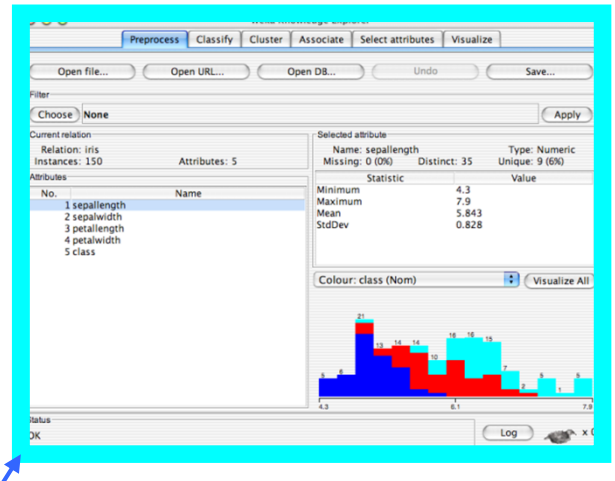
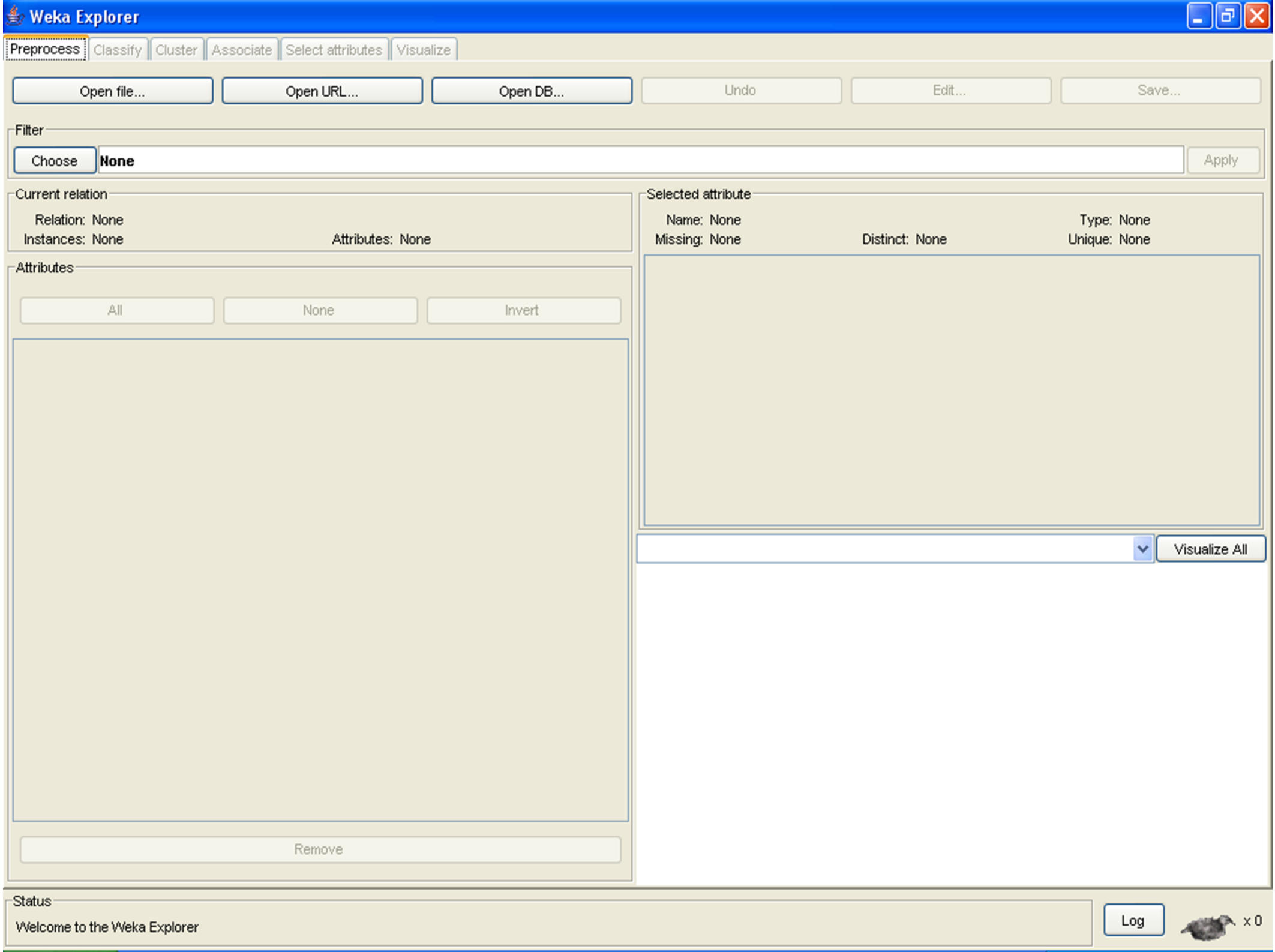
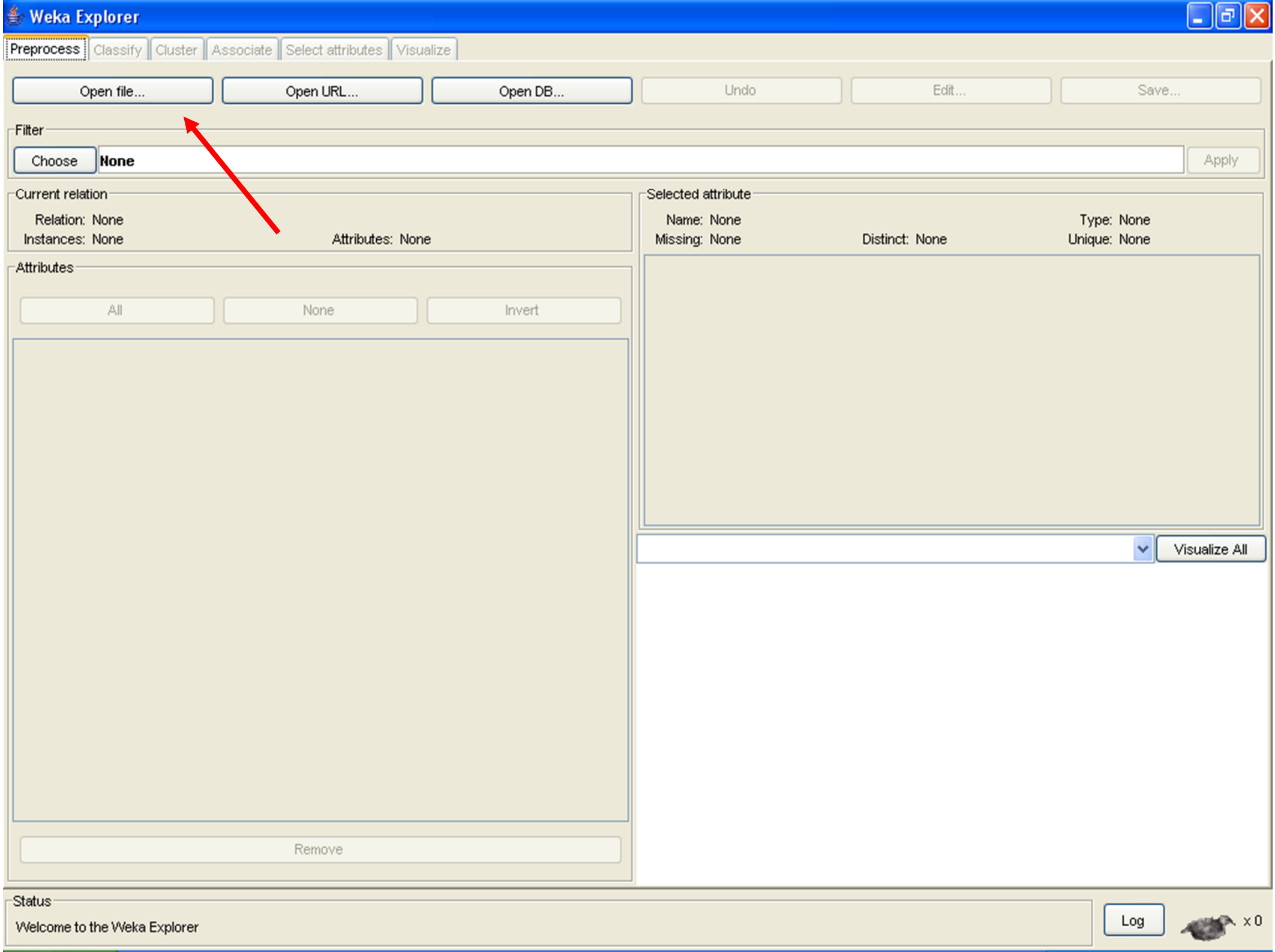
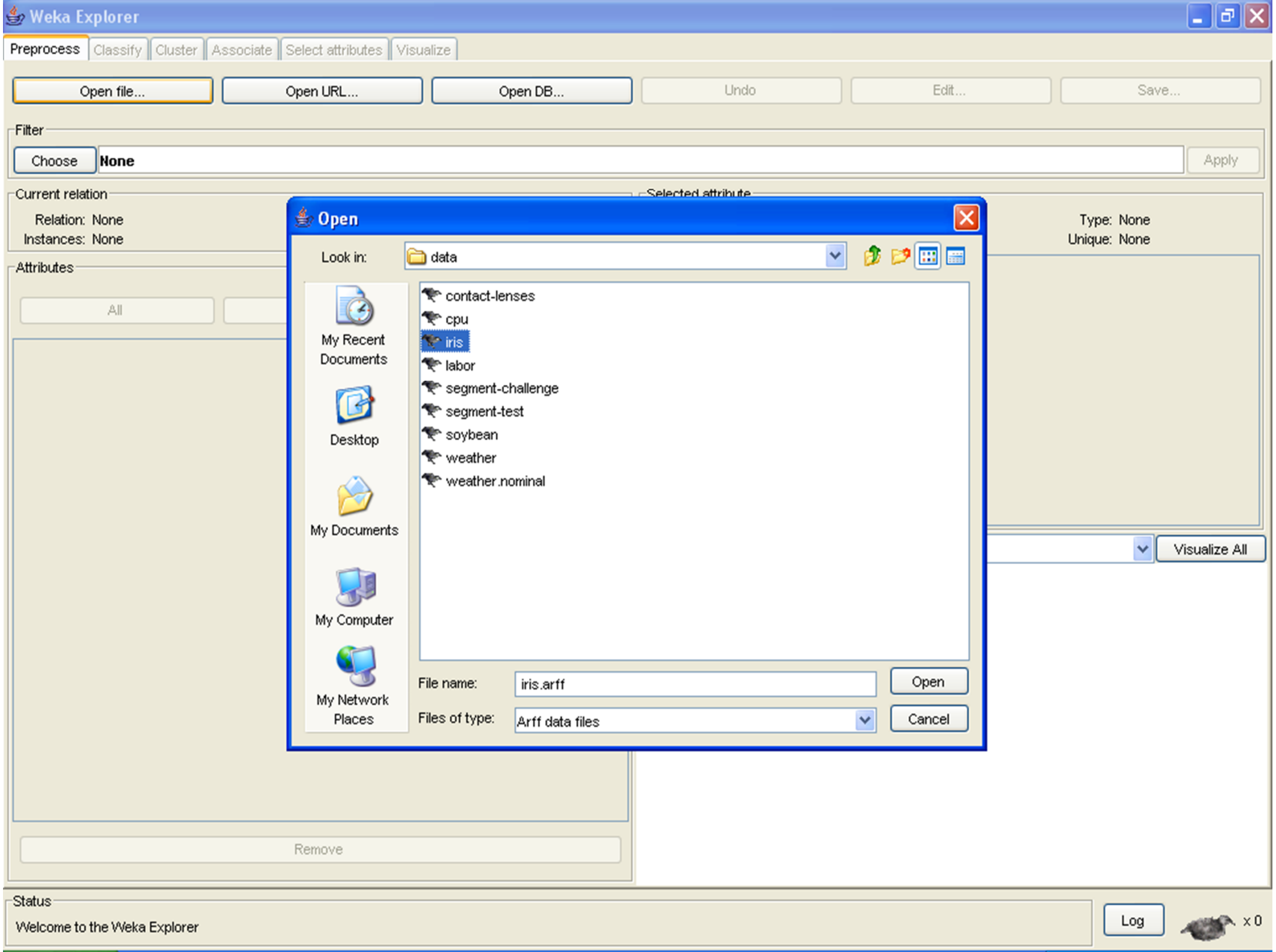
Neural Networks and SVM


Weka GUI Chooser
Waikato Environment for
Knowledge Analysis
Version 3.4.5
(c) 1999 - 2005
University of Waikato
New Zealand
GUI
Simple CLI
Explorer
Experimenter
KnowledgeFlow


Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Selected attribute
Name: sepallength
Type: Numeric
Missing: 0 (0%)
Distinct: 35
Unique: 9 (6%)
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Statistic
Value
Minimum
4.3
Maximum
7.9
Mean
5.843
StdDev
0.828
Colour: class (Nom)
Visualize All
Status
OK
Log


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: None
Instances: None
Attributes: None
Selected attribute
Name: None
Type: None
Missing: None
Distinct: None
Unique: None
Attributes
All
None
Invert
Remove
Visualize All
Status
Welcome to the Weka Explorer
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: None
Instances: None
Attributes: None
Selected attribute
Name: None
Type: None
Missing: None
Distinct: None
Unique: None
Attributes
All
None
Invert
Remove
Visualize All
Status
Welcome to the Weka Explorer
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: None
Instances: None
Selected attribute
Type: None
Unique: None
Attributes
All
Visualize All
Remove
Status
Welcome to the Weka Explorer
Log
x 0
Open
Look in:
data
contact-lenses
cpu
iris
labor
segment-challenge
segment-test
soybean
weather
weather.nominal
My Recent Documents
Desktop
My Documents
My Computer
My Network Places
File name:
iris.arff
Open
Files of type:
Arff data files
Cancel

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Undo | Edit... | Save...

Filter

Choose **None** | Apply

Current relation

Relation: iris
Instances: 150 Attributes: 5

Attributes

All | None | Invert

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

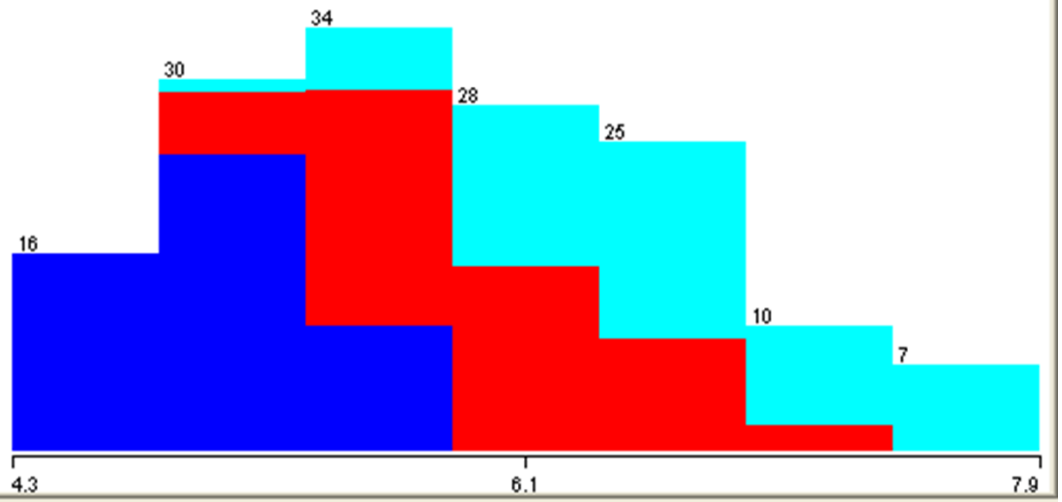
Remove

Selected attribute

Name: sepallength Type: Numeric
Missing: 0 (0%) Distinct: 35 Unique: 9 (6%)

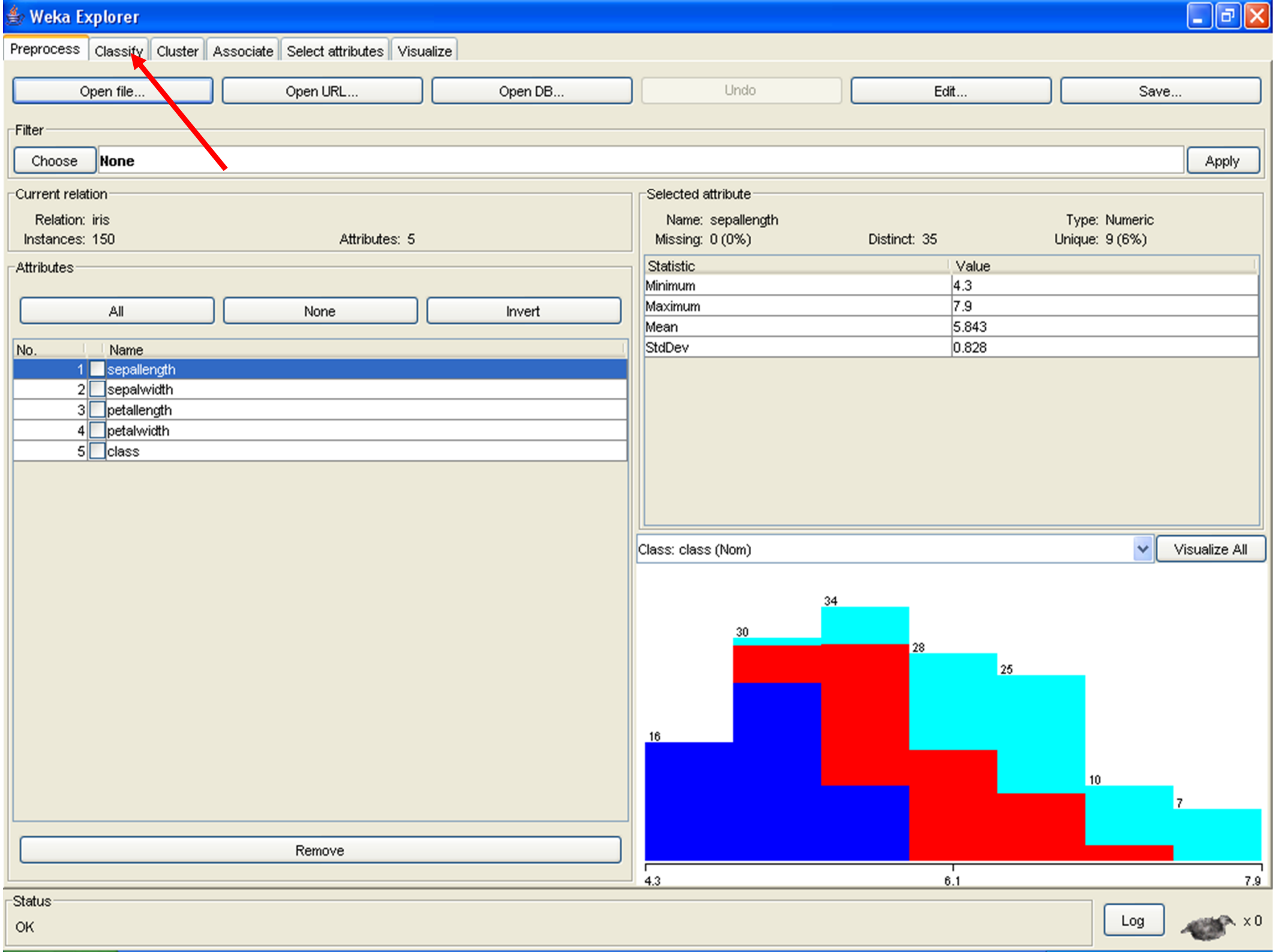
Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

