# Choosing the number of nodes for a neural network via the graphical jump method

Jing Chang and Herbert K. H. Lee

10/27/12

**Abstract**

A graphical tool for choosing the number of nodes for a neural network is introduced. The idea here is to fit the neural network with a range of numbers of nodes at first, and then generate a jump plot using a transformation of the mean square errors of the resulting residuals. A theorem is proven to show that the jump plot will select several candidate numbers of nodes among which one is the true number of nodes. Then a single node only test, which has been theoretically justified, will be used to rule out erroneous candidates. The method has a sound theoretical background, yields good results on simulated datasets, and shows wide applicability to datasets from real research.

Keywords: Neural Network; Model Selection; Jump Plot

## 1 Introduction

Determining the optimal number of hidden units for a neural network is a difficult problem. When there are too many nodes, although the output error is lower on the training data, the errors for predicting novel examples increase, which is called "overfitting". Selecting an

optimal number of hidden nodes allows a good fit to both training data and a hold-out test sample of data. Herein we consider only single hidden layer feedforward neural networks, although the methodology is extensible to other varieties. Thus we consider fitting models of the form: $y_i = \beta_0 + \sum_{j=1}^{k} \beta_j / [1 + exp(-\gamma_{j0} - \sum_{h=1}^{r} \gamma_{jh} x_{ih})] + \epsilon_i$ where $x_{ih}$ is the $h_{th}$ component of the $i_{th}$ sample of the inputs, $y$ is the output, and $\epsilon_i \sim N(0, \sigma^2)$.

A variety of approaches have been proposed to combat overfitting in neural networks, including early stopping (Sarle, 1995), weight decay, and Bayesian methods (Lee, 2004). Here we survey criteria-based methods, and then develop a new criterion based on a graphical interface. Two popular model selection criteria are Akaike's information criterion (AIC) (Akaike, 1973) and the Bayesian Information Criterion (BIC) (Schwarz, 1978) for choosing the best number of hidden units. Another related criterion for choosing the number of nodes is Mallow's $C_p$ statistic: $C_p = \frac{SS_E(p)}{\hat{\Sigma}^2} - n + 2p$, where $SS_E(p)$ is the sum of the squared errors of residuals, $n$ denotes the number of observations, $p$ denotes the number of parameters, and $\hat{\Sigma}^2$ is an unbiased estimate of the variance of an error term (Fogel, 1991). If $\hat{\Sigma}^2$ is known, any model which can estimate regression coefficients unbiasedly and include all critical regressors, has $C_p$ converging to the number of parameters when sample size is large (Gilmour, 1996).

In the context of neural networks, Murata et al. (1994) has studied the theoretical relationship between the training error and the generalization error with regard to the training examples and the complexity of the structure of a neural network, which reduces to the number of parameters in the mathematical expression of the AIC. The Network Information Criterion (NIC) chooses a specification for which the following expression takes a minimum: $NIC = -\frac{1}{T} lnL(\hat{w}) + \frac{tr[BA^{-1}]}{T}$. $T$ is the sample size. $L$ is the likelihood and $w$

denotes all the parameters. The matrix $A$ and $B$ are defined to be $A = -E[\Delta^2 lnL_t]$ and $B = E\left[\Delta lnL_t \Delta lnL_t'\right]$. If the class of models investigated includes the true model, $A = B$ asymptotically. Thus, $tr[BA^{-1}] = tr[I] = K$ is the effective number of model parameters, which is typically less than the nominal number because the parameters are dependent. However, this method can suffer from the problem of rejecting hidden units and choosing the least complex network architectures for model fitting (Anders and Korn, 1999).

In the field of choosing the number of clusters in a mixture model, (Sugar and James, 2003) proposed the jump method from the information theory point of view. By adapting ideas from rate distortion theory to clustering, the theory of the jump method investigates the functional form of the mean square error (MSE) curve in both the appearance and absence of clusters. Furthermore, they demonstrate both theoretically and empirically, that the MSE curve, when transformed to an appropriate negative power, will display a jump, reliably and accurately, at the true number of clusters. However, it is often arduous to designate the transformation parameter directly. (Chang and Sugar, 2008) proposed a graphical tool, christened the "graphical jump method", to ascertain the number of clusters. By changing the transformation parameter, the transformed MSE curve jumps at divergent numbers of clusters, called candidate numbers, amongst which one is the true number of clusters. If the candidate number is smaller than the true number of clusters, at least one cluster will accommodate more than one true cluster and yield a positive result on a test, which is dubbed the "cluster-existence test" and has been theoretically justified.

In this paper, the graphical jump method is extended to solve the problem of choosing the number of nodes for a neural network. Firstly, by using some theoretical results from Murata et al. (1994), a theorem is proved stating that after some boundary conditions are satisfied, there surely exists a transformation power by which the MSE can be transformed to exhibit

a jump at the true number of nodes. A "single node only test", which is also justified theoretically, is used to rule out erroneous candidates. The newly developed method for choosing the number of nodes makes limited parametric assumptions, which can be rigorously theoretically motivated using theorems from Murata et al. (1994), is both simple to understand and implement.

In Section 2, the theory and concrete steps of the graphical jump method are introduced in detail. In Section 3, simulation studies and results are elucidated. Section 4 describes the analysis of a real dataset. Section 5 lays out future research directions.

## 1.1 The introduction of the graphical jump method

Assume that the data are fitted to the following neural network:

$$y_i = \beta_0 + \sum_{j=1}^{k} \beta_j / [1 + exp(-\gamma_{j0} - \sum_{h=1}^{r} \gamma_{jh} x_{ih})] + \epsilon_i \tag{1}$$

where $\epsilon_i \sim N(0, \sigma^2)$. The MSE $d_k$ equals the variance of residuals generated by fitting the neural network model with $k$ nodes.

Assuming that the dataset is fitted by model (1), the graphical jump method has the following four basic steps for choosing the best number of nodes:

a. Calculate MSE $d_k$ for $K = 1...k_{max}$ by fitting the neural network model using $k$ nodes.

b. Choose a positive number $v > 0$, called the transformation power.

c. Calculate the jump score associated with $k$ nodes $J_k = d_k^{-v} - d_{k-1}^{-v}$, with $J_1 = d_1^{-v}$.

d. The best number of nodes is the number $k$ with the highest $J_k$.

To give a simple illustration of how the graphical jump method works, a simulated dataset

is generated with 100 observations from a $gamma(20, 40)$ distribution with supplementary standard normal noise and the true number of nodes of 4. The response, $Y1$, is the aggregate of the 4 different nodes: $Y2$, $Y3$, $Y4$ and $Y5$ (Figure (1)), whose coefficients are manifested at the top of the plots. The first node and the third node have active declining regions in the range of (-1.5, -0.3) and (0.5 ,1), while the second node and the fourth node have active increasing regions in the range of (-0.3 ,0.4) and (1.1, 1.9). Consequently, the final response variable, $Y1$, has 4 disconnected active regions with adjacent active regions in totally opposite directions, which necessitate 4 nodes to provide the best fit.

