Data Mining

SPSS Clementine 12.0

9. Neural Network Algorithms

Spring 2010

Instructor: Dr. Masoud Yaghini

Outline

- Introduction
- Field Encoding
- Neural Net Node Model Options
- Neural Net Node Additional Options
- Neural Net Node Expert Options
- Neural Net Model Nuggets
- References

Introduction

Introduction

- The **Neural Net node** is used to create and train a neural network.

Introduction

- **Requirements:**
  - There are no restrictions on field types.
  - Neural Net nodes can handle numeric, symbolic, or flag inputs and outputs.
  - The Neural Net node expects one or more fields with direction ***In*** and one or more fields with direction ***Out***.
  - Fields set to ***Both*** or ***None*** are ignored.

Introduction

- **Strengths:**
  - Neural networks are powerful general function estimators.
  - They usually perform prediction tasks at least as well as other techniques and sometimes perform significantly better.
  - Clementine incorporates several features to avoid some of the common pitfalls of neural networks, including:
    - sensitivity analysis (as indicated in the variable importance chart) to aid in interpretation of the network,
    - pruning and validation to prevent overtraining, and
    - dynamic networks to automatically find an appropriate network architecture.

Field Encoding

Introduction

- **Field Encoding:**
  - Scaling of Range Fields
  - Numeric Coding of Symbolic Fields
  - Binary Set Encoding of Symbolic Fields
  - Encoding of Flag Fields

Scaling of Range Fields

- In Clementine, range fields are rescaled to have values between 0 and 1.

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$$

  - x'_i is the rescaled value of input field x for record i
  - x_i is the original value of x for record i
  - x_{min} is the minimum value of x for all records
  - x_{max} is the maximum value of x for all records

Numeric Coding of Symbolic Fields

- Clementine recode a symbolic field as a group of numeric fields with one numeric field for each category or value of the original field.
- For each record, the value of the derived field corresponding to the category of the record is set to 1.0, and all the other derived field values are set to 0.0.
- Such derived fields are sometimes called **indicator fields**, and this recoding is called **indicator coding**.

Numeric Coding of Symbolic Fields

- Example:
  - consider the following data, where x is a symbolic field with possible values A, B, and C:

Record #	X	X_1'	X_2'	X_3'
1	B	0	1	0
2	A	1	0	0
3	C	0	0	1

Binary Set Encoding of Symbolic Fields

- The default encoding creates one input for each possible value of a set field.
- For large sets this can create a burdensome number of inputs which can bog down the network and increase memory requirements.
- **Binary Set Encoding**
  - The binary set encoding options use an encoding based on binary arithmetic to encode each set field as a group of numeric inputs.
  - Instead of *k* input units for a set field, where *k* is the number of possible values for the set, binary encoding uses $\log_2(k+1)$ input units (rounded up).

Binary Set Encoding of Symbolic Fields

- To get the encoded values, enumerate the possible set values in ascending order and convert the number of the value to be encoded into its binary (base-2) representation.
- Example:
  - consider a set field with three possible values, A, B, and C.
  - The field would be recoded as derived units, as illustrated in the table.

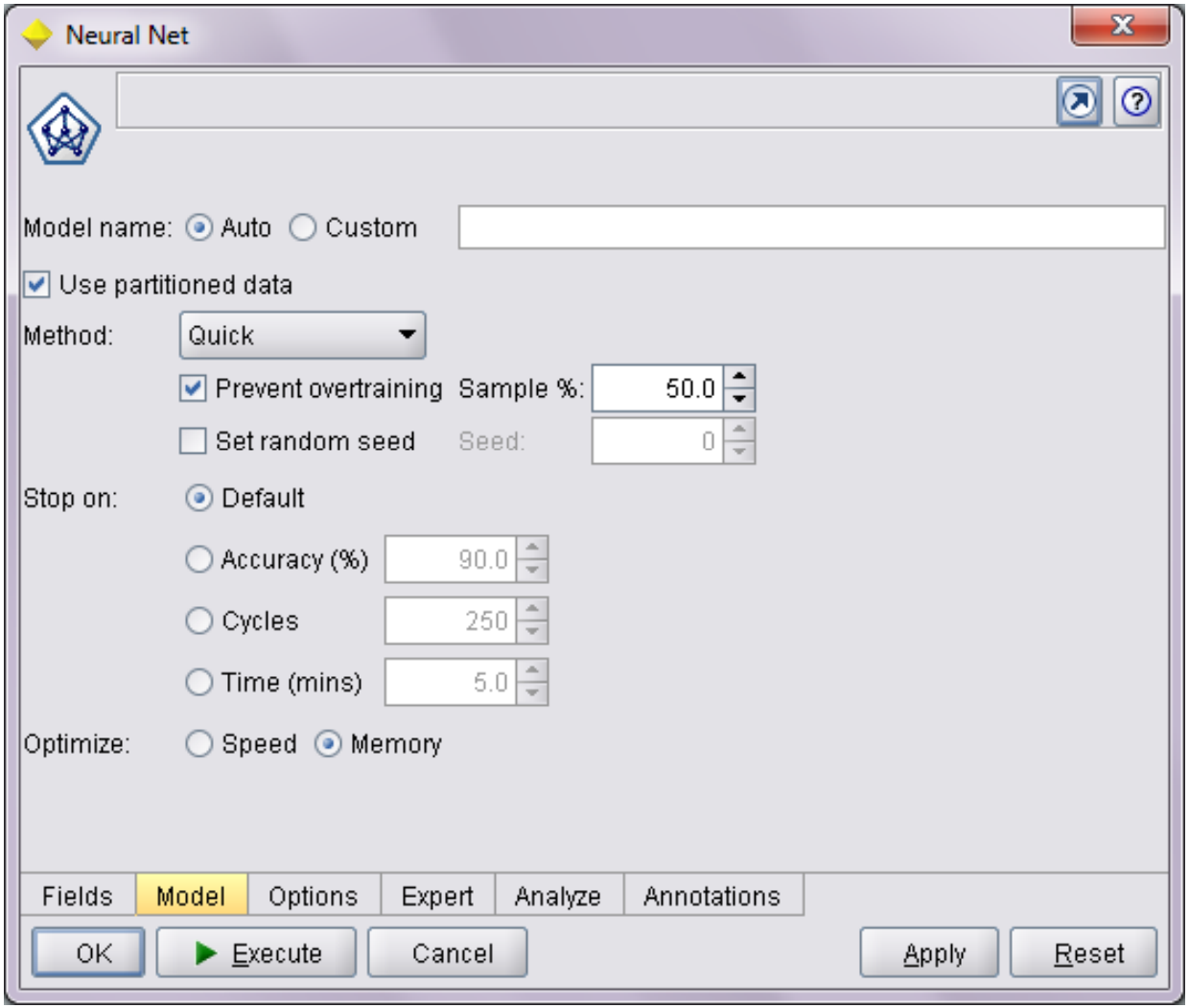
Record No.	X	X_1	X_2
1	A	0	1
2	B	1	0
3	C	1	1

Encoding of Flag Fields

- Flag fields are a special case of symbolic fields.
- Flag fields are represented by a single numeric field, taking the value of 1.0 for the “true” value and 0.0 for the “false” value.
- Blanks for flag fields are assigned the value 0.5.

