

3.17 Development to a Well-Designed Network Model

Choosing the Neural Network Architecture

- the number of input variables
- the type of neural network architecture
- the number of hidden layers
- the number of hidden units per hidden layer
- the type of combination, transfer or error function
- the type of optimization technique
- skip layers or direct connections

If you are very knowledgeable in the function you are fitting, you may be able to make some of the above choices based on its theoretical properties. Usually, trial-and-error is required to achieve a good network architecture. Understanding the structure of the data in preparing the data for analysis is extremely important to a well-designed neural network design. It's arbitrary to the selection of the various neural network architectures and configuration settings. Again, it's advised that building a well-designed neural network model takes patience and involves a large amount of time spent in preparing the data for network training. For example, similar to stepwise regression in traditional regression modeling, comparisons must be considered in the selection of the best linear combination of input variables that best describes and predicts the target variable in a neural network model. Also, it's recommended to sequentially fit numerous neural network models based on specifying several different starting values that are applied to the initial weight estimates and biases. In neural network modeling, there is no universal method that is known in selecting the best neural network model.

- Pre-processing of the data is one of the most important steps to a well-designed neural network model. That is, limiting the number of input variables in the model in order to reduce the effect in the "curse of dimensionality" in avoiding undesirable bad local minimums and accelerate convergence to the minimization process. And yet, retaining as much of the relevant information as possible with respect to the input variables included in the model that best explains the output responses. Also, it's important to exclude outliers or extreme values from the analysis and perform transformations to achieve normality in the input and target variables. You might also have to estimate the missing data points. Preparing the data for neural network modeling is usually one of the most complicated steps to the modeling design.
- It's important to standardize the input variables in the model to assure convergence in the optimization process and if the model is unsatisfactory then a transformation might be advised. Standardization is effective when the input variables are measured in different units.
- It's important that there is a sufficient number of observations in the training data. For a noise free distribution in the target values, a general rule is that there must be at least ten times as many records with comparison to the number of input variables in the model in order to get reasonably stable weight estimates i.e. assuming that there is small amount of variability in the underlying training data. *Stable estimates* meaning that partitioning the data and fitting the model multiple times will result in approximately the same weight estimates. For classification modeling, it is recommended that there should be at least 10 observations for each level of the categorical input variable in the neural network model. If the model is unsatisfactory then more data points might be required or bootstrapping techniques might be considered. The reason is because the hold out method works best when there is a sufficient amount of data or overfitting may occur due to a small sample size resulting in wild forecasting estimates based on fitting the network model beyond the range of the actual target values. Again, the number of cases needed depends on the complexity of the underlying function and the variability in the random error to the model.
- Even though there is no standard method in partitioning the data for the hold-out procedure it

might not be a bad idea in trying different partitioning schemas i.e. a different percentage of allocation to the partitioned data sets to avoid overfitting and generate better modeling estimates.

- It's important that the initial weight estimates and biases are set to some randomly selected small values. Too small of initial weight estimates will result in the network model failing to converge to the true parameter estimates and too large of initial weight estimates will result in a bad fit to the data. It is recommended that a well-designed neural network model consists of fitting several different models to the same data. That is, by initializing the network with different random weights in estimating the final network weights and biases in the neural network model is called *training the network*. And if the model is still unsatisfactory then preliminary runs are recommended in determining the most appropriate initial weight estimates to the training data set. The purpose of refitting the data numerous times is to alleviate the optimization problem in reaching an undesirable bad local minimum in the error function.
- The nonlinear transfer functions like the hyperbolic tangent function is usually applied to the input-to-hidden layer. The reason is because the general linear combination function is applied and also this same sigmoidal function has ideal convergence properties. It is not as critical that the appropriate transfer function is applied. However, it is important that the correct activation function i.e. transfer and combination function is applied.
- It might be appropriate to select an entirely different hidden or outer layer activation function in order to generate better forecasting estimates as opposed to the default neural network configurations. This idea was applied in a modeling comparison example presented later in the book to compare the similarity between a neural network design and a non-linear design. The hidden layer activation function that was selected to the neural network design was based on the functional relationship between the target variable and the input variable in the network model.
- The best neural network model design is usually the simplest model with a single hidden layer. And if the model is unsatisfactory then increase the complexity of the neural network model by adding additional hidden layer units to the design in order to increase the accuracy in the predictive modeling or classification performance. And yet, adding too many hidden layer units might lead to overfitting in the data. Therefore, a better idea is to form a "committee" of neural network models based on fitting numerous neural network models whose output is the weighted average of the individual neural network models assuming that the estimates have the same level of measurement and the variables are not based on entirely different attributes. At often times, the performance from a committee of neural network models will generate a better modeling performance with comparison to a single neural network model. One reason is because of the reduction in the error to the model is based on the overall average error of the individual models.
- It's important that the appropriate optimization technique is applied in order to assure that the weight estimates converge to the correct parameters. The optimization method to apply depends on the number of weights in the neural network model and the complexity of the error function.
- Using minimization algorithms based on the tuning constants then it's important that the tuning constants are accurately specified to assure convergence. Therefore, it's recommended to fit numerous network models and adjusting the tuning constants by trial-and-error that can result in a large amount of time spent determining the precise values. Usually the initial learning rate is set to some small positive number like .05 or .1 with a momentum parameter set to zero.
- Weight decay regularization might be considered in order to increase both the predictive and classification performance of the network design. Weight decay regularization can also overcome the overfitting phenomenon. That is, applying an appropriate weight decay estimate can result in excellent generalization results with the validation or test error staying fairly consistent as the number of hidden units increase by keeping the weight estimates small during network training.
- It's important that you carefully analyze the various modeling assessment statistics in measuring the accuracy in the modeling performance of the neural network model.