A graphical visualization is provided by Figure (2) which plots the successive jumps in the transformed MSE. In the plot, the possible number of nodes ranges from 1 to 10. The lower left plot of Figure (2) shows a jump at 4, which is the true number of nodes. Intuitively, this jump occurs because of the sharp increase in the jump scores that results from not modeling noise using additional nodes. Adding subsequent neural network nodes can not decrease, but increase the MSE of residuals and thus has a smaller contribution to the jump score. When the transformation powers change from 0.4, 1, 2 to 5, the highest jump scores occur at nodes of 1, 1, 4 to 10. A jump plot (Figure (3)) is generated to elucidate the functional relationship of the number of nodes selected versus transformation power used. As the transformation power increases from 0 to 20, the number of nodes selected changes from 1, 4 to 10.

As the transformation power $y$ approaches 0, the jump score for 1 node $J_1^{-y}$ approaches 1 while the jump score for $k$ nodes $J_k^{-y} - J_{k-1}^{-y}$ approaches 0. Thus the highest jump occurs at node one. As the transformation power approaches an enormous number, the jump score for 10 nodes $J_{10}^{-y} - J_9^{-y}$ will be the largest one. This is because the MSE of residuals is a decreasing function of the number of nodes, $J_{10}$ is smaller than $J_k$ for $1 \leq k \leq 10$. As $y$

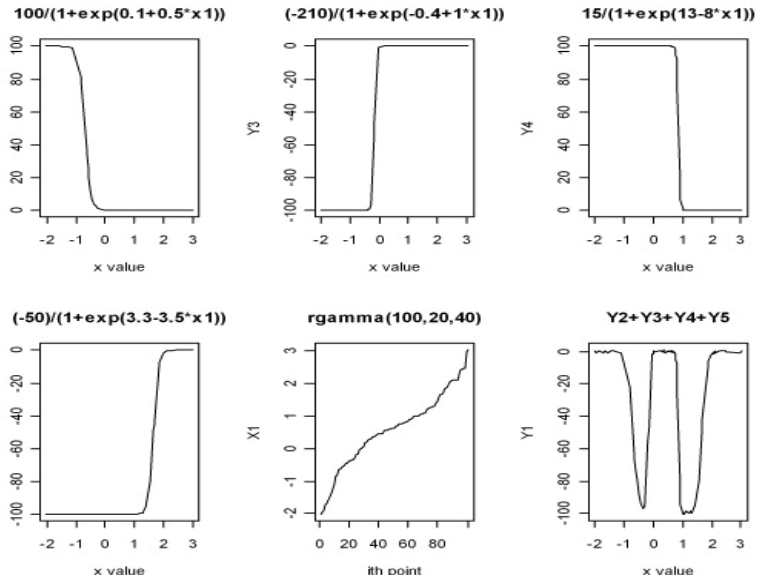



**Figure (1).The shape of each node for a simulated dataset. The first to the third nodes are described by the upper left, upper middle and upper right plots. The lower left plot illustrates the fourth node. The lower middle plot shows the $x$ values. The lower right plot illustrates the sum of the $4$ nodes.**

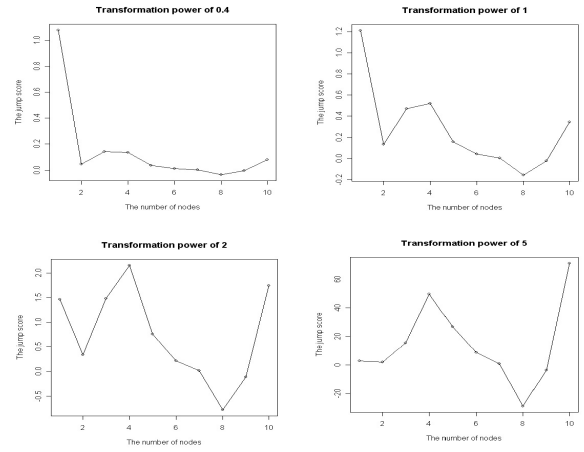
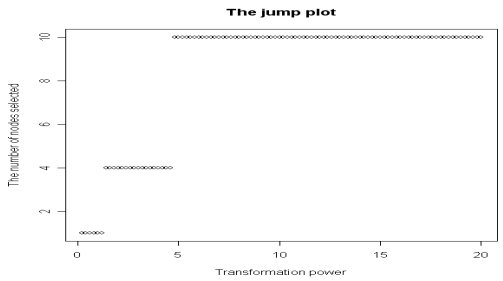



**Figure (2). Plots of the jump scores versus the number of nodes under different transformation powers for a simulated dataset.**

approaches infinity, $J_{10}^{-y}$ increases much faster than $J_k^{-y}$ for $1 \leq k \leq 10$. This leads to the fact that above some point of y, $J_{10}^{-y} - J_9^{-y}$ will be the highest jump score.

## 1.2 Summary of the Graphical Jump Method

By utilizing the graphical property of the jump plot, a graphical jump method is developed to choose the number of nodes more efficiently. Since a jump surely occurs at the best number of nodes, by choosing the candidates as the number of nodes where the jumps occurs, the range of candidate numbers of nodes are significantly narrowed down. When implementing the graphical jump method, a jump plot is sketched at first to illustrate all the candidate numbers of nodes, assuming there are a total of $g$ candidates. Secondly, organizing the $g$ candidate numbers of nodes from small to large, the dataset is modeled with these candidate numbers of nodes sequentially to produce $g$ consecutive sets of residuals, for which $g$ jump plots are produced to see whether each plot contains candidates with more than one node. The key idea is that if the best model has been found, there is nothing left to model in the residuals, whereas if there are not yet enough nodes in the model, then one can find this signal in the residuals by fitting one or more nodes to the residuals.




**Figure(3) The jump plot for the simulated dataset.**

If the candidate number of nodes is less than or equal to the best number of nodes minus two, then it can not account for the total variability of the dataset. The corresponding residuals will encompass variability that has to be explained by additional nodes, and thus it will test negative on the single node only test. This is caused by the fact that if they had produced positive result, then the total number of nodes needed to model the data is the candidate number of nodes plus one, which is less than the best number of nodes and contradicts the original assumption of the true size of the network.

Nevertheless, if a candidate number of nodes is adjacent to the best number of nodes, then the residuals of both numbers of nodes will present positive results on the "Single node only test". As a result, the first candidate number of nodes without an adjacent lower candidate, that demonstrates that its residuals need to be explained by a single node only, is the best number of nodes. If two earliest consecutive numbers of nodes, $N_i$, $N_i + 1$, both indicate that their residuals only need one node to count for the total variability, then the ratios of $d_{N_i+1}/d_{N_i}$ and $d_{N_i+2}/d_{N_i+1}$ are calculated. If $N_i$ is the best number of nodes, $d_{N_i+1}$ should be much larger than $d_{N_i}$ since the model changes from modeling the main effects to modeling the noise at this point. Therefore, the ratio of $d_{N_i+1}/d_{N_i}$ should be the larger one. In practice, the larger one of the two ratios corresponds to the true number of nodes.

# 2 Method

## 2.1 The Two Theorems

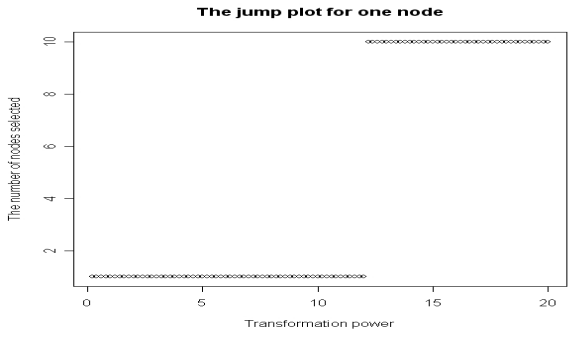
The following theorem provides an asymptotic result of the shape of the MSE curve after transformation, and thus it provides a theoretical explanation for the graphical jump method.