Neural Net Node Model Options

Neural Net Node Model Options



Clementine

Neural Net Node Model Options

- **Model name**
  - You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.
- **Use partitioned data**
  - If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Neural Net Node Model Options

- **Method.** There are six training methods for building neural network models:
  - **Quick**
  - **Dynamic**
  - **Multiple**
  - **Prune**
  - **RBFN**
  - **Exhaustive prune**

Neural Net Node Model Options

- **Quick**
  - This method uses rules of thumb and characteristics of the data to choose an appropriate shape (topology) for the network.
  - The method will generally produce smaller hidden layers that are faster to train and generalize better.
  - If you find you get poor accuracy with the default size, try increasing the size of the hidden layer on the **Expert tab** or try an alternative training method.

Neural Net Node Model Options

- **Dynamic**
  - This method creates an initial topology but modifies the topology by adding and/or removing hidden units as training progresses.
- **Multiple**
  - This method creates several networks of different topologies (the exact number depends on the training data).
  - These networks are then trained in a pseudo-parallel fashion.
  - At the end of training, the model with the lowest RMS error is presented as the final model.

Neural Net Node Model Options

- **Prune**
  - This method starts with a large network and removes (prunes) the weakest units in the hidden and input layers as training proceeds.
  - This method is usually slow, but it often yields better results than other methods.
- **RBFN**
  - The radial basis function network (RBFN) uses a technique similar to *k-means* clustering to partition the data based on values of the target field.

Neural Net Node Model Options

- **Exhaustive prune**
  - This method is related to the Prune method. It starts with a large network and prunes the weakest units in the hidden and input layers as training proceeds.
  - With **Exhaustive Prune**, network training parameters are chosen to ensure a very thorough search of the space of possible models to find the best one.
  - This method is usually the slowest, but it often yields the best results.
  - Note that this method can take a long time to train, especially with large datasets.

Neural Net Node Model Options

- **Prevent overtraining**
  - This option randomly splits the data into separate training and testing sets for purposes of model building.
  - The network is trained on the training set, and accuracy is estimated based on the test set.
  - Specify the proportion of the data to be used for training in the Sample % box in the Neural Net node, and the remainder of the data will be used for validation.

Neural Net Node Model Options

- **Set random seed**
  - If no random seed is set, the sequence of random values used to initialize the **network weights** will be different every time the node is executed.
  - This can cause the node to create different models on different runs, even if the node settings and data values are exactly the same.
  - By selecting this option, you can set the random seed to a specific value so the resulting model is exactly reproducible.
  - A specific random seed always generates the same sequence of random values, in which case executing the node always yields the same generated model.

Neural Net Node Model Options

- **Stop on.** You can select one of the following stopping criteria:
  - **Default**
  - **Accuracy (%)**
  - **Cycles**
  - **Time (mins)**
- **Default**
  - With this setting, the network will stop training when the network appears to have reached its optimally trained state.
  - If the default setting is used with the Multiple training method, the networks that fail to train well are discarded as training progresses.

Neural Net Node Model Options

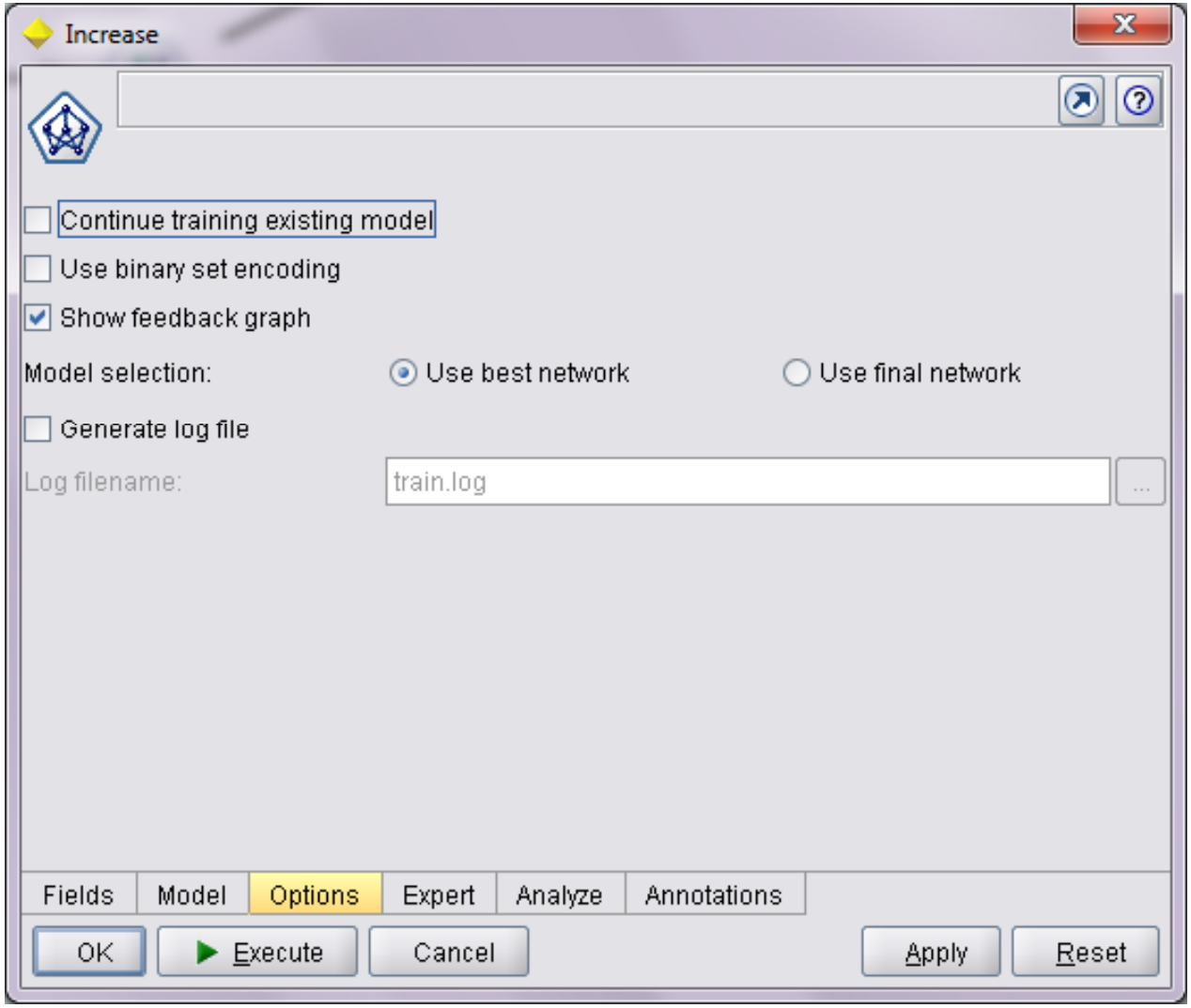
- **Accuracy (%)**
  - With this option, training will continue until the specified accuracy is attained.
  - This may never happen, but you can interrupt training at any point and save the net with the best accuracy achieved so far.
- **Cycles**
  - With this option, training will continue for the specified number of cycles (passes through the data).
- **Time (mins)**
  - With this option, training will continue for the specified amount of time (in minutes).