3.18 Advantages of Neural Network Modeling

Advantages of Neural Network Modeling

- **Enormous Number of Predictor Variables:** Neural network modeling is specifically designed to handle an enormous number of input variables to the nonlinear model. The reason is that some of the optimization techniques used in determining the smallest sum-of-squares error is especially designed in handling this dilemma. That is, assuming the error function that we want at a minimum is a relatively smooth function.
- **Unknown Functional Form:** Neural network modeling does not require any distributional assumptions between the input variables and the target variables to the model. MLP neural network modeling works best when the functional form is unknown between the input variables and the target variable in the multi-layer perceptron MLP architecture. The MLP architecture is the standard neural network design often used. One of the properties in the MLP design is that it ignores the functional relationship between the input variables and the target variable in the model. And unlike traditional regression modeling, neural network model building does not depend on the exact mathematical functional form between the target variable and the input variables to the model. However, similar to traditional regression modeling, it is critical to include the input variables that best describe or predict the target variable in the model. Also, it is important to identify the underlying distribution of the target variable in order to apply the correct transfer function and error function to the neural network model.
- **Universal Approximation:** Neural network modeling is designed to fit any functional form with any degree of accuracy assuming that there are a sufficient number of hidden layer units. Neural network modeling encompasses various statistical models. The flexibility to neural network modeling is that the estimates can be compared to a wide range of statistical modeling designs from multiple regression, logistic regression, nonlinear regression, time series, non-parametric modeling and classification modeling to any degree of accuracy. And adding more hidden units to a neural network architecture then the network model has the ability of fitting extremely complex nonlinear functions with a high degree of accuracy. For classification modeling, neural network models can approximate highly nonlinear decision boundaries with great precision given enough data and enough computational time. This will be presented at the end of the book.
- **Predictability:** There might be times when neural network modeling might perform better than traditional regression or classification modeling. That is, the main goal to forecast modeling or classification modeling is producing accurate forecasting or classification predictions.
- **Business Modeling to Maximize Profit or Minimize Loss:** Critical business decision can be made using both logistic regression and neural network modeling based on categorical-valued target variable. Both models can perform business modeling in order to maximize expected profit or minimize expected loss based on predetermined profit and cost amounts for each target-specific decision consequence or prior probabilities that represents the true proportion of occurrences to each level of the categorical-valued target variable. The predictive modeling design determines the best linear combination of parameter estimates that produces some combination of the posterior probabilities to each of the target levels in determining the largest expected profit or the smallest expected loss based on the separate decision levels that depends on accurately specified decision values and prior probabilities. Therefore, specifying the correct prior probabilities and target-specific decisions and assuming the predictive model is correctly specified in fitting the underlying distribution of the data will generally result in the correct decision results. The advantage in neural network modeling is that the network model might produce larger expected profits or smaller minimum losses as opposed to the logistic regression model or any other predictive model with a unique categorical response variable to predict.

- **Interaction or Nonlinear Terms:** Generally, MLP neural network designs generate better parameter estimates and a smaller error estimate with comparison to the traditional linear regression models with interaction, polynomial or nonlinear terms. One of the reasons why is because neural network modeling collapses some of the interaction terms and the error term from the regression model that represents the error term to the network model. That is, both the MLP and NRBF neural network designs can ignore some of the irrelevant terms in the predictive model as follows.

