Cross-Validation with LULOO

Paul Haase Sørensen & Magnus Nørgård
Department of Automation, B326
Technical University of Denmark
DK-2800 Lyngby, Denmark
Phones: (+45) 45253581, (+45) 45253565
Fax: (+45) 45881295
Emails: phs,pmn@iau.dtu.dk

Lars Kai Hansen & Jan Larsen
CONNECT, Department of Mathematical Modelling B349
Technical University of Denmark
DK-2800 Lyngby, Denmark
Phones: (+45) 45253889, (+45) 45253923
Fax: (+45) 45880117
Emails: lkhansen@ei.dtu.dk jl@imm.dtu.dk

***Abstract*— The leave-one-out cross-validation scheme for generalization assessment of neural network models is computationally expensive due to replicated training sessions. Linear unlearning of examples has recently been suggested as an approach to approximative cross-validation. Here we briefly review the linear unlearning scheme, dubbed LULOO, and we illustrate it on a system identification example. Further, we address the possibility of extracting confidence information (error bars) from the LULOO ensemble.**

1 Introduction

Consider nonlinear regression in which the output y is regressed nonlinearly on the input vector $\boldsymbol{x}$, this report concerns 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 [9], who derived estimators for the expected generalization error of regularized networks. These 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 [13], [15] 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.

Recently it was suggested to approximate the evaluation of the ensemble using *linear unlearning* of individual examples [4]. 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 [16], but different from the conventional estimators as FPE [1], Wahba's GCV, and GPE [9]. The proposed method is further related to NCV [10] which approximates leave-v-out cross-validation. We finally discuss the possibility of exploiting the ensemble of networks 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: $\boldsymbol{D} = \left[(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_N, y_N)\right]$. Training on the full set of examples provides a parameter vector denoted by $\widehat{\boldsymbol{w}}$; hence,

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

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,

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

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

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

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 Linear Unlearning LOO (LULOO) 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., [6], [9]) defined as the expected loss on a test sample, further taking the expectation w.r.t. to the training set distribution[5]:

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

where $\langle\cdot\rangle_{(x,y)}$ 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. $\langle\cdot\rangle_D$ denotes the expectation w.r.t. to all training sets of size N [6].

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

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

---

[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.

[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.

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

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

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 [16] 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

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

3.1 The LOO Test Error

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

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

Furthermore, it is often well motivated to invoke the so-called Gauss-Newton approximation for mean square error based problems (see e.g., [8]), 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., [8]) 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,

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

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 [9] and [6].

3.2 Prediction Confidence Intervals

The conditioned generalization error in the case of mean square error learning is given by

$$\sigma_e^2(\boldsymbol{x}) = E\left\{e^2|\boldsymbol{x}\right\} = E\left\{(y - F(\boldsymbol{x}, \widehat{\boldsymbol{w}}))^2|\boldsymbol{x}\right\} \tag{18}$$

which equals the conditional error variance. The prediction confidence interval is thus

$$y \in [\widehat{y} \pm c \cdot \sigma_e] \tag{19}$$

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

where c is a suitably constant. If the error is Gaussian distributed, $c = 1.96$ corresponds to a 95% confidence interval.

One might hypothesize that the individual predictions for a test input $\boldsymbol{x}$ could be used to obtain sensible error bars; however the LOO technique is not directly applicable for estimating the confidence interval, i.e., it is not possible to obtain a plain empirical estimate. Consequently, we turn to an asymptotic ($N \to \infty$) expression.

Error bars were mentioned in [2] who formulated the classical result see [12], p. 193 in a neural network context. The conditional error variance is approximated by:

$$\sigma_e^2(\boldsymbol{x}) = \sigma_n^2 + \boldsymbol{h}(\boldsymbol{x})^\top \langle \delta\boldsymbol{w}\delta\boldsymbol{w}^\top \rangle_D \boldsymbol{h}(\boldsymbol{x}) \tag{20}$$

where $\boldsymbol{h}(\boldsymbol{x}) = \partial F(\boldsymbol{x}, \widehat{\boldsymbol{w}})/\partial \boldsymbol{w}$ and $\delta\boldsymbol{w} = \widehat{\boldsymbol{w}} - \boldsymbol{w}^\circ$ is the weight fluctuations around the true weights $\boldsymbol{w}^\circ$ which minimizies the generalization error.

