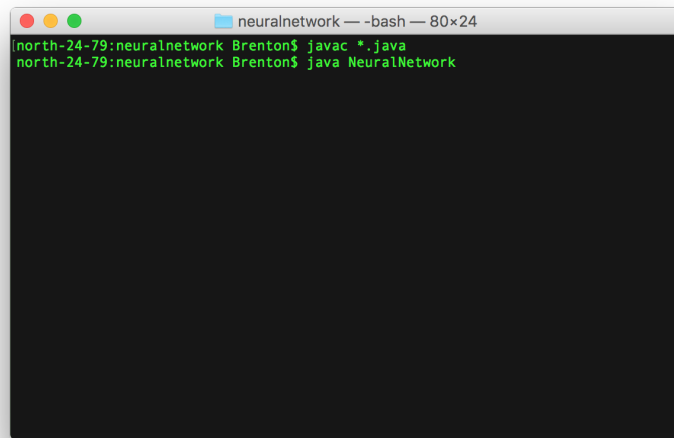


How to use the Application

Compiling and Launching the Application

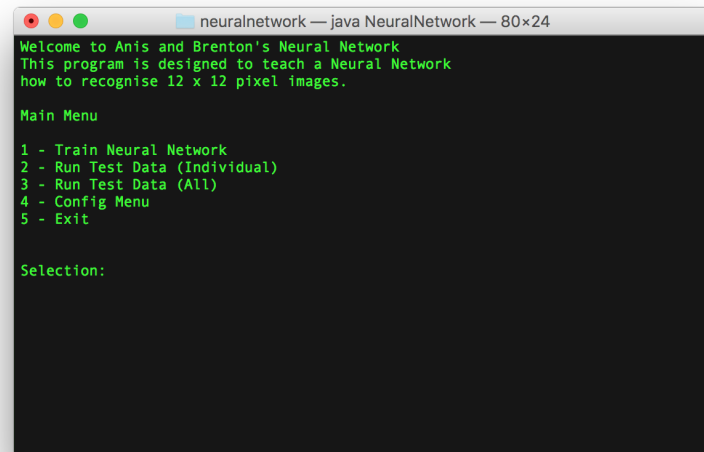
1. Unzip the project.
2. Open Terminal or command prompt.
3. Change the directory to the folder you unzipped.
4. Run the following command: `javac *.java`
5. On successful compile, run the following command: `java NeuralNetwork`

A screenshot of a terminal window titled "neuralnetwork — -bash — 80x24". The terminal shows two commands entered: `javac *.java` and `java NeuralNetwork`. The prompt for each command is `north-24-79:neuralnetwork Brenton$`.

```
north-24-79:neuralnetwork Brenton$ javac *.java
north-24-79:neuralnetwork Brenton$ java NeuralNetwork
```

Main Menu

This is the main menu of the application. From here you are able to perform all the main tasks of the application through input into the terminal. Selections are made by entering a number for the appropriate function you wish to perform and pressing enter.

A screenshot of a terminal window titled "neuralnetwork — java NeuralNetwork — 80x24". The text in the terminal is as follows:

```
Welcome to Anis and Brenton's Neural Network
This program is designed to teach a Neural Network
how to recognise 12 x 12 pixel images.

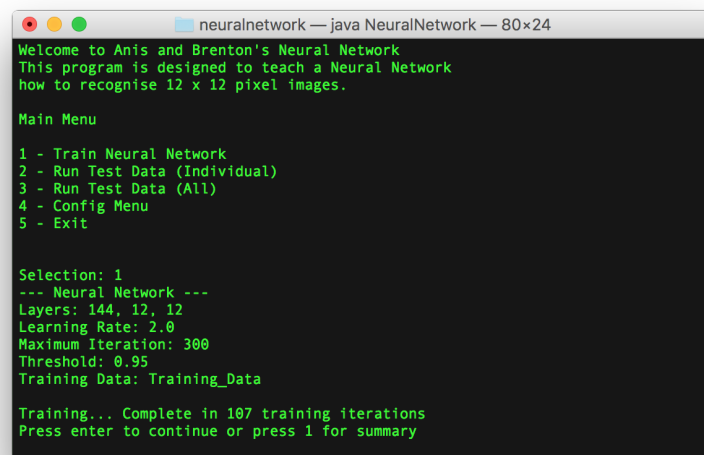
Main Menu

1 - Train Neural Network
2 - Run Test Data (Individual)
3 - Run Test Data (All)
4 - Config Menu
5 - Exit

Selection:
```

Training the Network

In order to use the network, it must first be trained. PLEASE NOTE: An error message will appear if an attempt is made to use the neural network before it is trained.

