Journal of the Nigerian Association of Mathematical Physics
Volume 22 (November, 2012), pp 335 – 340

On R^2 Contribution and Statistical Inference of the Change in the Hidden and Input Units Of The Statistical Neural Networks

[1]Christopher Godwin Udomboso, [2]Tolulope Olayemi James and [3]Mba Obasi Odim

[1]Department of Statistics
University of Ibadan, Ibadan, Nigeria.
[2]Department of Mathematics
Kebbi State University of Science and Technology,
Aliero, Kebbi State, Nigeria.
[3]Department of Mathematical Sciences
Redeemers University, Redemption Camp, Mowe, Nigeria.

Abstract

Determining the number of hidden units for obtaining optimal network performance has been a concern over the years despite empirical results showing that with higher neurons, the network error is reduced. This has led to indiscrimate increase in the hidden neurons, thereby bringing about overfitting. On the other hand, using too few hidden neurons leads to error bias, which can make neural network statistically unfit. In this paper, we developed a model for R^2 for investigating changes in hidden and input units, as well as developed tests that can be used in determining the number of hidden and input units to obtain optimal performance. The result of the analyses shows that there is effect on the network model when there is an increase in the number of hidden neurons, as well as the number of input units.

Keywords: Hidden Unit, Input Unit, R^2 change, F test.

1.0 Introduction

Many researchers have had problem in determining the number of hidden units to obtain optimal network performance (Panchal et al [1], Reed [2]). A test can be used in solving this problem. Empirical results have shown that with higher neurons, the network error is reduced. However, if care is not taken, one may be tempted to increase the hidden neurons indiscriminately. When this happens, overfitting occurs. Also, too few hidden neurons leads to error bias, which sometimes can be very embarrassing, thus making neural network statistically unfit. We approach this problem by assuming that the introduction of additional hidden neuron does not make any difference in the network. Also, input variables can have effect on the performance of the network. This makes the choice of variables of serious importance in neural network. Putting a redundant variable into the network can overfit the coefficient of determination. For this reason, we introduce a test for variable selection, which works by determining if a given input variable contributes to network optimal performance.

The number of hidden neurons affects how well the network is able to separate data. A large number of hidden neuron will ensure correct learning, and the network would be able to correctly predict the data it has been trained on, while with too few hidden neurons, the network may be unable to learn the relationships amongst the data and error will fail to fall below an acceptable level. Swanson and White [3, 4] examined the problem of forward interest rates in predicting future spot rates from a *model selection* perspective in order to shed some additional light to the classical hypothesis testing perspective approach by Mishkin [5]. They considered not only linear models, as in Mishkin [5], but also a class of flexible non-linear functional forms called artificial neural networks. The results reported provided additional support for the hypothesis that forward rates are indeed useful, and suggest that the class of non-linear models which is considered may also prove useful for forecasting interest rates. More specifically, they found that the premium of the forward rate over the spot rate helps to predict the sign of future changes in the interest rate.

[1]Corresponding author: ***Christopher Godwin Udomboso,*** E-mail: kris_ini@yahoo.com, Tel. +234 8037398736

Resop [6] used neural network models having no hidden neurons to those with enough hidden neurons enough to predict to the same accuracy as the statistical models he was comparing with. His work posed a question on what would happen when the hidden neurons are increased above that which predicts exact accuracy as the statistical models. Will there be a difference in the accuracy? In the work of Panchal et al [1], deciding the number of neurons in the hidden layers is seen as a very important part of deciding the overall neural network architecture. Though these layers do not directly interact with the external environment, they have a tremendous influence on the final output. Both the number of hidden layers and the number of neurons in each of these hidden layers must be carefully considered. Using too few neurons in the hidden layers will result in under fitting which makes the network difficult in adequately detecting the signals in a complicated data set. Likewise, using too many neurons in the hidden layers can result in over fitting, and increase in the time it takes to train the network.

This study centres on the Multi-Layer Perceptron (MLP) which happens to be the most commonly used type of ANN (Resop [6]). It has been found to be powerful in terms of model precision in the usage of homogeneous transfer functions (TFs), especially with complex or large data set. The choice of MLP is because it is the only ANN type that allows for statistical inference.

