major concern in neural network design. The use of *system identification* design tools in neural net learning has been pioneered by (Moody, 1991), who derived estimators for the expected generalization error of regularized networks. These

*Preprint, to appear in Advances in Computational Mathematics, 1996.

estimates, however, depend on a number of assumptions that can be quite hard to justify. Hence, it would be highly desirable to be able to perform an additional data-driven *consistency check* offered by the cross-validation technique.

The idea of cross-validation (Stone, 1974), (Toussaint, 1974) is based on training and testing on disjunct subsets resampled from the database, forming the *cross-validation ensemble* of models. The leave-one-out (LOO) ensemble of networks trained on all subsets leaving out one training example is an attractive — though computationally expensive — vehicle for generalization assessment of a neural network model. For the conventional neural net approaches unlearning of examples is not possible, and one basically has to train the full ensemble of networks, making the approach computationally unfeasible.

In this paper we suggest approximate evaluation of the ensemble using *linear unlearning* of individual examples. It is assumed that unlearning of a single example only affects the network weights slightly. Under this hypothesis we estimate the change in the network parameters within the quadratic approximation of the network cost function. Using the ensemble we derive an estimator for the test error of a regularized network which in fact is similar to an estimate due to (Wahba, 1990), but different from the conventional estimators as FPE (Akaike, 1969), Wahba's GCV, and GPE (Moody, 1991). The proposed method is further related to NCV (Moody, 1994) which approximates leave-v-out cross-validation. We finally discuss the possibility of exploiting the ensemble of networks for making ensemble predictions and for obtaining error bars on future examples.

The leave-one-out test error is compared to that obtained through linear unlearning on a benchmark case showing the viability of the approach.

2 Linear Unlearning

The network cost function is assumed to be a sum of the loss function $E(\boldsymbol{w})$ (additive in the example losses denoted ϵ [1].) and a regularization term $R(\boldsymbol{w})$, as shown by

$$C(\boldsymbol{w}) = E(\boldsymbol{w}) + R(\boldsymbol{w}) = \sum_{\alpha=1}^{N} \epsilon\left(y_\alpha, \widehat{y}_\alpha, \boldsymbol{w}\right) + R(\boldsymbol{w}) \tag{1}$$

where y_α is desired the output[2] (target) and N is the number of training examples, i.e., input-output pairs: $\boldsymbol{D} = [\,(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_N, y_N)\,]$. Training on the full set of examples provides a parameter vector denoted by $\widehat{\boldsymbol{w}}$; hence,

$$\frac{\partial C\left(\widehat{\boldsymbol{w}}\right)}{\partial \boldsymbol{w}} = \mathbf{0}. \tag{2}$$

[1] Most learning problems come with a natural loss measure, e.g., the squared error measure $\epsilon\left(y, \widehat{y}\right) = \left(y - \widehat{y}\right)^2$, where the desired target is denoted y and the network output is denoted $\widehat{y}$

[2] For simplicity we consider single output networks only. However, without further ado, the theory is valid for multiple output networks.

Likewise, a leave-one-out ensemble of network parameters, $\{\widehat{\boldsymbol{w}}_\beta\}_{\beta=1}^N$, is obtained by training on the N subsets, $\boldsymbol{D}_\beta$, containing $N-1$ examples:

$$C_\beta(\boldsymbol{w}) = \sum_{\alpha=1,\alpha\neq\beta}^{N} \epsilon\left(y_\alpha, \widehat{y}_\alpha, \boldsymbol{w}\right) + R(\boldsymbol{w}), \tag{3}$$

hence,

$$\frac{\partial C_\beta\left(\widehat{\boldsymbol{w}}_\beta\right)}{\partial \boldsymbol{w}} = \mathbf{0}. \tag{4}$$

We suggest to estimate the variation of the parameter vectors of the leave-one-out ensemble, $\Delta\boldsymbol{w}_\beta \equiv \widehat{\boldsymbol{w}}_\beta - \widehat{\boldsymbol{w}}$, by using a Taylor expansion of equation (4). Since

$$C_\beta(\boldsymbol{w}) = C(\boldsymbol{w}) - \epsilon\left(y_\beta, \widehat{y}_\beta, \boldsymbol{w}\right), \tag{5}$$

$\Delta\boldsymbol{w}_\beta$ satisfies

$$\boldsymbol{J}_\beta \Delta\boldsymbol{w}_\beta - \boldsymbol{g}_\beta + \boldsymbol{o}(\|\Delta\boldsymbol{w}_\beta\|) = \mathbf{0}. \tag{6}$$

where $\boldsymbol{o}(\cdot)$ is the vector order function. We further have defined the the *Hessian* of the regularized cost function, $\boldsymbol{J}_\beta$, and the *gradient* of the example loss, $\boldsymbol{g}_\beta$, by[3]:

$$\boldsymbol{J}_\beta = \frac{\partial^2 C_\beta\left(\widehat{\boldsymbol{w}}\right)}{\partial\boldsymbol{w}\partial\boldsymbol{w}^\top}, \qquad \boldsymbol{g}_\beta = \frac{\partial \epsilon\left(y_\beta, \widehat{y}_\beta, \widehat{\boldsymbol{w}}\right)}{\partial \boldsymbol{w}}. \tag{7}$$

Solving equation (6), with the additional assumption that the regularized Hessian is non-singular, we find the N weight vectors in the ensemble given by:

$$\widehat{\boldsymbol{w}}_\beta = \widehat{\boldsymbol{w}} + \boldsymbol{J}_\beta^{-1}\boldsymbol{g}_\beta + \boldsymbol{o}(\|\Delta\boldsymbol{w}_\beta\|). \tag{8}$$

With this ensemble in hand, we can get approximations of various interesting quantities which help us in validating the network model.