Neural Net Node Model Options

- **Optimize**
  - **Speed**. to instruct the algorithm to never use disk spilling in order to improve performance.
  - **Memory**. to instruct the algorithm to use disk spilling when appropriate at some sacrifice to speed.
    - This option is selected by default.

Neural Net Node Additional Options

Neural Net Node Additional Options



Clementine

Neural Net Node Additional Options

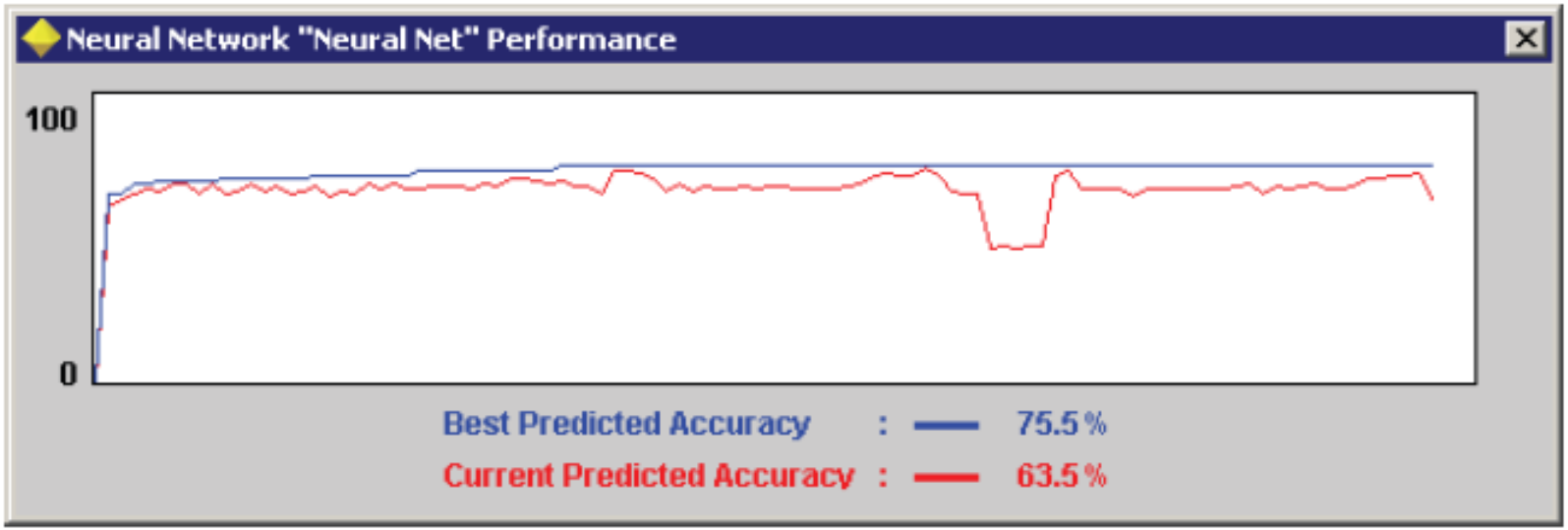
- **Continue training existing model**
  - By default, a completely new model is created each time a modeling node is executed.
  - If this option is selected, training continues with the last model successfully produced by the node.
  - This makes it possible to update or refresh an existing model without having to access the original data and may result in significantly faster performance since only the new or updated records are fed into the stream.
  - Details on the previous model are stored with the modeling node, making it possible to use this option even if the previous model nugget is no longer available in the stream or Models palette.

Neural Net Node Additional Options

- **Use binary set encoding**
  - If this option is selected, a compressed binary encoding scheme for set fields is used.
  - This option allows you to more easily build neural net models using set fields with large numbers of values as inputs.
  - However, if you use this option, you may need to increase the complexity of the network architecture (by adding more hidden units or more hidden layers) to allow the network to properly use the compressed information in binary encoded set fields.

Neural Net Node Additional Options

- **Show feedback graph**
  - If this option is selected, you will see a graph that displays the accuracy of the network over time as it learns.
  - In addition, if you have selected Generate log file, you will see a second graph showing the training set and test set metrics (defined below).
  - This feature can slow training time. To speed training time, deselect this option.



Neural Net Node Additional Options

- **Model selection**
  - By default, when training is interrupted, the node will return the best network as the generated net node.
  - You can request that the node return the final model instead.