From the theory of the so-called Jackknife estimator (see e.g., [3], [12]), 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 $\sqrt{N-1}$, thus

$$\langle \delta\boldsymbol{w}\delta\boldsymbol{w}^\top \rangle_D \approx \sum_{\beta=1}^{N} (\Delta\boldsymbol{w}_\beta - \overline{\Delta\boldsymbol{w}})(\Delta\boldsymbol{w}_\beta - \overline{\Delta\boldsymbol{w}})^\top \tag{21}$$

where $\overline{\Delta\boldsymbol{w}} = N^{-1}\sum_{\beta=1}^{N} \Delta\boldsymbol{w}_\beta$.

An estimate of the noise variance[8] σ_n^2 can be obtained by various standard methods, see e.g., [6], [10], [12], or by using

$$\widehat{\sigma_n^2} = N^{-1}\left(E(\widehat{\boldsymbol{w}}) + \frac{1}{2}\mathrm{Trace}[\boldsymbol{H} \cdot \langle \delta\boldsymbol{w}\delta\boldsymbol{w}^\top \rangle_D] \right), \quad \boldsymbol{H} = \frac{\partial^2 E}{\partial\boldsymbol{w}\partial\boldsymbol{w}^\top}(\widehat{\boldsymbol{w}}) \tag{22}$$

4 Example

The leave-one-out cross validation scheme has been evaluated on data obtained from a non-linear simulation model of a pneumatic servo mechanism. A detailed description of the system can be found in [14].

The servo mechanism consists of a linear compressed air cylinder lifting an inertial weight. The cylinder is fed from a system of servo valves which open proportional to a control signal. The servo valve opening characteristics are approximately linear, so that the main nonlinear behavior is due to the cylinder itself. The cylinder chamber's compression dynamics are position dependent and the servo valve flow characteristic is nonlinear. The sampling period of the control system is selected to 0.1 seconds. Furthermore, measurement noise is present in the position measurement data. We seek a neural network model of the position servo, where the system input is the servo valve control input and the output of the system is the piston position.

In order to train a neural network model of the servomechanism, an experiment has initially been performed to collect a set of data. Since the servomechanism contains an integration, the experiment was carried out in closed-loop using a manually tuned PI-controller. A small random signal was added to the control signal to ensure identifiability. A data set of 3000 samples was generated. The relatively large data set is necessary in order to make sure that all of the operating range of the system was contained in the data. Fig. 1 displays the entire data set.

In figure 2 we show how the individual test errors (squared residuals) entering the test error estimate (17) correlate with the results of a full leave-one-out procedure, i.e., the result of training $N = 500$ networks on the corresponding subsets of the training set. To illustrate the capacity of the LULOO ensemble for representing the distribution of predictions on test inputs we show in figure 3 the ensemble error bars evaluated on a portion of the test set.

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 a system identification problem demonstrates the viability of this approach. We have also investigated the LULOO ensemble error bars.

Acknowledgments

This research was partly 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.

[8] We assume that the noise variance is homogeneous.