2.1 Average Generalization Error Estimate

A common measure of the quality of a neural model is the expected generalization error[4] (see e.g., (Larsen & Hansen, 1994), (Moody, 1991)) defined as the expected loss on a test sample, further taking the expectation w.r.t. to the training set distribution[5]:

$$\langle E_{\text{test}}(\widehat{\boldsymbol{w}})\rangle_{\boldsymbol{D}} = \left\langle \langle \epsilon(y, \widehat{y}, \widehat{\boldsymbol{w}})\rangle_{(\boldsymbol{x},y)} \right\rangle_{\boldsymbol{D}} = \int \left[\int \epsilon(y, \widehat{y}, \widehat{\boldsymbol{w}}) \cdot p(\boldsymbol{x}, y)\, d\boldsymbol{x}\, dy\right] p(\boldsymbol{D})\, d\boldsymbol{D}. \tag{9}$$

where $\langle\cdot\rangle_{(\boldsymbol{x},y)}$ is the expectation w.r.t. to the joint input-output probability density $p(\boldsymbol{x}, y)$, and $p(\boldsymbol{D})$ is the joint probability density of the training data. $\langle\cdot\rangle_{\boldsymbol{D}}$ denotes the expectation w.r.t. to all training sets of size N[6].

[3] Here we implicitly assume that the cost function is twice continuously differentiable.

[4] Also known as the expected test error or the expected prediction risk.

[5] By assumption all expectations exist, i.e., $E_{\text{test}} < \infty$.

[6] Note, for notational convenience, we do not explicitly distinguish between the particular realization of the data set and the data set regarded as a random variable.

Since $p(\boldsymbol{x}, y)$ is unknown we seek for an estimate of $\langle E_{\text{test}} \rangle_{\boldsymbol{D}}$, like the leave-one-out test error given by,

$$E_{\text{LOO}} = \frac{1}{N} \sum_{\beta=1}^{N} \epsilon\left(y_\beta, \widehat{y}_\beta, \widehat{\boldsymbol{w}}_\beta\right). \tag{10}$$

In general, it is difficult to give quantitative results on the LOO test error as an estimator of $\langle E_{\text{test}} \rangle_{\boldsymbol{D}}$; however, one simple theorem applies:

Theorem 1 *If the training data are* ***independently*** *distributed,* E_{LOO} *is an unbiased estimate of* $\langle E_{\text{test}}(\widehat{\boldsymbol{w}}_\beta) \rangle_{\boldsymbol{D}_\beta}$ *where* $\boldsymbol{D}_\beta$ *is the training data with sample* β *left out, and* $\widehat{\boldsymbol{w}}_\beta$ *is the estimate obtained by training on* $N-1$ *examples.*

Proof Assume training data independence, then $p(\boldsymbol{D}) = p(\boldsymbol{D}_\beta) \cdot p(\boldsymbol{x}_\beta, y_\beta)$. The proof simply follows by evaluating $\langle E_{\text{LOO}} \rangle_{\boldsymbol{D}}$. ■

Theorem 2 *An* $o(1/N)$ *approximation of the LOO test error (10) is given by*

$$\widehat{E}_{\text{LOO}} = \frac{1}{N} \sum_{\beta=1}^{N} \left[\epsilon\left(y_\beta, \widehat{y}_\beta, \widehat{\boldsymbol{w}}\right) + \boldsymbol{g}_\beta^\top \boldsymbol{J}_\beta^{-1} \boldsymbol{g}_\beta\right]. \tag{11}$$

Proof From (6) it is easy to verify that $\Delta \boldsymbol{w}_\beta = O(1/N)$, where $O(\cdot)$ is the Landau order function. For consistency, the approximation of the LOO estimator should not include terms of $O(1/N^i)$, $i \geq 2$. Thus expanding the LOO test error (10) linearly in $\Delta \boldsymbol{w}_\beta$ and using (7), (8) we get the desired result. Note that $o(1/N)$ is the order function, i.e., if $a(N) = o(1/N)$ then $a(N)/N \to 0$ as $N \to \infty$. ■

Since only one data example is left out when resampling, we generally expect the $o(1/N)$ approximation to be fairly good — even for moderate training set sizes. Only in the case of a network which is linear in the parameters and trained with a quadratic cost function[7], it is possible to obtain an exact expression (see further (Wahba, 1990) and section 3.1).