Neural Net Node Additional Options

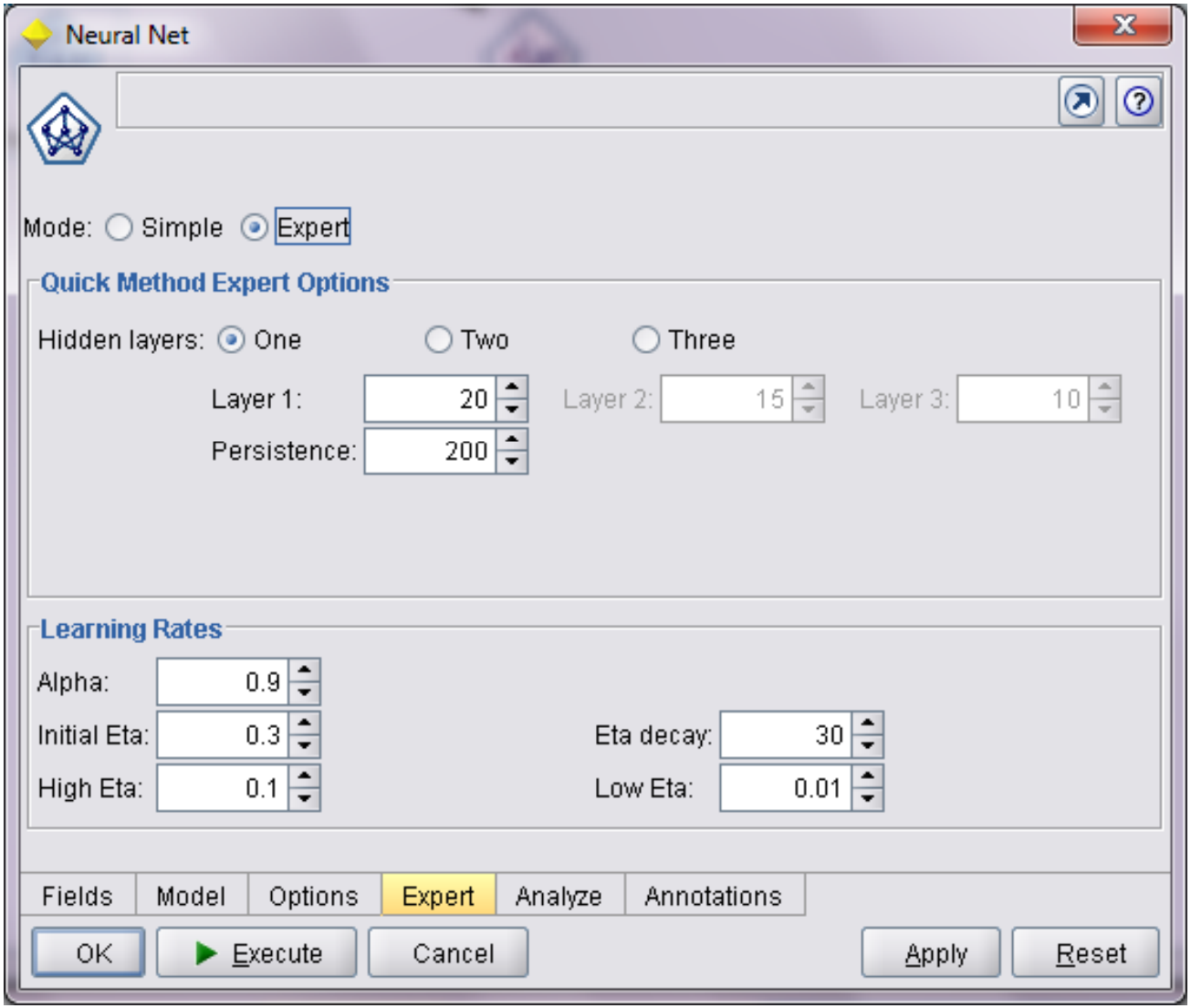
- **Generate log file**.
  - If this option is selected, information on training progress will be written to the specified log file.
  - To change the log file, enter a log filename or use the File Chooser button (labeled with an ellipsis) to select a location.
  - If you select a file that already exists, the new information will be appended to the file.

Neural Net Node Expert Options

Neural Net Node Expert Options

- For those with detailed knowledge of neural networks, expert options allow you to fine-tune the training process.
- To access expert options, select the desired training method on the Model tab, and set the Mode to Expert on the Expert tab.
- Specific options depend on the training method you have selected.

Quick Method Expert Options



Quick Method Expert Options

- **Hidden layers**
  - Select the number of hidden layers for the neural network.
  - More hidden layers can help neural networks learn more complex relationships, but they also increase training time.
  - For each layer, specify the number of hidden units to include.
  - More hidden units per layer can also help in learning complex tasks, but as with additional hidden layers, they also increase training time.

Quick Method Expert Options

- **Persistence**
  - Specify the number of cycles for which the network will continue to train if no improvement is seen.
  - Applies when using the Default stopping model.
  - Higher values can help networks escape local minima, but they also increase training time.
- **Alpha and Eta**
  - These parameters control the training of the network.

Quick Method Expert Options

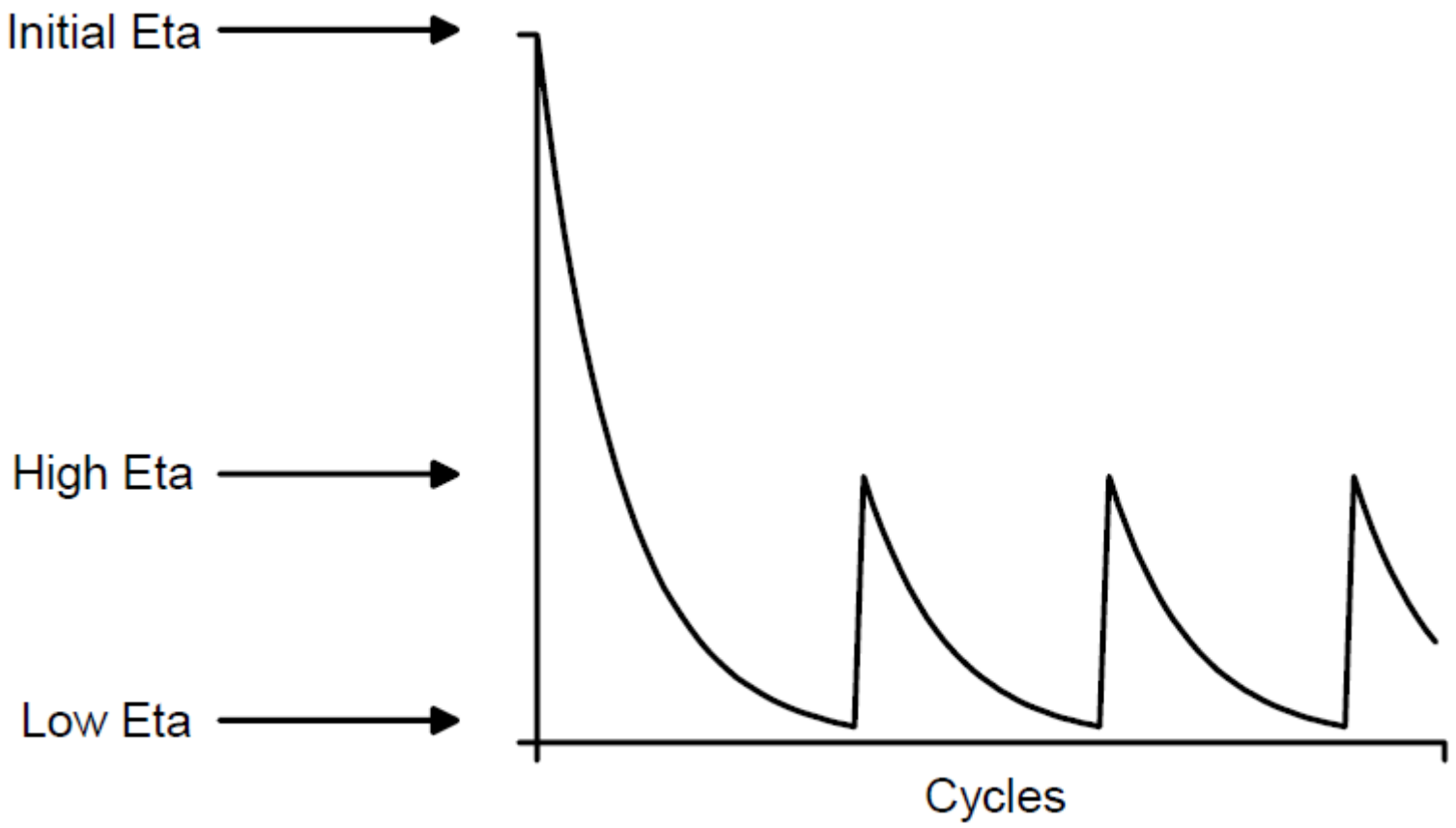
- **Alpha**
  - A momentum term used in updating the weights during training.
  - Momentum tends to keep the weight changes moving in a consistent direction.
  - Specify a value between 0 and 1.
  - Higher values of alpha increase momentum, decreasing the tendency to change direction based on local variations in the data.