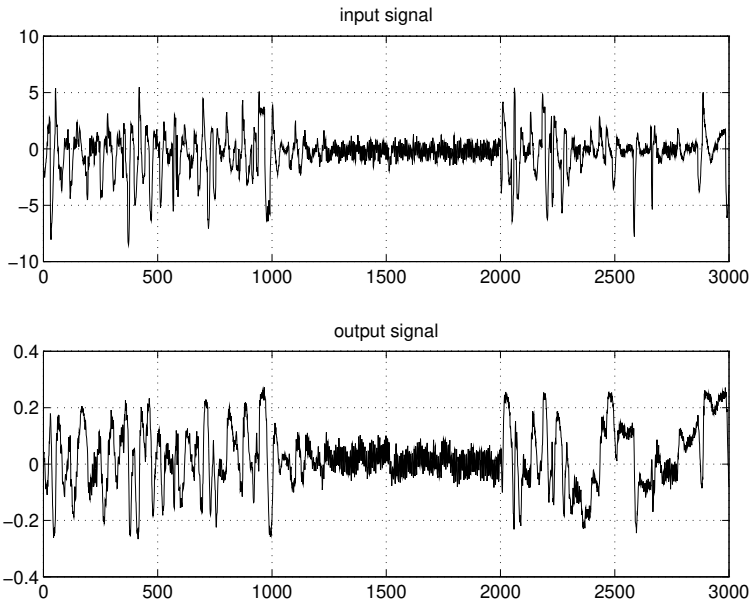

Figure 1: The entire data set used for the identification of the servo system.

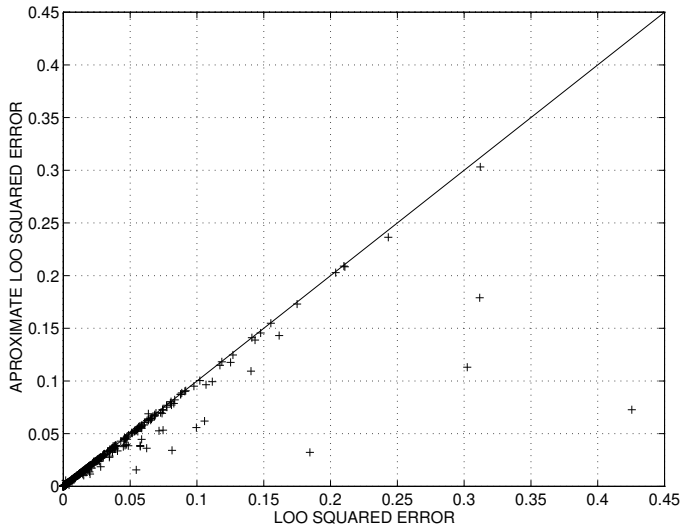

Figure 2: Correlation of individual squared errors (losses) in the leave-one-out (10) with the linear unlearned leave-one-out estimates (17) on the training set.

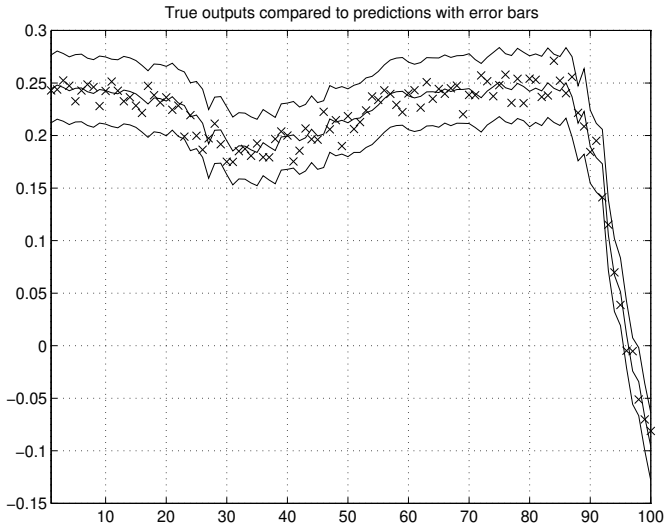

Figure 3: We show the predictions and the LULOO ensemble error bars on part of the test set. Crosses mark the test data values, the central solid line is the prediction and the standard deviation confidence interval is between the two solid lines

References

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

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

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

[4] L.K. Hansen and J. Larsen: "Linear Unlearning for Crossvalidation" *Advances in Computational Mathematics*, to appear (1996).

[5] 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.

[6] 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).

[7] 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).

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

[9] 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).

[10] 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).

[11] 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).

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

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

[14] Sørensen, P.H., P.K. Sinha, K. Al-Mutib : "Identification of a Pneumatic Servo Mechanism using Neural Networks." *Proc. Int. Conf. on Machine Automation*, Tampere, Finland, Vol.2, pp. 499-512, (1994).

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

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

[17] 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).