3 Mean Square Error Learning

Our scheme can be applied to any cost function and network type requiring the cost to be twice continuously differentiable in the weights. Here we consider the standard case of a regression net trained with the mean square error measure. Let $F(\boldsymbol{x}, \boldsymbol{w})$ be the network function, then the loss is the squared error between the output and the predicted output, as follows:

$$\epsilon\left(y_\alpha, \widehat{y}_\alpha, \boldsymbol{w}\right) = \left(y_\alpha - F(\boldsymbol{x}_\alpha, \boldsymbol{w})\right)^2. \tag{12}$$

Introducing the gradient of the network function, $\boldsymbol{h}_\beta = \partial F(\boldsymbol{x}_\beta, \widehat{\boldsymbol{w}})/\partial \boldsymbol{w}$ we use (7) and find

$$\boldsymbol{g}_\beta = -2\left(y_\beta - F(\boldsymbol{x}_\beta, \widehat{\boldsymbol{w}})\right) \boldsymbol{h}_\beta. \tag{13}$$

[7] That is, $\epsilon = (y - \widehat{y})^2$ and $R(\boldsymbol{w}) \propto \boldsymbol{w}^\top \boldsymbol{w}$. Furthermore, (10) should be expanded quadratically in $\Delta \boldsymbol{w}_\beta$.

3.1 The LOO Test Error

If (13) is inserted in (11) we get:

$$\widehat{E}_{\mathrm{LOO}} = \frac{1}{N} \sum_{\beta=1}^{N} (y_\beta - F(\boldsymbol{x}_\beta; \widehat{\boldsymbol{w}}))^2 \left[1 + 4\boldsymbol{h}_\beta^\top \boldsymbol{J}_\beta^{-1} \boldsymbol{h}_\beta\right]. \tag{14}$$

Furthermore, it is often well motivated to invoke the so-called Gauss-Newton approximation for mean square error based problems (see e.g., (Ljung, 1987)), in which,

$$\boldsymbol{J} = \frac{\partial^2 C(\widehat{\boldsymbol{w}})}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top} = \frac{\partial^2 E(\widehat{\boldsymbol{w}})}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top} + \frac{\partial^2 R(\widehat{\boldsymbol{w}})}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top} \approx 2 \sum_{\alpha=1}^{N} \boldsymbol{h}_\beta \boldsymbol{h}_\beta^\top + \frac{\partial^2 R(\widehat{\boldsymbol{w}})}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top} \tag{15}$$

Within this approximation the estimator takes a particular simple form. Using, $\boldsymbol{J}_\beta = \boldsymbol{J} - 2\boldsymbol{h}_\beta \boldsymbol{h}_\beta^\top$, and the matrix inversion lemma (see e.g., (Ljung, 1987)) we find:

$$\boldsymbol{J}_\beta^{-1} = \boldsymbol{J}^{-1} + \frac{2\boldsymbol{J}^{-1} \boldsymbol{h}_\beta \boldsymbol{h}_\beta^\top \boldsymbol{J}^{-1}}{1 - 2\boldsymbol{h}_\beta^\top \boldsymbol{J}^{-1} \boldsymbol{h}_\beta}. \tag{16}$$

Inserting this expression into the estimate (14) we get the remarkable simple result,

$$\widehat{E}_{\mathrm{LOO}} = \frac{1}{N} \sum_{\beta=1}^{N} (y_\beta - F(\boldsymbol{x}_\beta; \widehat{\boldsymbol{w}}))^2 \, \frac{1 + 2\boldsymbol{h}_\beta^\top \boldsymbol{J}^{-1} \boldsymbol{h}_\beta}{1 - 2\boldsymbol{h}_\beta^\top \boldsymbol{J}^{-1} \boldsymbol{h}_\beta}. \tag{17}$$

With a pointer to classical test error estimators, (17) may be interpreted as a modified "example based" FPE. Thus the term, $2\boldsymbol{h}_\beta^\top \boldsymbol{J}^{-1} \boldsymbol{h}_\beta$, corresponds to the effective number of parameters divided by the training set size for the particular example β. With this construction one may hope that the statistical properties of the input distribution are reflected in the estimator. In the conventional asymptotically estimators the properties of the input distribution are eliminated from the theory by invoking the limit of large training sets. For further reference, see (Moody, 1991) and (Larsen & Hansen, 1994).

There is a close connection with (17) and the "leaving-one-out lemma" (Stone, 1974), (Wahba, 1990) which relates the LOO errors $\epsilon(y_\beta, \widehat{y}_\beta, \widehat{\boldsymbol{w}}_\beta)$ with the losses from the full set model, $\epsilon(y_\beta, \widehat{y}_\beta, \widehat{\boldsymbol{w}})$. In fact, if the model is linear in the weights and weight decay regularization is used (i.e., a quadratic regularizer) then the estimate coincides with the "leaving-one-out lemma" except that one has to take the regularization into account. Suppose the network is linear, i.e., $F(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{x}$, then the LOO test error reads:

$$E_{\mathrm{LOO}} = \frac{1}{N} \sum_{\beta=1}^{N} \frac{\left(y_\beta - \widehat{\boldsymbol{w}}^\top \boldsymbol{x}_\beta\right)^2}{\left(1 - 2\boldsymbol{h}_\beta^\top \boldsymbol{J}^{-1} \boldsymbol{h}_\beta\right)^2}. \tag{18}$$

Notice that this is an exact expression unlike the result of theorem 2.

3.2 Ensemble Of Networks

With the ensemble of LOO estimates $\{\widehat{\boldsymbol{w}}_\beta\}_{\beta=1}^N$ one might consider the ensemble network output which results by combining $F(\boldsymbol{x}, \widehat{\boldsymbol{w}}_\beta)$.

