Prediction Intervals for Neural Network Models

ACHILLEAS ZAPRANIS
Department of Accounting and Finance
University of Macedonia
156 Egnatia St.
PO Box 1591
540 06 Thessaloniki
GREECE

EFSTRATIOS LIVANIS
Department of Applied Informatics
University of Macedonia
156 Egnatia St.
PO Box 1591
540 06 Thessaloniki
GREECE

Abstract: Neural networks are a consistent example of non-parametric estimation, with powerful universal approximation properties. However, the effective development and deployment of neural network applications, has to be based on established procedures for estimating confidence and especially prediction intervals. This holds particularly true in cases where there is a strong culture for testing the predictive power of a model, e.g., in financial applications. In this paper we review the major state-of-the-art approaches for constructing confidence and prediction intervals for neural networks, discuss their assumptions, strengths and weaknesses and we compare them in the context of a controlled simulation. Our preliminary results, which are being presented in this paper, indicate a clear superiority of the combination of the bootstrap and maximum likelihood approaches in constructing prediction intervals, relative to the analytical approaches.

Key-Words: neural networks, confidence intervals, prediction intervals, bootstrap, maximum likelihood.

1 Introduction

The efficient utilization of neural networks, especially in financial applications, requires a confidence measure of their predictive behavior in the statistical sense. Neural network predictions suffer from uncertainty due to: *i*) inaccuracies in the training dataset and *ii*) limitations of the model and the training algorithm. The fact that the training dataset is typically noisy and incomplete, while all the possible realizations of the dependent variable are not available, contributes to the total prediction variance a component known as *data noise variance*, σ_ε^2. Moreover, the limitations of the model and the training algorithm introduce further uncertainty to the network's predictions. It is called *model uncertainty* and its contribution to the total prediction variance is called *model uncertainty variance*, σ_m^2. These two uncertainty sources are assumed to be independent and the *total prediction variance* σ_p^2 is given by the sum of their variances, i.e., σ_ε^2 and σ_m^2 [10].

If the above variance estimates σ_m^2 and σ_p^2 are available, we can form confidence and prediction intervals. In the case of confidence intervals we focus on σ_m^2, since we are interested in the difference between the predicted output $\hat{y}_i$ and the unknown function $\varphi(\mathbf{x}_i)$, which generated the available observations $(\mathbf{x}_i, y_i)$. In the case of prediction intervals we focus on σ_p^2, since we are interested in the difference between the predicted output $\hat{y}_i$ and the realized observation y_i.

In section 2 of this paper, we examine in more detail the difference between confidence and prediction intervals. In section 3, we examine the analytical approach in constructing confidence and prediction intervals, which basically extends the nonlinear regression theory in the nonparametric setting. The Bayesian approach takes a different view of this problem, but since it is basically inappropriate for multidimensional problems, we do not examine it any further here. In section 4, we examine the use of maximum likelihood techniques for providing local error bars (local estimates of a variable noise variance). In section 5, we examine the use of bootstrap, as a typical resampling technique employed by ensemble methods (i.e., *bagging* and *balancing*) for constructing confidence intervals. In

section 6, in the context of a controlled simulation we contrast the synergistic use of bootstrap and maximum likelihood approaches with the analytical approach. Finally, in section 7 we conclude.

2 Confidence Intervals versus Prediction Intervals

Suppose we have a set of observations $D_n = (\mathbf{x}_i, y_i)$, $1 \le i \le n$, that satisfy the nonlinear neural model:

$$y_i = g_\lambda\left(\mathbf{x}_i; \mathbf{w}_0\right) + \varepsilon_i \qquad (1)$$

where y_i is the output of the neural network $g_\lambda(\mathbf{x}_i; \mathbf{w}_0)$ and $\mathbf{w}_0$ represents the "true" vector of the network's parameters $\mathbf{w}$ for the unknown function $\varphi(\mathbf{x}_i)$, which is being estimated by the network. In this setting, it is true that:

$$g_\lambda\left(\mathbf{x}_i; \mathbf{w}_0\right) \approx \varphi\left(\mathbf{x}_i\right) \equiv E\left[y_i \mid \mathbf{x}_i\right] \qquad (2)$$