Quick Method Expert Options

- **Eta**
  - The learning rate, which controls how much the weights are adjusted at each update.
  - Eta changes as training proceeds for all training methods except RBFN, where eta remains constant.
  - **Initial Eta.** is the starting value of eta.
  - **Low & High Eta.** During training, eta starts at Initial Eta, decreases to Low Eta, then is reset to High Eta and decreases to Low Eta again. The last two steps are repeated until training is complete.
  - **Eta decay.** specifies the rate at which eta decreases, expressed as the number of cycles to go from High Eta to Low Eta.

Quick Method Expert Options

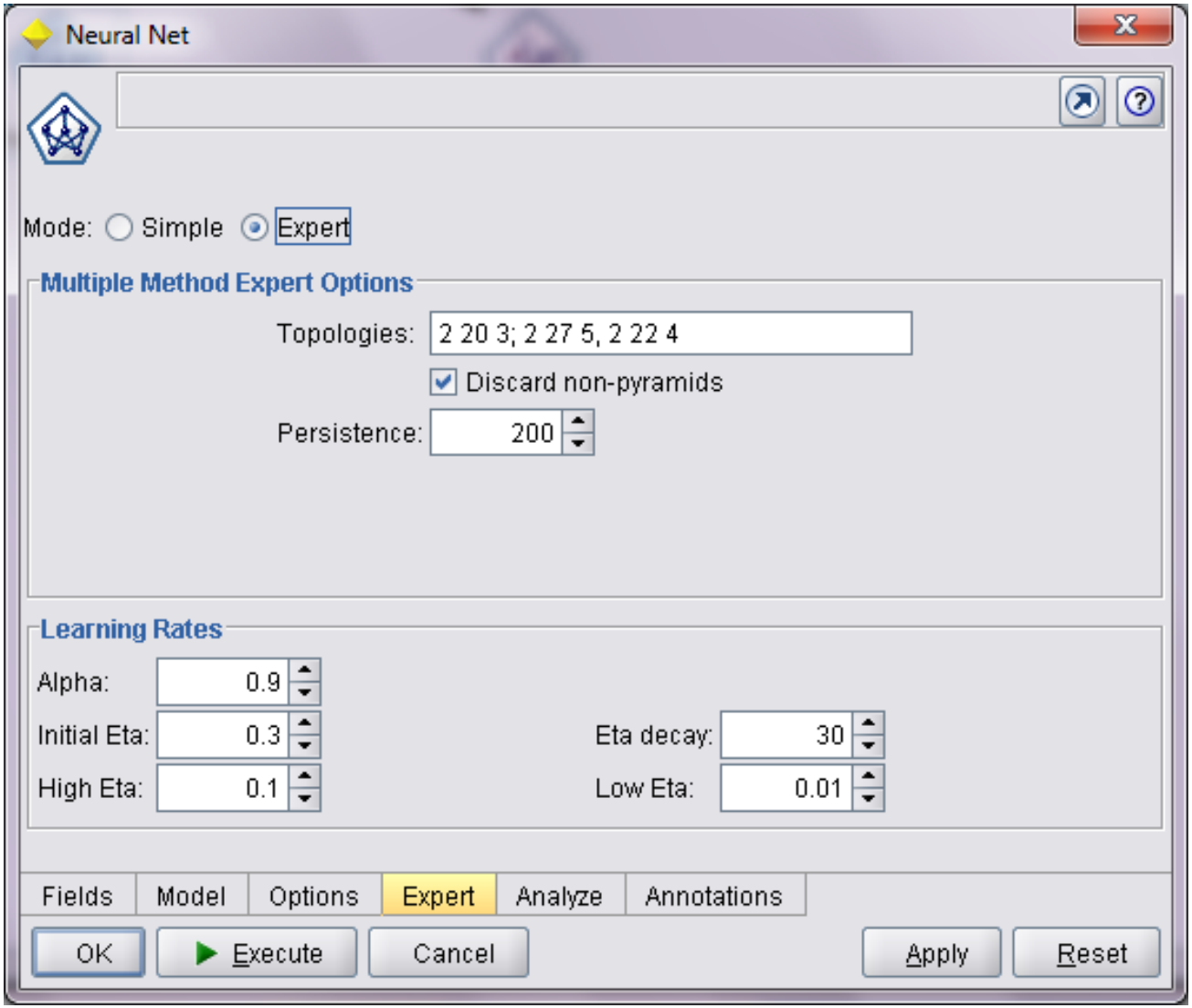
- How **eta** changes during neural network training:



Dynamic Method Expert Options

- There are no expert options for the dynamic method in the Neural Net node.

Multiple Method Expert Options



Multiple Method Expert Options

- **Topologies**
  - Specify the topologies of the networks to be trained.
  - A topology is given by specifying the number of hidden units in each layer, separated by commas.
  - Topologies can specify one, two, or three hidden layers by using the appropriate number of parameters.
  - For example, a network with one hidden layer of 10 units would be specified as 10; a network with three hidden layers of 10, 12, and 15 units would be specified as 10, 12, 15.

Multiple Method Expert Options

- **Topologies - a range of numbers**
  - You can also specify **a range of numbers** for hidden units in a layer by providing two or three numbers separated by **spaces**.
  - If two numbers are given, separate networks are created with a number of hidden units equal to each integer between the first and second number (inclusive).
  - For example, to generate networks having 10, 11, 12, 13, and 14 hidden units in a single layer, specify 10 14.
  - To generate networks with two hidden layers where the first layer varies from 10 to 14 and the second layer varies from 8 to 12, specify 10 14, 8 12.

Multiple Method Expert Options

- **Topologies - a range of numbers (cont.)**
  - In this case, networks are generated that contain all possible combinations of values.
  - If a third value is given, it is used as an increment for counting from the first value to the second.
  - For example, to generate networks with 10, 12, 14, and 16 hidden units, specify 10 16 2.