Now, for the sake of generality, we consider leave-v-out cross-validation. That is, split the training set $\boldsymbol{D}$ into K disjoint cross-validation sets of size v, with $N = Kv$ [8], and train on the remaining $N - v$ samples. The training sets are denoted by $\boldsymbol{D}_\beta$, $\beta \in [1; K]$.

Considering the ensemble network, we state the following

Theorem 3 *Assume a leave-v-out cross-validation scenario, and let $C_\beta(\boldsymbol{w})$ be the cost function (1) evaluated on the training data $\boldsymbol{D}_\beta$, $\beta \in [1; K]$. The loss is the mean square error, and the weight estimates are defined as $\widehat{\boldsymbol{w}}_\beta = \arg\min_{\boldsymbol{w}} C_\beta(\boldsymbol{w})$.*

Secondly, assume that the data are generated according to $y_\alpha = \phi(\boldsymbol{x}_\alpha) + n_\alpha$ where $\phi(\cdot)$ is a nonlinear function, and n_α is zero mean white noise with variance $\sigma_n^2 < \infty$, independent of the input. Further, that the neural model is complete, i.e., $\exists \boldsymbol{w}^\circ, \forall \boldsymbol{x} : \; F(\boldsymbol{x}, \boldsymbol{w}^\circ) \equiv \phi(\boldsymbol{x})$.

Thirdly, assume that the ensemble network is defined by

$$\bar{F}(\boldsymbol{x}, \widehat{\boldsymbol{w}}_1, \cdots, \widehat{\boldsymbol{w}}_K) = \sum_{\beta=1}^{K} u_\beta F(\boldsymbol{x}, \widehat{\boldsymbol{w}}_\beta) \tag{19}$$

where $\{u_\beta\}_{\beta=1}^K$ is a set of weights, independent on the training data[9], satisfying $\sum_\beta u_\beta = 1$.

The following properties then applies to the expected generalization error (9):

1. *The expected generalization error is minimized when $u_\beta = 1/K$, $\beta \in [1; K]$.*

2. *In the $o(1/N)$ approximation[10], the expected generalization error of the ensemble network equals that of using a single network trained on all data.*

Proof Using a $o(1/N)$ approximation of the expected generalization error, (Larsen & Hansen, 1994) showed for a single network trained on all N data:

$$\langle E_{\text{test}} \rangle_{\boldsymbol{D}} = \sigma_n^2 \left(1 + m_{\text{eff}}/N\right) + o(1/N) \tag{20}$$

where m_{eff} is the effective number of weights,

$$m_{\text{eff}} = \operatorname{tr}\left[\widetilde{\boldsymbol{H}}_\circ \widetilde{\boldsymbol{J}}_\circ^{-1} \widetilde{\boldsymbol{H}}_\circ \widetilde{\boldsymbol{J}}_\circ^{-1}\right], \tag{21}$$

$\operatorname{tr}[\cdot]$ is the trace operator, and the "true" scaled Hessians[11] are defined by:

$$\widetilde{\boldsymbol{H}}_\circ = \left\langle \frac{\partial^2 E_{\text{test}}(\boldsymbol{w}^\circ)}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top} \right\rangle_{(\boldsymbol{x},y)}, \quad \widetilde{\boldsymbol{J}}_\circ = \widetilde{\boldsymbol{H}}_\circ + \frac{1}{N} \frac{\partial^2 R(\boldsymbol{w}^\circ)}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top}. \tag{22}$$

[8] The results presented are easily modified to deal with the case of v not being a divisor of N.

[9] The case where the weights depend on data is e.g., treated in (Krogh & Vedelsby, 1995).

[10] See e.g., (Larsen & Hansen, 1994), (Moody, 1991).

[11] Note that $\widetilde{\boldsymbol{J}}_\circ$ does not scale with N, whereas $\boldsymbol{J}$ does.

$\langle\cdot\rangle_{(\boldsymbol{x},y)}$ denotes the expectation w.r.t. the joint input-output density $p(\boldsymbol{x},y)$ and $\boldsymbol{w}^\circ$ are the optimal weights.

Define the weight fluctuations $\delta\boldsymbol{w}_\beta = \boldsymbol{w}^\circ - \widehat{\boldsymbol{w}}_\beta$ and let $\langle E_{\text{test}}(\widehat{\boldsymbol{w}}_1,\cdots,\widehat{\boldsymbol{w}}_K)\rangle_{\boldsymbol{D}}$ denote the expected generalization error of the ensemble network (19). Using a technique similar to that reported in (Larsen & Hansen, 1994), we perform a second order Taylor series expansion of the expected generalization error around $\boldsymbol{w}_\beta = \boldsymbol{w}^\circ$, as follows:

$$\begin{aligned}\langle E_{\text{test}}(\widehat{\boldsymbol{w}}_1,\cdots,\widehat{\boldsymbol{w}}_K)\rangle_{\boldsymbol{D}} &= E_{\text{test}}(\boldsymbol{w}^\circ,\cdots,\boldsymbol{w}^\circ) + \sum_{\beta=1}^{K} \frac{\partial E_{\text{test}}(\boldsymbol{w}^\circ,\cdots,\boldsymbol{w}^\circ)}{\partial \boldsymbol{w}_\beta^\top}\langle\delta\boldsymbol{w}_\beta\rangle_{\boldsymbol{D}} \\ &+\frac{1}{2}\sum_{\gamma=1}^{K}\sum_{\beta=1}^{K}\operatorname{tr}\left[\frac{\partial^2 E_{\text{test}}(\boldsymbol{w}^\circ,\cdots,\boldsymbol{w}^\circ)}{\partial\boldsymbol{w}_\gamma\partial\boldsymbol{w}_\beta^\top}\langle\delta\boldsymbol{w}_\gamma\delta\boldsymbol{w}_\beta^\top\rangle_{\boldsymbol{D}}\right] \\ &+\sum_{\beta=1}^{K} o\left(\|\delta\boldsymbol{w}_\beta\|^2\right). \qquad (23)\end{aligned}$$