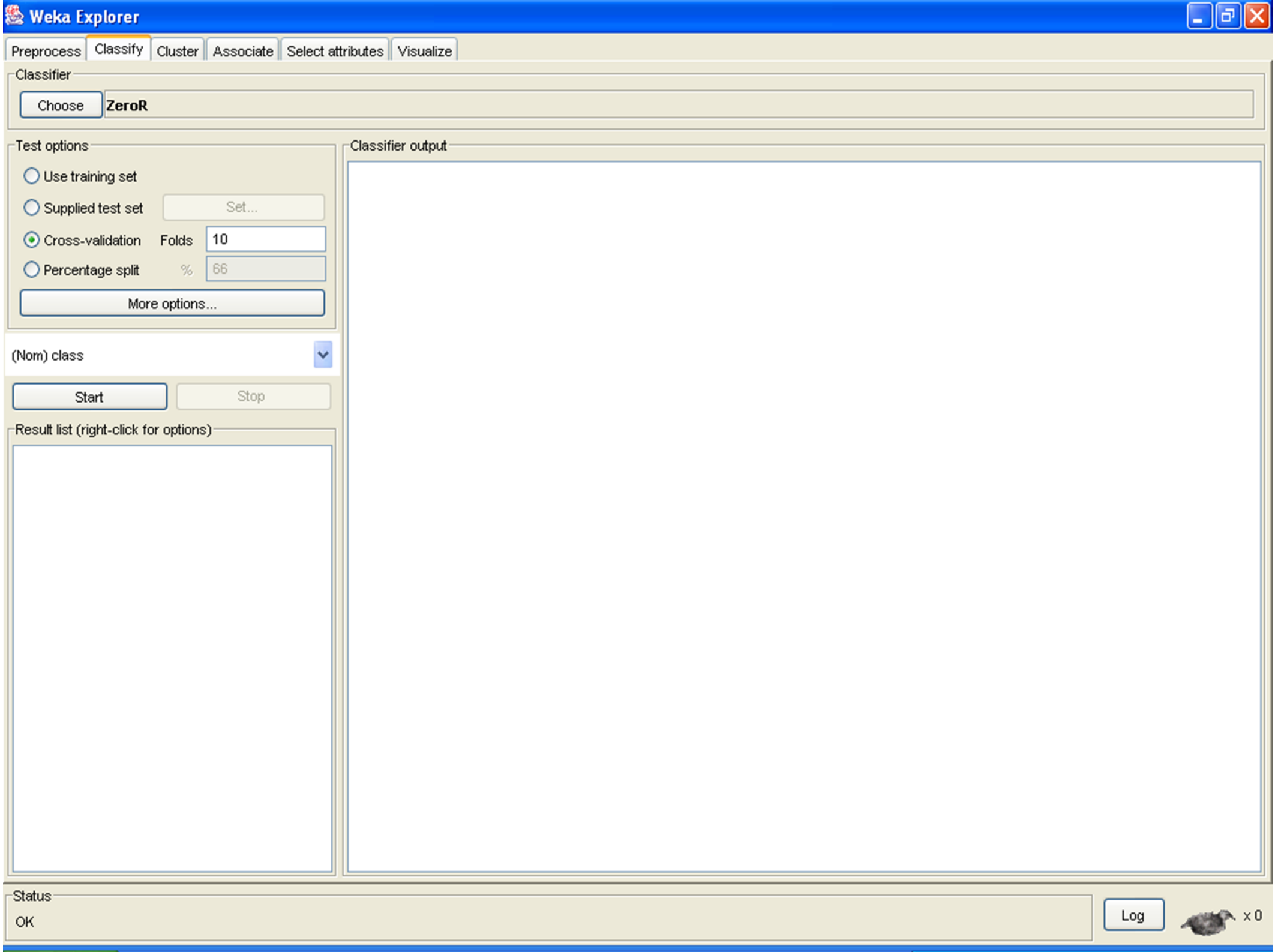
Class: class (Nom) | Visualize All

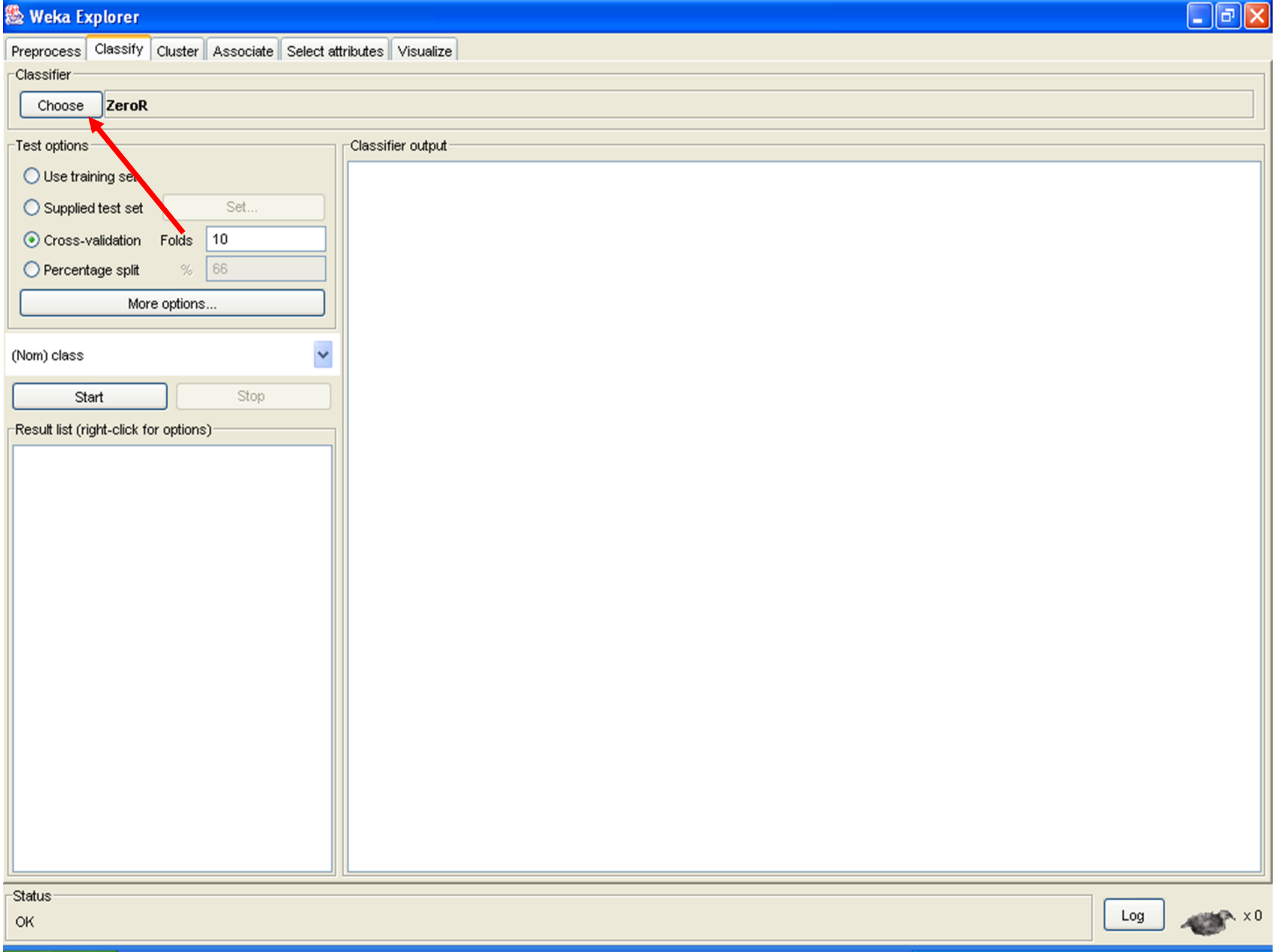


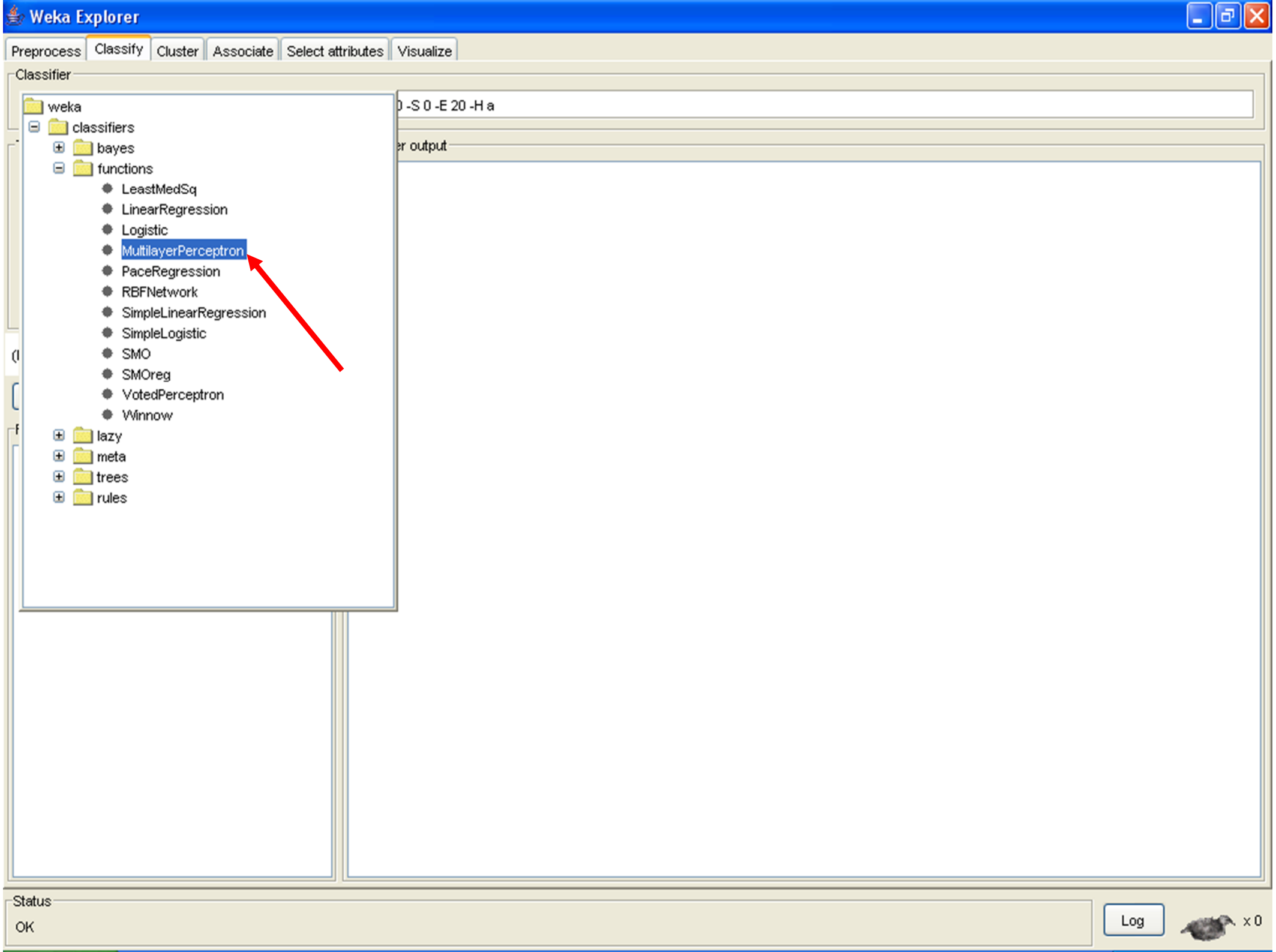
Status

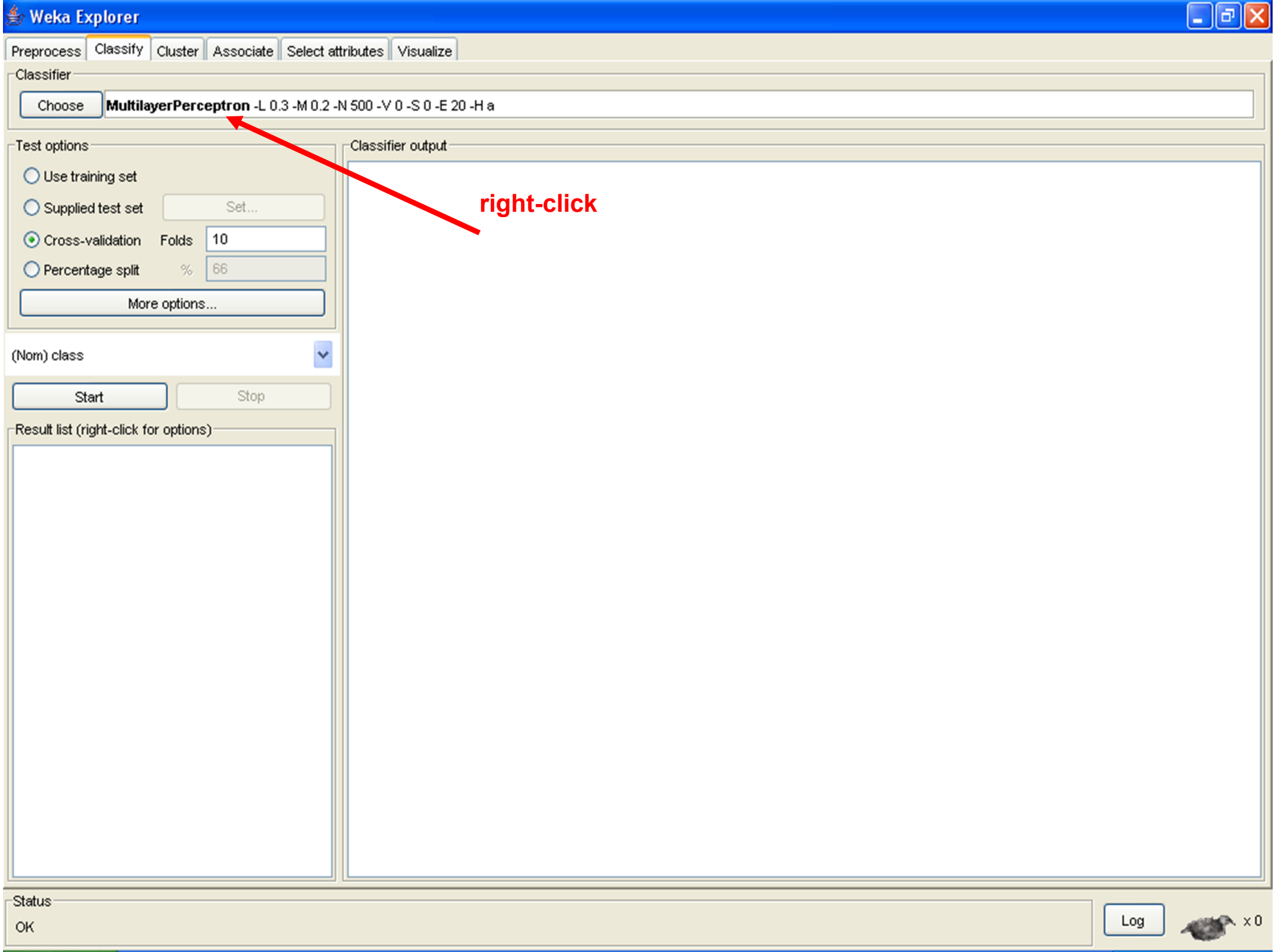
OK | Log | x 0

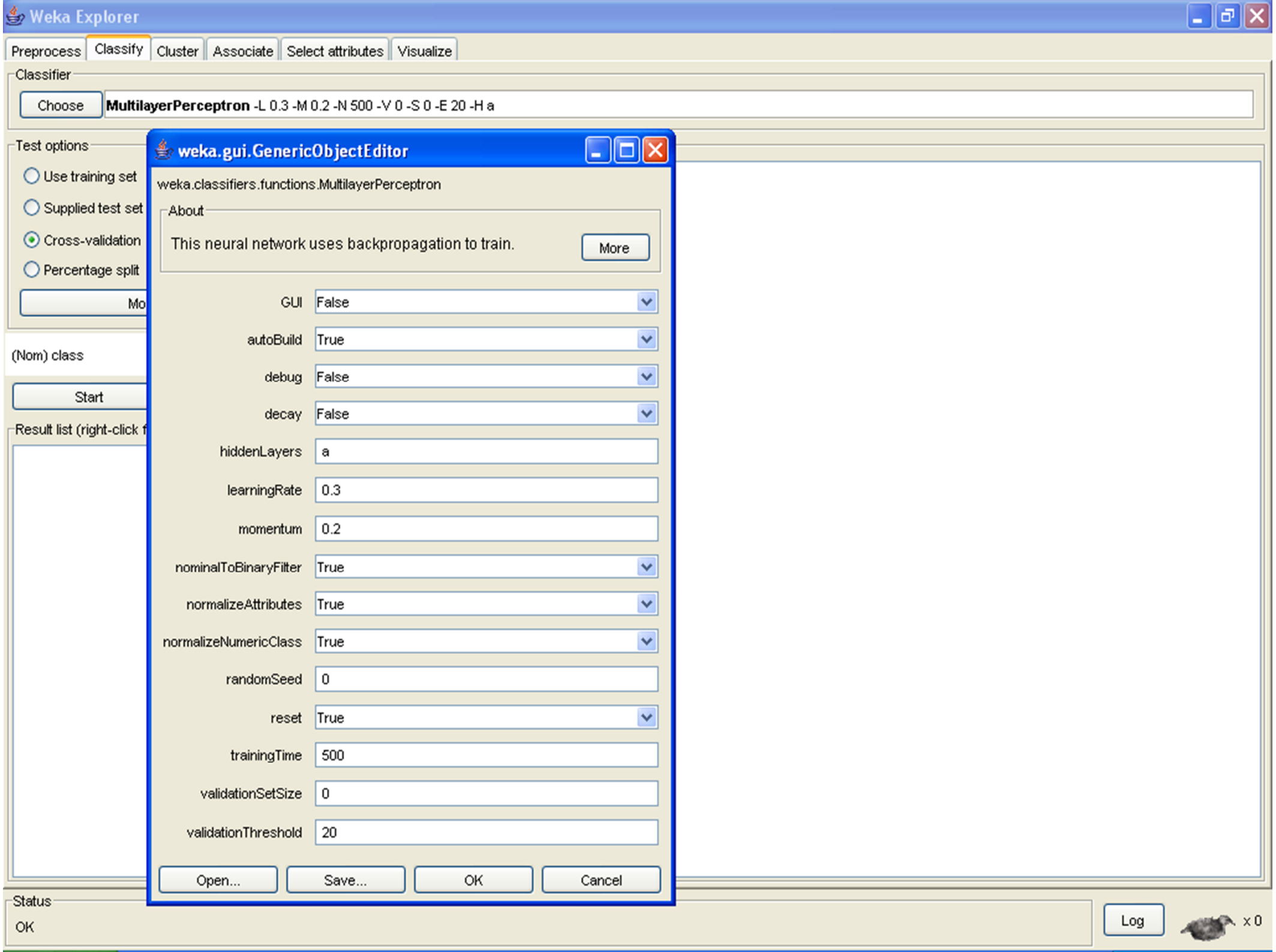

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Attributes
All
None
Invert
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Remove
Selected attribute
Name: sepallength
Type: Numeric
Missing: 0 (0%)
Distinct: 35
Unique: 9 (6%)
Statistic
Value
Minimum 4.3
Maximum 7.9
Mean 5.843
StdDev 0.828
Class: class (Nom)
Visualize All
34
30
28
25
16
10
7
4.3
6.1
7.9
Status
OK
Log
x 0