**Theorem 1:** Assume the dataset $y$ follows model (1). Define $K_{max}$ as the maximum number of nodes investigated, $t$ as the sample size. Assume the dataset is composed of $G$ nodes and that the likelihood ratio test of $H_0$ : the dataset is composed of $G$ nodes, versus $H_A$: the dataset is composed of less than $G$ nodes, yields a positive result. Define $m^*$ as the number of parameters in one node. If $\frac{t}{2Chisq(m^*,0.05)} < v < \frac{t^2}{6m^*(K_{max}-G)}$, as $t \longrightarrow \infty$, the jump method always selects the true number of nodes as a candidate (proof in appendix).

By the aforementioned theorem, when changing the transformation power from small to large, a jump surely exists at the best number of nodes. Nonetheless, sometimes a correct transformation power is difficult to identify due to the deviation of the dataset from the theoretical model, measurement error caused by experiment operators and random errors, etc. Ergo, the graphical property of the jump plot is explored to provide a tool for choosing the number of nodes based on the jump plot. The nodes at which jumps appear are called candidate numbers of nodes, $N_1, N_2, ..., N_g$. Theorem 2 provides a theoretical foundation for examining whether a dataset needs to be fitted by the neural network model with more than one node, dubbed "Single node only test". It is not onerous to conjecture that, if $N_i$ is the best number of nodes, after fitting the data with $N_i$ nodes, the residuals follow a standard normal distribution and result in a positive test in the "Single node only test". Consequently, the candidate numbers of nodes in the jump plot will be fitted to the dataset with the neural network models one by one to figure out which residuals will give positive results in the test.

**Theorem 2:** Assume that $y \overset{iid}{\sim} N(0,1)$. Define $K_{max}$ as the maximum number of nodes investigated, $t$ as the sample size, and $m^*$ as the number of parameters in one node. If $v < \frac{t^2}{4K_{max}m^*}$, as $t \longrightarrow \infty$, the jump plot will only select one node as the candidate (proof in appendix).

Theorem 2 is illustrated by Figure (4), which is a jump plot generated by a dataset simulated from a standard normal distribution. Theorem 2 demonstrates that the jump plot designates one node as the solitary candidate in a long range of transformation power. However, the length of this range depends of the actually dataset generating the jump plot. Empirically, this range would always include (0, 2), i.e, the length of this range would be larger than 2 units for the majority of the datasets. Therefore, if the true number of nodes is one, the jump plot would select one as the candidate number of node in a range, including (0, 2), as the solo candidate. Nevertheless, by Theorem 1, if the true number of nodes is larger than one, the jump plot would select a candidate number of node bigger than one within this range. For instance, in Figure (3), the dataset is composed of 4 nodes and 4 nodes are pinpointed in the the jump plot within the range of (1.2, 4.6), which overlaps the range (0, 2) at (0.8, 2). Therefore, in all, if the jump plot picks a candidate number of nodes (larger than one) within the range of (0, 2), then the dataset is deemed as composed of more than one node and one node otherwise. As a result, in Figure (4), the dataset is regarded as composed of one node.




**Figure(4) The jump plot for the simulated dataset.**

## 2.2 Connection between the theory of the NIC criterion and the graphical jump method

There are connections between the theory of the NIC criterion and the graphical jump method. Consider a stochastic system which has an input vector $x \in R^k$ and produces an output vector $y \in R^l$. An input vector $x$ is generated according to the probability $q(x)$ and an output vector $y$ is generated according to a conditional probability $q(y|x)$ specified by $x$. $q(x, y)$ is the product of $q(x)$ and $q(y|x)$. A network is considered to have a conditional distribution $p(y|x, \theta)$, where $\theta \in R^m$ is an m-dimensional parameter vector that describes the network, such as a set of weights and thresholds. $q$ represents the true distribution of $p(y|x, \theta)$. Let $f(x, y)$ be the density function for neural network models. The dataset is assumed to obey the model: $y = f(x, \theta) + \xi(x)$, where $\xi(x)$ is noise and $E(\xi(x)) = 0$. To calculate the goodness of fit of a neural network, a discrepancy function $D(q, p(\theta))$ is designed to measure the difference between $q(y|x)$ and $p(y|x, \theta)$. $d(x, y, \theta)$ is a loss function, typically, it could be square error loss or negative log likelihood. The square error loss is defined as:

$$d(x, y, \theta) = (\|y - f(x, \theta)\|)^2$$

The discrepancy function we use is :

$$D(q, p(\theta)) \equiv \int d(x, y, \theta) q(x) q(y \mid x) dx dy$$

In order to minimize the discrepancy function, the true probability distribution $q(x, y)$ of the target system needs to be known. However, it is not possible to identify $q(x, y)$ in reality. Frequently, the empirical distribution $q^*(x, y) \equiv \frac{1}{t} \sum_{i=1}^{t} \delta(x - x_i, y - y_i)$ is used instead. It is well known that the empirical distribution approximates the true distribution $q(x, y)$ in the weak sense when sample size is large, and hence it is reasonable to evaluate the network

model by $q^*(x,y)$ instead of $q(x,y)$. $p(\theta)$ is the distribution of $\theta$. For square error loss,

$$D(q^*, p(\theta)) \equiv \sum_{i=1}^{t} \|y_i - f(x_i, \theta)\|^2$$

The following is an important result on which the graphical jump method depends:

$$D(q, p(\theta_{opt})) = min_\theta D(q, p(\theta))$$

where $\theta_{opt}$ is the value of $\theta$ when $D(q, p(\theta))$ achieves the minimum.

$$D(q^*, p(\theta^*)) = min_\theta D(q^*, p(\theta))$$

where $\theta^*$ is the value of $\theta$ when $D(q^*, p(\theta))$ attains the minimum. Let

$$R_{opt} \equiv V_q[\nabla d(x, y, \theta_{opt})]$$

i.e. for the true distribution of $\theta$, $R$ is the variance of the first derivative of n $d(x, y, \theta_{opt})$ with respect to $\theta$. Let $m$ be the number of parameters in the model, $R_{opt}$ is of $m \times m$ dimensions. Let

$$Q_{opt} \equiv E_q[\nabla\nabla d(x, y, \theta_{opt})]$$

i.e. for the true distribution of $\theta$, $Q_{opt}$ is the expectation of the second derivative of $d(x, y, \theta_{opt})$ with respect to $\theta$. Let $m$ be the number of parameters in the model, $Q_{opt}$ is of dimension $m \times m$.

Theorem 3: The average discrepancy between the system $q(x,y) = q(y|x)q(x)$ and the machine $p(y|x, \tilde{\theta})$ learned from t examples is given by $< D(q, p(\tilde{\theta})) >=< D(q^*, p(\tilde{\theta})) > + \frac{1}{t} tr(R_{opt} Q_{opt}^{-1}) + O(t^{-\frac{3}{2}})$,

where $< . >$ denotes the expectation with respect to the distribution of $\tilde{\theta}$, the parameter after sufficient learning of $\theta$ with the machine, and $\theta^*$ (Murata, et al., 1994, page 868).