Since $\boldsymbol{w}^\circ$ defines the optimal weight vector, the following facts are easily recognized:

- $E_{\text{test}}(\boldsymbol{w}^\circ,\cdots,\boldsymbol{w}^\circ) = \sigma_n^2$
- $\boldsymbol{w}^\circ = \arg\min_{\boldsymbol{w}} E_{\text{test}}(\boldsymbol{w})$; hence, $\partial E_{\text{test}}(\boldsymbol{w}^\circ,\cdots,\boldsymbol{w}^\circ)/\partial\boldsymbol{w}_\beta = \mathbf{0}$.

Furthermore, straightforward calculations show

$$\frac{\partial^2 E_{\text{test}}(\boldsymbol{w}^\circ,\cdots,\boldsymbol{w}^\circ)}{\partial\boldsymbol{w}_\gamma\partial\boldsymbol{w}_\beta^\top} = u_\gamma u_\beta \widetilde{\boldsymbol{H}}_\circ. \qquad (24)$$

Thus (23) reads:

$$\langle E_{\text{test}}(\widehat{\boldsymbol{w}}_1,\cdots,\widehat{\boldsymbol{w}}_K)\rangle_{\boldsymbol{D}} = \sigma_n^2 + \frac{1}{2}\sum_{\gamma=1}^{K}\sum_{\beta=1}^{K} u_\gamma u_\beta \operatorname{tr}\left[\widetilde{\boldsymbol{H}}_\circ\langle\delta\boldsymbol{w}_\gamma\delta\boldsymbol{w}_\beta^\top\rangle_{\boldsymbol{D}}\right] + \sum_{\beta=1}^{K} o\left(\|\delta\boldsymbol{w}_\beta\|^2\right). \qquad (25)$$

What remains is to find the covariance matrix of the weight fluctuations. Expanding $\partial C_\beta(\boldsymbol{w})/\partial\boldsymbol{w}$ to first order in $\delta\boldsymbol{w}_\beta$ and using the fact that $\partial C_\beta(\widehat{\boldsymbol{w}}_\beta)/\partial\boldsymbol{w} = \mathbf{0}$ one gets[12]

$$\delta\boldsymbol{w}_\beta = \left[\frac{\partial^2 C_\beta(\boldsymbol{w})}{\partial\boldsymbol{w}\partial\boldsymbol{w}^\top}\right]^{-1} \sum_{\alpha\in\boldsymbol{D}_\beta} 2\frac{\partial F(\boldsymbol{x}_\alpha,\boldsymbol{w}^\circ)}{\partial\boldsymbol{w}} n_\alpha - \frac{\partial R(\boldsymbol{w}^\circ)}{\partial\boldsymbol{w}}. \qquad (26)$$

Using this expression it can be shown that

$$\begin{aligned}&\langle\delta\boldsymbol{w}_\gamma\delta\boldsymbol{w}_\beta^\top\rangle_{\boldsymbol{D}} = \\ &\quad u_\gamma u_\beta\cdot\widetilde{\boldsymbol{J}}_\circ^{-1}\left[\frac{4}{(N-v)^2}\sum_{\alpha_1\in\boldsymbol{D}_\gamma}\sum_{\alpha_2\in\boldsymbol{D}_\beta}\left\langle\frac{\partial F(\boldsymbol{x}_{\alpha_1},\boldsymbol{w}^\circ)}{\partial\boldsymbol{w}}n_{\alpha_1}\frac{\partial F(\boldsymbol{x}_{\alpha_2},\boldsymbol{w}^\circ)}{\partial\boldsymbol{w}^\top}n_{\alpha_2}\right\rangle_{\boldsymbol{D}}\right]\widetilde{\boldsymbol{J}}_\circ^{-1} \\ &\quad +o(1/N). \qquad (27)\end{aligned}$$

[12]The notation $\alpha\in\boldsymbol{D}_\beta$ means that the summation runs over the indices of the data in training set $\boldsymbol{D}_\beta$.