Initially, we assume that the error ε is i.i.d. with zero mean and constant variance σ_ε^2. The vector $\hat{\mathbf{w}}_n$ is the least squares estimate of $\mathbf{w}_0$ obtained by minimizing the error function:

$$SSE = \sum_{i=1}^{n}\left(y_i - g_\lambda\left(\mathbf{x}_i; \mathbf{w}\right)\right)^2 \qquad (3)$$

The predicted output of the network for the input vector $\mathbf{x}_i$ and the weight vector $\hat{\mathbf{w}}_n$, is:

$$\hat{y}_i = g_\lambda\left(\mathbf{x}_i; \hat{\mathbf{w}}_n\right) \qquad (4)$$

In this framework, a *confidence interval* is concerned with the accuracy of our estimate of the true but unknown function $\varphi(\mathbf{x}_i)$, i.e., it is concerned with the distribution of the quantity:

$$\varphi\left(\mathbf{x}_i\right) - g_\lambda\left(\mathbf{x}_i; \hat{\mathbf{w}}_n\right) \equiv \varphi\left(\mathbf{x}_i\right) - \hat{y}_i \qquad (5)$$

On the other hand, the much more important notion of a *prediction interval* is concerned with the accuracy of our estimate of the predicted output of the network, i.e., it is concerned with the distribution of the quantity:

$$y_i - g_\lambda\left(\mathbf{x}_i; \hat{\mathbf{w}}_n\right) \equiv y_i - \hat{y}_i \qquad (6)$$

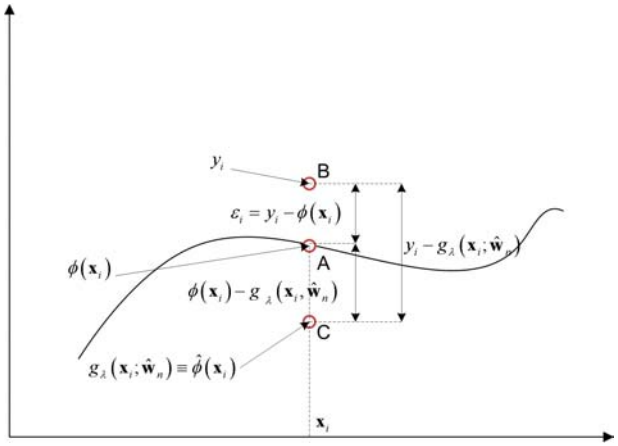

Fig. 1: Relationship between the network's prediction, the observation y_i and the underlying function $\varphi(\mathbf{x}_i)$, which has created the observation with the addition of the stochastic component ε_i.

From Fig. 1 and equations (5) and (6) it follows that:

$$\left(y_i - \hat{y}_i\right) = \left(\varphi\left(\mathbf{x}_i\right) - \hat{y}_i\right) + \varepsilon_i \qquad (7)$$

As we can see from equation (7) the confidence interval is enclosed in the prediction interval.

3 Analytical Methods

Let us denote with $(\mathbf{x}^*, y^*)$ an observation which has not been used for the training of the network (e.g., a future observation), that satisfies the following relationship:

$$y^* = g_\lambda\left(\mathbf{x}^*; \mathbf{w}_n\right) + \varepsilon^* \qquad (8)$$