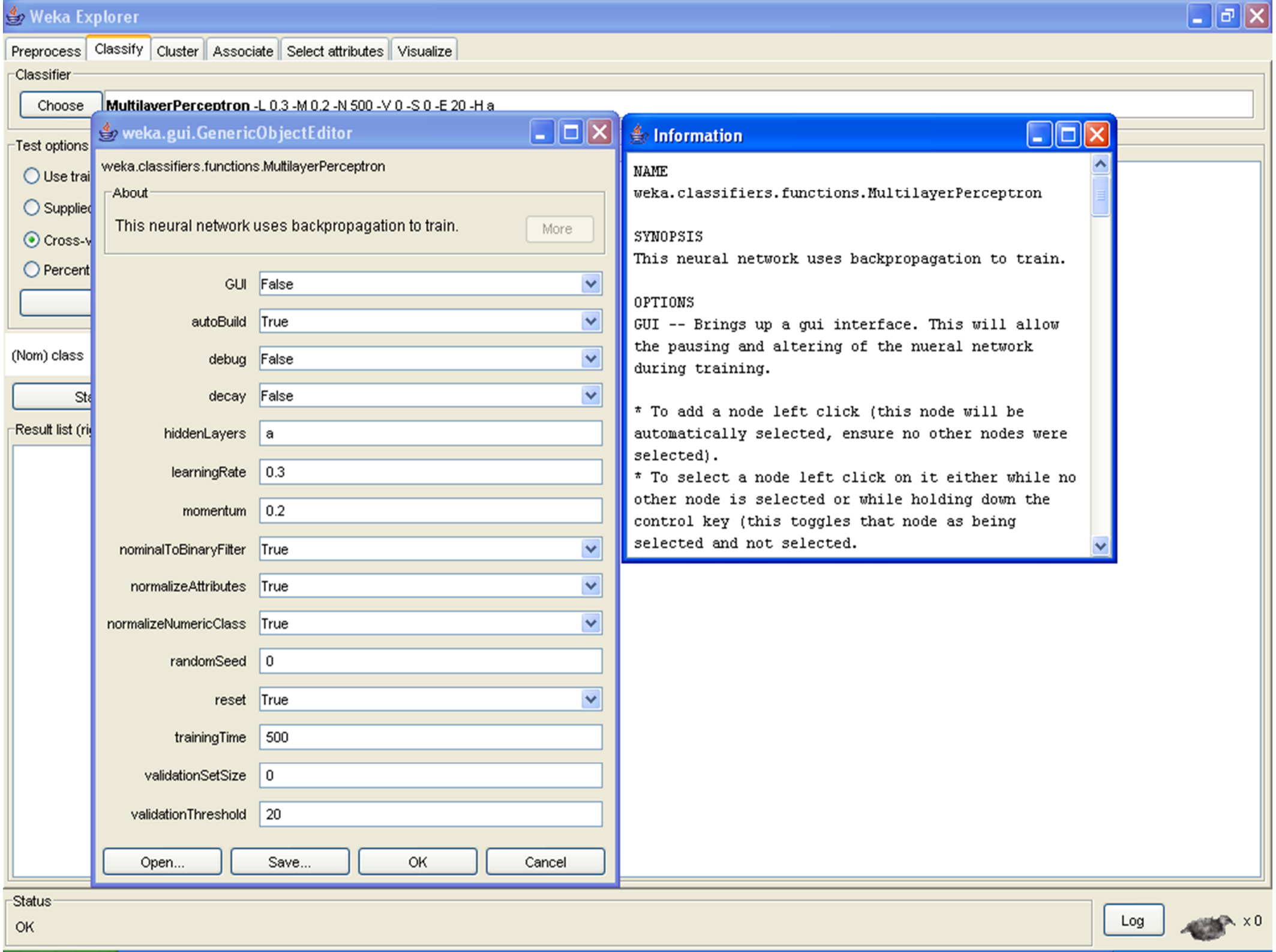

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ZeroR
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
(Nom) class
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ZeroR
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
(Nom) class
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
weka
classifiers
bayes
functions
LeastMedSq
LinearRegression
Logistic
MultilayerPerceptron
PaceRegression
RBFNetwork
SimpleLinearRegression
SimpleLogistic
SMO
SMOreg
VotedPerceptron
Winnow
lazy
meta
trees
rules
0 -S 0 -E 20 -H a
er output
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
right-click
(Nom) class
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
Mo
(Nom) class
Start
Result list (right-click f
Status
OK
Log
x 0
weka.gui.GenericObjectEditor
weka.classifiers.functions.MultilayerPerceptron
About
This neural network uses backpropagation to train.
More
GUI
False
autoBuild
True
debug
False
decay
False
hiddenLayers
a
learningRate
0.3
momentum
0.2
nominalToBinaryFilter
True
normalizeAttributes
True
normalizeNumericClass
True
randomSeed
0
reset
True
trainingTime
500
validationSetSize
0
validationThreshold
20
Open...
Save...
OK
Cancel


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Test options
Use trai
Supplied
Cross-v
Percent
(Nom) class
Sta
Result list (ri
weka.gui.GenericObjectEditor
weka.classifiers.functions.MultilayerPerceptron
About
This neural network uses backpropagation to train.
More
GUI
False
autoBuild
True
debug
False
decay
False
hiddenLayers
a
learningRate
0.3
momentum
0.2
nominalToBinaryFilter
True
normalizeAttributes
True
normalizeNumericClass
True
randomSeed
0
reset
True
trainingTime
500
validationSetSize
0
validationThreshold
20
Open...
Save...
OK
Cancel
Information
NAME
weka.classifiers.functions.MultilayerPerceptron
SYNOPSIS
This neural network uses backpropagation to train.
OPTIONS
GUI -- Brings up a gui interface. This will allow the pausing and altering of the nueral network during training.
* To add a node left click (this node will be automatically selected, ensure no other nodes were selected).
* To select a node left click on it either while no other node is selected or while holding down the control key (this toggles that node as being selected and not selected.
Status
OK
Log
x 0

Information

NAME
weka.classifiers.functions.MultilayerPerceptron

SYNOPSIS
This neural network uses backpropagation to train.

OPTIONS
GUI -- Brings up a gui interface. This will allow the pausing and altering of the nueral network during training.

* To add a node left click (this node will be automatically selected, ensure no other nodes were selected).
* To select a node left click on it either while no other node is selected or while holding down the control key (this toggles that node as being selected and not selected.
* To connect a node, first have the start node(s) selected, then click either the end node or on an empty space (this will create a new node that is connected with the selected nodes). The selection status of nodes will stay the same after the connection. (Note these are directed connections, also a connection between two nodes will not be established more than once and certain connections that are deemed to be invalid will not be made).
* To remove a connection select one of the connected node(s) in the connection and then right click the other node (it does not matter whether the node is the start or end the connection will be removed).
* To remove a node right click it while no other nodes (including it) are selected. (This will also remove all connections to it)
.* To deselect a node either left click it while holding down control, or right click on empty space.
* The raw inputs are provided from the labels on the left.
* The red nodes are hidden layers.
* The orange nodes are the output nodes.
* The labels on the right show the class the output node represents. Note that with a numeric class the output node will automatically be made into an unthresholded linear unit.

Alterations to the neural network can only be done while the network is not running, This also applies to the learning rate and other fields on the control panel.

* You can accept the network as being finished at any time.
* The network is automatically paused at the beginning.
* There is a running indication of what epoch the network is up to and what the (rough) error for that epoch was (or for the validation if that is being used). Note that this error value is based on a network that changes as the value is computed. (also depending on whether the class is normalized will effect the error reported for numeric classes.
* Once the network is done it will pause again and either wait to be accepted or trained more.

Note that if the gui is not set the network will not require any interaction.

autoBuild -- Adds and connects up hidden layers in the network.

Information

debug -- If set to true, classifier may output additional info to the console.

decay -- This will cause the learning rate to decrease. This will divide the starting learning rate by the epoch number, to determine what the current learning rate should be. This may help to stop the network from diverging from the target output, as well as improve general performance. Note that the decaying learning rate will not be shown in the gui, only the original learning rate. If the learning rate is changed in the gui, this is treated as the starting learning rate.

hiddenLayers -- This defines the hidden layers of the neural network. This is a list of positive whole numbers. 1 for each hidden layer. Comma seperated. To have no hidden layers put a single 0 here. This will only be used if autobuild is set. There are also wildcard values 'a' = (attribs + classes) / 2, 'i' = attribs, 'o' = classes , 't' = attribs + classes.

learningRate -- The amount the weights are updated.

momentum -- Momentum applied to the weights during updating.

nominalToBinaryFilter -- This will preprocess the instances with the filter. This could help improve performance if there are nominal attributes in the data.

normalizeAttributes -- This will normalize the attributes. This could help improve performance of the network. This is not reliant on the class being numeric. This will also normalize nominal attributes as well (after they have been run through the nominal to binary filter if that is in use) so that the nominal values are between -1 and 1

normalizeNumericClass -- This will normalize the class if it's numeric. This could help improve performance of the network, It normalizes the class to be between -1 and 1. Note that this is only internally, the output will be scaled back to the original range.

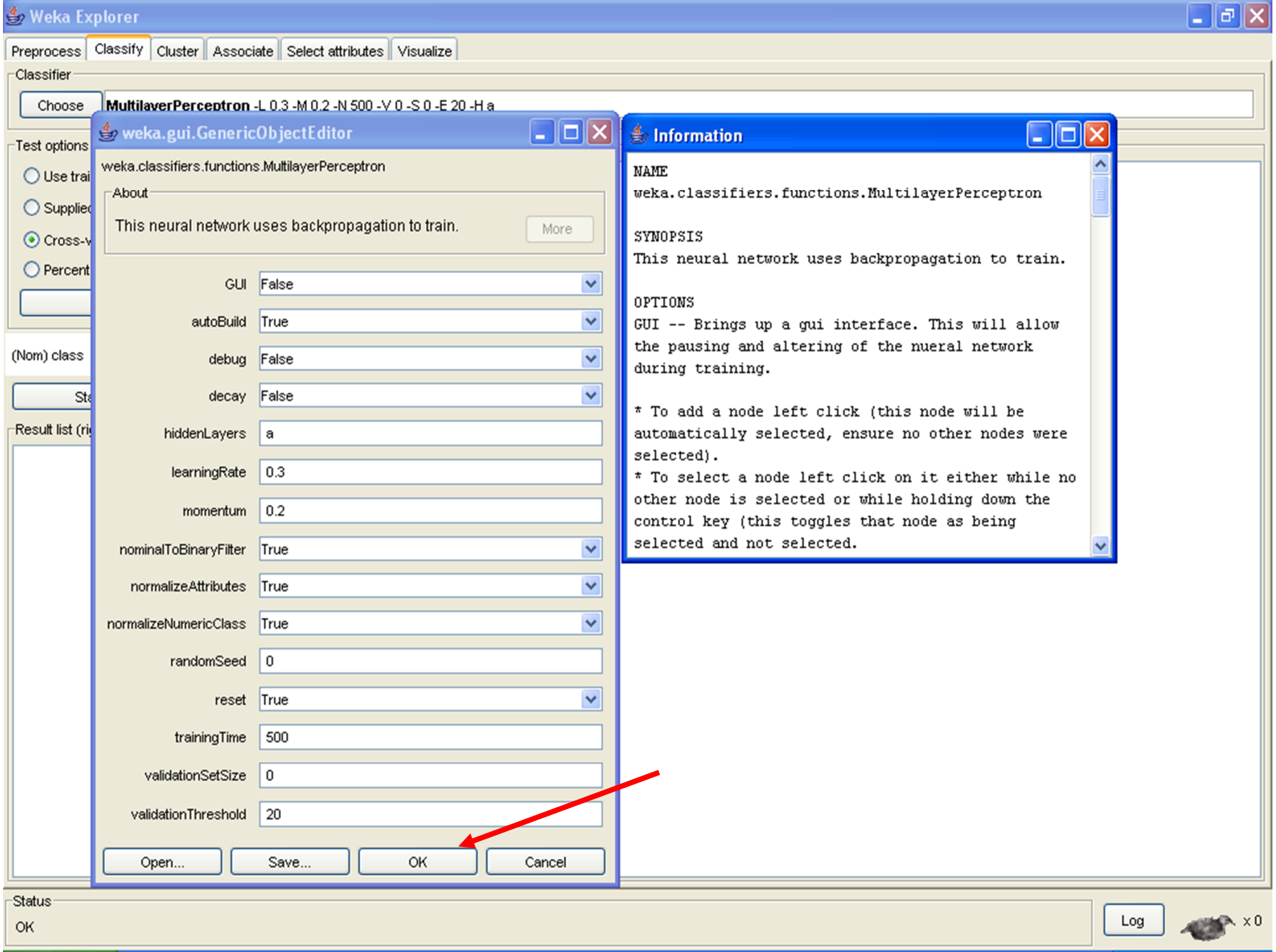
randomSeed -- Seed used to initialise the random number generator.Random numbers are used for setting the initial weights of the connections betweem nodes, and also for shuffling the training data.

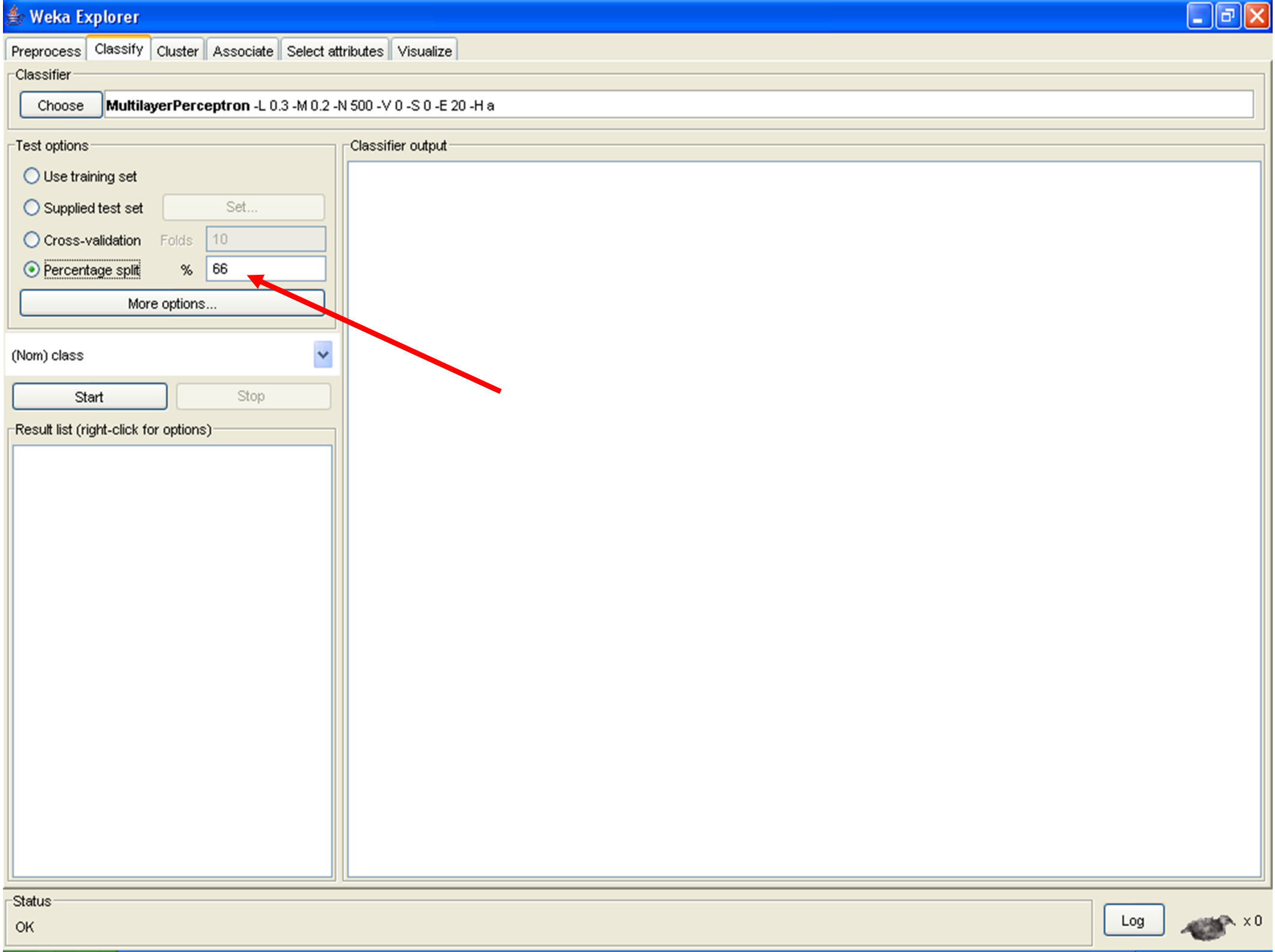
reset -- This will allow the network to reset with a lower learning rate. If the network diverges from the answer this will automatically reset the network with a lower learning rate and begin training again. This option is only available if the gui is not set. Note that if the network diverges but isn't allowed to reset it will fail the training process and return an error message.

trainingTime -- The number of epochs to train through. If the validation set is non-zero then it can terminate the network early

validationSetSize -- The percentage size of the validation set.(The training will continue until it is observed that the error on the validation set has been consistently getting worse, or if the training time is reached).
If This is set to zero no validation set will be used and instead the network will train for the specified number of epochs.

validationThreshold -- Used to terminate validation testing.The value here dictates how many times in a row the validation set error can get worse before training is terminated.


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Test options
Use trai
Supplie
Cross-v
Percent
(Nom) class
Sta
Result list (ri
weka.gui.GenericObjectEditor
weka.classifiers.functions.MultilayerPerceptron
About
This neural network uses backpropagation to train.
More
GUI
False
autoBuild
True
debug
False
decay
False
hiddenLayers
a
learningRate
0.3
momentum
0.2
nominalToBinaryFilter
True
normalizeAttributes
True
normalizeNumericClass
True
randomSeed
0
reset
True
trainingTime
500
validationSetSize
0
validationThreshold
20
Open...
Save...
OK
Cancel
Information
NAME
weka.classifiers.functions.MultilayerPerceptron
SYNOPSIS
This neural network uses backpropagation to train.
OPTIONS
GUI -- Brings up a gui interface. This will allow the pausing and altering of the nueral network during training.
* To add a node left click (this node will be automatically selected, ensure no other nodes were selected).
* To select a node left click on it either while no other node is selected or while holding down the control key (this toggles that node as being selected and not selected.
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
(Nom) class
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier

Choose **MultilayerPerceptron** -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 66

More options...