A screenshot of a terminal window titled "neuralnetwork — java NeuralNetwork — 80x24". The text in the terminal is as follows:

```
Welcome to Anis and Brenton's Neural Network
This program is designed to teach a Neural Network
how to recognise 12 x 12 pixel images.

Main Menu

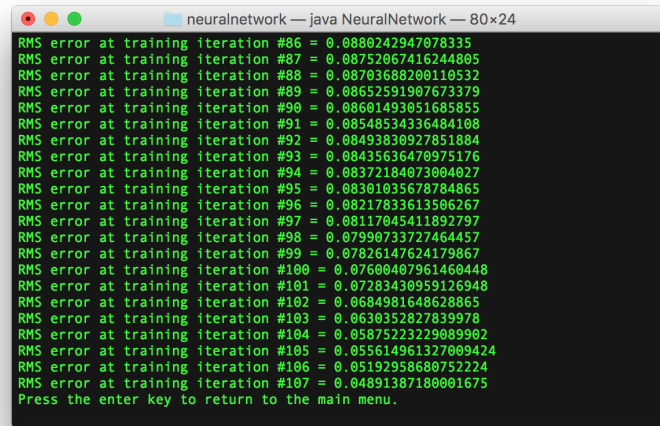
1 - Train Neural Network
2 - Run Test Data (Individual)
3 - Run Test Data (All)
4 - Config Menu
5 - Exit

Selection: 1
--- Neural Network ---
Layers: 144, 12, 12
Learning Rate: 2.0
Maximum Iteration: 300
Threshold: 0.95
Training Data: Training_Data

Training... Complete in 107 training iterations
Press enter to continue or press 1 for summary
```

Once the network is trained, a message appears confirming that the network is trained. The application then asks the user if they wish to print a summary of the training iterations. This

shows the rms error after each training iteration. The user can either press enter to continue or press 1 and then enter to print the summary.



```
neuralnetwork — java NeuralNetwork — 80x24
RMS error at training iteration #86 = 0.0880242947078335
RMS error at training iteration #87 = 0.08752067416244805
RMS error at training iteration #88 = 0.08703688200110532
RMS error at training iteration #89 = 0.08652591907673379
RMS error at training iteration #90 = 0.08601493051685855
RMS error at training iteration #91 = 0.08548534336484108
RMS error at training iteration #92 = 0.08493830927851884
RMS error at training iteration #93 = 0.08435636470975176
RMS error at training iteration #94 = 0.08372184073004027
RMS error at training iteration #95 = 0.08301035678784865
RMS error at training iteration #96 = 0.08217833613506267
RMS error at training iteration #97 = 0.08117045411892797
RMS error at training iteration #98 = 0.07990733727464457
RMS error at training iteration #99 = 0.07826147624179867
RMS error at training iteration #100 = 0.07600407961460448
RMS error at training iteration #101 = 0.07283430959126948
RMS error at training iteration #102 = 0.0684901648628865
RMS error at training iteration #103 = 0.0630352027839978
RMS error at training iteration #104 = 0.05875223229089902
RMS error at training iteration #105 = 0.055614961327009424
RMS error at training iteration #106 = 0.05192958680752224
RMS error at training iteration #107 = 0.04891387180001675
Press the enter key to return to the main menu.
```

When the summary is printed to the screen, the application also creates a CSV file containing all the RMS Errors for that training set. This is saved to the location where the application is being executed from.

Testing the Network

To test the neural network, the user has the choice to either run an individual piece of test data or run all the test data that is currently loaded into the system. The location of test data files or training data can be changed through the config menu. (See Below: Set Training Data and Set Test Data)

