

Problem Set 11: Artificial Neural Networks and Supervised Learning Computational Perception and Artificial Intelligence

Description:

The Artificial Neural Network (ANN) models the connections between perceptrons that are designed to mimic the interconnectedness of a nervous system. Supervised learning involves taking a large set of data organized into attributes and classes (Xi, Y) and then passing these data through the ANN to build a Hypothesis (predicting model).

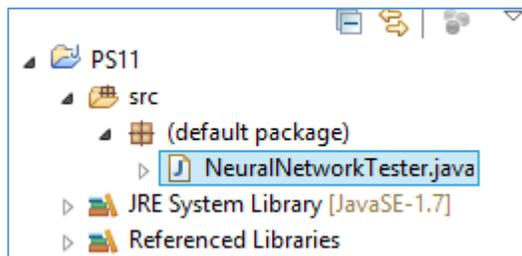
This exercise will guide you through the following:

- A. Downloading and Installing WEKA – [A Machine Learning tool from The University of Waikato](#)
- B. Creating a Java project in Eclipse to build and test a Neural Network.
- C. Placing the weka.jar archive on the build path of your Java ANN project
- D. Running an ANN on a sample set of classification data.

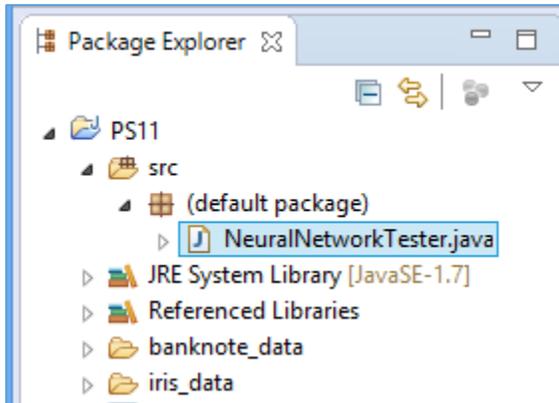
This is meant to be a ‘taste’ of Machine Learning and ANN. Machine Learning is a significant topic of study and can take many years of research and study to fully grasp the concepts. However, this will open your eyes and mind to the world of ‘big data’ and Artificial Intelligence.

Setup:

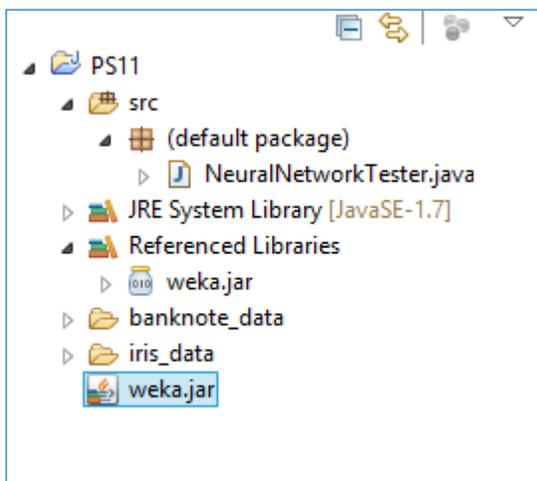
1. Download [ps11.zip](#) and extract into your Computational Perception folder
2. Run the weka-3-7-12-x64.exe installer program. Do not worry if Viewfinity blocks installation. (We can download this software another way . .)
3. Start Eclipse and set your workspace to the ps11 folder.
4. Start a new Java Project called PS11
5. Create a new class called “NeuralNetworkTester.java”



6. Copy the banknote_data folder from into your PS11 Java Project
7. Copy the iris_data folder into your PS11 Java Project. (Both of these where located in downloads)



8. Copy the 'weka.jar' file into your PS11 project. Right click on the weka.jar and select 'Build Path – Add to build path'. Your project should have the structure below.