Theorem 1 studies the difference of $< D(q, p(\tilde{\theta})) > and < D(q^*, p(\tilde{\theta})) >$ in terms of the ensemble average of training sets. Nevertheless, when using this criterion for model selection, we need to evaluate $< D(q, p(\tilde{\theta})) >$ and $< D(q^*, p(\tilde{\theta})) >$ for one particular training set. A "subset" for a single layer feedforward neural network is defined as following: if the first model has fewer hidden units than the second model, then it is deemed a submodel of the second one. The submodel can be obtained from the full model by setting the connection weights and thresholds of the extra units equal to 0. $< D(q, p(\tilde{\theta})) >$ can be decomposed into the following:

Let $M_i = \{p_i(y|x, \theta_i); \theta_i \in R^{m_i}\}$ be a hierarchical series of models: $M_1 \subset M_2 \subset M_3 \subset ...$, where $M_i$ is a submodel of $M_j$,$(i < j)$.

For only one training set,

$$D(q, p(\widetilde{\theta})) = D(q*, p(\widetilde{\theta})) + U\frac{1}{\sqrt{t}} + \frac{1}{t}tr(R_{opt}Q_{opt}^{-1}) + O(t^{-3/2}) \qquad (2)$$

where $U = \sqrt{t}D(q, p(\theta_{opt})) - D(q^*, p(\theta_{opt}))$, is a random variable of order 1 with zero mean. $U$ is common to all the models within a hierarchical structure, such as single layer neural network models with the same dimensions for the input and the output vectors, see Murata et al. (1994) (page 869).

Obviously, the discrepancy $D(q, p(\widetilde{\theta}))$ achieves the minimum at the best numbers of nodes,

resulting in a sequence of inequality equations composed of the right side of formula (2). For negative log likelihood, $R = Q$, which makes $tr(R_{opt}Q_{opt}^{-1})$ reduce to the number of parameters in the corresponding neural network. $U$ is common to all the models within a hierarchical structure. $D(q*, p(\widetilde{\theta}))$ can be expressed as a function of MSE of residuals under different numbers of nodes. Thus, the inequality equations reduce to the inequality relationship of MSE of residuals under different numbers of nodes. By utilizing those inequality relationships of MSE of residuals and with the help of a Taylor expansion, the necessary conditions of the jump method, which are inequality relationships of MSE after transformation, are proved and two related Theorems are established. The proofs of the two Theorems are attached in the appendix.

# 3 Simulation Study

## 3.1 One dimensional data

Simulation studies are first performed with one $x$ variable, and with 100, 200, and 300 observations. For one $x$ variable, data are simulated with 4 nodes, which have distinct active regions as shown in Figure (1). The $x$ variable is generated from a gamma distribution with shape parameter of 20 and rate parameter of 40, plus a noise variable with standard Gaussian distribution. From nodes 1 to 4, the $y$ variables are generated using the following formula:

1. $y_1 = \frac{100}{1+\exp(7+10\times x)}$ 2. $y_2 = \frac{-100}{1+\exp(5+30\times x)}$

3. $y_3 = \frac{100}{1+\exp(-30+35\times x)}$ 4. $y_4 = \frac{-100}{1+\exp(-20+12\times x)}$

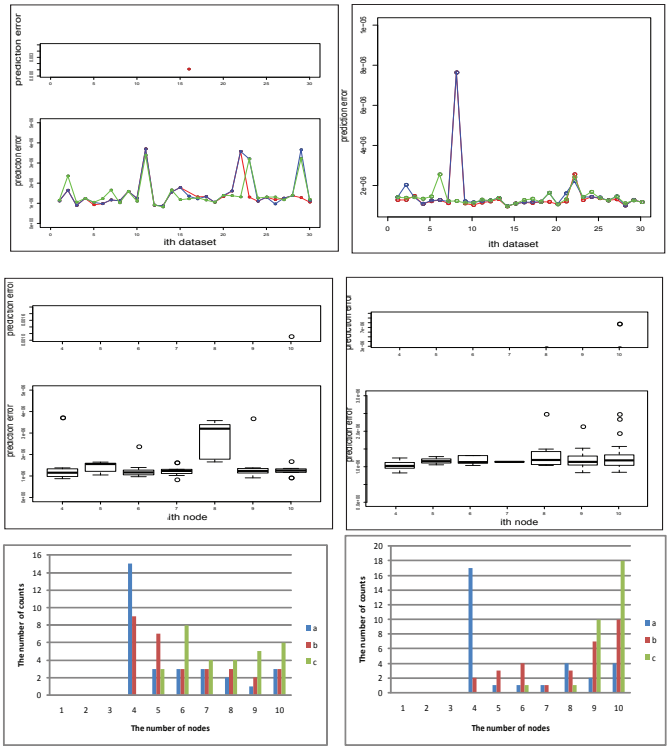
The final $Y = y_1 + y_2 + y_3 + y_4 + \epsilon$, where $\epsilon \sim N(0, 0.1)$.

Table (1) demonstrates simulation results for one dimensional data for sample sizes of 100, 200 and 300. The first 3 rows are the percentages of the number of nodes correctly chosen for

each method and each scenario. The last 3 rows are the mean and standard deviation of the prediction errors for each method for the 30 datasets in each scenario. The three methods for comparison are the graphical jump method, cross validation and BIC. The goals of the graphical jump method and cross validation are different. The former is for the purpose of model selection, which corresponds to the top 3 rows of table, and the latter is for the purpose of prediction, which corresponds to the bottom 3 rows of the table. The predictive accuracy is obtained by comparing the prediction results to the known true response values in the simulated examples. The total number of datasets is 30 for all the scenarios in this Chapter. The graphical jump method performs the best in choosing the correct number of nodes. The percentages of the number of nodes correctly chosen for the graphical jump method are well above those of the other two methods. For the prediction errors,the graphical jump method leads the other two methods for 300 sample size scenario, leads BIC for 200 sample size scenario and performs the worst for 100 sample size scenario. However, for 100 sample size scenario, there is an outlier in the prediction errors of the graphical jump method, as shown in Figure (5). This outlier is $1.11 \times 10^{-3}$. Since all the other prediction errors in this scenario are around $10^{-6}$, this outlier dramatically elevates the mean of the prediction errors of the graphical jump method, which should be $1.41 \times 10^{-6}$ without the outlier and lower than the means of the prediction errors of the other two methods. For 200 sample size scenario, there is also an outlier, $7.70 \times 10^{-6}$, in the prediction errors of the graphical jump method. Without this outlier, the mean should be $1.13 \times 10^{-6}$, which will be the lowest among the three.

Figure (5) illustrates the results of the simulation studies for one dimensional data. The upper left plot is the scatter plot of the prediction errors versus the $i_{th}$ dataset (There are 30 simulated datasets for each scenario and for each dataset, there is a prediction error generated for each method). There are two parts in the plot, the first part has y axis ranges from

$5 \times 10^{-6}$ to 0.005 and x-axis indicating the sequence of the 30 dataset. The second part has $y$ axis ranges from 0 to $5 \times 10^{-6}$ and the same $x$ axis as the first part.