(Nom) class

Start | Stop

Result list (right-click for options)

09:17:47 - functions.MultilayerPerceptron

Classifier output

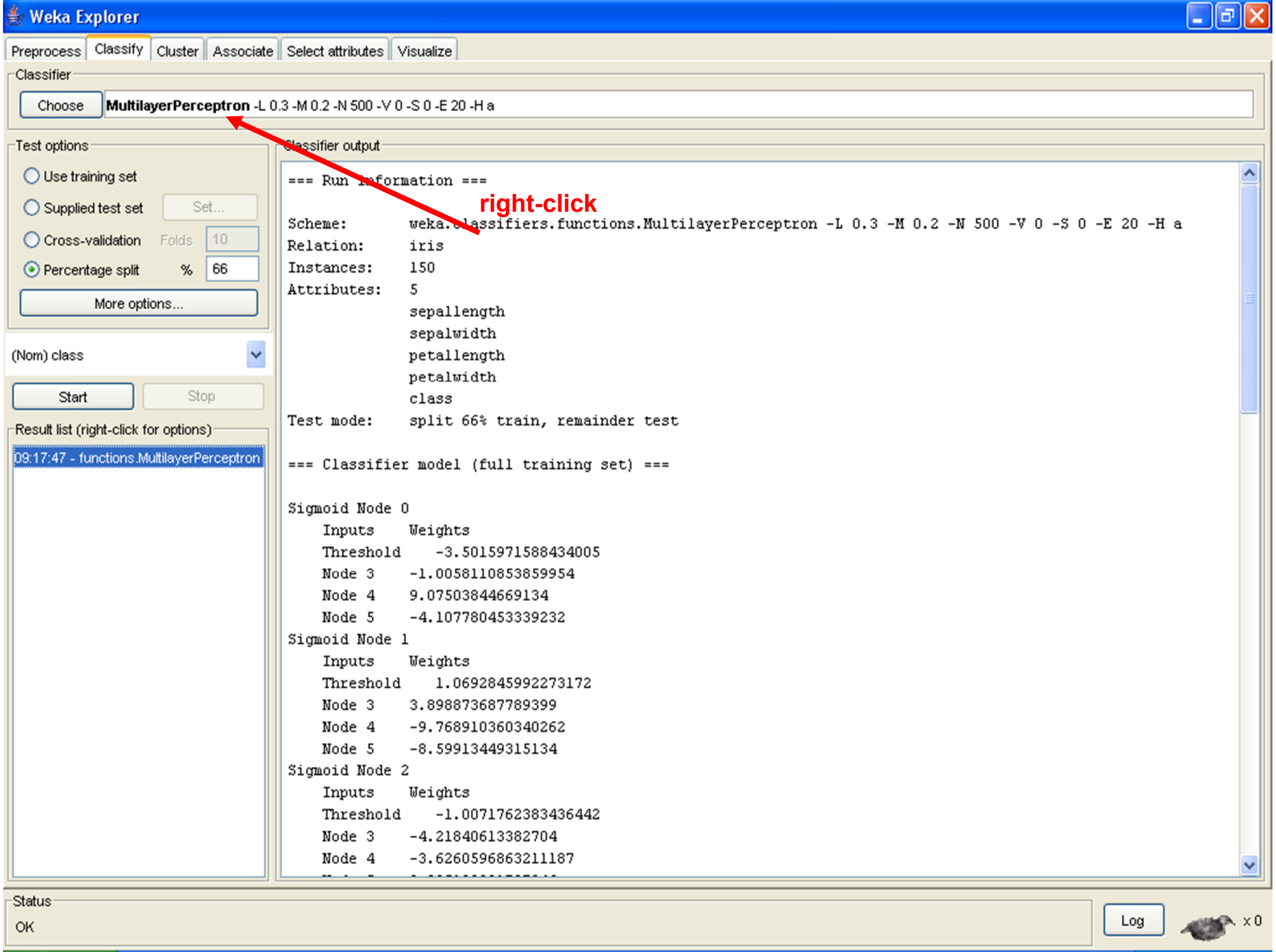
```
=== Run information ===

Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    split 66% train, remainder test

=== Classifier model (full training set) ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    -3.5015971588434005
    Node 3    -1.0058110853859954
    Node 4    9.07503844669134
    Node 5    -4.107780453339232
Sigmoid Node 1
    Inputs    Weights
    Threshold    1.0692845992273172
    Node 3    3.898873687789399
    Node 4    -9.768910360340262
    Node 5    -8.59913449315134
Sigmoid Node 2
    Inputs    Weights
    Threshold    -1.0071762383436442
    Node 3    -4.21840613382704
    Node 4    -3.6260596863211187
```

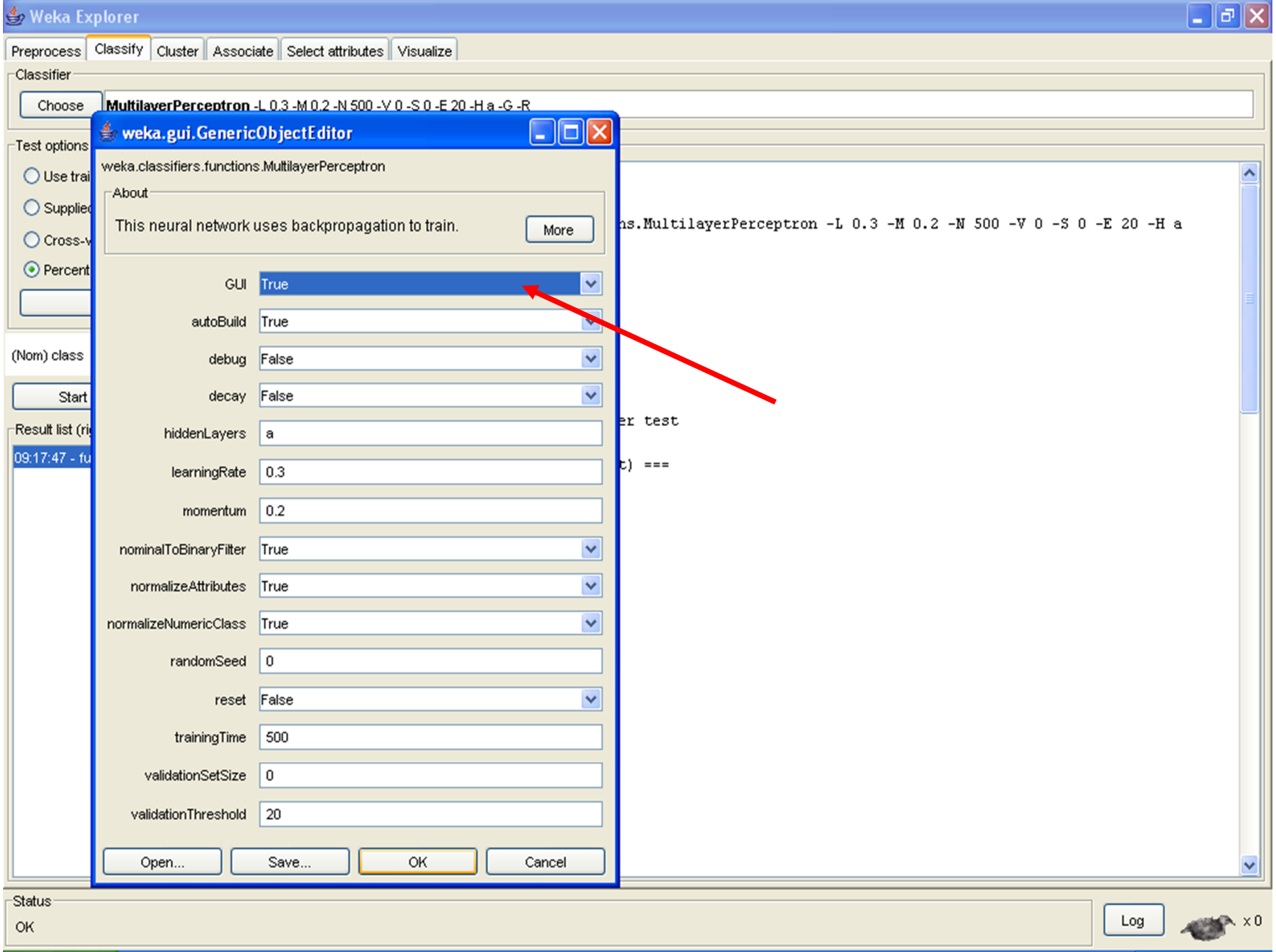
Status

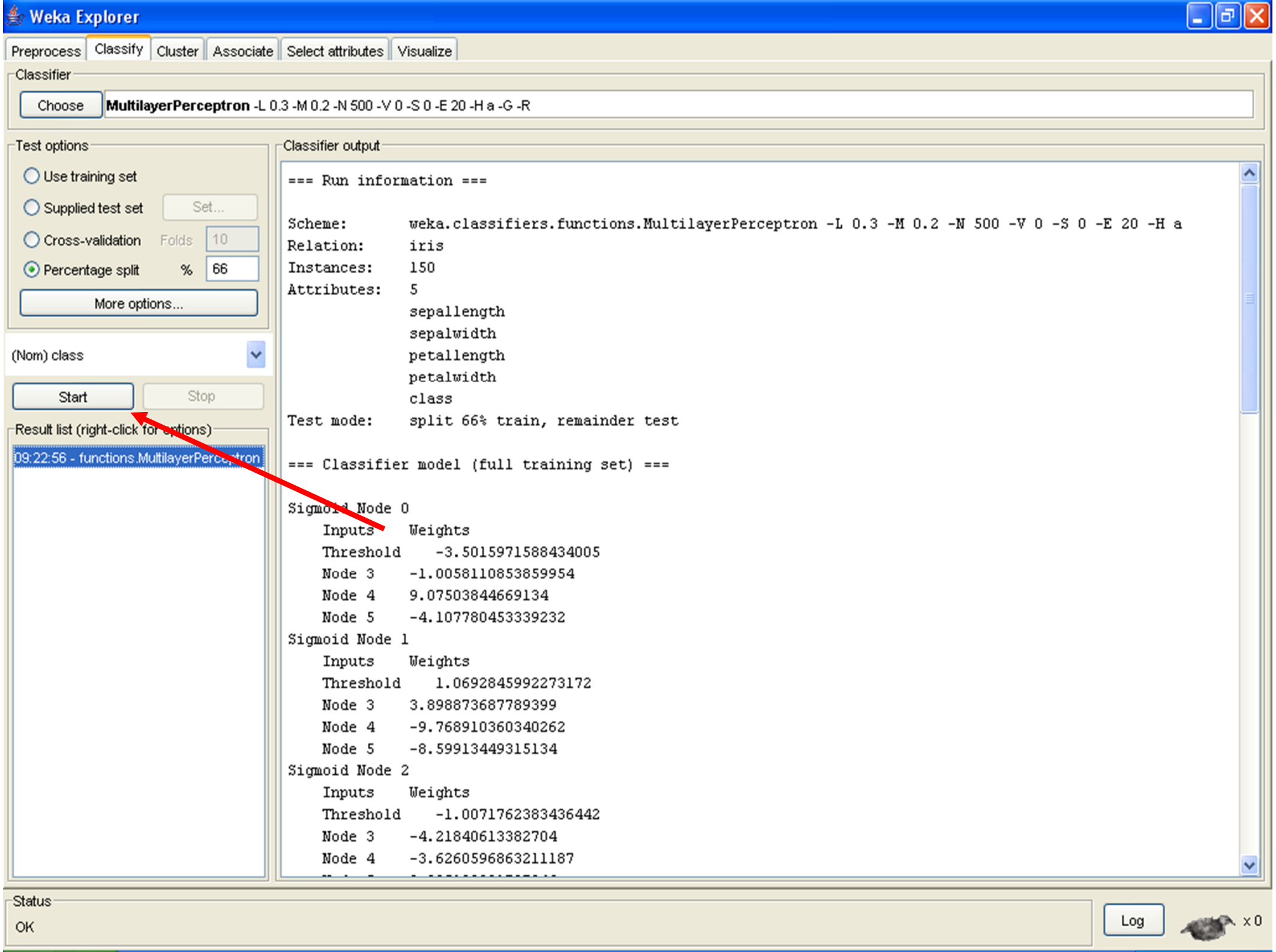
OK

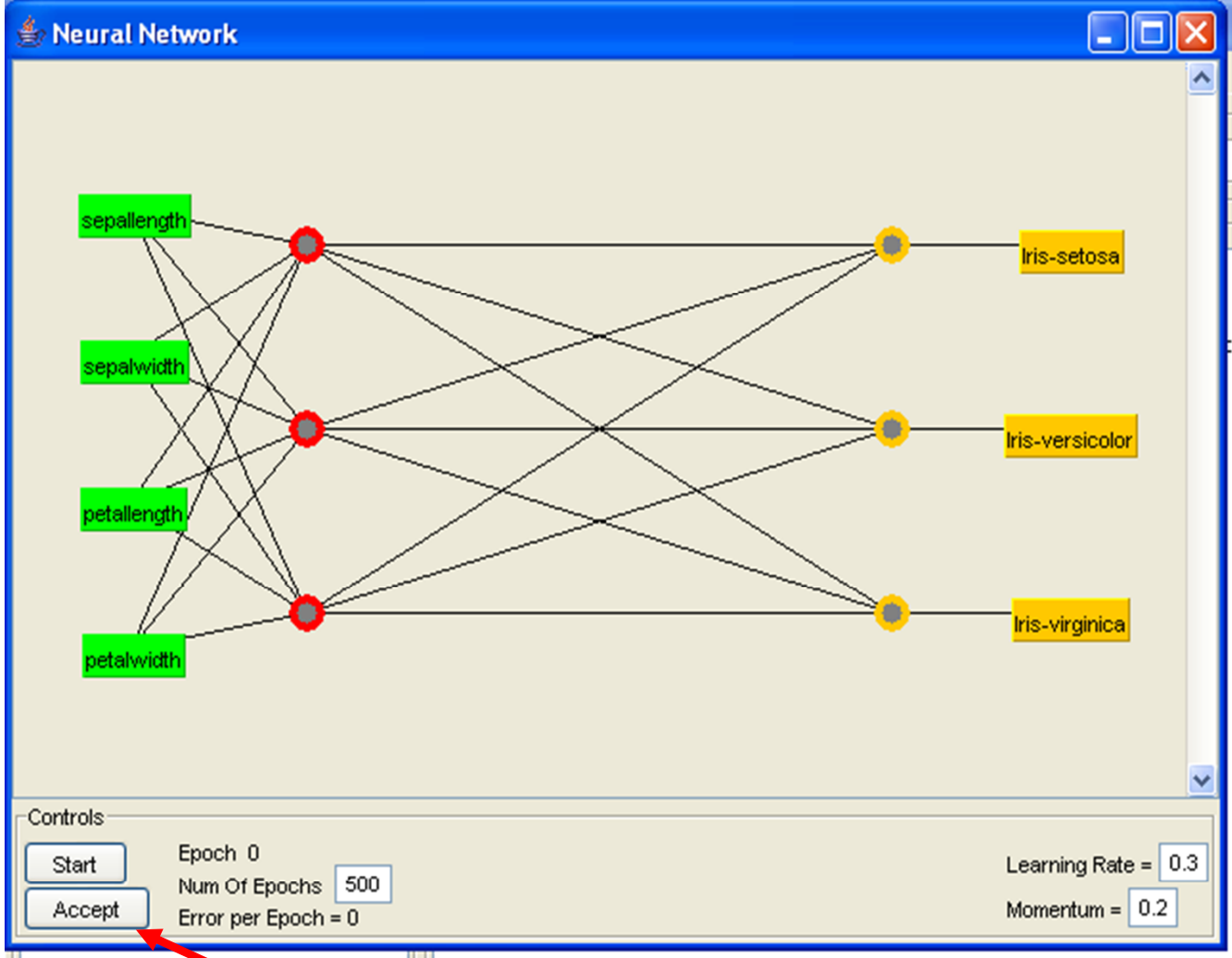
Log

x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
right-click
=== Run Information ===
Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
class
Test mode: split 66% train, remainder test
=== Classifier model (full training set) ===
Sigmoid Node 0
Inputs Weights
Threshold -3.5015971588434005
Node 3 -1.0058110853859954
Node 4 9.07503844669134
Node 5 -4.107780453339232
Sigmoid Node 1
Inputs Weights
Threshold 1.0692845992273172
Node 3 3.898873687789399
Node 4 -9.768910360340262
Node 5 -8.59913449315134
Sigmoid Node 2
Inputs Weights
Threshold -1.0071762383436442
Node 3 -4.21840613382704
Node 4 -3.6260596863211187
(Nom) class
Start
Stop
Result list (right-click for options)
09:17:47 - functions.MultilayerPerceptron
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R
Test options
Use trai
Supplied
Cross-v
Percent
(Nom) class
Start
Result list (ri
09:17:47 - fu
weka.gui.GenericObjectEditor
weka.classifiers.functions.MultilayerPerceptron
About
This neural network uses backpropagation to train.
More
GUI
True
autoBuild
True
debug
False
decay
False
hiddenLayers
a
learningRate
0.3
momentum
0.2
nominalToBinaryFilter
True
normalizeAttributes
True
normalizeNumericClass
True
randomSeed
0
reset
False
trainingTime
500
validationSetSize
0
validationThreshold
20
Open...
Save...
OK
Cancel
ns.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
er test
t) ===
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
09:22:56 - functions.MultilayerPerceptron
Classifier output
=== Run information ===
Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
class
Test mode: split 66% train, remainder test
=== Classifier model (full training set) ===
Sigmoid Node 0
Inputs Weights
Threshold -3.5015971588434005
Node 3 -1.0058110853859954
Node 4 9.07503844669134
Node 5 -4.107780453339232
Sigmoid Node 1
Inputs Weights
Threshold 1.0692845992273172
Node 3 3.898873687789399
Node 4 -9.768910360340262
Node 5 -8.59913449315134
Sigmoid Node 2
Inputs Weights
Threshold -1.0071762383436442
Node 3 -4.21840613382704
Node 4 -3.6260596863211187
Status
OK
Log
x 0



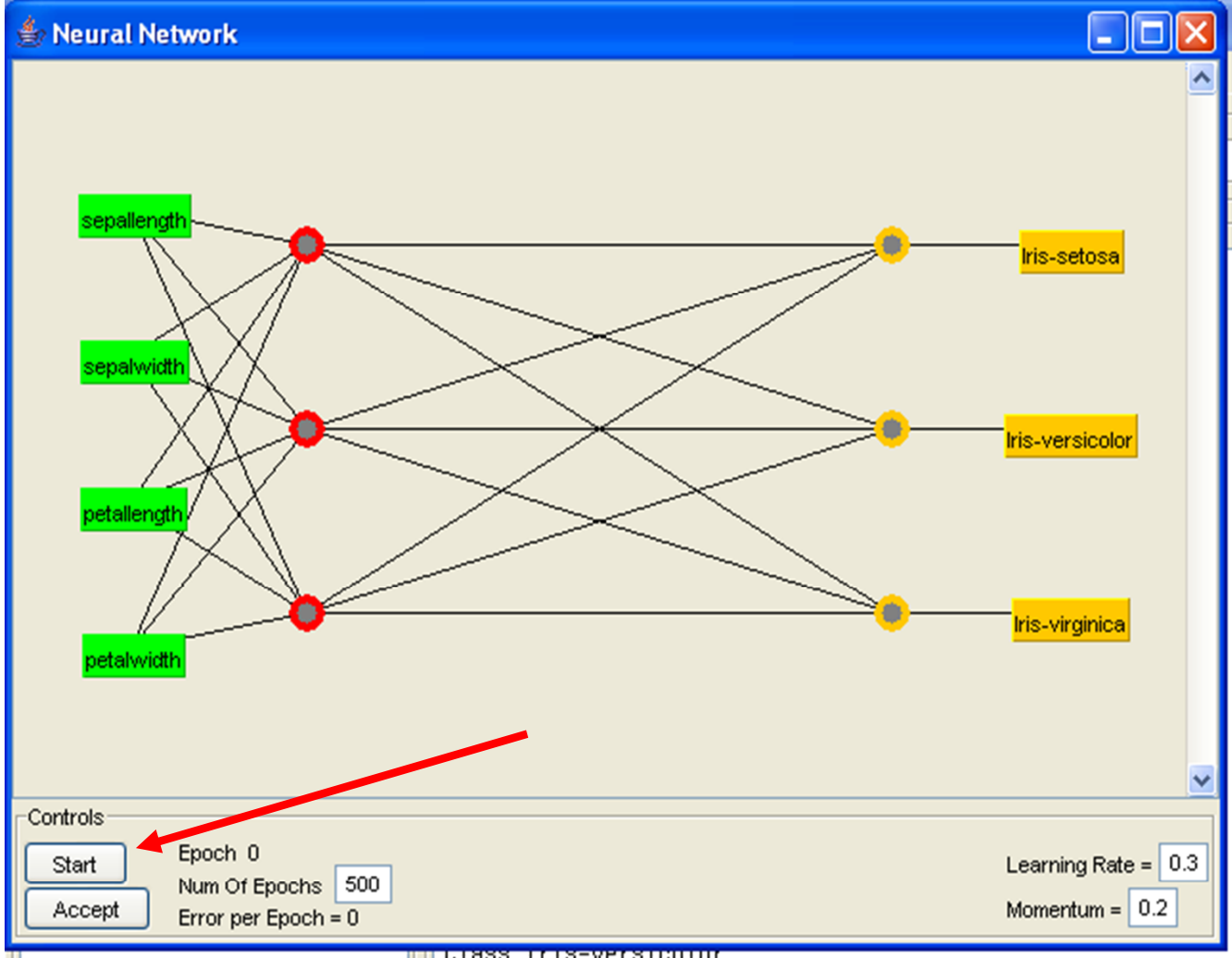
ton -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R

Status

Building model on training data...