Since the noise is white, the expectation $\langle\cdot\rangle_{\boldsymbol{D}}$ only gives a non-zero contribution, viz. σ_n^2, when $\alpha_1 = \alpha_2$. When $\gamma = \beta$ this occurs $N - v$ times, whereas when $\gamma \neq \beta$ it occurs $N - 2v$ times; equal to the overlap between two different training sets. In consequence, (25) becomes

$$\langle E_{\text{test}}(\widehat{\boldsymbol{w}}_1, \cdots, \widehat{\boldsymbol{w}}_K)\rangle_{\boldsymbol{D}} = \sigma_n^2 + \frac{\sigma_n^2 m_{\text{eff}}}{N - v}\left[\sum_{\beta=1}^{K} u_\beta^2 + \frac{N-2v}{N-v}\sum_{\gamma=1}^{K}\sum_{\beta=1,\beta\neq\gamma}^{K} u_\gamma u_\beta.\right] \tag{28}$$

In order to find the weights u_β which minimizes (28) under the constraint $\sum_\beta u_\beta = 1$, we apply a Lagrange technique. That is, minimize instead the Lagrange function $L = \langle E_{\text{test}}(\widehat{\boldsymbol{w}}_1, \cdots, \widehat{\boldsymbol{w}}_K)\rangle_{\boldsymbol{D}} + \lambda(\sum_\beta u_\beta - 1)$ with λ denoting the Lagrange multiplier. It is easily verified that $u_\beta = 1/K$, $\beta \in [1; K]$, minimizes the Lagrange function. Moreover, straightforward manipulations shows that $\langle E_{\text{test}}(\widehat{\boldsymbol{w}}_1, \cdots, \widehat{\boldsymbol{w}}_K)\rangle_{\boldsymbol{D}} = \langle E_{\text{test}}(\widehat{\boldsymbol{w}})\rangle_{\boldsymbol{D}}$. ■

Although the ensemble network does not deliver better generalization performance, one might hypothesize that the individual predictions for a test input $\boldsymbol{x}$ could be used to obtain sensible error bars. Error bars were mentioned in (Buntine & Weigend, 1991) who formulated the classical result see (Seber & Wild, 1989), p. 193 in a neural network context.

If we return to the specific LOO case, approximate predictions for a new test input, $\boldsymbol{x}$, are obtained by expanding the network function $F(\boldsymbol{x}, \boldsymbol{w})$, as follows:

$$F(\boldsymbol{x}, \widehat{\boldsymbol{w}}_\beta) = F(\boldsymbol{x}, \widehat{\boldsymbol{w}}) + \frac{\partial F(\boldsymbol{x}, \widehat{\boldsymbol{w}})}{\partial \boldsymbol{w}^\top}\Delta \boldsymbol{w}_\beta, \tag{29}$$

hence, using (6), (16),

$$F(\boldsymbol{x}, \widehat{\boldsymbol{w}}_\beta) = F(\boldsymbol{x}, \widehat{\boldsymbol{w}}) - (y_\beta - F(\boldsymbol{x}_\beta, \widehat{\boldsymbol{w}}))\,\frac{\boldsymbol{h}^\top(\boldsymbol{x})\boldsymbol{J}^{-1}\boldsymbol{h}_\beta}{1 - \boldsymbol{h}_\beta^\top \boldsymbol{J}^{-1}\boldsymbol{h}_\beta} \tag{30}$$

where $\boldsymbol{h}(\boldsymbol{x}) = \partial F(\boldsymbol{x}, \widehat{\boldsymbol{w}})/\partial \boldsymbol{w}$.

Unfortunately, it turns out that that the error bars which can be formed from (30) are only qualitative by nature, i.e., they may indicate in which parts of the input region high errors can be expected. The reason for this statement should be sought in the following: First the fluctuations among the individual predictions only reflect the variations due to the the fact that we estimate from a finite training set. That is, the noise inherent in the data generating system n_α is not included. The latter is easily incorporated by estimating the noise variance by other means, e.g., $\sigma_n^2 = E(\widehat{\boldsymbol{w}})/(N - m'_{\text{eff}})$, where m'_{eff} reflects an effective number of weights which differs slightly from m_{eff}, as reported by (Larsen & Hansen, 1994). More importantly, the fluctuations in the predictions do not scale properly with N. From the theory of the so-called Jackknife estimator (see e.g., (Fox *et al.*, 1980), (Seber & Wild, 1989)), it is known that in order to estimate the covariance matrix of the weight fluctuations from the LOO ensemble, we need to multiply the LOO fluctuations $\Delta \boldsymbol{w}_\beta$ in (6) by a factor of $\sqrt{N-1}$. We are currently pursuing this topic further, see (Larsen & Hansen, 1995).

4 Numerical Example

For illustration of the test error estimate and the ensemble predictions we study the well-known "sunspot" prediction benchmark (Weigend *et al.*, 1990). The network is a *tapped delay line architecture* with $I = 12$ input units, $H = 3$ hidden sigmoid units and a single linear output unit. The network function can be written as:

$$F\left(\boldsymbol{x}_\alpha, \boldsymbol{w}\right) = \sum_{j=1}^{H} w_j^H \tanh\left(\sum_{i=0}^{I-1} w_{ij}^I x_{\alpha-i} + w_{i0}^I\right) + w_0^H \tag{31}$$

where $\boldsymbol{x}_\alpha = [x_\alpha, \cdots, x_{\alpha-I+1}]$ is the input vector with x_α denoting the sunspot activity in the years 1700–1979 and $\boldsymbol{w} = [\boldsymbol{w}^I, \boldsymbol{w}^H]$ are the network parameters. The loss function is the squared error, and the regularization is a simple weight decay, i.e., $R(\boldsymbol{w}) = \kappa|\boldsymbol{w}|^2$, with $\kappa = 0.01$. We used a second order batch mode Gauss-Newton algorithm for training.