Our aim is to construct a prediction interval for y^* and a confidence interval for $\varphi(\mathbf{x}^*)$, which is basically the conditional expectation of y^* given $\mathbf{x}^*$. We assume that ε is i.i.d. with zero mean and constant variance σ_ε^2. The vector of the network parameters is being estimated by minimizing the sum of

squared errors (3). For a large number of training patterns and for a neural network which provides a good approximation of the underlying function $\varphi(\mathbf{x}^*)$, the estimated vector $\hat{\mathbf{w}}_n$ will be close to the "true" parameter vector $\mathbf{w}_0$. Then a first order Taylor expansion can be used in order to obtain a linear approximation of the neural network function around $\mathbf{x}^*$:

$$\begin{aligned} \hat{y}^* &\equiv g_\lambda\left(\mathbf{x}^*;\hat{\mathbf{w}}_n\right) \\ &\approx g_\lambda\left(\mathbf{x}^*;\mathbf{w}_0\right)+\mathbf{f}_*^{\mathrm{T}}\left(\hat{\mathbf{w}}_n-\mathbf{w}_0\right) \end{aligned} \tag{9}$$

where:

$$\mathbf{f}_*^{\mathrm{T}}=\left(\frac{\partial g_\lambda\left(\mathbf{x}^*;\mathbf{w}_0\right)}{\partial w_1},\ \ ...,\ \ \frac{\partial g_\lambda\left(\mathbf{x}^*;\mathbf{w}_0\right)}{\partial w_p}\right) \tag{10}$$

Then the 100(1-α)% ***confidence interval*** for $\varphi(\mathbf{x}^*)$ is given by [2]:

$$\hat{y}^* \pm t_{a/2,n-p}\hat{\sigma}_\varepsilon\left[\mathbf{f}_*^{\mathrm{T}}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\mathbf{f}_*\right]^{1/2} \tag{11}$$

The matrix $\mathbf{F}$ is the $(n \times p)$ Jacobian matrix, where n is the number of samples used to estimate $\hat{\mathbf{w}}_n$, p is the number of the network parameters and $\hat{\sigma}_\varepsilon$ is the estimate of the standard deviation of the error term. For a network with m input units and λ hidden units, the number of network weights in (11) is $p = \lambda(m + 2) + 1$.

For a neural network with irrelevant connections (unneeded connections for the task at hand), the number of the parameters is not equal to the number of the network weights. There is an "effective" number of parameters $p_{effective} < p$, which corresponds to an equivalent solution (in terms of SSE) to the initial one. Huang and Ding [2] showed that if the network is trained to convergence, then equation (11) is valid for large training samples, even if we set the number of the network parameters equal to the number of the connections.

Furthermore, if we assume that the error term is normally distributed as $N(0, \sigma_\varepsilon^2)$, then the 100(1-$\alpha$)% ***prediction interval*** for y^* is given by [2]:

$$\hat{y}^* \pm t_{a/2,n-p}\hat{\sigma}_\varepsilon\left[1+\mathbf{f}_*^{\mathrm{T}}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\mathbf{f}_*\right]^{1/2} \tag{12}$$

De Veaux *et al* [3] showed that the above method for computing the prediction interval works well when the training dataset is large. However, when the training dataset is small and the network is trained to convergence the matrix $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ can be nearly singular. In this case, the estimated prediction intervals are not reliable. On the other hand, stopping the training prior to convergence, to avoid overfitting, reduces the effective number of parameters and can lead to prediction intervals that are too wide. A solution to this problem is given by employing connection pruning techniques, such as the *Irrelevant Connection Elimination* scheme (ICE) [13]. After the convergence of the training algorithm to a solution, ICE eliminates the network connections that can be presumed redundant. Another approach is to deactivate irrelevant connections during training using a *weight decay* method [12]. In this case, the error function which is being minimized has the form:

$$\sum_{i=1}^{n}\left(y_i-g_\lambda\left(\mathbf{x}_i;\mathbf{w}\right)\right)^2+c\sum_{i=1}^{p}w_i \tag{13}$$

where $c > 0$ is a weight decay parameter. The ***prediction interval*** in this case becomes [3]:

$$\begin{aligned} \hat{y}^* \pm t_{a/2,n-p}\hat{\sigma}_\varepsilon[1+\mathbf{f}_*^{\mathrm{T}}&\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}+c\mathbf{I}\right)^{-1} \\ &\mathbf{F}^{\mathrm{T}}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}+c\mathbf{I}\right)^{-1}\mathbf{f}_*\,]^{1/2} \end{aligned} \tag{14}$$