Log

x 1



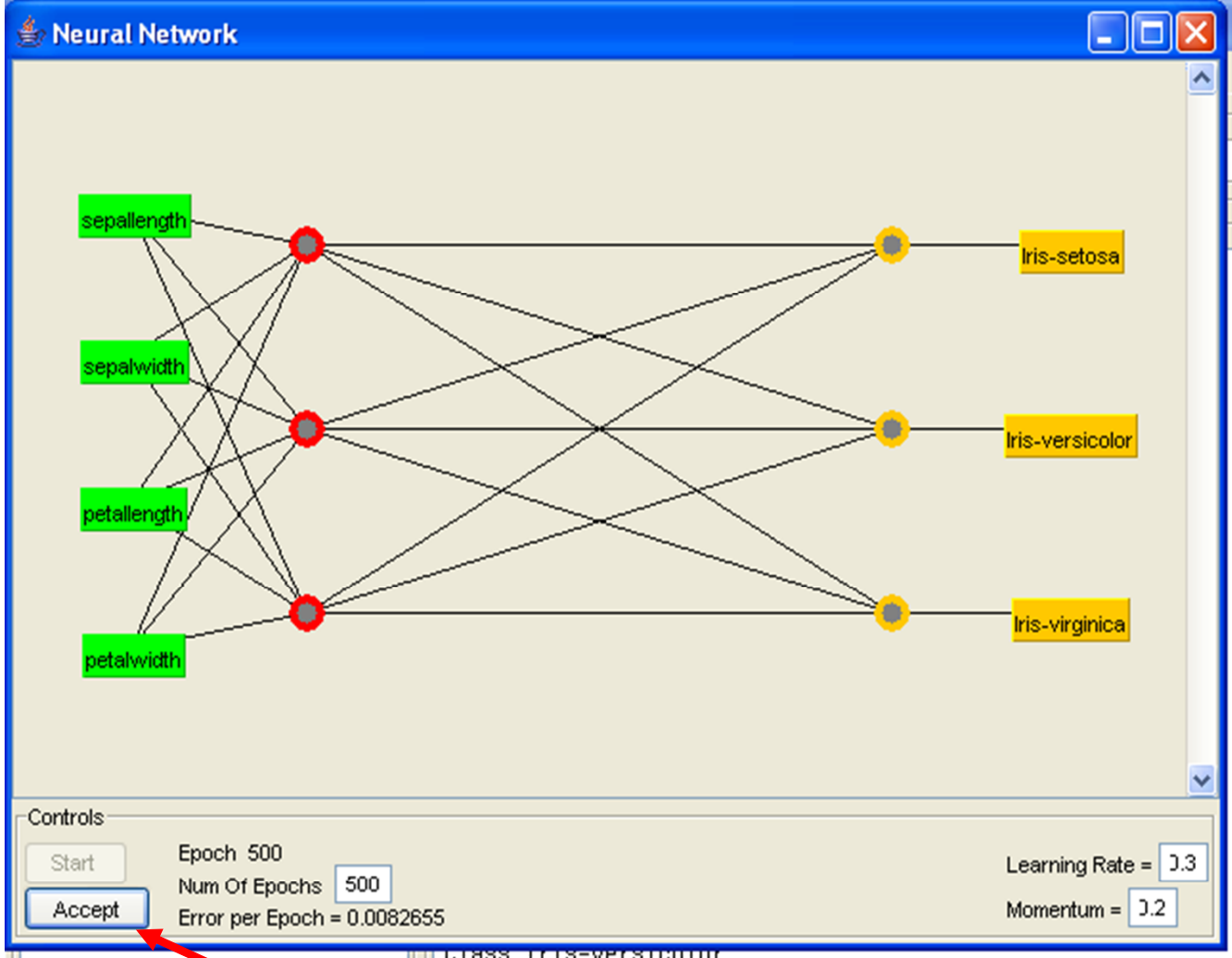

```
Class Iris-versicolor
    Input
    Node 1
Class Iris-virginica
    Input
    Node 2


Time taken to build model: 92.3 seconds
```

Status

Building model on training split...

Log

x 1




```
Class Iris-versicolor
    Input
    Node 1
Class Iris-virginica
    Input
    Node 2


Time taken to build model: 92.3 seconds
```

Status

Building model on training split...

Log

x 1

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier

Choose **MultilayerPerceptron** -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 66

More options...

(Nom) class

Start | Stop

Result list (right-click for options)

10:07:52 - functions.MultilayerPerceptron
10:08:05 - functions.MultilayerPerceptron

Classifier output

```
Time taken to build model: 92.3 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          50               98.0392 %
Incorrectly Classified Instances         1                1.9608 %
Kappa statistic                          0.9704
Mean absolute error                      0.0239
Root mean squared error                  0.1101
Relative absolute error                  5.3594 %
Root relative squared error             23.2952 %
Total Number of Instances               51

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  1          0          1         1         1        Iris-setosa
  1          0.031      0.95      1         0.974    Iris-versicolor
  0.941      0          1         0.941     0.97     Iris-virginica

=== Confusion Matrix ===

  a  b  c   <-- classified as
 15  0  0 |  a = Iris-setosa
  0 19  0 |  b = Iris-versicolor
  0  1 16 |  c = Iris-virginica
```

Status

OK

Log

x 0

Try with different network topologies

Other Neural Network Software

- SNNS- Stuttgart Neural Network Simulator

http://www-ra.informatik.uni-tuebingen.de/SNNS/

- Alyuda NeuroIntelligence

http://www.alyuda.com/neural-networks-software.htm

- "backprop.cpp" from V. Rao and H. Rao

C++ Neural Network and Fuzzy Logic

Backpropagation Simulation version 1

Using “backprop.cpp”

- Download the executable file as well as the example training and testing data sets
- The training and data set were obtained from the UCI Machine Learning Repository’s Wine Recognition Database
- The original data set was randomly divided to create the training (2/3) and testing (1/3) data sets

Using “backprop.cpp”

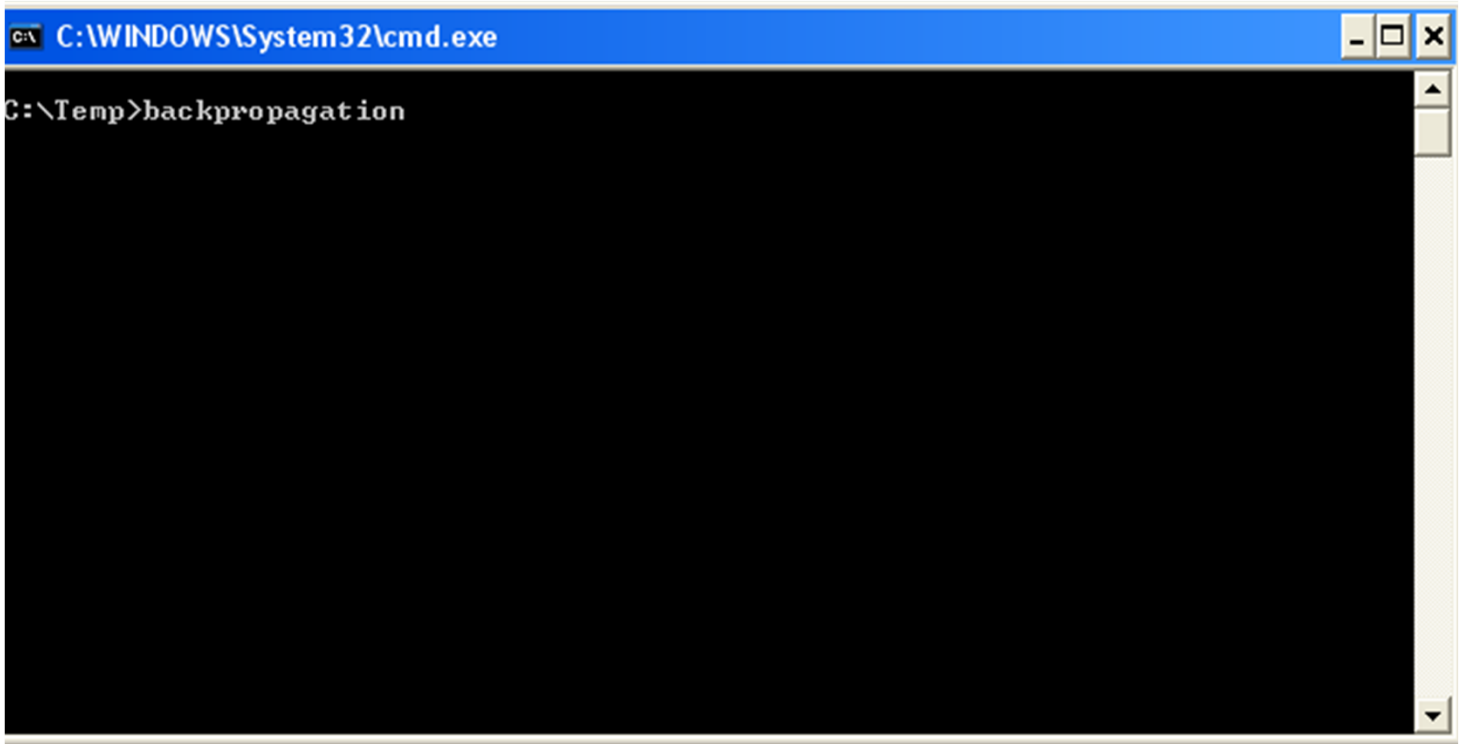
- The training data set contains the input and output values.
- The testing data set contains only the input values

Using “backprop.cpp”

- Please note the changes made to the output values for this particular example.
  - Class 1: output node 1 = 0; output node 2 = 0
  - Class 2: output node 1 = 0; output node 2 = 1
  - Class 3: output node 1 = 1; output node 2 = 1

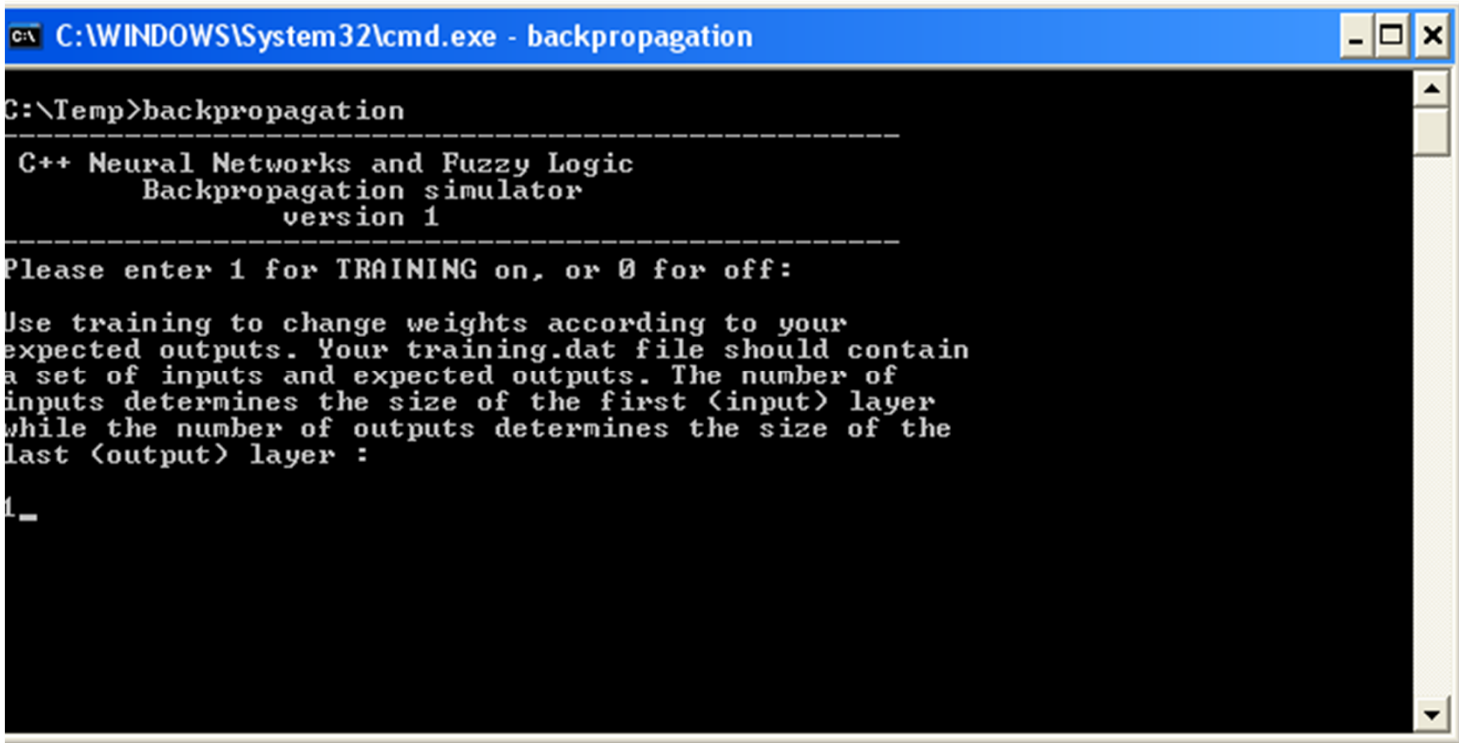
Using “backprop.cpp” (TRAINING)

- On the DOS prompt, type “backpropagation” to run the application



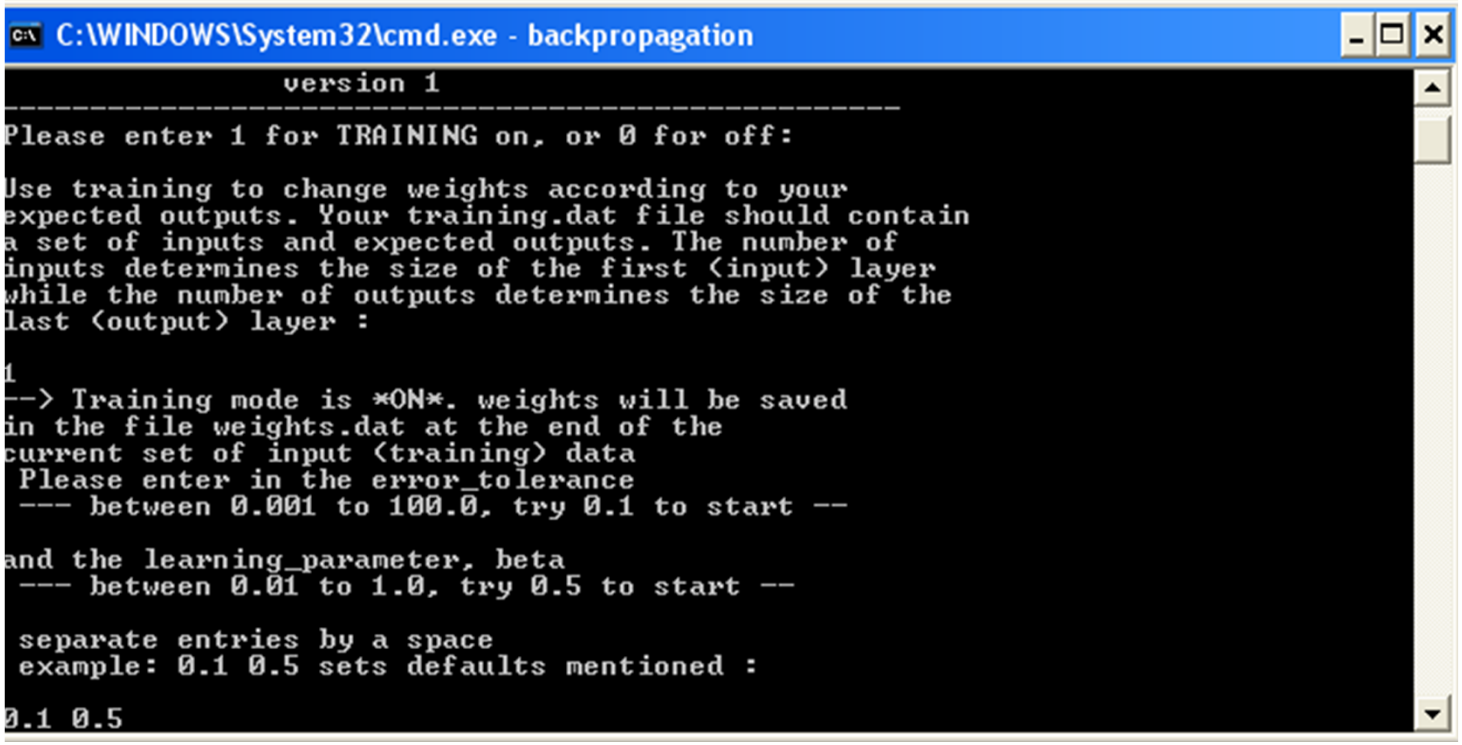
Using “backprop.cpp” (TRAINING)

- Enter “1” for TRAINING



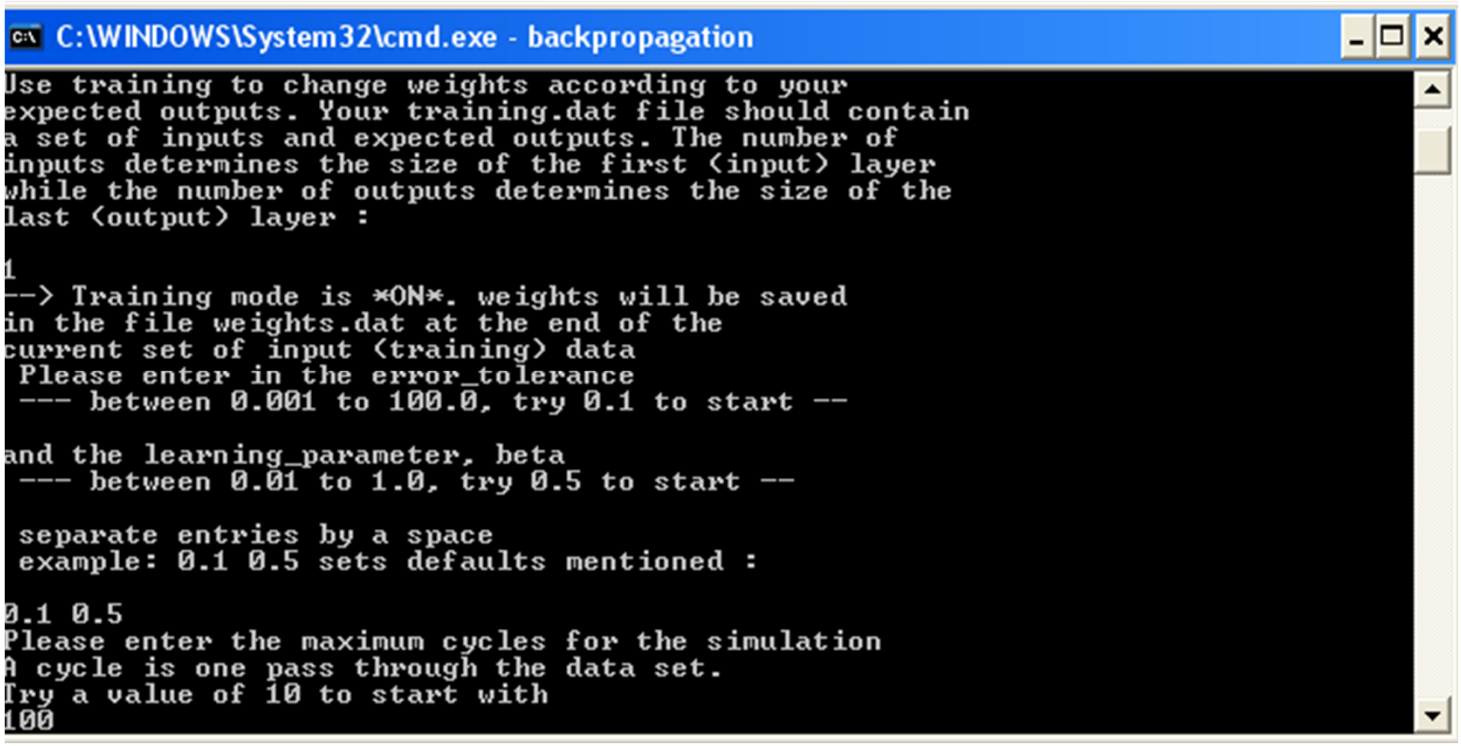
Using “backprop.cpp” (TRAINING)