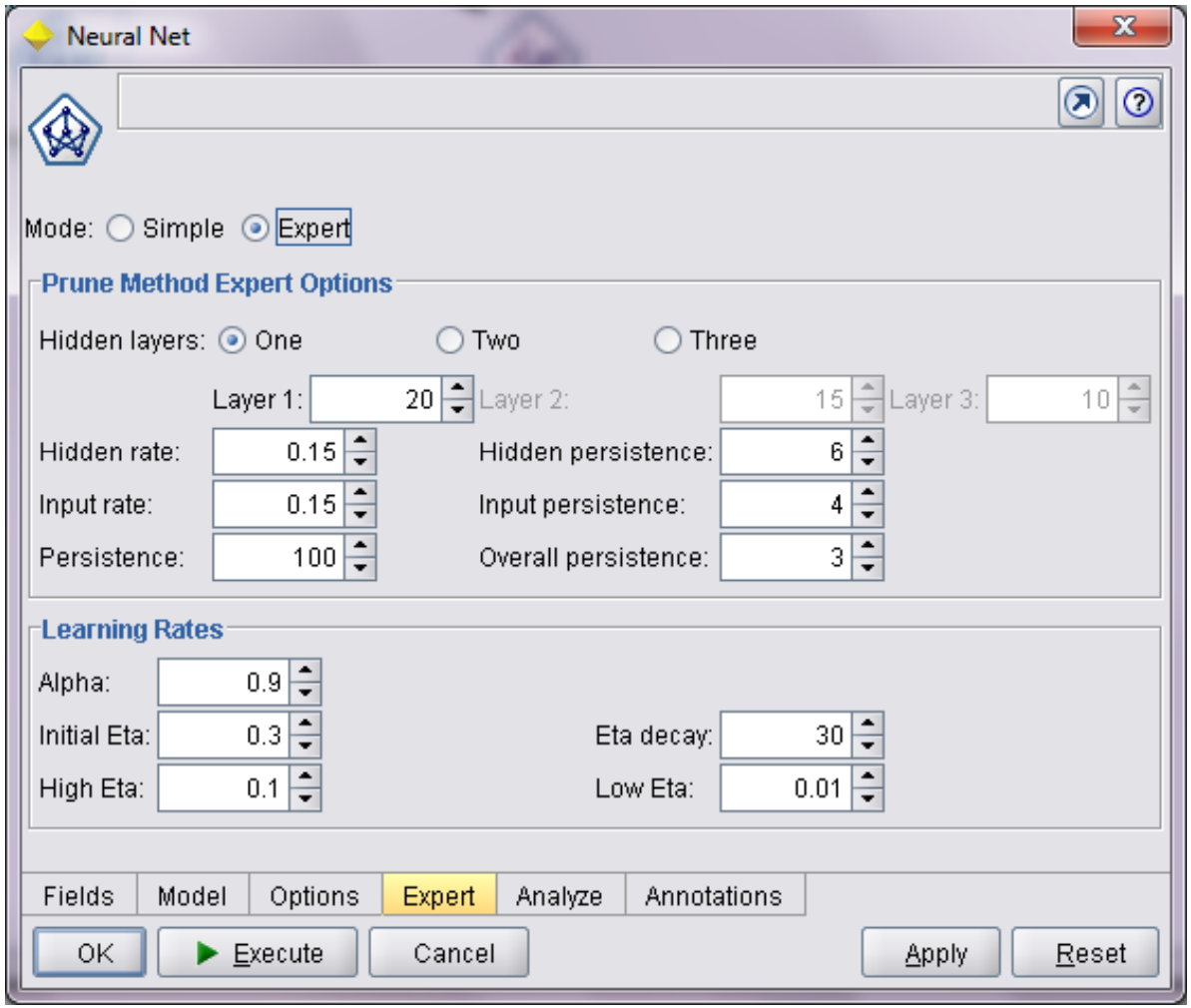
Multiple Method Expert Options

- **Multiple network topologies**
  - you can provide multiple network topologies, separated by semicolons.
  - For example, to generate networks with one hidden layer of 10, 12, 14, and 16 hidden units, and networks having two hidden layers of 10 hidden units and 7 to 10 hidden units, respectively, specify 10 16 2; 10, 7 10.

Multiple Method Expert Options

- **Discard non-pyramids**
  - Pyramids are networks where each layer contains the same number or fewer hidden units than the preceding layer.
  - Such networks usually train better than non-pyramidal networks.
  - Selecting this option discards networks that are not pyramids.

Prune Method Expert Options



Prune Method Expert Options

- **Hidden layers**
  - Select the number of hidden layers for the initial network before pruning.
- **Hidden rate**
  - Specify the number of hidden units to be removed in a single hidden unit pruning.
- **Hidden persistence**
  - Specify the number of hidden unit pruning operations to perform if no improvement is seen.