4 Maximum Likelihood Methods

In contrast with analytical methods, here we do not assume a constant error variance. Maximum likelihood methods do not impose this restrictive condition, but instead they try to estimate $\sigma_\varepsilon^2(\mathbf{x})$ as a function of $\mathbf{x}$. Just as in the case of analytical methods, we assume that the estimated neural network provides a good approximation of the unknown underlying function, that is the expectation $E[y|\mathbf{x}]$ – see equation (2). From this equation it follows that the variance can be approximated by training a second neural network $f_v(\mathbf{x};\mathbf{u})$ (where v is the number of hidden units and $\mathbf{u}$ is the weight vector of the new network), using squared residuals ($g_\lambda(\mathbf{x};\mathbf{w}_0)$ –

$y)^2$ as the target values. In this case the error function that is being minimized is:

$$\sum_{i=1}^{n}\left\{\left(g_\lambda\left(\mathbf{x}_i;\mathbf{w}_0\right)-y_i\right)^2-f_\nu\left(\mathbf{x}_i;\mathbf{u}_0\right)\right\}^2 \quad (15)$$

and $\hat{\sigma}_\varepsilon^2\left(\mathbf{x}_i\right)\approx f_\nu\left(\mathbf{x}_i;\mathbf{u}_0\right)$.

Rather than using two separate networks, Nix and Weigend [9] proposed a single network with one output for $\varphi(\mathbf{x}) = E[y|\mathbf{x}]$ and another for $\sigma_\varepsilon^2(\mathbf{x})$. Using the sum of squares error function to obtain $\mathbf{u}^*$ in $f_\nu(\mathbf{x};\mathbf{u})$, and thus $\sigma_\varepsilon^2(\mathbf{x})$, is equivalent to using maximum likelihood estimation, and for this reason these methods are called maximum likelihood methods.

5 Ensemble Methods

The rapid increase in computing power of modern computers made realistic the use of neural network ensemble methods for estimating confidence and prediction intervals for neural networks [6], [11].

In these techniques estimates from a number of neural networks are combined to provide generalization performance superior to that provided by a single network. Some of the most popular varieties such as *bagging* and *balancing* [6], use *bootstrap* [5] to generate the training datasets for the ensemble approach. Both of these techniques attempt to "stabilize" high-variance predictors such as neural networks by generating multiple bootstrap versions of the predictor and then combining the outputs of these individual versions to form "smoother" predictions. However, *bagging* and *balancing* differ in the way the predictions are combined. Bootstrap creates a set Ψ of B new datasets, by repeatedly sampling by replacement from the original data set in a random manner:

$$\Psi=\left\{g_\lambda\left(\mathbf{x};\hat{\mathbf{w}}^{(*i)}\right)\right\}_{i=1}^{B} \quad (16)$$

These datasets are being used for training a set of B networks. The output of the network for the input vector $\mathbf{x}$ will be the average of the B network outputs:

$$g_{\lambda,avg}\left(\mathbf{x}\right)=\frac{1}{B}\sum_{i=1}^{B}g_\lambda\left(\mathbf{x};\hat{\mathbf{w}}^{(*i)}\right) \quad (17)$$

To generate confidence and prediction intervals we assume that the neural network provides an unbiased estimation of the true regression $\varphi(\mathbf{x}) \equiv E[y|\mathbf{x}]$. This means that the distribution $P(\varphi(\mathbf{x})|g_{\lambda,avg}(\mathbf{x}))$ is centred on the estimate $g_{\lambda,avg}(\mathbf{x})$. Our assumption here is that the bias component in the confidence interval is minimal in comparison to the variance component. If we also assume that the distribution of $P(\varphi(\mathbf{x})|g_{\lambda,avg}(\mathbf{x}))$ is Gaussian, then the variance of this distribution can be estimated by calculating the variance across the B outputs:

$$\hat{\sigma}_m^2\left(\mathbf{x}\right)=\frac{1}{B-1}\sum_{i=1}^{B}\left(g_\lambda\left(\mathbf{x};\hat{\mathbf{w}}^{(*i)}\right)-g_{\lambda,avg}\left(\mathbf{x}\right)\right)^2$$