- Enter the error tolerance and the learning rate



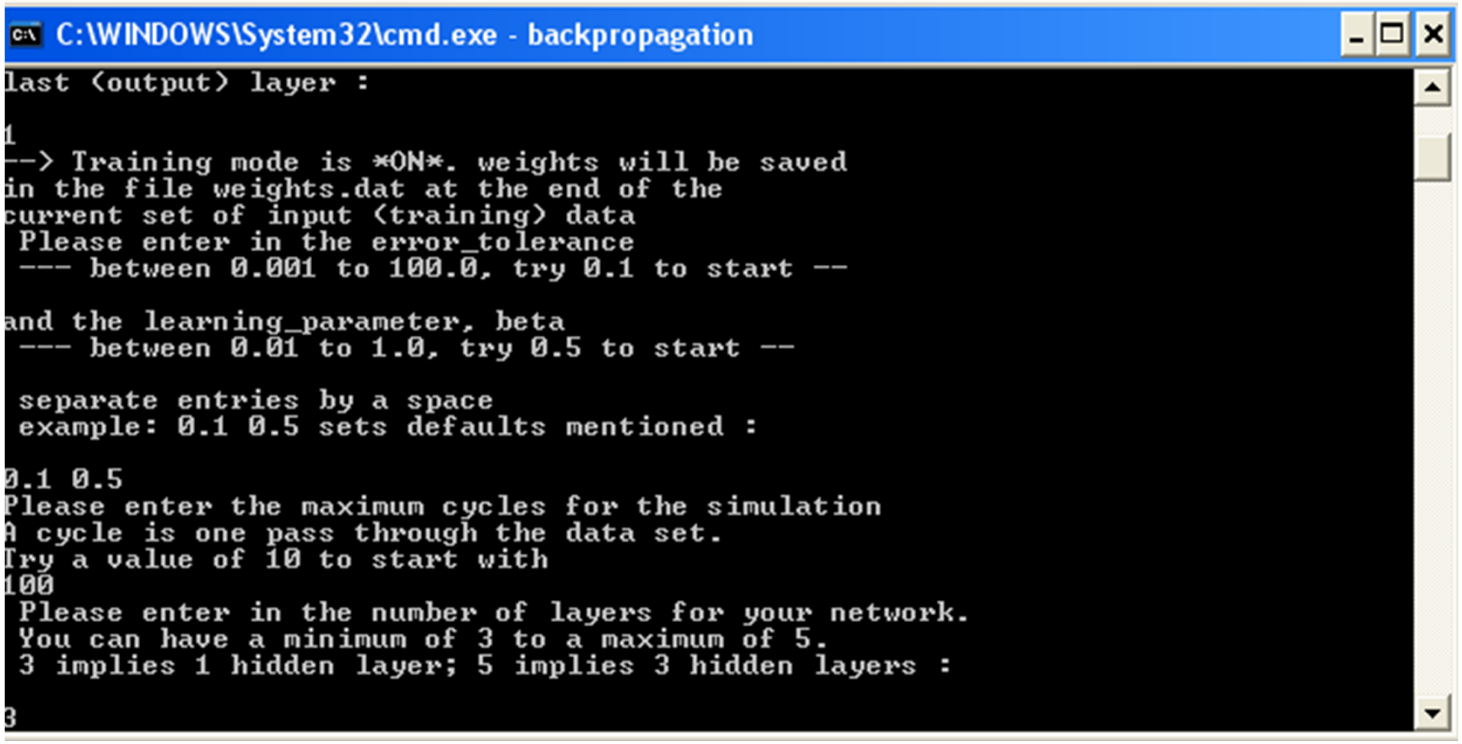
Using “backprop.cpp” (TRAINING)

- Enter the maximum number of cycles. One cycle is one pass through the data set.



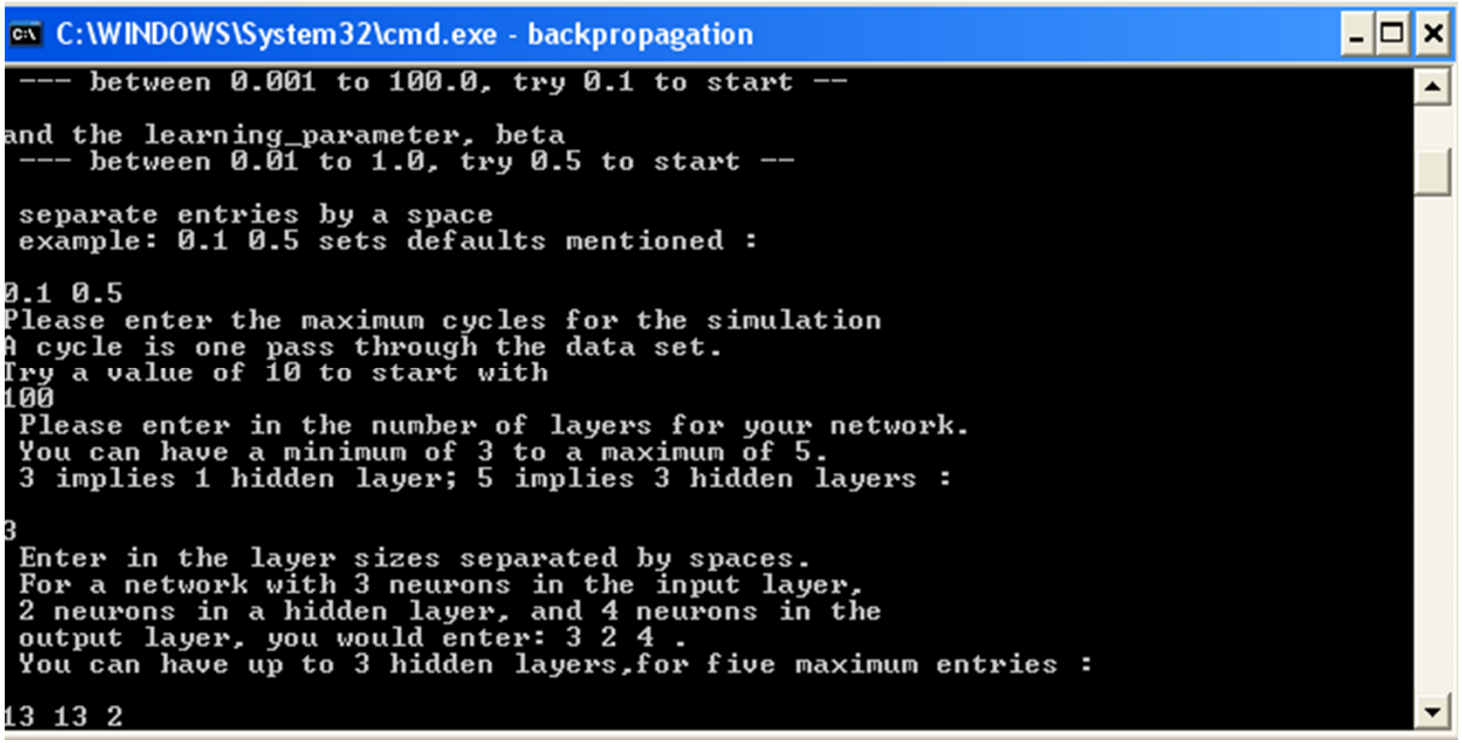
Using “backprop.cpp” (TRAINING)

- Enter the number of layers for your network.



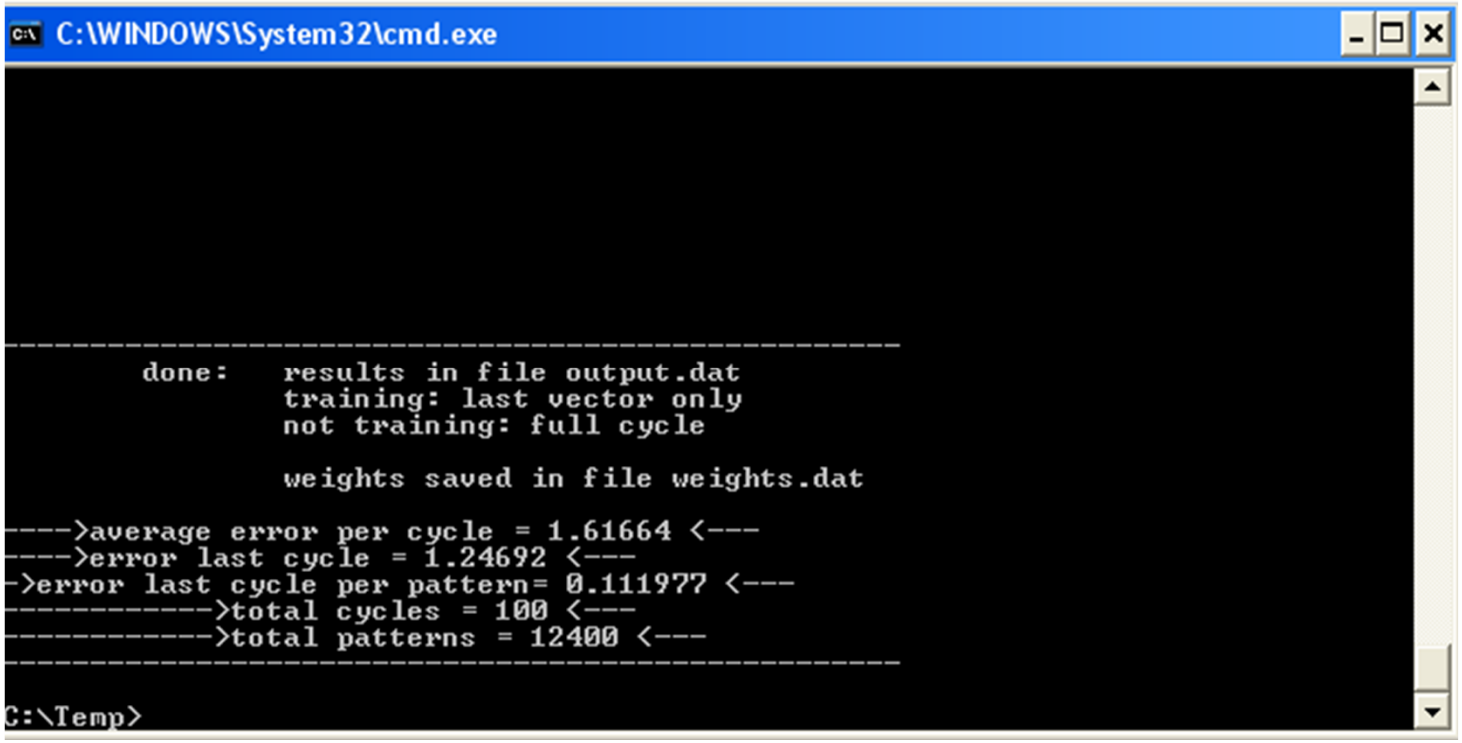
Using “backprop.cpp” (TRAINING)

- Enter the layer sizes separated by spaces.
- IMPORTANT: the number of input and output nodes must match the training data (on the “training.dat” file)



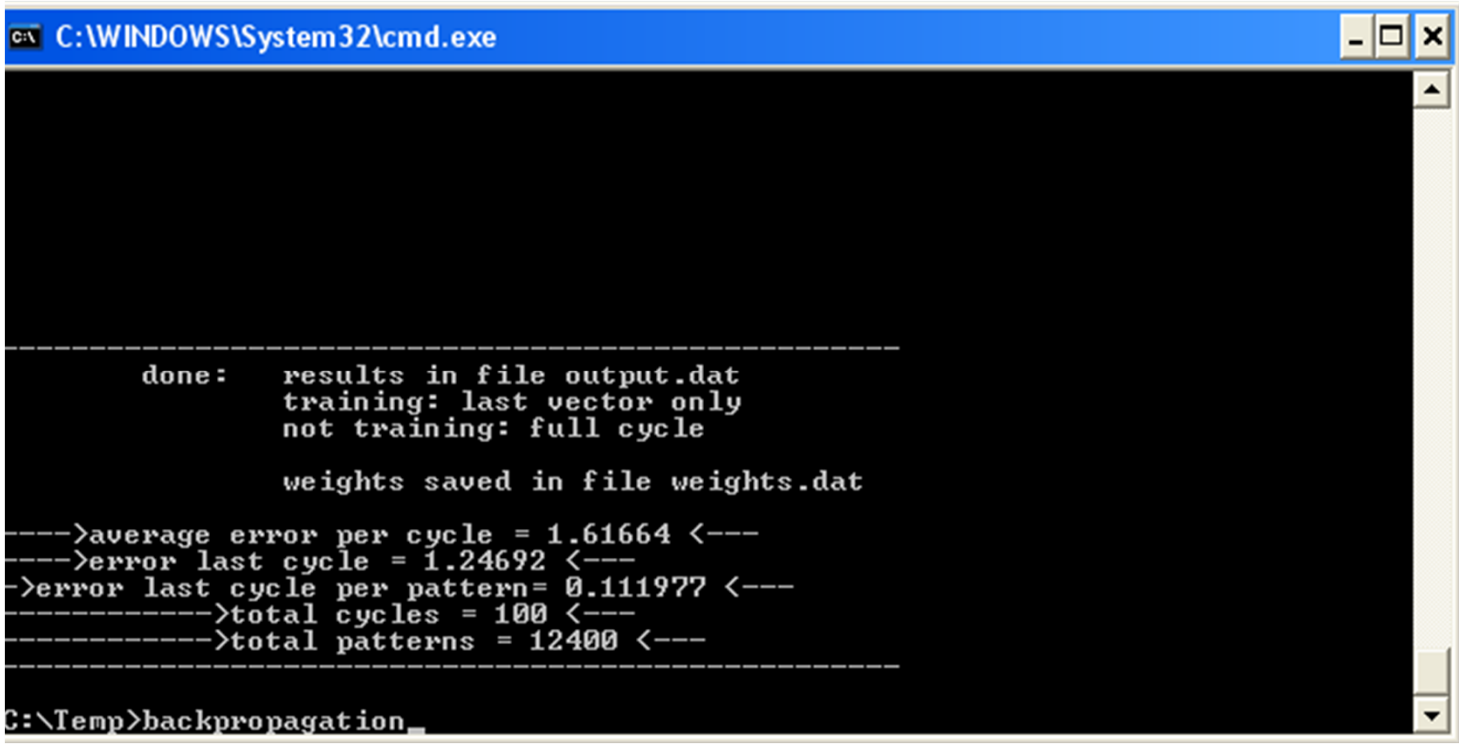
Using “backprop.cpp” (TRAINING)

- Run the neural network. Weights were saved on the “weights.dat” file. They will be used later during testing



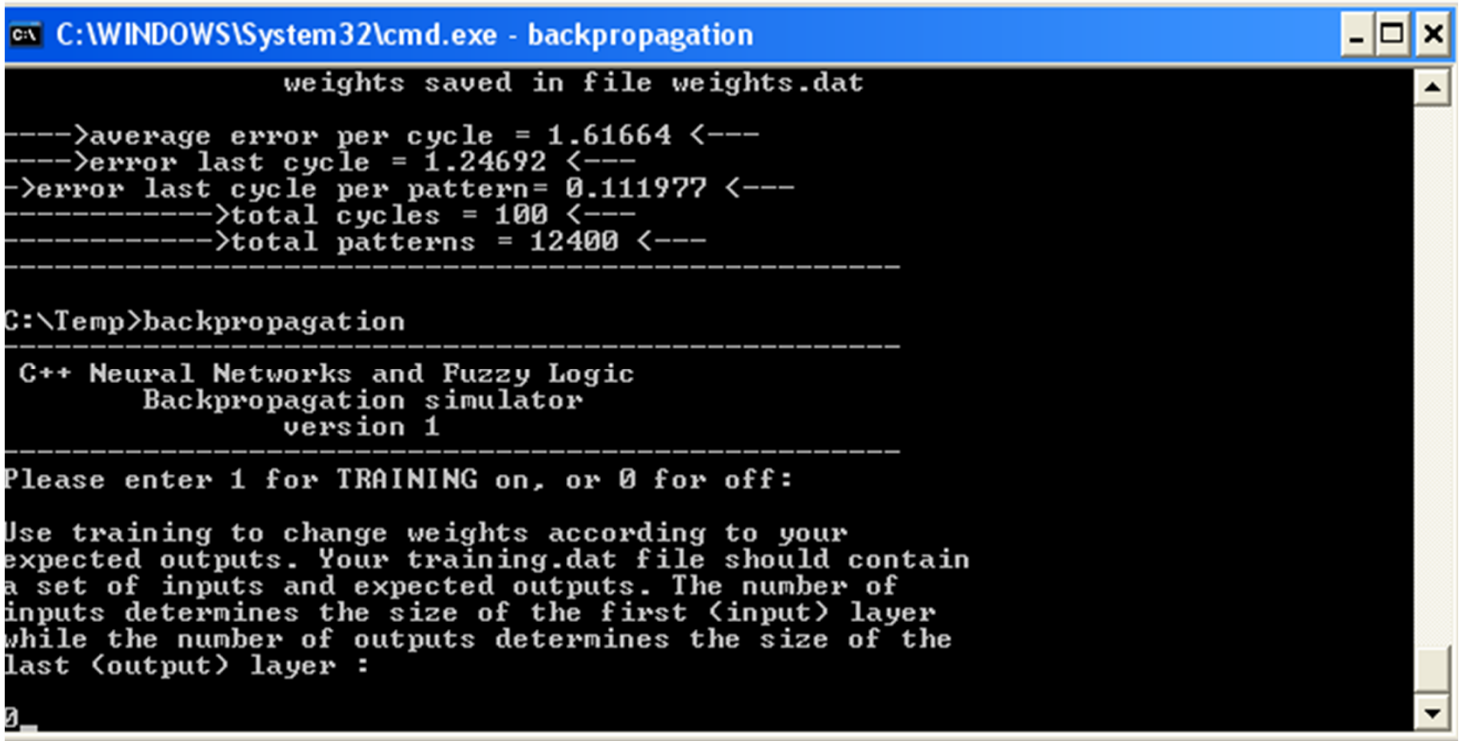
Using “backprop.cpp” (TESTING)

- On the DOS prompt, type “backpropagation” to run the application again



Using “backprop.cpp” (TESTING)

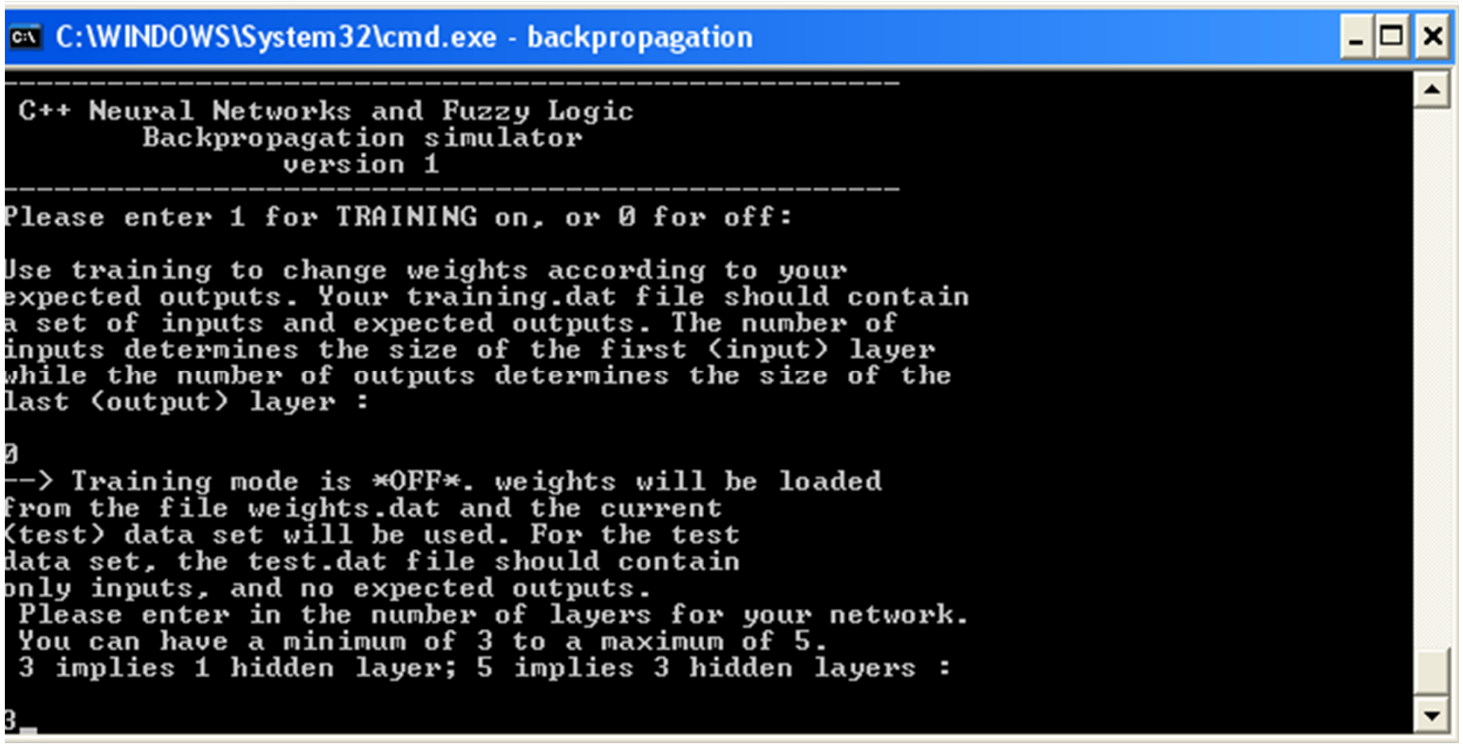
- Enter “0” for TESTING



Using “backprop.cpp” (TESTING)

- Enter the number of layers for your network.