2.0 Theoretical Analysis

The statistical neural network model proposed by Anders [7] is given as

$$y = f(\mathrm{X}, \mathrm{w}) + e \tag{1}$$

where y is the dependent variable, $\mathrm{X} = (x_0 \equiv 1, x_1, \ldots, x_I)$ is a vector of independent variables, $\mathrm{w} = (\alpha, \beta, \gamma)$ is the network weight: 'α' is the weight of the input unit, 'β' is the weight of the hidden unit, and 'γ' is the weight of the output unit, and e_i is the stochastic term that is normally distributed (that is, $e_i \sim N(0, \sigma^2 I_n)$).

For an i^{th} case, it can be shown that (2)

$$\widehat{\mathrm{w}} = \frac{\sum_{i=1}^{n} y_i^* h_i}{\sum_{i=1}^{n} h_i^2} \tag{3}$$

where $\widehat{\mathrm{w}}$ is the estimate of the network model, $y_i^* = y_i - y^0$, $h_i = f'(x_1, \mathrm{w}^0)$, y^0 is the initial derivative of the Gauss-Newton iterative algorithm, and $\mathrm{w}^0 = 0$. $\widehat{\mathrm{w}}$ can thus be expressed as

$$\widehat{\mathrm{w}} = \left(\widehat{\alpha} = \frac{\sum_{i=1}^{n} y_i^* h_{i(\alpha)}}{\sum_{i=1}^{n} h_{i(\alpha)}^2}, \widehat{\beta} = \frac{\sum_{i=1}^{n} y_i^* h_{i(\beta)}}{\sum_{i=1}^{n} h_{i(\beta)}^2}, \widehat{\gamma} = \frac{\sum_{i=1}^{n} y_i^* h_{i(\gamma)}}{\sum_{i=1}^{n} h_{i(\gamma)}^2}\right)$$

So that we can write the estimated function of (1) as

$$y_i^* = h_i \mathrm{w} + e_i \tag{4}$$

Given n observations, we can write (4) in matrix form

$$Y^* = \mathrm{H}\mathrm{W} + U \tag{5}$$

where Y^* is an $\mathrm{n} \times 1$ matrix, H is an $\mathrm{n} \times \mathrm{k}$ matrix, W is an $\mathrm{k} \times 1$ matrix, U is an $\mathrm{n} \times 1$ matrix. We note that $U \sim N(0, \sigma^2 I_n)$.

The sum of squared residual is

$$\sum_{i=1}^{n} U^2 = U'U = Y^{*\prime}Y^* - 2\widehat{\mathrm{W}}'\mathrm{H}'Y^* + \widehat{\mathrm{W}}'\mathrm{H}'\mathrm{H}\widehat{\mathrm{W}}$$

$\widehat{\mathrm{W}}'\mathrm{H}'Y^*$ is a scalar which equals its transpose $Y^{*\prime}\widehat{\mathrm{W}}\mathrm{H}$.

Minimizing the squared residual, we have that

$$\widehat{\mathrm{W}} = \left(\mathrm{H}'\mathrm{H}\right)^{-1}\mathrm{H}'Y^* \tag{6}$$

This implies that

$$\widehat{Y}^* = \mathrm{H}\widehat{\mathrm{W}} \tag{7}$$

We can decompose the y vector of the network model, represented by Y^* into the part explained by the regression and also the unexplained part. We can write Y^* as

$$Y^* = \widehat{Y}^* + \varepsilon = \mathrm{H}\widehat{\mathrm{W}} + \varepsilon \tag{8}$$

Taking the square of (7), we have

$$Y^{*\prime}Y^* = \widehat{\mathrm{W}}'\mathrm{H}'\mathrm{H}\widehat{\mathrm{W}} + \varepsilon'\varepsilon \tag{9}$$

