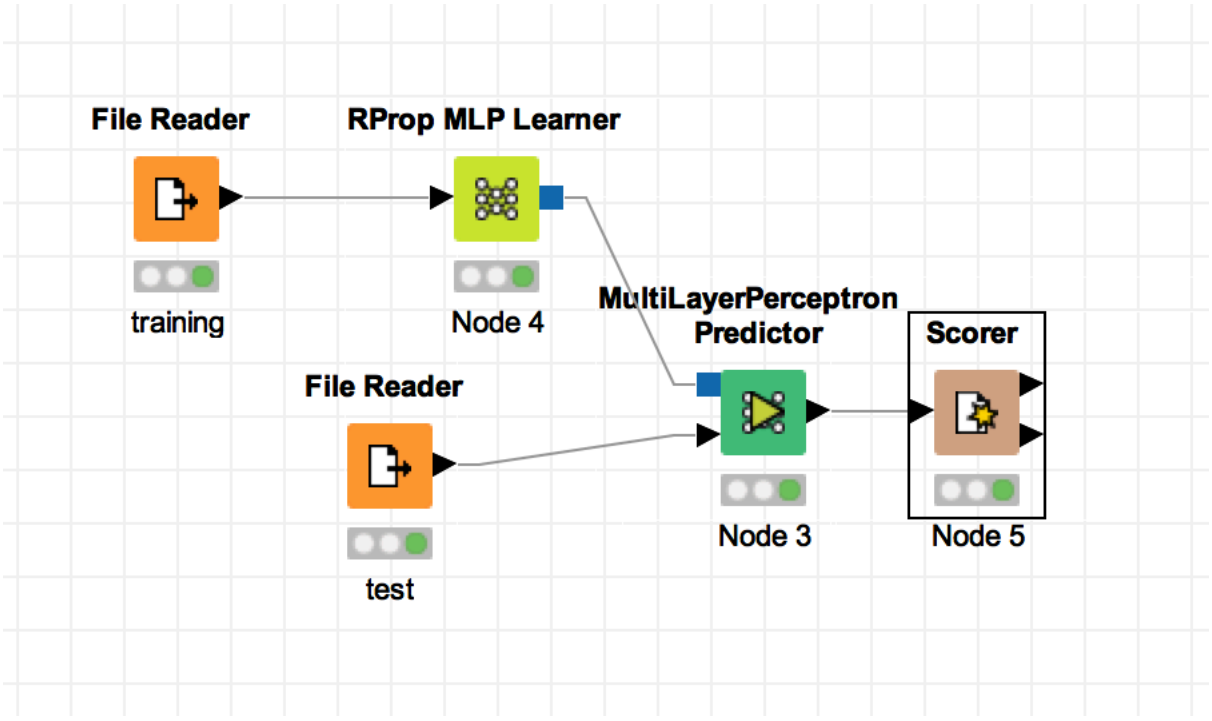
I used KNIME to implement a tool-based ANN:



Col4 \ Pre...	Iris-setosa	Iris-versic...	Iris-virginica
Iris-setosa	10	0	0
Iris-versic...	0	9	1
Iris-virginica	0	2	8

Correct classified: 27	Wrong classified: 3
Accuracy: 90 %	Error: 10 %
Cohen's kappa (κ) 0.85	

The problem with KNIME was that I couldn't change many parameters. This means that the comparison isn't completely fair, as I couldn't remove the one node in the hidden layer that the KNIME MLP automatically has.

Performance wise, KNIME correctly classified 27/30 -> accuracy of 90%, while my perceptron only came up to 63.666 % where it completely ignored "iris versicolor" class and instead seemed to either conclude that a datapoint was "iris setosa" or "iris virginica".
I included an example test run in the zip file, where my perceptron does slightly worse, but at least doesn't completely ignore any class.

Generally, both models have no trouble separating "iris setosa" from the others after very little training - as we know it is linearly separable this is no surprise.

I suspect that if KNIME had allowed me to create a model without any hidden layer nodes, it would have performed worse and closer to my own implementation.