In figure 1 we show how the individual test errors (squared residuals) entering the

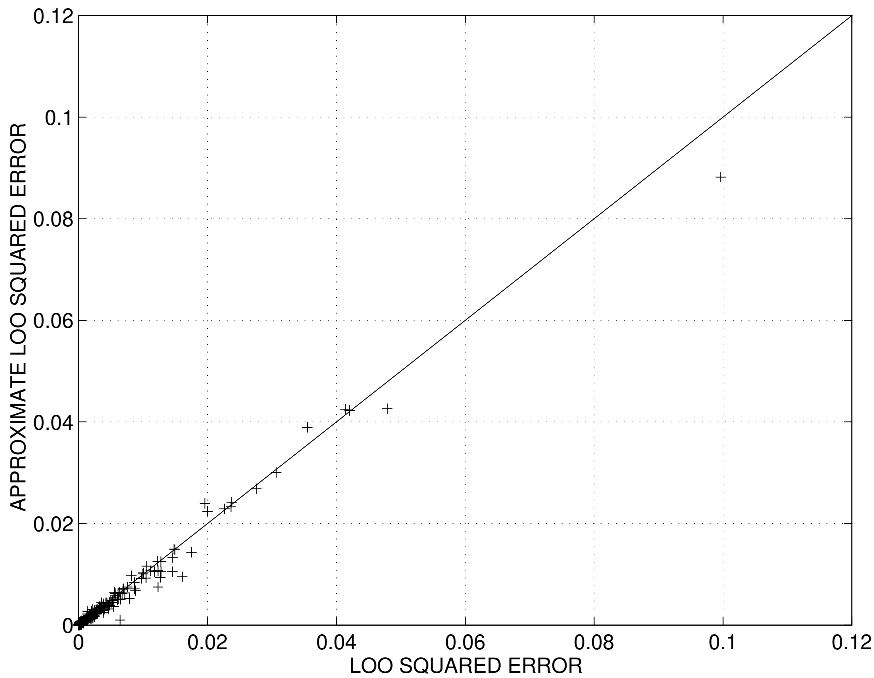

Figure 1: Correlation of individual squared errors (losses) in the leave-one-out (10) with the linear unlearned leave-one-out estimates (17) on the $N = 209$ sunspot training set.

test error estimate (17) correlate with the results of a full leave-one-out procedure, i.e., the result of training $N = 209$ networks on the corresponding subsets of the sunspot training set.

The LOO test error estimates (10), (17) are:

$$E_{\mathrm{LOO}} = 0.0050, \qquad \widehat{E}_{\mathrm{LOO}} = 0.0047 \tag{32}$$

which gives a 6% discrepancy.

To illustrate the capacity of the ensemble for representing the distribution of predictions on test inputs we show in figure 2 the ensemble evaluated on the sunspot "test set 1" (years 1921–1955). In the upper panel is shown the exact leave-one-out ensemble predictions, while in the lower panel we show the predictions of the approximate linearly unlearned ensemble. Recall that the fluctuations on these figures are merely qualitative by nature, i.e., in the regions where large fluctuations occurs we believe that the true error is high.

5 Conclusion

This paper suggested to use linear unlearning of examples to approximate the computationally expensive leave-one-out cross-validation technique. Numerical studies on the sunspot time series prediction benchmark demonstrated the viability of this approach.

We analyzed the possibility of employing the ensemble of networks produced by the cross-validation scheme for constructing an ensemble predictor. Considering a linear combination of networks, it was shown that the generalization performance is identical to that of using a single network trained on the full set of data.

Acknowledgments

This research was supported by the Thomas B. Thriges Foundation and the Danish Natural Science and Technical Research Councils through the Computational Neural Network Center (CONNECT). JL furthermore acknowledge the Radio Parts Foundation for financial support.

References

H. Akaike: "Fitting Autoregressive Models for Prediction," *Annals of the Institute of Statistical Mathematics*, **21**, 243–247, (1969).

W.L. Buntine & A.S. Weigend: "Bayesian Back-Propagation," *Complex Systems*, **5**, 603–643, (1991).

T. Fox, D. Hinkley & K. Larntz: "Jackknifing in Nonlinear Regression," *Technometrics*, **22**(1), 29–33, (1980).

A. Krogh & J. Vedelsby: "Neural Network Ensembles, Cross Validation, and Active Learning," in G. Tesauro *et al.* (eds.), *Advances in Neural Information Processing Systems 7*: MIT Press, Cambridge, Massachusetts, 1995.

J. Larsen & L.K. Hansen: "Generalization Performance of Regularized Neural Network Models," in J. Vlontzos, J.-N. Hwang & E. Wilson (eds.), *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing IV*, Piscataway, New Jersey: IEEE, 42–51, (1994).

J. Larsen & L.K. Hansen: "Empirical Generalization Assessment of Neural Network Models," in F. Girosi, J. Makhoul, E. Manolakos & E. Wilson (eds.).

Proceedings of the IEEE Workshop on Neural Networks for Signal Processing V, Piscataway, New Jersey: IEEE, 30–39, (1995).

L. Ljung: *System Identification: Theory for the User*, Englewood Cliffs, New Jersey: Prentice-Hall, (1987).

J. Moody: "Note on Generalization, Regularization, and Architecture Selection in Nonlinear Learning Systems," in B.H. Juang, S.Y. Kung & C.A. Kamm (eds.) *Proceedings of the first IEEE Workshop on Neural Networks for Signal Processing*, Piscataway, New Jersey: IEEE, 1–10, (1991).