Traditional regression modeling:

$$Y = X_1X_2 + X_3X_4 + \varepsilon_{\text{reg}}$$

Neural network modeling:

$$Y = X_1X_2 + \varepsilon \text{ where } \varepsilon = X_3X_4 + \varepsilon_{\text{reg}}$$

- **Transformed Data:** Neural network modeling is not as affected as traditional regression modeling when it comes to transforming the input variables in achieving a better fit. The reason is that neural network applies nonlinear functions to the process. But, it is important that the appropriate transformation is applied to accelerate convergence in the iterative process in improving network estimation and generalization performance. On the other hand the appropriate transformation to the target variable is important in achieving normality and constant variance in the iterative process resulting in improved network generalization. In Enterprise Miner, the **Transform Variables** node performs data transformations to both the input and target variables.
- **Issues with Multicollinearity:** The error estimates and test statistics in the more traditional forecasting methods like multiple regression and logistic regression can be inflated or introduce bias to the forecasting model because of multicollinearity. Thereby, making the statistical inference or statistical analysis invalid. In neural network designs, if multicollinearity exists between the input variables in the model then it will cause the weight estimates to become large due to ill-conditioning in the Hessian matrix leading to bad generalization. An ill-conditioned matrix means that the determinant of the Hessian matrix will be zero, therefore the inverse to the matrix is undefined with no unique solution to the parameter estimates in the predictive model. That is, there are multiple input variables in the model that are identical and therefore the system of least-squares equations cannot be solved. However, assuming a small number of weights in the neural network design and a sum-of-squares error function to the design then a Levenberg-Marquardt optimization technique may be used. The advantage to this method is that it performs well even when the parameter estimates are highly correlated to one another. Although, it is always a good idea to identify and eliminate the highly correlated input variables in advance thereby the network can determine the most appropriate weights more efficiently. That is, the first-order optimization techniques used in network training are affected by ill-conditioning resulting in slow convergence. Whereas, the second-order optimization techniques are more well-behaved under ill-conditioning. Also, the preferable methods under ill-conditioning are the algorithms that require a smaller number of weights.
- **Validating:** The purpose in performing neural network predictive modeling might be to compare the traditional regression, time series, logistic regression or discriminant analysis estimates with the neural network estimates to either validate the estimates from the predictive or classification model or even adopt the neural network model with either a better classification performance with comparison to the logistic regression model or the Fisher's discriminant model or better prediction estimates with comparison to the traditional regression estimates either at certain data points of interest or beyond the range of the actual data points.

Disadvantages of Neural Network Modeling

- **Weights Uninterruptible:** The one of the biggest criticism to neural network modeling is that the weights are uninterruptible. That is, unlike traditional regression, in neural network modeling the weights parameters are uninterruptible due to the inputs standardized, the hidden layer weights and the nonlinear hidden layer activation function applied. The hidden-to-target weight estimates do not explain the relationship in the effect or the rate of change with respect to the target variable and the input variables. Assessing the importance of the inputs based on the size of the weights is not easy. The reason is that the inputs are also related to the input-to-hidden layer weights and the hidden-to-target weights. Yet, if all the input-to-hidden weights are close to zero with respect to the input variable, it usually indicates that the variable is not an important effect in predicting the target. Although, at times there might exist large hidden-to-target weight estimates. Therefore, one goal in neural network designing is to determine the association between the input variables to output weights by interpreting the value of the weight connections between the units. One way in measuring the importance of the inputs in a neural network model is by observing the increase in the error function by removing the input variable in the predictive model. Other modeling assessment tactics include pruning inputs, sensitivity pruning or sensitivity analysis by creating various conditional effects plots. Some neural network experts have conducted principal component analysis of the network weight estimates in order to make interpretations of the parameter estimates.
- **Convergence Unassured:** There is no universal optimization technique that is applied in computing the optimum weight estimates to a MLP network design. And there is no guarantee that the iteration algorithm applied to the validation data set will converge to the most desirable minimum error in determining the best model i.e. with the best linear combination of parameter estimates from the training data set. This is especially true if the initial weight vector is not even near the correct values or when the nonlinear error surface has multiple saddle points, i.e. local minimums and maximums, due to overfitting with too many weights in the network model.
- **Needs Enormous Data:** Neural network modeling works best when there exists an enormous amount of data for the split sampling or hold out technique and applying the iterative neural network algorithm in computing the weight estimates to the model. That is, assuming that there is enough data points allocated to the training data set in calculating the weight estimates along with a sufficient amount of data partitioned to the validation data set in computing the smallest error or largest expected profit with the remaining data set aside to the test data set in assessing the accuracy of the neural network modeling fit. At times, the network model will not generalize well resulting in wild forecasting estimates due to a small sample size.
- **Excessive Computational Time and Memory:** Considerable amount of computational time and memory resources is often required performing the iterative grid search routine in determining the best linear combination of neural network weights, particularly if the initial starting weight estimates are not a close approximation to the true parameter estimates and fitting the neural network model to a highly nonlinear error function.
- **No Diagnostic Statistics:** Currently, there does not seem to be any diagnostic statistics that are available in testing for lack of fit to the neural network model, identifying influential and outlier data points and performing significant tests of the weight estimates in measuring the importance of the input variables in the neural network model. That is, currently Enterprise Miner does not calculate prediction intervals based on the neural network estimates in assessing the accuracy of the predictive model since the standard errors to the weight estimates are undefined.
- **No Standard Method for Partitioning:** As explained previously, the neural network hold out method is based on the partitioning of the training data. But, there is no general rule that is known in the precise allocation of the training data in order to create the partitioned data sets.