We note that the sum of squares of the Y^* values is

$$Y^{*\prime}Y^* = \sum_{i=1}^{n} Y_i^{*2}$$

and subtracting the mean from the values of Y_i^*,

$$\sum_{i=1}^{n} (Y_i^* - \overline{Y}^*)^2 = \sum_{i=1}^{n} Y_i^{*2} - n\overline{Y}^{*2}$$

$n\overline{Y}^{*2}$ is known as the correction factor, which on subtraction from both sides of (9) gives

$$\left(Y^{*\prime}Y^{*} - n\bar{Y}^{*2}\right) = \left(\widehat{\mathbf{W}}'\mathbf{H}'\mathbf{H}\widehat{\mathbf{W}} - n\bar{Y}^{*2}\right) + \varepsilon'\varepsilon$$

which can also be written as,

$$SST_N = SSR_N + SSE_N \tag{10}$$

where,

SST_N is the Total Sum of Squares of the network in Y^*.
SSR_N is the Explained (Regression) Sum of Squares of the network.
SSE_N is the Unexplained (Error) Sum of Squares of the network.

The performance of a network can be determined by the coefficient of determination, R^2 [3, 4, 8 - 11]. We can further obtain an expression to show the performance of the network whenever the number of hidden or input units or change. It is known that R^2 is a measure of the fit of a model. It is essentially the proportion of the total variation in Y^* that is accounted for by variation in the regression.

That is,

$$R^2 = \frac{\widehat{\mathbf{W}}'\mathbf{H}'\mathbf{H}\widehat{\mathbf{W}} - n\bar{Y}^{*2}}{Y^{*\prime}Y^{*} - n\bar{Y}^{*2}} = 1 - \frac{\varepsilon'\varepsilon}{Y^{*\prime}AY^{*}}$$

where $A = 1 + \frac{n\bar{Y}^{*2}}{Y^{*\prime}Y^{*}}$

Thus

$$R^2 = 1 - \frac{SSE_N}{SST_N} \quad \text{or} \quad R^2 = \frac{SSR_N}{SST_N} \tag{11}$$

Alternatively, we can express the coefficient of determination as the squared correlation between the observed values of Y^* and the predictive values produced by the estimated regression equation.

That is,

$$R^2 = \frac{\left[\sum_{i=1}^{n}(Y_i^{*} - \bar{Y}^{*})(\hat{Y}_i^{*} - \bar{Y}^{*})\right]^2}{\left[\sum_{i=1}^{n}(Y_i^{*} - \bar{Y}^{*})^2\right]\left[\sum_{i=1}^{n}(Y_i^{*} - \bar{Y}^{*})^2\right]} = \frac{(\sum y^{*}\hat{y}^{*})^2}{(\sum y^{*2})(\sum \hat{y}^{*2})} \tag{12}$$

Let R_h^2 denote the coefficient of determination of a network with a given number of hidden units, and $R_{h_{(i)}}^2$ denote the coefficient of determination of a network given a change in the number of hidden units. We note here that the change, $h_{(i)}$, can be an increase, h_{+1} or a decrease, h_{-1}. The error produced by this change is given as

$$e_{h(i)} = y_h^{*} - \hat{y}_{h_{(i)}}^{*}$$

y_h^{*} is the output with a given number of hidden units, and $\hat{y}_{h_{(i)}}^{*}$ is the output given a change in the number of hidden units.

Then, $R_h^2 = \frac{(\sum y^{*}\hat{y}_h^{*})^2}{(\sum y^{*2})(\sum \hat{y}_h^{*2})}$, and $R_{h_{(i)}}^2 = \frac{\left(\sum y^{*}\hat{y}_{h_{(i)}}^{*}\right)^2}{(\sum y^{*2})\left(\sum \hat{y}_{h_{(i)}}^{*2}\right)}$

We denote the difference in the coefficient of determination of the network as $R_{h^*}^2$, and express as,

$$R_{h^*}^2 = R_h^2 - R_{h_{(i)}}^2$$

$$R_{h^*}^2 = \frac{(\sum y^{*}\hat{y}_h^{*})^2\left(\sum \hat{y}_{h_{(i)}}^{*2}\right) - \left(\sum y^{*}\hat{y}_{h_{(i)}}^{*}\right)^2(\sum \hat{y}_h^{*2})}{(\sum y^{*2})(\sum \hat{y}_h^{*2})\left(\sum \hat{y}_{h_{(i)}}^{*2}\right)} \tag{13}$$