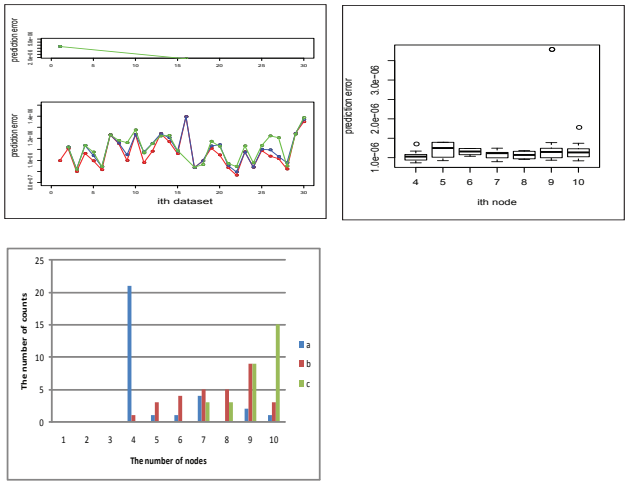



**Figure(5) Simulation study results for one dimensional data of 100 and 200 sample sizes.**

The two parts display the different appearance of the scatter plot in the corresponding ranges of the $y$ axes. In the second part of the plot, the major part, the red, blue and green curves connect points of prediction errors for the graphical jump method, BIC and cross validation separately. This color representation is used for all the scatter plots in this Chapter. In the first part of the plot, only the outlier $1.11 \times 10^{-3}$ appears.

**Table (1). The results of simulation studies for one dimension scenario.**

| number of datapoints | 100 | 200 | 300 |
|---|---|---|---|
| graphical jump method (percentage) | 50 | 56.7 | 70 |
| BIC (percentage) | 30 | 6.7 | 3.3 |
| 10 (percentage)fold cross validation | 0 | 0 | 0 |
| graphical jump method (mean(sd))) | 3.82e-05(0.00020) | 1.34e-06 (1.23e-06) | 1.07e-06(1.34e-07) |
| BIC (mean(sd)) | 1.55e-06(8.30e-07) | 1.42e-06 (1.22e-06) | 1.19e-06(5.07e-07) |
| 10 fold cross validation (mean(sd)) | 1.48e-06 (6.70e-07) | 1.25e-06 (3.51e-07) | 1.22e-06 (5.13e-07) |




**Figure(6) Simulation study results for one dimensional data of sample size of 300.**

The middle left plot contains several box plots. Similarly as that in the upper left plot, it has two parts with the y-axis ranges from $5 \times 10^{-6}$ to 0.002 and from 0 to $5 \times 10^{-6}$ separately. To draw the box plot, the prediction errors for all the three methods in this scenario are com-

bined and then categorized by their corresponding number of nodes chosen. For example, for all the datasets where the graphical jump method chooses the number of nodes of three, the prediction errors generated by the graphical jump method are combined into one group. Similarly, the prediction errors whose corresponding number of nodes chosen are three for the BIC method and the cross validation method are categorized into the same group as that for the graphical jump method. For other numbers of nodes, data are categorized similarly. In the box plot, as the number of nodes chosen increases, the mean value of the box plot also increases generally. The fact that the box plot of the 8th node has the highest mean value may be caused by random effect. The outlier is located at the bottom of the first part of the plot.

The lower left plot is a $2-D$ histogram generated by excel. The $x$-axis indicates the number of nodes chosen. The $y$-axis indicates the number of datasets which choose the corresponding number of nodes in the $x$-axis. For all the $2-D$ histograms in this Chapter, the blue, red and green cylinders compose the histograms for the graphical jump method, BIC and cross validation separately. For the lower left plot, the graphical jump method chooses the correct number of nodes, i.e. four nodes, with the highest rate. BIC chooses four nodes and five nodes with the highest rates.

The right three plots are generated for sample size of 200. They are similarly as that for the left three plots except that the upper right plot is a normal scatter plot. For most part of the upper right plot, the red curves are the lowest among the three, which means that for most of the datasets, the prediction errors generated by the graphical jump method are the lowest. For the plot in the middle right, the outlier is located in the first part of the plot and the mean values of the box plots in the second part have increase trend along the x-axis. The $2-D$ histogram in the lower right shows that the graphical jump method chooses 4

nodes with the highest rate while BIC and cross validation choose 9 nodes and 10 nodes with the highest rates.

Figure (6) also has three plots. The upper left, upper right and lower left plots contain scatter plots, box plots and $2-D$ histograms separately. The scatter plot is also composed of two parts with the $y$-axes ranges from 0 to $2 \times 10^{-6}$ and from $2 \times 10^{-6}$ to $5 \times 10^{-6}$. For most of the scatter plot, the red curve is the lowest, which means the graphical jump method generates the lowest prediction errors most of the time. The upper right plot shows that the mean value of the box plot increases with the number of nodes chosen generally. For the $2-D$ histogram, the graphical jump method chooses 4 nodes as the best number of nodes most of the time. However, BIC and cross validation choose 9 or 10 nodes with the highest percentages, which is similar to that of 200 sample size scenario. This is because for a small sample size, such as the 100 sample size scenario, 4 nodes can explain most of the variability of the dataset. Adding more nodes can not decrease the MSE of residuals too much. Hence BIC and cross validation will choose a node count close to 4 nodes. When the sample size becomes larger, there are more data points. 4 nodes is not enough to explain most of the variability of the dataset. Adding more nodes makes the MSE of the residuals decrease a lot. Hence BIC and cross validation will choose 9 and 10 nodes as the best number of nodes with high rates.

## 3.2 Three dimensional dataset

Finally, simulations are performed for three variables. A cube described by the three variables is generated. For each node, the active region is located at a corner of the cube (See Figure (7)). The faces labeled by "F" are the front faces. Figure (7) demonstrates the positions of the active regions relative to the front faces. For the aggregate of the four nodes,

the active regions are located at the 4 different corners of the cube. The nodes are then generated as following: firstly, $n \times 100$ observations with gamma distribution, with shape parameter 20 and rate parameter 40, plus a standard Gaussian noise term, are generated. After sorting them from small to large, the $n \times 100$ observations are divided into $n$ subgroups with consecutive 100 observations being in the same subgroup. Therefore, $n$ subgroups are produced. $n$ $x_1$ observations are produced with each observation equaling to the mean of each subgroup. $n$ $x_2$ observations and $n$ $x_3$ observations are generated similarly. Then $n^3$ observations, $X_1$,...,$X_{n^3}$, are produced with the 3 coordinates being a combination of each $x_1$ (first coordinate), $x_2$ (second coordinate) and $x_3$ (third coordinate) values. The first to fourth nodes are simulated by the following formula:

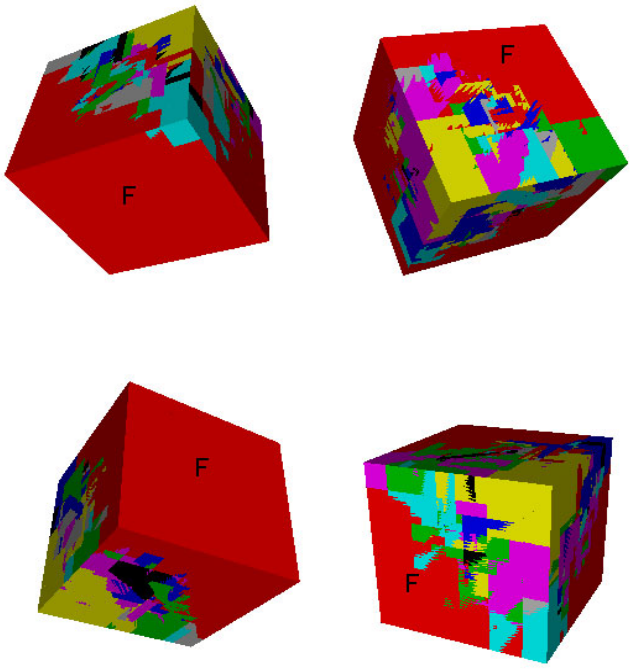
(1) $y_1 = \frac{100}{1+\exp(20-5X_1-5X_2-5*X_3)}$
(2) $y_2 = \frac{100}{1+\exp(8+5X_1+5X_2+5X_3)}$
(3) $y_3 = \frac{100}{1+\exp(10+5X_1-5X_2+5X_3)}$
(4) $y_4 = \frac{100}{1+\exp(15-5X_1+5X_2-5X3)}$

where $X_1$, $X_2$ and $X_3$ are the first, second and third coordinates of $X$s. The final $Y$ values are generated by $Y = y_1 + y_2 + y_3 + y_4 + \epsilon$, where $\epsilon \sim N(0, 1)$. Simulations are done for n values equaling 5 and 6.