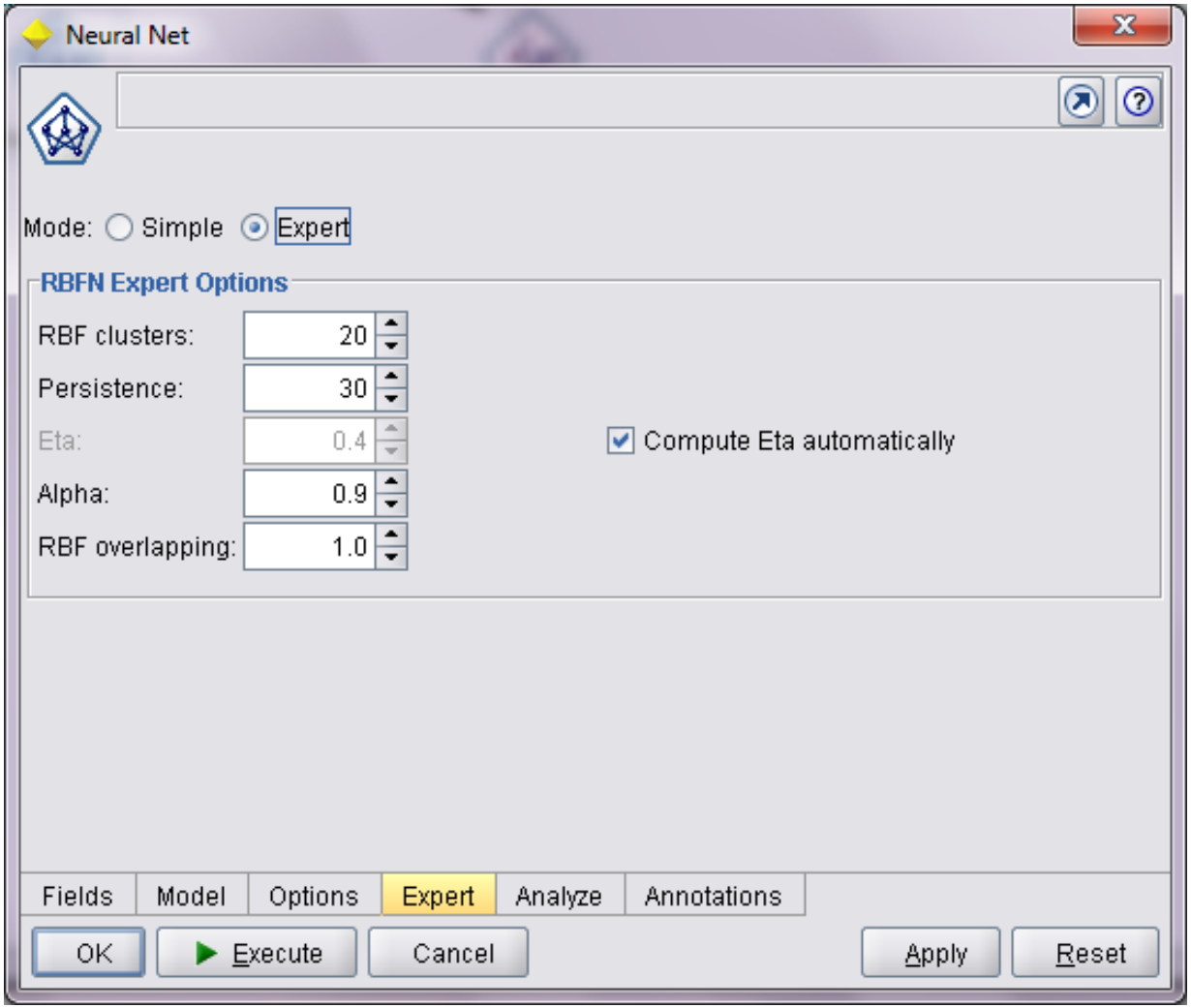
Prune Method Expert Options

- **Input rate**
  - Specify the number of input units to be removed in a single input pruning.
- **Input persistence**
  - Specify the number of input pruning operations to be performed if no improvement is seen.
- **Persistence**
  - Specify the number of cycles for which the network will train before attempting to prune if no improvement is seen.

Prune Method Expert Options

- **Overall persistence**
  - Specify the number of times to go through the hidden unit prune/input prune loop if no improvement is seen.
  - Applies when using the Default stopping model.

RBFN Method Expert Options



Clementine

RBFN Method Expert Options

- **RBF clusters**
  - Specify the number of radial basis functions or clusters to use.
  - This corresponds to the size of the hidden layer.
- **Persistence**
  - Specify the number of cycles for which the network will continue to train if no improvement is seen.

RBFN Method Expert Options

- **Eta**
  - For RBFNs, eta remains constant.
  - By default, eta will be computed automatically, based on the first two cycles.
  - To specify the value for eta, deselect Compute Eta automatically and enter the desired value.
- **Alpha**
  - A momentum term used in updating the weights during training.

Prune Method Expert Options

- **RBF overlapping**
  - The hidden units in an RBFN represent radial basis functions that define clusters or regions in the data.
  - This parameter allows you to control how much those regions or clusters overlap.
  - Normally during training, records affect only the cluster(s) to which they are closest.
  - By increasing this parameter, you increase the size of the region associated with each hidden unit, allowing records to affect more distant clusters.
  - Specify a positive real value.

Exhaustive Prune Method Expert Options

- There are no expert options for the Exhaustive Prune method in the Neural Net node.

Neural Net Model Nuggets

Neural Net Model Nuggets

- Neural Net model nuggets contain all of the information captured by the trained network, as well as information about the neural network's characteristics, such as accuracy and architecture.

Neural Net Model Nuggets

- **Generating new field**
  - When you execute a stream containing a Neural Net model nugget, a new field is added to the stream for each output field from the original training data.
  - The new field contains the network's prediction for the corresponding output field.
  - The name of each new prediction field is the name of the output field being predicted, with ***$N-*** added to the beginning.
  - For example, for an output field named ***profit***, the predicted values would appear in a new field called ***$N-profit***.

Neural Net Model Nuggets

- **Generating a Filter node**
  - The Generate menu allows you to create a new Filter node to pass input fields based on the results of the model.

Neural Net Model Nuggets

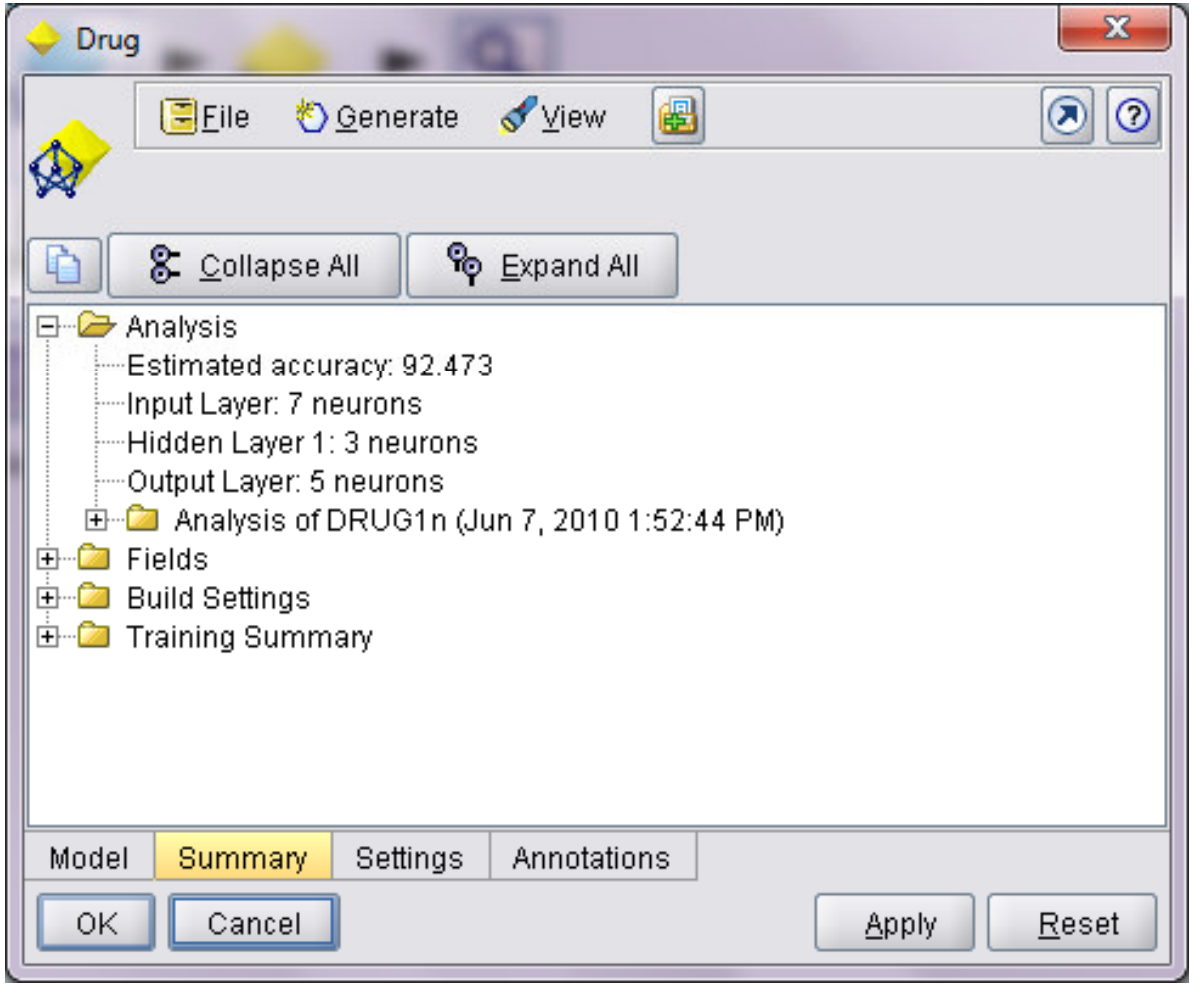
- **Variable Importance**
  - Optionally, a chart that indicates the relative importance of each variable in estimating the model may also be displayed on the Model tab.
  - Typically you will want to focus your modeling efforts on the variables that matter most and consider dropping or ignoring those that matter least.
  - Note this chart is only available if Calculate variable importance is selected on the Analyze tab before generating the model.