In terms of the sum of squares,

(for simplicity, we denote SST_{N_h} as SST_h, SSE_{N_h} as SSE_h, as well as $SST_{N_{h_{(i)}}}$ as $SST_{h_{(i)}}$, and $SSE_{N_{h_{(i)}}}$ as $SSE_{h_{(i)}}$, and the descriptions are equivalent to that used for the outputs)

$$R_h^2 = 1 - \frac{SSE_h}{SST_h} = \frac{SST_h - SSE_h}{SST_h}, \text{ and } R_{h_{(i)}}^2 = 1 - \frac{SSE_{h_{(i)}}}{SST_{h_{(i)}}} = \frac{SST_{h_{(i)}} - SSE_{h_{(i)}}}{SST_{h_{(i)}}}$$

Then,

$$R_{h^*}^2 = \frac{SST_h SSE_{h_{(i)}} - SST_{h_{(i)}} SSE_h}{SST_h SST_{h_{(i)}}} \tag{14}$$

Similarly, for a change in input units, the difference in R^2 contribution is,

$$R_{in^*}^2 = \frac{(\sum y^{*}\hat{y}_{in}^{*})^2\left(\sum \hat{y}_{in_{(i)}}^{*2}\right) - \left(\sum y^{*}\hat{y}_{in_{(i)}}^{*}\right)^2(\sum \hat{y}_{in}^{*2})}{(\sum y^{*2})(\sum \hat{y}_{in}^{*2})\left(\sum \hat{y}_{in_{(i)}}^{*2}\right)} \tag{15}$$

where y_{in}^{*} is the output with a given number of input units, and $\hat{y}_{in_{(i)}}^{*}$ is the output given a change in the number of input units.

Similarly, in terms of sum of squares,

$$R^2_{in^*} = \frac{SST_{in}SSE_{in_{(i)}} - SST_{in_{(i)}}SSE_{in}}{SST_{in}SST_{in_{(i)}}} \tag{16}$$

where the symbols are as described earlier for the outputs.

It should be noted that $\frac{SSR_N/_{k-1}}{SSE_N/_{n-k}} \sim F = \frac{SSR_N(n-k)}{SSE_N(k-1)} = \frac{R^2(n-k)}{(1-R^2)(k-1)}$

which we can also write in terms of R^2 as $F = \frac{R^2(n-k)}{(1-R^2)(k-1)}$

where F is the Fisher statistic.
Without conflict of symbol, we define the following for hidden units:

$$F_h = \frac{SSR_h(n-k)}{SSE_h(k-1)} = \frac{R^2_h(n-k)}{(1-R^2_h)(k-1)}, \text{ and } F_{h_{(i)}} = \frac{SSR_{h_{(i)}}(n-k)}{SSE_{h_{(i)}}(k-1)} = \frac{R^2_{h_{(i)}}(n-k)}{\left(1-R^2_{h_{(i)}}\right)(k-1)}$$

It should be noted that h and $h_{(i)}$ are as defined above in the coefficient of determination.
Now, $F_{h^*} = F_h - F_{h_{(i)}}$
Combining these equations results in

$$F_{h^*} = \frac{(n-k)R^2_{h^*}}{(k-1)(1-R^2_h)\left(1-R^2_{h_{(i)}}\right)} \tag{17}$$

This result can also be shown in terms of the sum of squares.
That is,

$$F_{h^*} = \left(\frac{n-k}{k-1}\right)\left[\frac{SST_{h_{(i)}}}{SSE_{h_{(i)}}} - \frac{SST_h}{SSE_h}\right] \tag{18}$$

Using (8), it implies that (7) can be rewritten as,

$$F_{h^*} = \left(\frac{n-k}{k-1}\right)\left[\left(1-R^2_{h_{(i)}}\right)^{-1} - (1-R^2_h)^{-1}\right] \tag{19}$$