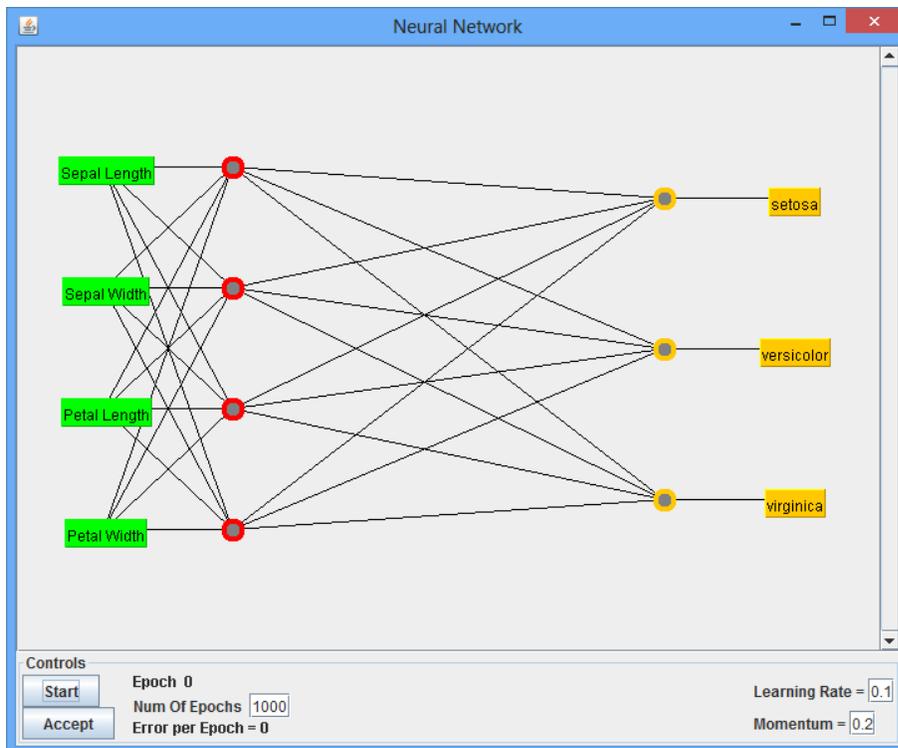


Requirements:

1. Implement an Artificial Neural Network in the NeuralNetworkTester.java class. We will cover this in class. However, if you missed this you may click below and copy the code the class.

[NeuralNetworkTester.java](#)

2. Run the code. When prompted with the GUI of the Neural Network, click 'Start' to begin training the network.



3. When the training is finished (It will count to 1000). Click Accept. The resulting output in the terminal should look like:

```

<terminated> NeuralNetworkTester [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Nov 5, 2015, 1
Testing Neural Network
---Registering Weka Editors---
Trying to add database driver (JDBC): jdbc.idbDriver - Error, not in CLASSPATH?
0.013333333333333334

Correctly Classified Instances      148          98.6667 %
Incorrectly Classified Instances    2            1.3333 %
Kappa statistic                    0.98
Mean absolute error                0.0262
Root mean squared error            0.0919
Relative absolute error             5.8889 %
Root relative squared error        19.5053 %
Total Number of Instances          150

50 0 0
0 49 1
0 1 49

```

4. In the 'bank_data' folder, there is another data set 'banknote_training.csv'. Run the testNeuralNetwork function using 'banknote_training.csv'. You will need to open the banknote_training file to see what the index of the Classification data is. Make note of the results of the Neural Network.

5. Now that we have practiced with some simple data sets, find your own data sets to test. You will need to make sure your data sets has the following characteristics: (Excel is very useful for formatting .csv files)

- a. It should be in .csv format
- b. It should have attributes and one class (labeled as top of columns)
- c. The classification should be in text format
- d. The data should be sorted by classification

	A	B	C	D	E
1	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
2	5	3.4	1.6	0.4	setosa
3	5.4	3.9	1.7	0.4	setosa
4	5	3	1.6	0.2	setosa
5	4.9	3.1	1.5	0.1	setosa
6	4.8	3.1	1.6	0.2	setosa
7	4.8	3	1.4	0.1	setosa
8	5.2	4.1	1.5	0.1	setosa
9	5.5	3.5	1.3	0.2	setosa
10	4.9	3	1.4	0.2	setosa
11	5	3.5	1.3	0.3	setosa
12	4.6	3.2	1.4	0.2	setosa
13	4.3	3	1.1	0.1	setosa

Here are some sites to find data:

<http://archive.ics.uci.edu/ml/> (UC Irvine Machine Learning Repository)

<https://www.kaggle.com/> (Contest page for Big Data – This would make a great final project!)

6. Download, format, create folders, and run the Neural Network Learner on two data sets you find. Write a summary of the results of your tests. Include the output from the Java console and any observations you make (How accurate was the Learner? How long did the Learner take to analyze the data?).

Save the summary as “PS11 Analysis.docx” and email to Mr. Michaud.