Neural Net Model Nugget Summary

- The **Summary tab** for a neural net model displays information about the estimated accuracy and the architecture or topology of the network.
- In addition, if you have executed an **Analysis node** attached to this modeling node, information from that analysis will also appear in this section.

Neural Net Model Nugget Summary

- Sample generated net node, Summary tab



Neural Net Model Nugget Summary

- **Estimated accuracy**
  - This is an index of the accuracy of the predictions.
  - For symbolic outputs, this is simply the percentage of records for which the predicted value is correct.
  - For numeric targets, the calculation is based on the differences between the predicted values and the actual values in the training data.
  - Because these estimates are based on the training data, they are likely to be somewhat optimistic.
  - The accuracy of the model on new data will usually be somewhat lower than this.

Neural Net Model Nugget Summary

- **Input, Hidden, and Output Layers**
  - The number of units is listed separately for each layer in the network.

References

References

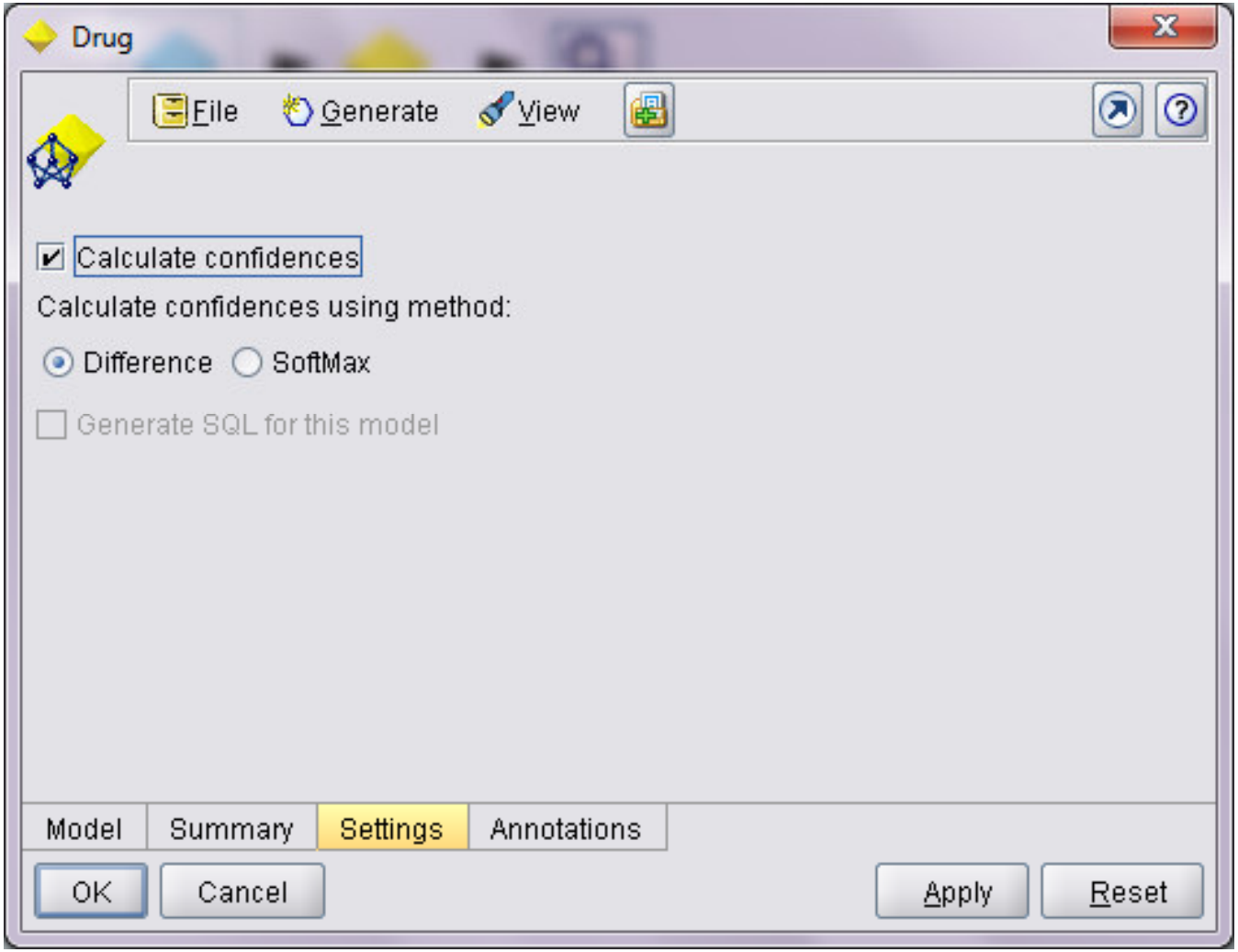
- Integral Solutions Limited., **Clementine® 12.0 User's Guide**, 2007.

The end

Neural Net Model Nugget Settings

- The **Settings tab** for a neural net model specifies how confidences are calculated and whether SQL is generated to take advantage of in-database mining.
- This tab is only available after the model nugget has been added to a stream.

Neural Net Model Nugget Settings



Neural Net Model Nugget Settings

- **Calculate confidences** (flag or set targets only)
  - For flag and set targets, you can specify whether confidences are calculated.