Equations (17), (18) and (19) are different expressions for the F-test for change in hidden units in the neural network.
The hypothesis for this problem is formulated as follows:
$H_0: \beta_h = 0,\ H_1: \beta_h \neq 0$
where β_h is the parameter of the hidden unit, and $h = 1, 2, \ldots, H$.
We reject the null hypothesis if $|F_{h^*}| > |F|$.
In the same, without conflict of symbols, we define the following for input units:

$$F_{in} = \frac{SSR_{in}(n-k)}{SSE_{in}(k-1)} = \frac{R^2_{in}(n-k)}{(1-R^2_{in})(k-1)}, \text{ and } F_{in_{(i)}} = \frac{SSR_{in_{(i)}}(n-k_{(i)})}{SSE_{in_{(i)}}(k_{(i)}-1)} = \frac{R^2_{in_{(i)}}(n-k_{(i)})}{\left(1-R^2_{in_{(i)}}\right)(k_{(i)}-1)}$$

where $k_{(i)}$ is the number of input parameters or variables after a change (that is, increment or decrement).
This follows that

$$F_{in^*} = F_{in} - F_{in_{(i)}} = \frac{(n-1)(k-k_{(i)})R^2_{in}R^2_{in_{(i)}} + [n(k-1)-k(k_{(i)}-1)]R^2_{in^*}}{(k-1)(k_{(i)}-1)(1-R^2_{in})\left(1-R^2_{in_{(i)}}\right)} \tag{20}$$

The hypothesis is set up by assuming that an input variable $\boldsymbol{x}$, has no effect on the output $\boldsymbol{y}^*$. That is,
$H_0: \alpha_i = 0, H_1: \alpha_i \neq 0$
where α_i is the parameter of the input unit, and $i = 1, 2, \ldots, I$.
We reject the null hypothesis if $|F_{in^*}| > |F|$.

3.0 Results And Discussion

This section discusses the results of analysis of the derivations in the previous section. We considered stepwise increase of the hidden neurons from 1 to 10, keeping the input unit constant for the first set of analyses, while the second set considers input units from 2 to 6, keeping the hidden neurons constant.

For change in hidden neurons, the hypothesis is given as follows:
$H_0: \beta_h = 0,\ H_1: \beta_h \neq 0$
The hypothesis simply states that there is no effect in the neural network model when the hidden neuron is increased.

Table 1: Values of R^2 and R^2 Change for change in Hidden Neurons

HN	R^2	HN change	R^2 change
1	0.478346	1 to 2	0.332489
2	0.810835	2 to 3	-0.02534
3	0.785492	3 to 4	0.087695
4	0.873187	4 to 5	0.011567
5	0.884754	5 to 6	-0.12842
6	0.756336	6 to 7	0.092212
7	0.848547	7 to 8	0.046143
8	0.894691	8 to 9	0.084534
9	0.979224	9 to 10	-0.20962
10	0.769609		

From Table 1, it is noticed that except for hidden neurons changes from 2 to 3, 5 to 6, and 9 to 10, which shows negative contribution of the R^2 change, other results shows a positive contribution of the R^2 change.
The inference results in Table 2 shows rejection of hidden neurons contribution to the network model at hidden neurons change from 1 to 2 and 9 to 10. This implies that there was no significant effect on the network model when the number of hidden neurons is increased to 2 and 10 respectively.