By assuming the distribution $P(\varphi(\mathbf{x})|g_{\lambda,avg}(\mathbf{x}))$ is Gaussian, then its inverse distribution $P(g_{\lambda,avg}(\mathbf{x})|\varphi(\mathbf{x}))$ is also Gaussian. While we do not know the distribution of inputs and outputs, the best that we can do is to estimate the distribution $P(g_{\lambda,avg}(\mathbf{x})|\varphi(\mathbf{x}))$ from the distribution $P(g_\lambda(\mathbf{x})|g_{\lambda,avg}(\mathbf{x}))$. So given the observation $(\mathbf{x}^*, y^*)$ using bootstrap we can construct the following ***confidence interval***:

$$g_{\lambda,avg}\left(\mathbf{x}^*\right)\pm t_{a/2,B}\hat{\sigma}_m\left(\mathbf{x}^*\right) \quad (19)$$

where the estimation of the model uncertainty variance σ_m^2 is given from equation (18). However, this variance estimate will be biased. For most input vectors $\mathbf{x}$ will be over-estimated and so the confidence interval (19) will also be over-estimated. Carney *et. al.* [1] proposed a method to deal with this problem. They divide the number of bootstrap networks for the ensemble into M smaller ensembles generating a set of M $g_{\lambda,avg}(\mathbf{x})$ values.

From the set of these values we approximate a more accurate variance measure for the distribution $P(g_{\lambda,avg}(\mathbf{x})|\varphi(\mathbf{x}))$. The variance estimate is not computing only from the M ensemble outputs, while in this case the variance measure itself would be highly variable and unreliable. Instead we form new bootstrap re-sampled sets of the M $g_{\lambda,avg}(\mathbf{x})$ values.

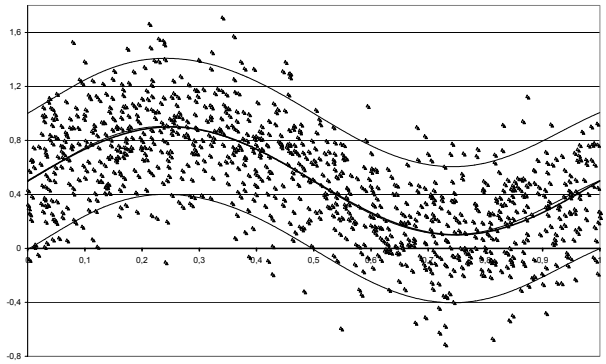

Fig.2 : The 95% prediction interval for normal distribution of y, using the algebraic estimation of σ_p. The number of hidden units is $\lambda = 3$. The synthetic data set was sampled by the Gaussian distribution $N(0.5+0.4\sin(2\pi x), 0.3^2)$. The noise variance is constant. The PICP is 89.6%.

We calculate a variance measure for each of these sets and then calculate an average of these to provide a smoother, lower variance estimate of the variance of the distribution $P(g_{\lambda,avg}(\mathbf{x})|\varphi(\mathbf{x}))$.

This process is not computationally intensive since there are no networks to train. If we assume Gaussian distribution we can construct a ***confidence interval*** in the usual fashion:

$$g_{\lambda,avg}\left(\mathbf{x}^*\right) \pm z_{a/2}\hat{\sigma}_m\left(\mathbf{x}^*\right) \qquad (20)$$

where $(1 - \alpha)100\%$ is the level of confidence.