Table (2) is presented the same way as for one dimensional case. The graphical jump method leads the other two methods in both picking the correct number of nodes and making predictions.

For Figure (8), everything else is the same as those from sample size of 100 and dimension of one except that the plot containing box plots is divided into two parts with different ranges in $x$-axis. Since the mean values of prediction errors for 3 nodes are much higher than those

of the rest numbers of nodes, they are sketched separately at the left parts. In the right parts of the plots, the box plots have increased mean values along the $x$-axis.




**Figure(7) Plots of the shape of each node for the simulated data with three explanatory variables.**

For each of the scenarios above, another set of explanatory variables and response variable are generated the same way as that in each scenario. The newly generated explanatory variables are used to make predictions and the predicted values are compared to the corresponding newly generated response variable. This provides a way of keeping a hold-out sample for examine predictive accuracy (Draper and Krnjajic, 2007).

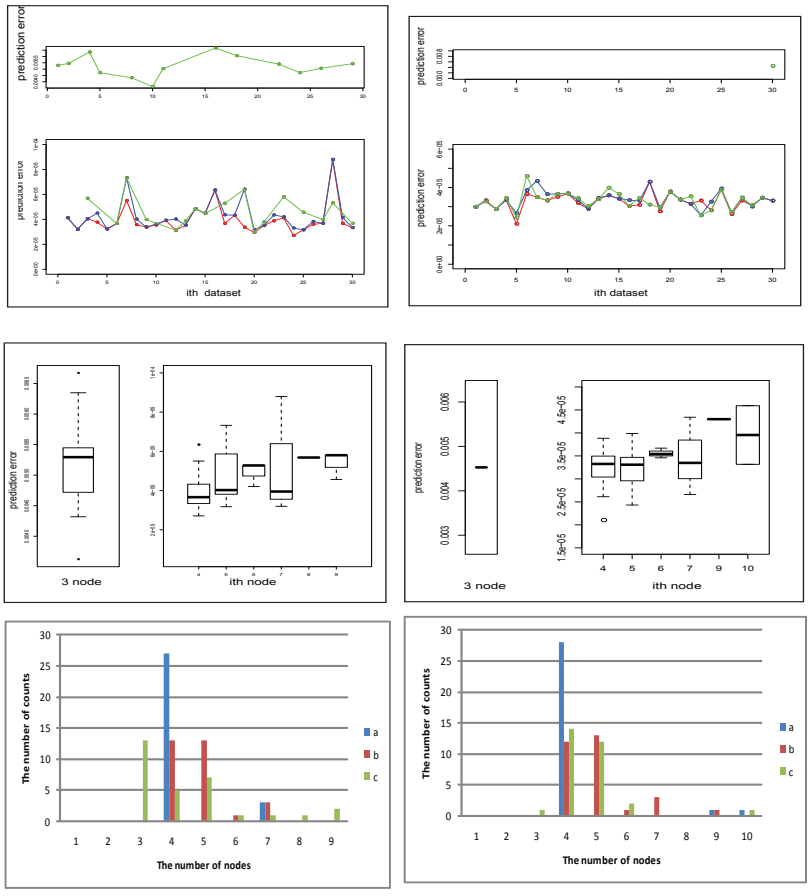
**T**able (2). The results of simulation studies for three dimensional scenarios.

| number of datapoints | 125 | 216 |
|---|---|---|
| graphical jump method | 90 | 93.3 |
| BIC | 43.3 | 40 |
| 10 fold cross validation | 16.7 | 46.7 |
| graphical jump method (mean(sd)) | 3.97e-05(1.18e-05) | 3.27e-05(4.21e-06) |
| BIC (mean(sd)) | 4.30e-05 (1.30e-05) | 3.37e-05(4.36e-06) |
| 10 fold cross validation (mean(sd)) | 0.0023 (0.0027) | 0.00018 (0.00082) |

The variance of the difference of the two are used to measure the performance of the various methods. Then the graphical jump method is used to choose the best number of nodes for functions $x^2, x^3, x^4$ and $sin(x)$ with noise, which obeys a normal distribution having mean 0 and standard deviation 0.1, and without noise. For each scenario, 10 datasets are analyzed. For $sin(x)$, the $x$ variable ranges from -1.57 to 8.8 and have 100 observations. For $x^2, x^3$ and $x^4$, the $x$ variables range from -1 to 1 and each have 400 observations.

The graphical jump method performs excellently in picking the number of nodes for functions $sin(x)$, $x^2$, $x^3$ and $x^4$ (Table (3)). The graphical jump method has excellent performance in choosing the right number of nodes for functions such as $sin$, $x^2$, $x^3$ and $x^4$. However, for $x^3$ without noise, the correct number of nodes chosen is 3 while it is 2 for $x^3$ with noise. The cause of this difference is explained by Figure (9) and Figure (10) (the numbers are the values of MSE of those points). In Figure (9), the fitted curves for 2, 3 and 4 nodes are very similar in shape for all four plots, which means the change in MSE is big from 1 node to 2 nodes, but not from 2 nodes to 3 nodes. Hence, it is not difficult to understand that the best number of nodes chosen is 2 for $x^4$ with and without noise, and $x^3$ with noise. In the left plot of Figure (10), the MSE decreases a lot when the number of nodes changes from 1 to

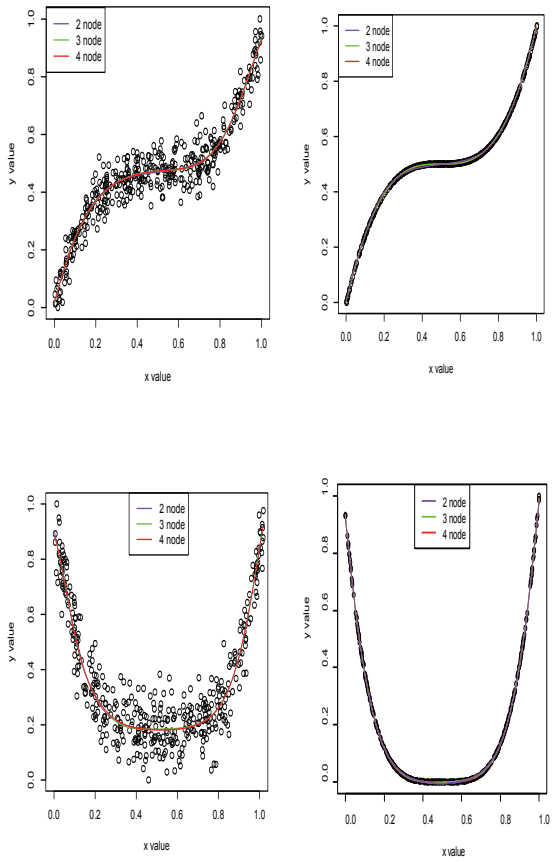
2 and much slower after. Therefore the best number of nodes chosen is 2 for $x^3$ with noise. In the right plot of Figure (10), the MSE decreases a lot when the number of nodes changes from 2 to 3 and much slower when the number of nodes changes from 3 to 4. Consequently, the best number of nodes chosen is 3 for $x^3$ without noise. The critical point is that without noise, $x^3$ is best fit with 3 nodes. The MSE of the residuals changes quite a lot when the number of nodes changes from 2 to 3. However, for the noise case, the noise level is too high, it masks the big changes in MSE of the residuals from 2 nodes to 3 nodes. Thus, for the noise case, only the big change of MSE of residuals from 1 node to 2 nodes is obvious to see.