Table 2: Values of F and F Change for change in Hidden Neurons

HN	F	HN change	F change	$\|F\|$ change	Decision on H_0
1	7.335835	1 to 2	26.95519	26.95519	*Reject*
2	34.28647	2 to 3	-4.99637	4.99637	*Accept*
3	29.29465	3 to 4	25.79043	25.79043	*Accept*
4	55.08508	4 to 5	6.331858	6.331858	*Accept*
5	61.40357	5 to 6	-36.5849	36.5849	*Accept*
6	24.83205	6 to 7	19.98969	19.98969	*Accept*
7	44.81431	7 to 8	23.14491	23.14491	*Accept*
8	67.96664	8 to 9	309.0962	309.0962	*Reject*
9	377.0628	9 to 10	-350.339	350.339	*Accept*
10	26.72359				

For change in input units, the hypothesis is given as follows:
That is,
$H_0: \alpha_i = 0, H_1: \alpha_i \neq 0$
where α_i is the parameter of the input unit, and $i = 1, 2, \ldots, I$.
The hypothesis simply assumes that increase in the input variable $\boldsymbol{x}$, has no effect on the output $\boldsymbol{y}^*$.

Table 3: Values of R^2 and R^2 Change for change in Input Units

K	R^2	K change	R^2 change
2	0.769609	2 to 3	0.219852
3	0.989462	3 to 4	-0.00063
4	0.988834	4 to 5	0.009836
5	0.99867	5 to 6	-0.01089
6	0.98778		

Table 3 shows that at change from an even input unit to an odd input unit R^2 change is positive, while at change from an odd input unit to even input unit, we have negative R^2 change. This is attested to in Table 4 where there is rejection of the contribution the input units whenever it is increased from even to odd, and acceptance whenever the input unit is increased from odd to input.

Table 4: Values of F and F Change for change in Input Units

K	F	K change	F change	$\|F\|$ change	Decision on H_0
2	26.72359	2 to 3	724.4002	724.4002	*Reject*
3	327.8361	3 to 4	-18.6582	18.6582	*Accept*
4	176.722	4 to 5	1324.805	1324.805	*Reject*
5	938.7028	5 to 6	-837.658	837.658	*Accept*
6	64.66858				

4.0 Conclusion

In this study, we have been able to investigate the contribution of increase in hidden neurons and input units in a given statistical neural network model. The result of the analyses shows that there is effect on the network model when there is an increase in the number of hidden neurons, as well as the number of input units. At some point, indiscriminate increase in the number of hidden neurons is of no use. We notice especially that at increase from even to odd input units, the model has no significance difference. But on the other hand, an increase from odd to even shows a remarkable significance difference

Acknowledgement

This study received much assistance from Professor G. N. Amahia of the Department of Statistics, University of Ibadan, Ibadan, Nigeria, as well as Professor I. K. Dontwi of the Department of Mathematical Sciences, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana.

References

[1] Panchal G., Ganatra A. Kosta Y. P., and Panchal D (2011): Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers. International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April 2011

[2] Reed R. (1993): Pruning algorithms - A survey, IEEE Transactions on Neural Networks, 4, 740–747.

[3] Swanson N. R. and White H. (1995): A model-selection approach to assessing the information in the term structure using linear models and artificial neural networks, Journal of Business and Economic Statistics, 13, 265–275.

[4] Swanson N. R. and White H. (1997): A model-selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks, Review of Economic and Statistics, 79, 540–550.

[5] Mishkin F.S. (1988): The Information in the Term Structure: Some Further Results. Journal of Applied Econometrics, 3, pp. 307-14.

[6] Resop J. P. (2006): A Comparison of Artificial Neural Networks and Statistical Regression with Biological Resources Applications. Published Thesis of the University of Maryland, College Park, USA.

[7] Anders U. (1996): Statistical Model Building for Neural Networks. AFIR Colloqium. Nunberg, Germany.

[8] Suhartono, Subanar, Guritno S. (2003): Model Selection in Neural Networks by using Inference of R2 incremental, PCA and SIC Criterion for Time Series Forecasting. PhD Thesis Extract, Mathematics Department, Gadjah Mada University, Indonesia

[9] Kaashoek J. F. and Van Dijk H.K. (2002): Neural Network Pruning Applied to Real Exchange Rate Analysis, Journal of Forecasting, 21, 559–577.

[10] Gujarati D. N. (1995): Basic Econometrics, Third edition, New York: McGraw Hill International, 10.

[11] Kutner M. H., Nachtsheim C.J. and Neter J. (2004): Applied Linear Regression Models, NewYork: McGraw Hill International.