To estimate prediction intervals, we must compute an estimate of the *prediction variance* σ_p^2, which is given by the sum of the *model uncertainty variance* σ_m^2 and the *data noise variance* σ_ε^2. For the estimation of σ_ε^2 we can use maximum likelihood techniques or analytical methods [1], [6]. For the observation $(\mathbf{x}^*, y^*)$ which has not used for the training of the network, the ***prediction interval*** is given by:

$$g_{\lambda,avg}\left(\mathbf{x}^*\right) \pm t_{a/2,B}\hat{\sigma}_p\left(\mathbf{x}^*\right) \qquad (21)$$

In the next section we compare the aforementioned approaches in the context of a controlled simulation.

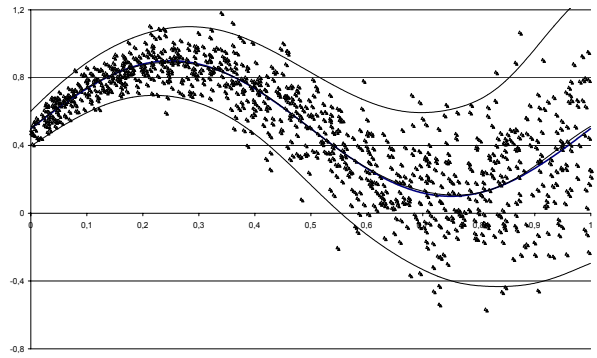

Fig. 3: The 95% prediction interval for normal distribution of y, using the combination of the bootstrap and maximum likelihood estimation of σ_p . The number of hidden units is $\lambda = 3$. The synthetic data set was sampled by the Gaussian distribution $N(0.5+0.4\sin(2\pi x), 0.052+0.1x^2)$. The noise variance is function of x. The PICP is 95.9%.

6 A Controlled Simulation

We generated two synthetic datasets: α) with constant noise term variance and β) with noise term variance which is a function of x. The first dataset was created by employing random sampling from the Gaussian distribution $N(0.5+0.4\sin(2\pi x), 0.3^2)$, while the second dataset was created by employing random sampling from the Gaussian distribution $N(0.5+0.4\sin(2\pi x), 0.052+0{,}1x^2)$.

In both cases a neural model with one hidden layer and $\lambda = 3$ hidden units was selected, on the basis of the Zapranis' and Refenes' framework for "neural model identification, selection and adequacy" [14].

In Fig. 2 we can see the 95% prediction interval for the synthetic dataset α and the analytical approach (12). As we have already discussed the analytical approach can only handle constant error variance. The *Prediction Interval Correct Percentage* (PICP) in this case is 89.6%. Since, its nominal value is 95%, for prediction intervals of good quality we expect the value of PICP to be systematically around 95%.

In Fig. 3 we can see the 95% prediction interval (21) for the synthetic dataset β. The local estimates of the data noise variance, $\sigma_\varepsilon^2(x)$, were obtained by using the ML approach, while the local estimates of the model uncertainty variance, $\sigma_m^2(x)$, were obtained by using the bootstrap approach. The total prediction variance, $\sigma_p^2(x)$, in (21) is simply the

sum of $\sigma_m^2(x)$ and $\sigma_\varepsilon^2(x)$. As we can see the PICP in this case is much improved (95.9%).

7 Summary and Conclusions

Neural networks are a research field which has enjoyed rapid expansion and increasing popularity in both the academic and industrial research communities. However, their efficient utilization requires dependable confidence and especially prediction intervals. In this paper, we examined the state-of-the-art approaches for confidence and prediction interval estimation, and we compared the analytical approach and the synergistic use of the ML and bootstrap approaches for constructing prediction intervals, in the context of a controlled simulation.

The analytical approach we presented here was based on the first order Taylor expansion of the neural estimator. Other analytical approaches are the *delta* estimator (first order Taylor expansion which uses the Hessian matrix) and the *sandwich* estimator (second order Taylor expansion using the Hessian matrix). The sandwich estimator is considered to tolerate better model misspecification. But on the other hand, delta and sandwich estimators require the computation and inversion of the Hessian matrix, a procedure which, under certain circumstances, can be very unstable. In an empirical investigation in [2] it is reported that the use of the Hessian matrix does not improve the accuracy of the estimation. In any case, the analytical approaches can not handle non constant noise variance.