- **No Universal Input Variable Selection Routine:** Like traditional regression modeling, the accuracy in neural network modeling depends on the appropriate selection of the input variables to the model. The standard variable selection technique is called *stepwise-regression* that identifies a combination of important input variables to the model from a large number of potential input variables. Another method to apply is *input pruning* or *sensitivity analysis* i.e. sequentially eliminating input variables in the model in order to reduce the complexity in the network model. *Variable Clustering* analysis formulates unique groups of input variables that are as correlated as possible with each other and different clusters are created that are unrelated to each other. The next step is selecting the best input variables within each cluster that is created. This method is based on principal components. That is, the principal components are a linear combination of the inputs with weights that are chosen which explains the largest amount of variability in the data. That is, where the first component explains the most variability, the second principal component explaining the next largest variability and so on. The basic idea is then to select the best input variables within each cluster that is created. This method selects the best inputs to the predictive model based on an algorithm starting off with all the input variables in one cluster. Principal components is then performed that examines the eigenvalues computed for each component. The eigenvalues measures the amount of variability explained by each component. The algorithm examines each successive eigenvalue that is greater than some predetermined threshold value where the initial cluster is split into two separate groups. This recursive splitting is performed with the number of clusters selected until the second eigenvalue drops below the threshold rate. That is, specifying a larger threshold value will result in fewer clusters and less variability explained by the input variables. Conversely, specifying a smaller threshold value will result in more clusters with the linear combination of inputs explaining more variability in the data. *Decision tree* analysis can also select the best inputs to the predictive model that performs the best splits to the target values. Decision tree techniques are designed in pruning the decision tree by determining the optimum reduction in the tree structure based on the best splits located at the beginning of the root node where a majority of the data exists.
- **Excessive Hidden Units Leads to Overfitting:** Selecting too many hidden units leads to overfitting and there is no standard method in selecting the appropriate number of hidden units. A basic architecture selection strategy is to fit a MLP design with one hidden layer and a fixed set of input variables then iteratively increase the number of hidden units to the model until the various modeling assessment statistics begin to increase called *constructive learning*. A type of architecture selection technique that is applied to a network design is starting with a small model and increasing the complexity of the model at each step called *sequential network construction* or *cascade correlation*. Sequential network construction starts with one hidden unit and using the final weight estimates at each step as starting values in subsequent steps. Cascade correlation is basically fitting both a single hidden layer unit and a (link) linear design. The network design begins with a skip layer design i.e. a single input unit directly connected to the target unit. And in subsequent steps, one hidden layer unit is added to the design where each additional hidden layer unit is connected to the existing input unit and the existing hidden layer units previously added are then frozen. *Frozen weights* are weights once they are fitted never change. These added hidden layers act like a new input variable to the design after they have been trained. This process is repeated until there is a sufficiently small validation error that is achieved.
- **No Universal Method for the Initial Weight Estimates:** There is no universal method that is known in calculating the initial weight estimates in the neural network model. Compounded by the fact that the various optimization techniques depend on the starting weight estimates being a reasonably close approximation to the correct parameter values.
- **Selection of Activation and Error Function:** Neural network modeling depends on a correctly specified activation function, i.e. transfer and combination function, and error function that depends on the level of measurement of the target variable in the neural network model.