J. Moody: "Prediction Risk and Architecture Selection for Neural Networks" in V. Cherkassky, J. H. Friedman & H. Wechsler (eds.) *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, Series F, **136**, Berlin, Germany: Springer-Verlag, (1994).

N. Murata, S. Yoshizawaand & S. Amari: "Network Information Criterion — Determining the Number of Hidden Units for an Artificial Neural Network Model," *IEEE Transactions on Neural Networks*, **5**(6), 865–872, (1994).

G.A.F. Seber & C.J. Wild: *Nonlinear Regression*, New York, New York: John Wiley & Sons, (1989).

M. Stone: "Cross-validatory Choice and Assessment of Statistical Predictors," *Journal of the Royal Statistical Society B*, **36**(2), 111–147, (1974).

G.T. Toussaint: "Bibliography on Estimation of Misclassification," *IEEE Transactions on Information Theory*, **20**(4), 472–479, (1974).

G. Wahba: "Spline Models for Observational Data," *CBMS-NSF Regional Conference Series in Applied Mathematics*, SIAM **59**, (1990).

A.S. Weigend, B.A. Hubermann & D.E. Rumelhart: "Predicting the Future: A Connectionist Approach," *International Journal of Neural Systems*, **1**(3), 193–209, (1990).

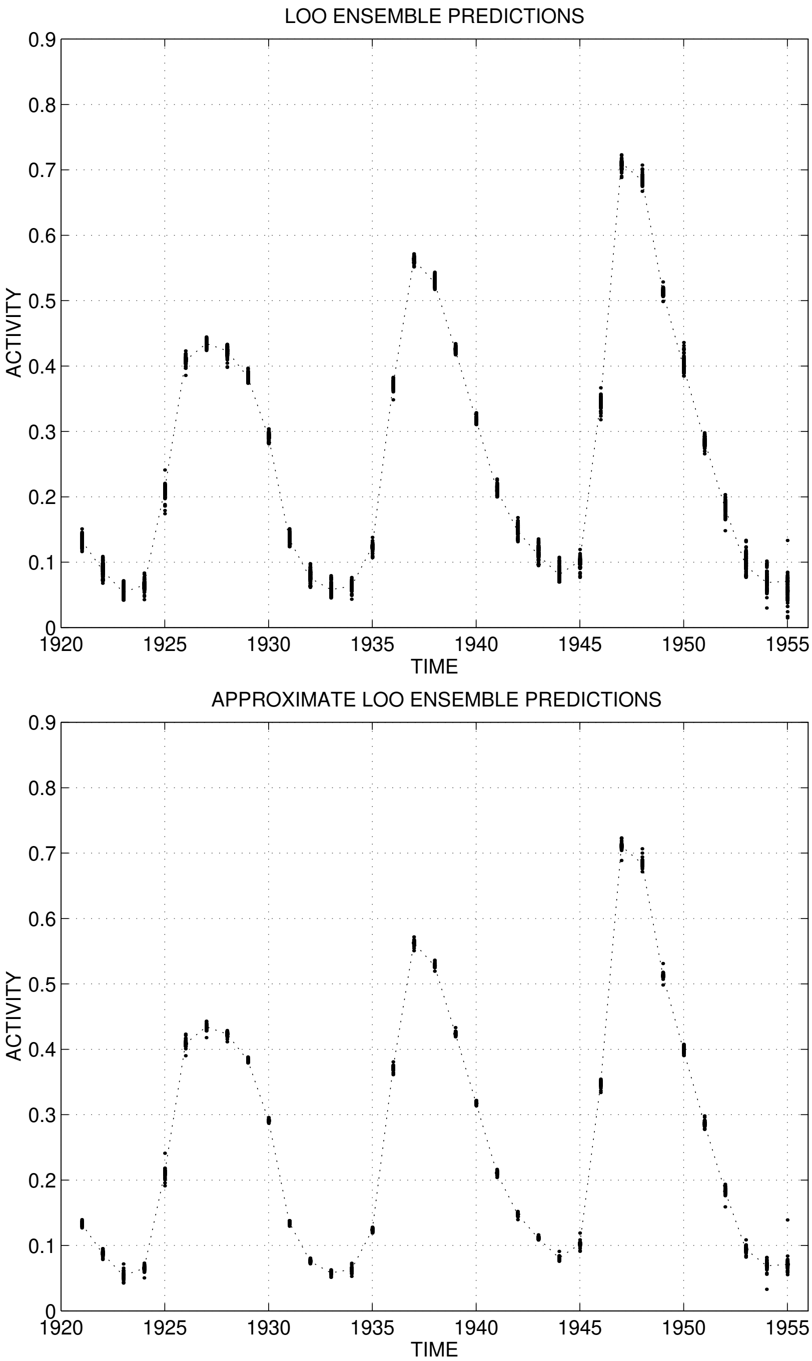

Figure 2: Estimates of the leave-one out ensemble on the sunspot "test set 1" (1921–1955) (above). Estimates of the linearly unlearned ensemble on "test set 1" (1921–1955) (below). The 209 network predictions are indicated by individual heavy dots, and the dotted line indicates the predicted output $F(\boldsymbol{x}, \widehat{\boldsymbol{w}})$.