The maximum likelihood approach can be used for estimating local error bars which are a function of x, i.e., $\sigma_\varepsilon^2(x)$. However, these cannot be used for constructing neither confidence, nor prediction intervals, by themselves. Moreover, the ML approach underestimates the "true" noise variance, since the neural network f_v in (15) interpolates between the errors and does not pass through all of them.

The ensemble methods attempt to stabilize the high variance of the neural network predictors using bootstrap to generate multiple versions of the model and then combining the network outputs. The bootstrap approach can be used to obtain local estimates of the model uncertainty variance, $\sigma_m^2(x)$, and thus for constructing confidence intervals. By adding to $\sigma_m^2(x)$ the local noise variance estimate, $\sigma_\varepsilon^2(x)$, we can estimate the total prediction variance, $\sigma_p^2(x)$, and thus obtain a prediction interval from equation (21).

As we have seen that approach gave as PICP equal to 95.9% for the synthetic dataset with noise variance which was a function of x. This compares very favourably to the PICP of 89.6% for the synthetic dataset with constant variance and the analytical approach.

8 References

[1] Carney J. G., Cunningham P., Bhagwan U., "Confidence and prediction intervals for neural network ensembles", *in Proc. of the International Joint Conference of Neural Networks (IJCNN'99), 1999, Washington DC, USA.*

[2] Chryssolouris G., Lee M., Ramsey A., "Confidence interval prediction for neural networks models", *IEEE Trans. Neural Networks*, 7 (1), 1996, pp. 229-232.

[3] De Veaux R. D., Schumi J., Schweinsberg J., Ungar L. H., "Prediction intervals for neural networks via nonlinear regression", *Technometrics*, 40 (4), 1998, pp. 273 – 282.

[4] Dybowski R., "Assigning confidence intervals to neural network predictions", *Neural Computing Applications Forum (NCAF) Conference*, London, 1997.

[5] Efron B., Tibshirani R. J., *An Introduction to the Bootstrap*, Chapman and Hall, 1993.

[6] Heskes T., "Practical confidence and prediction intervals", In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, vol. 9, 1997, pp. 176-182, The MIT Press.

[7] Hwang J. T. G., Ding A. A., "Prediction intervals for artificial neural networks", *Journal of the American Statistical Association*, 92 (438), 1997, pp. 748 -757.

[8] Neal R. M., "Bayesian learning for neural networks", *PhD Thesis, Dept. of Computer Science*, University of Toronto, 1994.

[9] Nix D. A., Weigend A. S., "Learning local error bars for non – linear regression", *Proceedings of NIPS 7*, 1995, pp. 489 – 496.

[10] Papadopoulos G., Edwards P. J., Murray A. F., "Confidence estimation methods for neural networks: A practical comparison", in Proc. *ESANN 2000*, pp. 75-80.

[11] Tibshirani R., "A comparison of some error estimates for neural network models", *Neural Computation*, 8, 1996, pp. 152 – 163.

[12] Yang L., Kavli T., Carlin M., Clausen S., De Groot P. F. M.,"An evaluation of confidence bound estimation methods for neural networks", *ESIT 2000*, 14 – 15 September 2000, Aachen, Germany.
[13] Zapranis, A.D. and G. Haramis, "An algorithm for controlling the complexity of neural learning: The irrelevant connection elimination scheme", *in Proc. The Fifth Hellenic European Research on Computer Mathematics and Its Applications Conference (HERCMA)*, 2001, Athens, 20-22 September.
[14] Zapranis, A.D., Refenes, A-P.N., *Principles of Neural Model Identification, Selection and Adequacy: With Applications to Financial Econometrics*, London, Springer-Verlag, 1999.