**Figure(8) Plots of the shape of each node for the simulated data with three explanatory variables.**

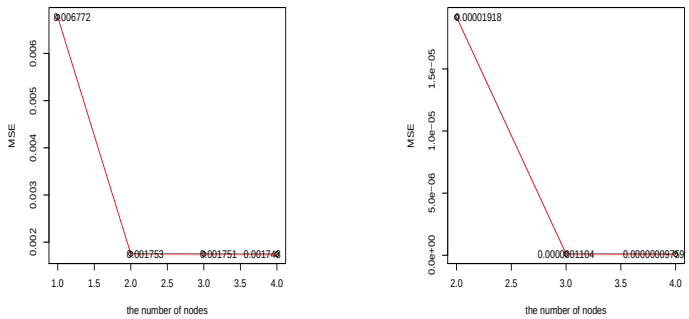
**Table (3) The results of simulation studies for several functions.**

| variable | sin with noise | sin without noise | $x^2$ with noise | $x^2$ without noise | $x^3$ with noise | $x^3$ without noise | $x^4$ with noise | $x^4$ without noise |
|---|---|---|---|---|---|---|---|---|
| the number of nodes chosen | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 2 |
| count | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 9 |

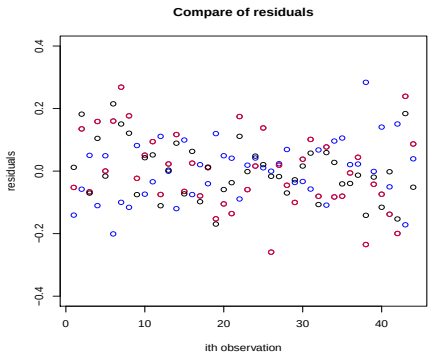



**Figure(9) Plots of fitted curves for functions $x^3$ and $x^4$ with and without noise for 2, 3 and 4 nodes (The left upper and left lower plots are for $x^3$ and $x^4$ with noises situations, The right upper and right lower plots are for $x^3$ and $x^4$ without noises situations).**

# 4 Cancer and Smoking Example




**Figure(10) Plots of MSE versus the number of nodes for simulated datasets of function $x^3$ (The left plot is for with noise situation, the right plot is for without noise situation).**

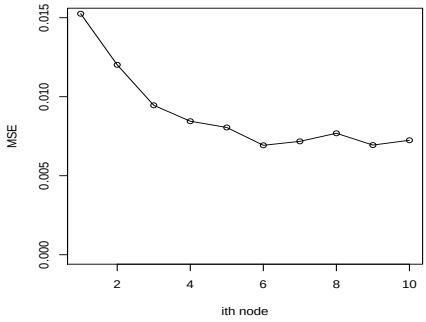



**Figure(11) Comparison of residuals generated by fitting the numbers of nodes selected by different methods (red dots are residuals of 1 node, blue dots are residuals of 6 nodes and black dots are residuals of 10 nodes).**

We now show our method in use on a real dataset. The data are per capita numbers of cigarettes sold (which is a surrogate for the number smoked) in 43 states and the District of Columbia in 1960 together with death rates per thousand population from lung cancer and bladder cancer (J.F.Fraumeni, 1960). When using the cigarette smoking rate as the response, and lung cancer and bladder cancer as the explanatory variables, the graphical jump method chooses 6 nodes as the best number of nodes. The BIC chooses 10 nodes as the best

number of nodes. The cross validation method chooses one node as the best number of nodes.

According to the residual plot (Figure (11)), the residuals for one node spread much broader than the residuals for 6 nodes and 10 nodes, and the spread of residuals for 6 nodes looks similar as that for 10 nodes, which means that increasing from 1 node to 6 nodes improves the result significantly, but not for increasing the number of nodes from 6 to 10.




**Figure(12) MSE curve fitted with different number of nodes selected by different methods for the cancer smoking dataset.**

For the MSE plot (Figure (12)), when the number of nodes increases to 6 nodes, the curve bends which means increasing the number of nodes further does not decrease the MSE of residuals as fast as before. This implies that 6 nodes might be the best number of nodes. In combination with the residual plot, the conclusion is that 6 nodes is the best number of nodes and the graphical jump method picks this answer.

# 5 Discussion and Future Directions

We introduced the graphical jump method for model selection in neural network models. Several theorems show the theoretical validity of the method, and simulations show the practical efficacy. Results were also demonstrated on a real dataset.

The graphical jump method has the potential to be extended to solve other problems for choosing numbers, such as choosing the number of species in environmental studies, or choosing the number of neighbors in graphical model. Also we can try to choose the number of nodes for other types of neural networks such as recurrent neural networks (encompassing simple recurrent networks, Hopfield networks, Echo state networks), radial basis function networks, and stochastic neural networks (including the Boltsmann machine).

# Appendix A: Proofs

### Proof of theorem 1

Let $\theta_n$ be estimated parameter after $n$ modifications by using $t$ samples repeatedly, where in each modification,

$$\theta_{n+1} = \theta_n - \epsilon(y - f(x, \theta_n))^T \nabla f(x, \theta_n)$$

It has been shown that $\theta_n$ approaches $\theta^*$ as $n \to \infty$ and $\epsilon \to 0$.
Let $\pi_n(\theta_n)$ be the probability distribution of $\theta_n$. $p(\widetilde{\theta})$ obeys

$$\widetilde{\pi}(\theta) = lim_{n\to\infty}\pi_n(\theta)$$

where $\widetilde{\theta}$ is deemed as the limits of $\theta_n$ see Murata et al. (1994) (page 867).

If $d(x, y, \theta_{opt}) = -logp(y|x, \theta_{opt})$,

$$Q_{opt} = E_q[-log''p(y|x, \theta_{opt})]$$

$$R_{opt} = V_q[-log'p(y|x, \theta_{opt})] = E_q[(-log'p(y|x, \theta_{opt}))^2] - (E_q[-log'p(y|x, \theta_{opt})])^2$$

since

$$E_q[log''p(y \mid x, \theta_{opt})] = -E_q[log(p'(y \mid x, \theta_{opt}))]^2$$

and

$$E_q[-log'p(y|x, \theta_{opt})] = 0$$

(see (Murata N, etc., 1994)).

$$Q_{opt} = E_q[-log''p(y|x,\theta_{opt})] = E_q[(-log'p(y|x,\theta_{opt}))^2] = R_{opt}$$

Without loss of generality, assume $G$ is the true number of nodes, $MSEq(G) = 1$. $MSEq(k)$, $MSEq^*(k)$ are the MSE under $q$ and $q^*$ distribution when fitting $k$ nodes to the neural network model separately. $m(k)$ is the number of parameters for $k$ nodes and $m^*$ is the number of parameters in one node.