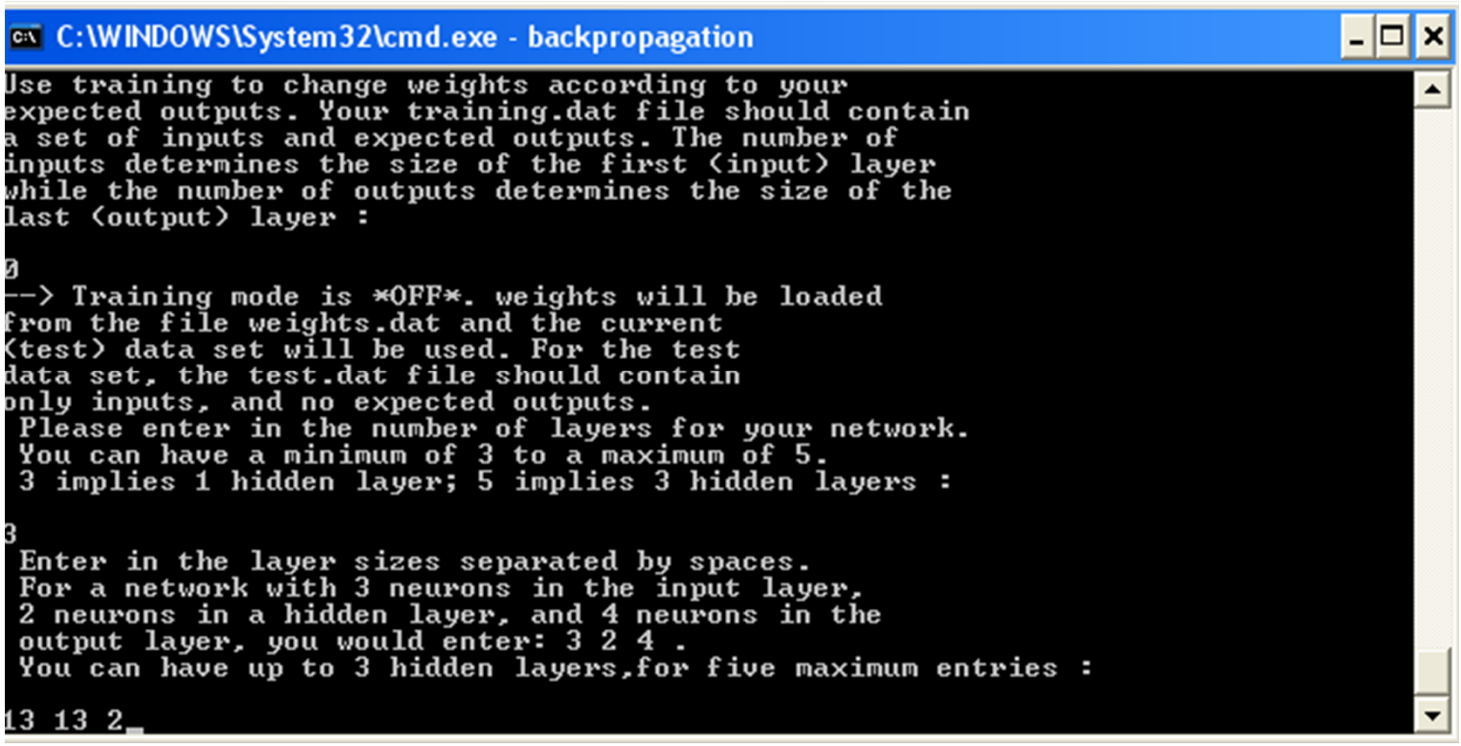
IMPORTANT: This must be the same used for training



Using “backprop.cpp” (TESTING)

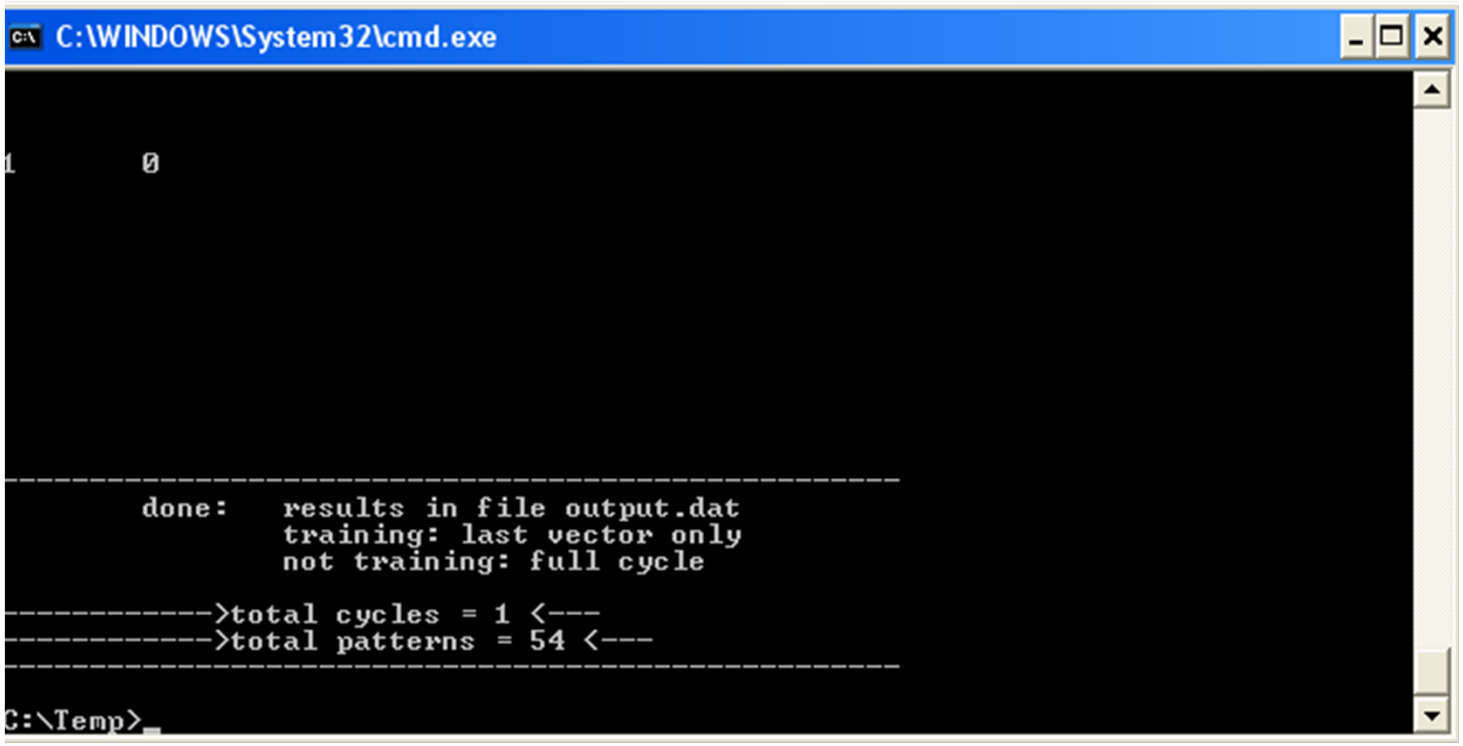
- Enter the layer sizes separated by spaces.

IMPORTANT: This must be the same used for training

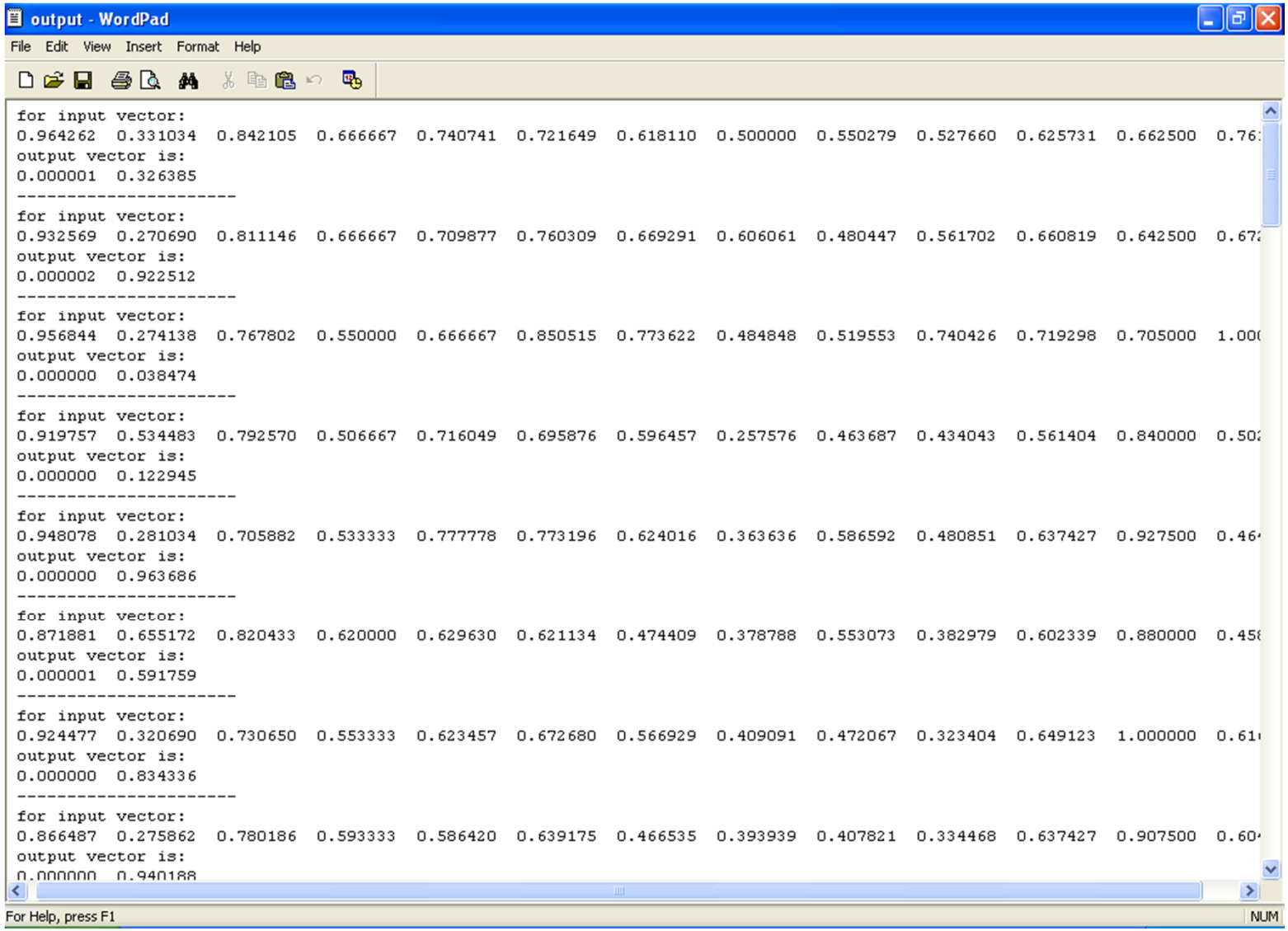


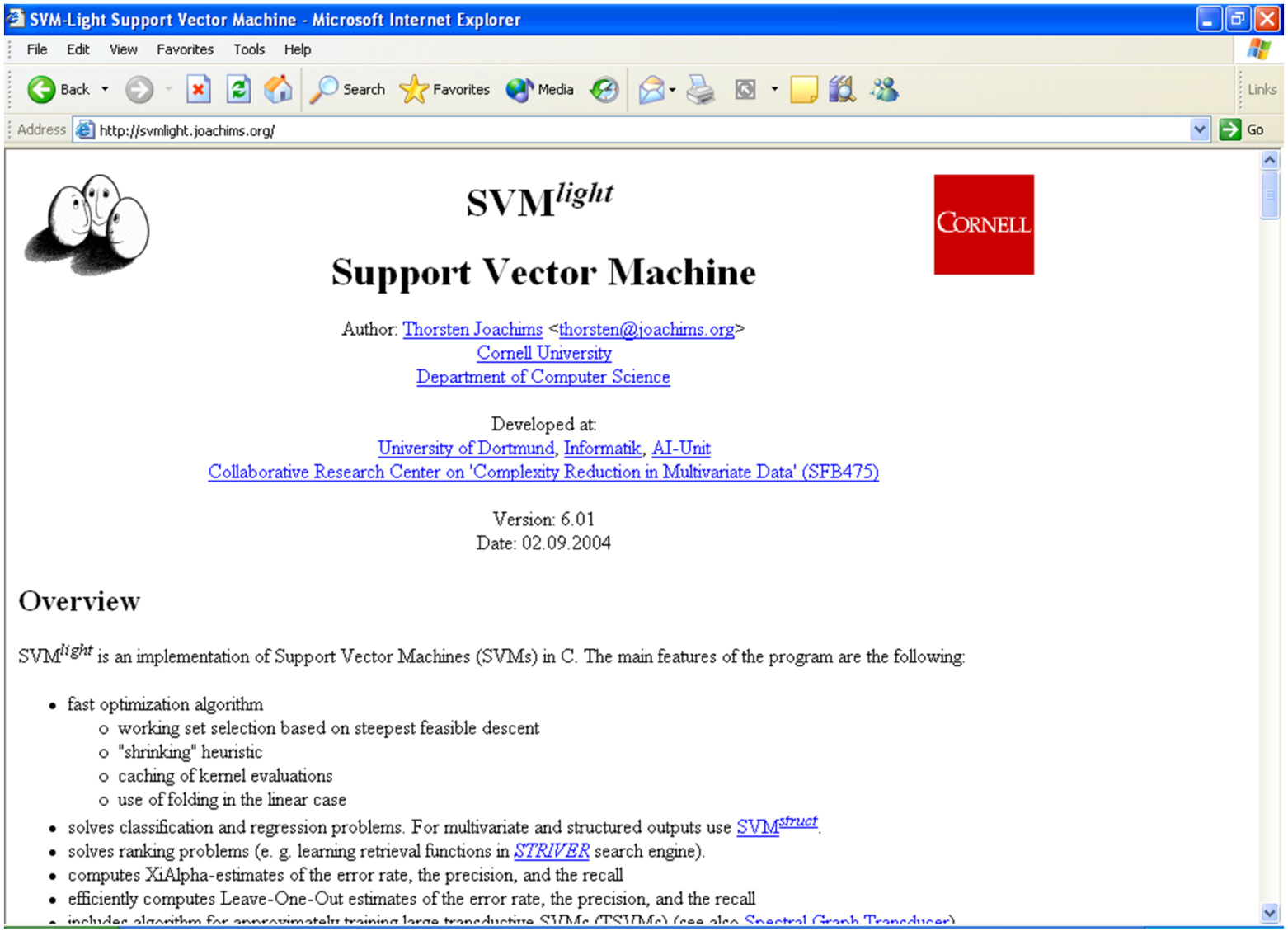
Using “backprop.cpp” (TESTING)

- Run the neural network. Output results from the testing data set were saved on the “output.dat” file.