In the following proof, we all assume $t \longrightarrow \infty$, $O(1/t/sqrt(t)) \longrightarrow 0$ and $O(1/t^{5/2}) \longrightarrow 0$. Since the minimum discrepancy $D(q, p(\widetilde{\theta})) = MSEq(G)$ should occur at the true number of nodes,

Since

$$L = \prod_{i=1}^{t} \frac{1}{\sqrt{2\pi}\sigma} exp^{[-(y_i - \beta z_i)^2]/(2\sigma^2)}$$

where $z_{ij} = [1 + exp(-\gamma_{j0} - \sum_{h=1}^{r} \gamma_{jh} x_{ih})]^{-1}$. $\Rightarrow$

$$log(\widehat{L}) = -t/2 - tlog(\widehat{\sigma})$$

where $\widehat{L}$ and $\widehat{\sigma}$ are maximum likelihood estimates. If

$$d(x, y, \theta) = -log(p(y \mid x, \theta))$$

$$-log(\widehat{L}_k) = t/2 + \frac{t}{2} log(\widehat{MSE}_k) \qquad (2)$$

Since $-log(\widehat{L}_k)$ is a function of $\widehat{MSE}$, when the minimum of $\frac{1}{2}\widehat{MSE}$ occurs at $G$ nodes, the minimum of $-log(\widehat{L}_k)$ also occurs at $G$ nodes. And $R_{opt} = Q_{opt}$, $R_{opt}^{-1} = Q_{opt}^{-1}$, if $m$ is the number of parameters in the corresponding model, $R_{opt}$ is of dimensions $mxm$,

$tr(R_{opt}Q_{opt}^{-1}) = tr(R_{opt}R_{opt}^{-1}) = tr(I_{mxm}) = m$, (see (Murata, et al., 1994 (page 868))). Therefore,

by (2), let $D(q, p(\widetilde{\theta})) = -log(\widehat{L}_{G+k}(q, p(\widetilde{\theta})))$

$-log(\widehat{L}_k(q, p(\widetilde{\theta}))) = -log(\widehat{L}_k(q*, p(\widetilde{\theta}))) + U\frac{1}{\sqrt{t}} + \frac{1}{t}tr(R_{opt}Q_{opt}^{-1}) + O(t^{-3/2})$
$= -log(\widehat{L}_k(q*, p(\widetilde{\theta}))) + U\frac{1}{\sqrt{t}} + \frac{m(k)}{t} + O(t^{-3/2})$

Since the minimum discrepancy occurs at $G$ nodes,
$-log(\widehat{L}_{G+k}(q, p(\widetilde{\theta}))) > -log(\widehat{L}_G(q, p(\widetilde{\theta})))$
by (2), $-log(\widehat{L}_G(q*, p(\widetilde{\theta}))) + U\frac{1}{\sqrt{t}} + \frac{m(G)}{t} + O(t^{-3/2}) < -log(\widehat{L}_{G+k}(q*, p(\widetilde{\theta}))) + U\frac{1}{\sqrt{t}} + \frac{m(G+k)}{t} + O(t^{-3/2})$

$$-log(\widehat{L}_{G+k}) + \frac{m(G+k)}{t} + O(\frac{1}{t^{3/2}}) > -log(\widehat{L}_G) + \frac{m(G)}{t}$$

by(2)
$\frac{t}{2} + \frac{t}{2}log(\widehat{MSE}q^*(G+k)) + U\frac{1}{\sqrt{t}} + \frac{m(G+k)}{t} + O(\frac{1}{t^{3/2}}) >$
$\frac{t}{2} + \frac{t}{2}log(\widehat{MSE}q^*(G)) + U\frac{1}{\sqrt{t}} + \frac{m(G)}{t} + O(\frac{1}{t^{3/2}})$
$\Rightarrow \frac{-t}{2}log(MSEq^*(G+k)) + \frac{t}{2}log(MSEq^*(G)) < \frac{km^*}{t} + O(\frac{1}{t^{3/2}})$
$log(\frac{MSEq^*(G)}{MSEq^*(G+k)}) < \frac{2m^*k}{t^2} + O(\frac{1}{t^{5/2}})$
$\frac{MSEq^*(G)}{MSEq^*(G+k)} < exp(\frac{2m^*k}{t^2} + O(\frac{1}{t^{5/2}}))$ (3)

Then by Taylor Series Expansion, we can prove that

$$\frac{MSEq^*(G)}{MSEq^*(G+k)} < 1 + \frac{2m^*k}{t^2} + O(\frac{1}{t^{5/2}}) \qquad (4)$$

Then, majorly with the help of Taylor series expansion, we can prove that the highest jump

score occurs at $G$ nodes,

$$MSEq^*(G)^{-v} - MSEq^*(G-1)^{-v} > MSEq^*(k)^{-v} - MSEq^*(k-1)^{-v}$$

for any $k$ for certain range of $v$.

Since the prove is too long, please refer to my thesis for details.

**Proof of theorem 2**

Without loss of generality, assume $MSE_q(1) = 1$

Similarly with the help of Taylor series expansion, we can prove

$$MSE_{q^*}^{-v}(k+1) - MSE_{q^*}^{-v}(k) < MSE_{q^*}^{-v}(1)$$

where $k \geq 1$ for certain range of $v$. Please refer to my thesis for details.

# References


Akaike, H. (1973), "Information Theory and an Extension of the Maximum Likelihood Principle," *Second International Symposium on Information Theory*, 1, 267–281.

Anders, U. and Korn, O. (1999), "Model selection in neural networks," *Neural Networks*, 12, 309–323.

Chang, J. and Sugar, A. C. (2008), "Choosing the number of clusters via the graphical jump method," .

Draper, D. and Krnjajic, M. (2007), "Bayesian model specification." Tech. Rep. 9, Department of Applied Mathematics and Statistics, University of California-Santa Cruz.

Fogel, D. (1991), "An information criterion for optimal neural network selection," *IEEE Transactions on Neural Networks*, 5, 490–497.

Gilmour, S. G. (1996), "The interpretation of Mallows's $C_p$ – *statistics*,″ *The Statistician*, $45, 49 - -56$.

J.F.Fraumeni (1960), "Cigarette Smoking and Cancers of the Urinary Tract: Geographic Variations in the United States," *Journal of the National Cancer Institute*, 41, 1205–1211.

Lee, H. K. H. (2004), *Bayesian Nonparametrics via Neural Networks*, Alexandria, Virginia: American Statistical Association, and Philadelphia, Pennsylvania: Society for Industrial and Applied Matematics.

Murata, N., Yoshizwa, S., and Amari, S. (1994), "Network information criteria: determine the number of hidden units for an artificial neural network model," *IEEE Transactions on Neural Networks*, 5, 865–872.

Sarle, W. (1995), "Stopped Training and Other Remedies for Overfitting," in *Proceedings of the 27th Symposium on the Interface*, pp. 352–360.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Sugar, C. and James, G. (2003), "Finding the number of clusters in a data set: An information theoretic approach," *Journal of the American Statistical Association*, 98, 750–763.