Using “backprop.cpp” (OUTPUT)





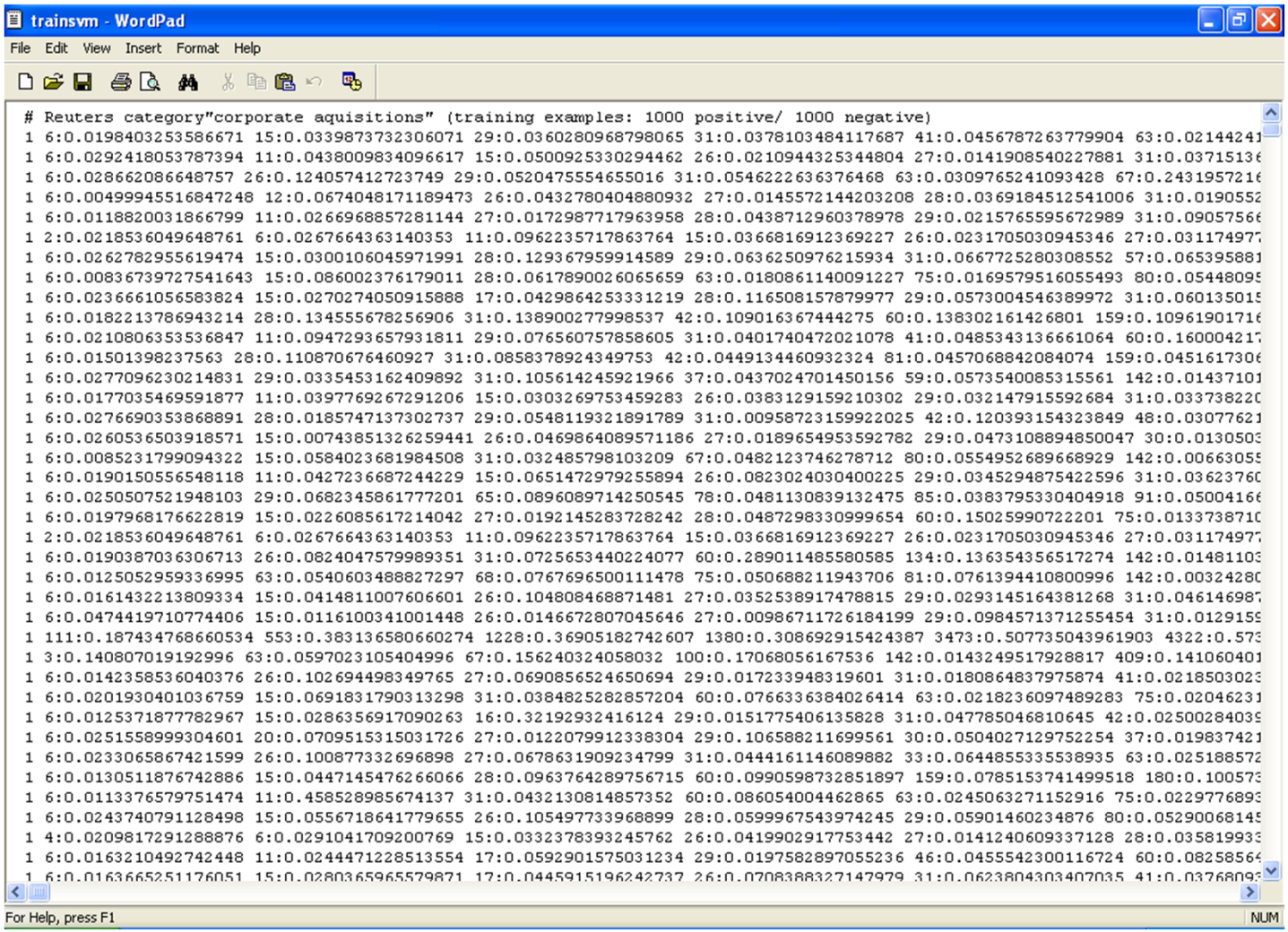
SVM light (http://svmlight.joachims.org/)

Using "SVM*light*"

- Download the executable file as well as the example training and testing data sets
  - Reference : T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

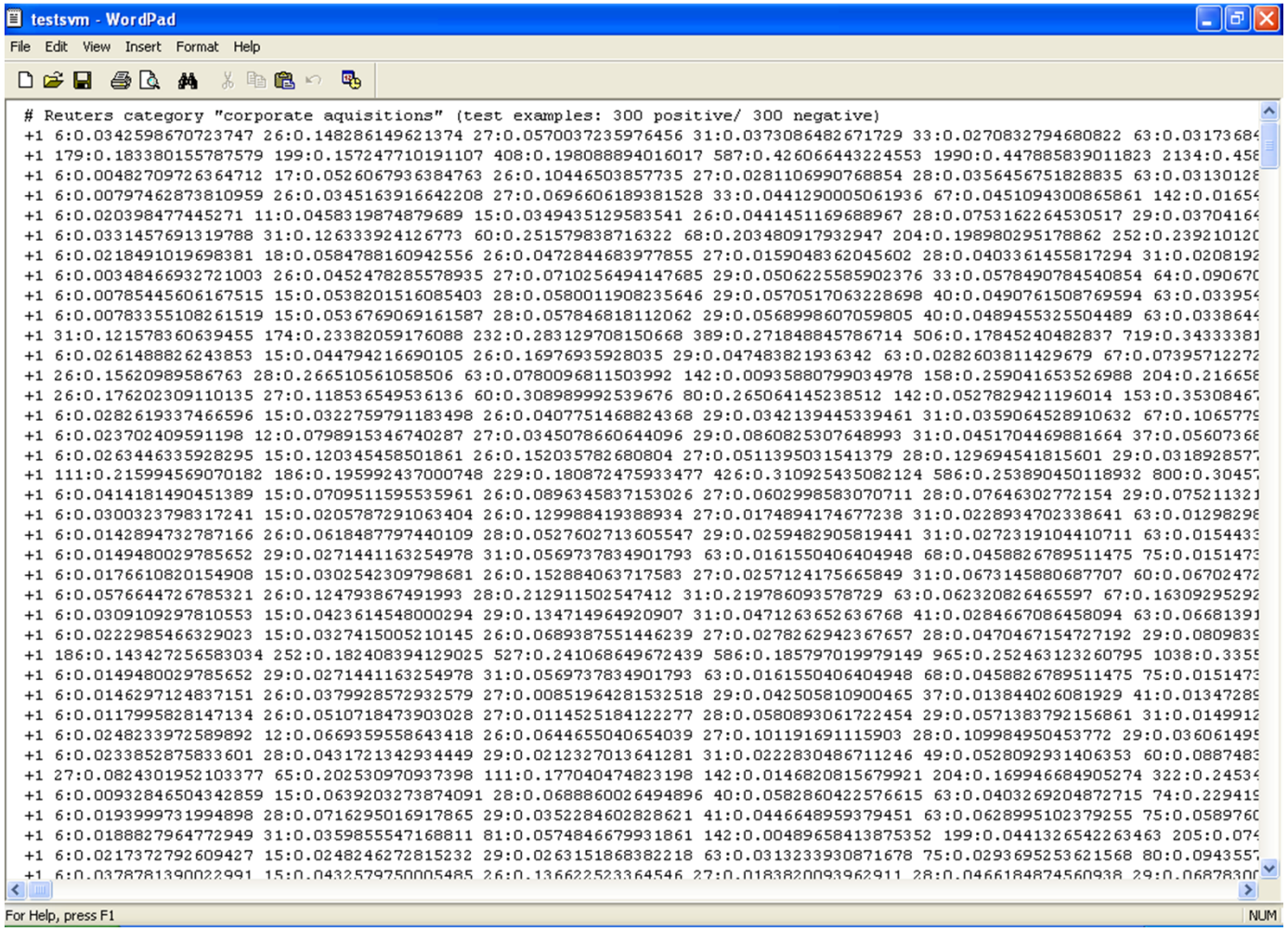
Using "SVM*light*"

- Training file ("trainsvm.dat"):



Using "SVM*light*"

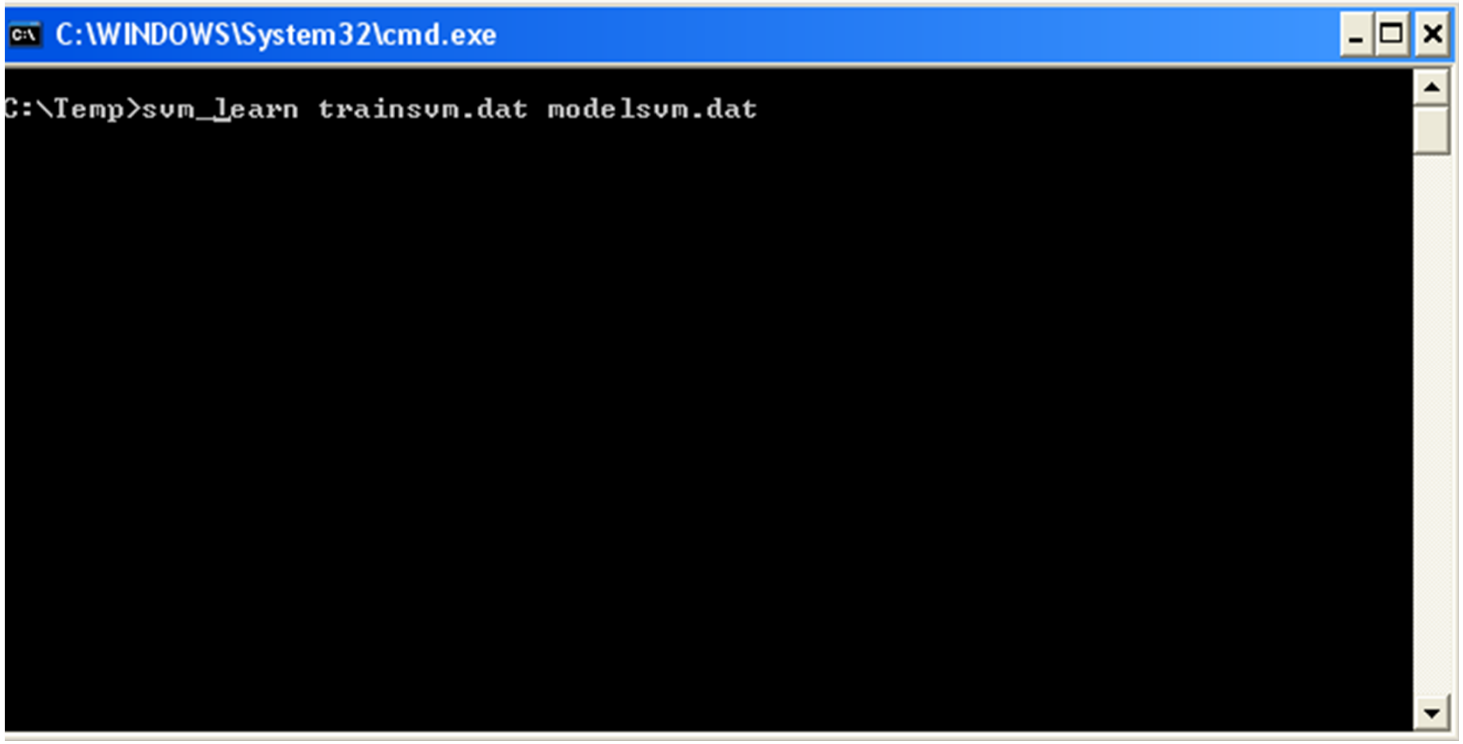
- Testing file ("testsvm.dat"):



Using "SVM*light*" (TRAINING)

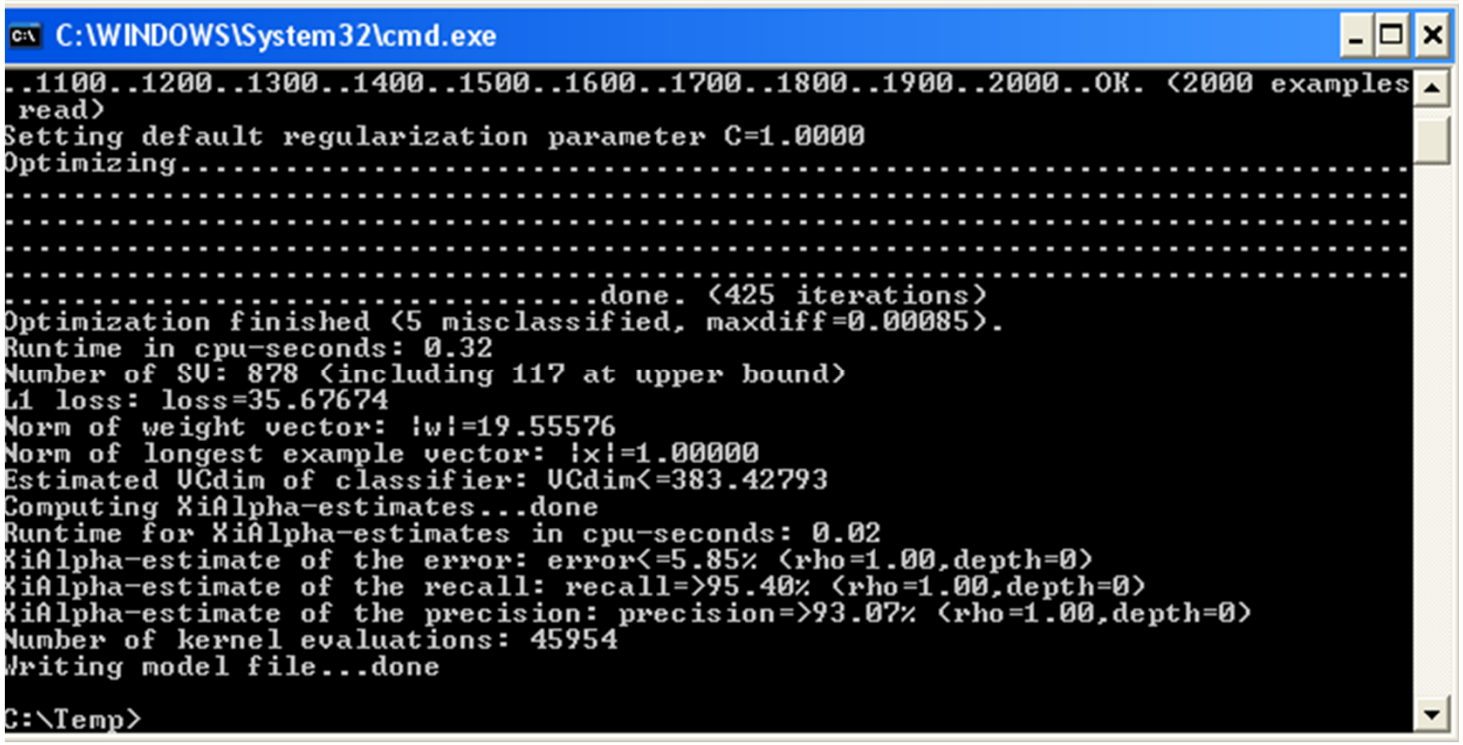
svm_learn is called with the following parameters:

svm_learn [options] trainingexamples_file model_file



Using "SVM*light*" (TRAINING)

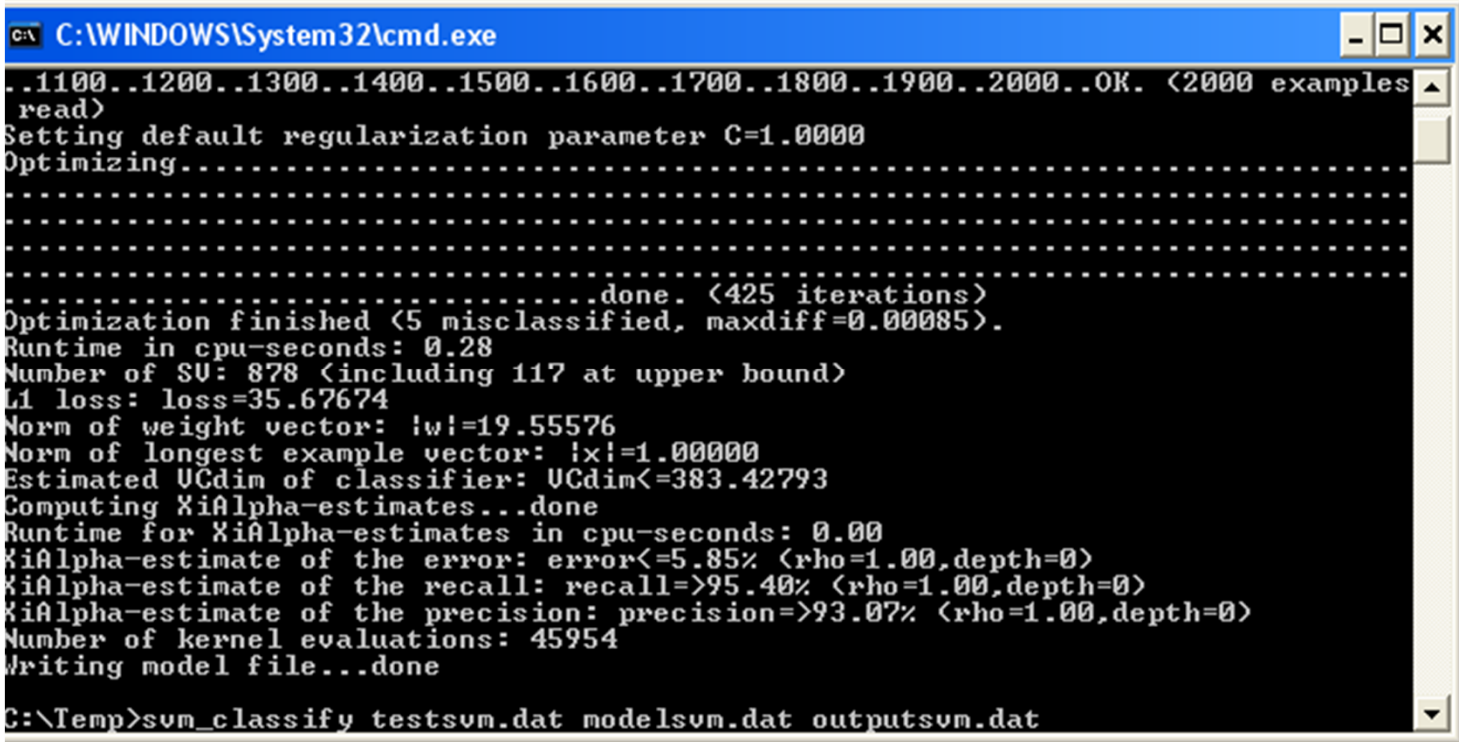
Run svm_learn. The model is saved at the "model_file". Later, this file will be used for testing



Using "SVM*light*" (TESTING)

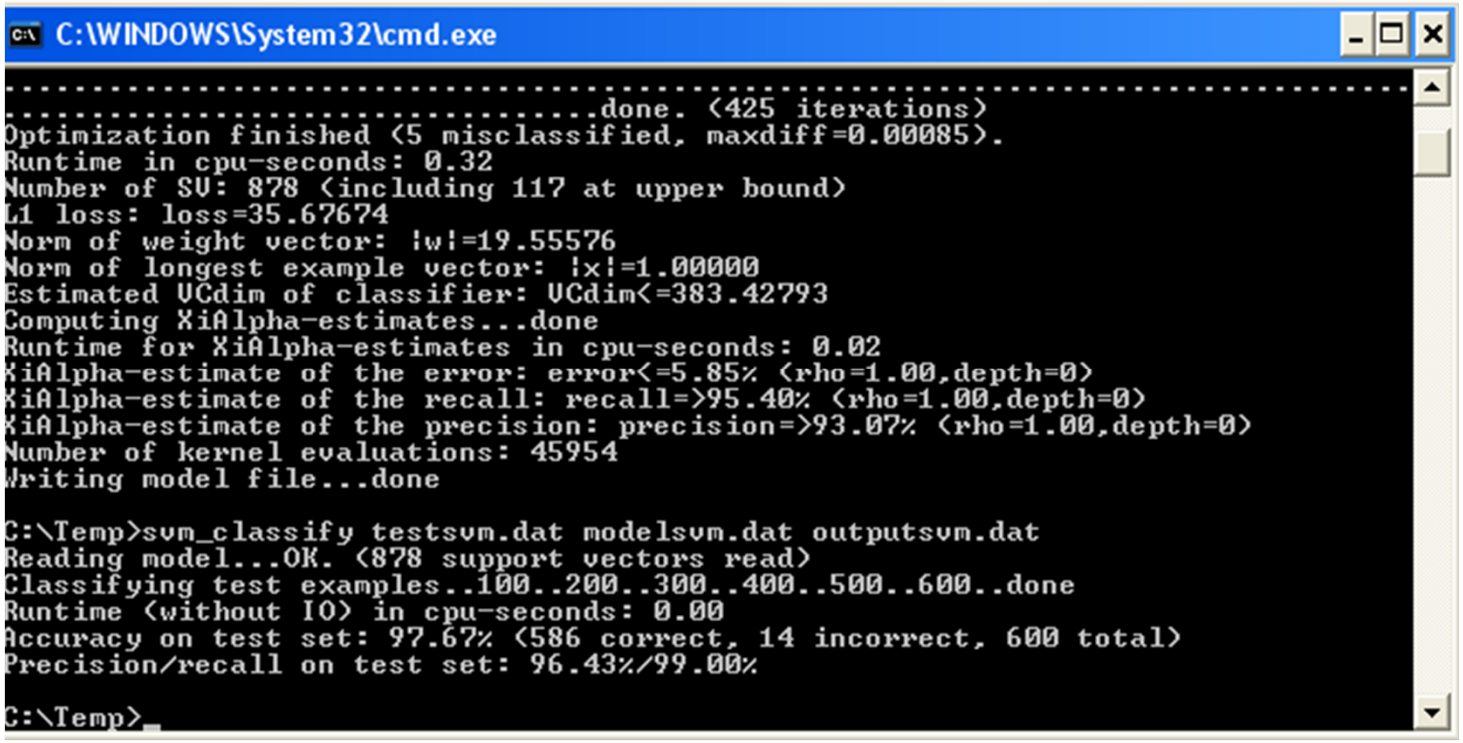
svm_classify is called with the following parameters:

svm_classify [options] testingexamples_file model_file output_file



Using "SVM*light*" (TESTING)

Run svm_classify. Results will be saved at the "output_file"



Using "SVM*light*" (RESULTS)

- Output file ("outputsvm.dat"):