Run Individual Files

Selecting this option will allow the user to then chose any one of the test data files the system found. The user simply selects which file to run by entering its corresponding number and pressing the enter key.

```
neuralnetwork — java NeuralNetwork — 80x24

2 - Clock.noise.5.txt
3 - Cross.noise.5.txt
4 - Cross.noise10.txt
5 - Exclamation.noise.10.txt
6 - Exclamation.noise.5.txt
7 - Face.noise.10.txt
8 - Face.noise.5.txt
9 - Giveaway.noise.10.txt
10 - Giveaway.noise.5.txt
11 - House.noise.10.txt
12 - House.noise.5.txt
13 - Info.noise.10.txt
14 - Info.noise.5.txt
15 - Smile.noise.10.txt
16 - Smile.noise.5.txt
17 - Stand.noise.10.txt
18 - Stand.noise.5.txt
19 - Stop.noise.10.txt
20 - Stop.noise.5.txt
21 - Tick.noise.10.txt
22 - Tick.noise.5.txt
23 - Walk.noise.10.txt
24 - Walk.noise.5.txt
Selection:
```

```
neuralnetwork — java NeuralNetwork — 80×24
```

```
21 - Tick.noise.10.txt  
22 - Tick.noise.5.txt  
23 - Walk.noise.10.txt  
24 - Walk.noise.5.txt  
Selection: 1
```

```
    ****  
   *  *  *  *  
  *      *  *  
 *        *  *  
*          *  *  
 *         *  *  
  *       *  *  
   *****  
  *      *  *  
 *        *  *  
*          *  *  
 *         *  *  
  *       *  *  
   *****  
  *      *  *  
 *        *  *  
*          *  *  
 *         *  *  
  *       *  *  
   *****  
  *      *  *  
 *        *  *  
*          *  *
```

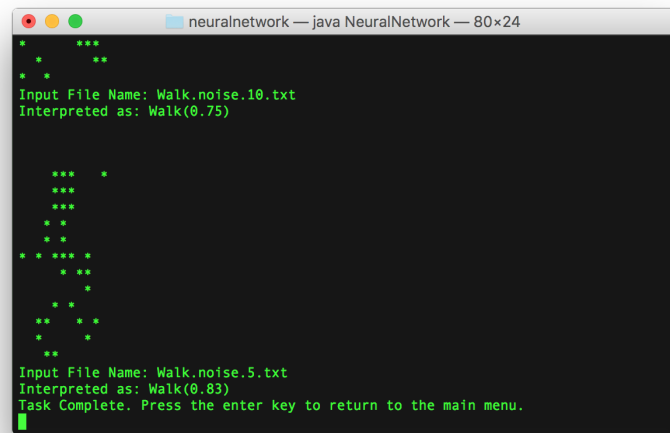
```
Input File Name: Clock.noise.10.txt  
Interpreted as: Clock(0.87)  
Task Complete. Press the enter key to return to the main menu.
```

As you can see above, it displays back to the user the name of the input file and what the neural network interpreted the image as. If any other images in the neural network have an output

above 0.5, they will then be listed in descending order.

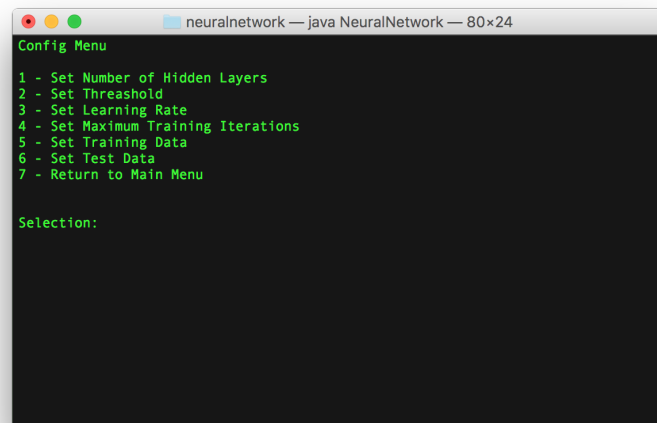
Run All Files

This option will run all the files currently loaded into the program against the neural network.



Config Menu

The config menu allows the user to change various settings within the application.



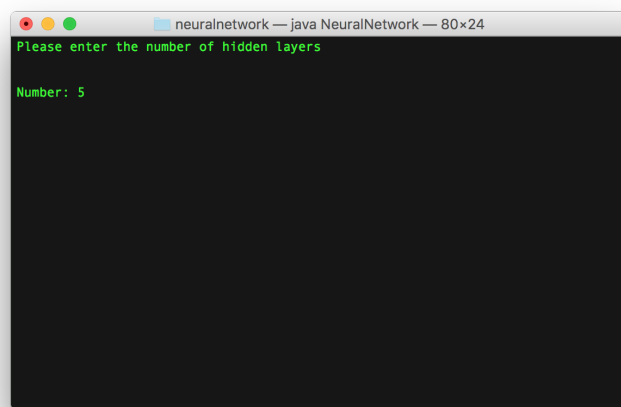
The user has the ability to change:

- The number of hidden layers including the number of neurons per layer.

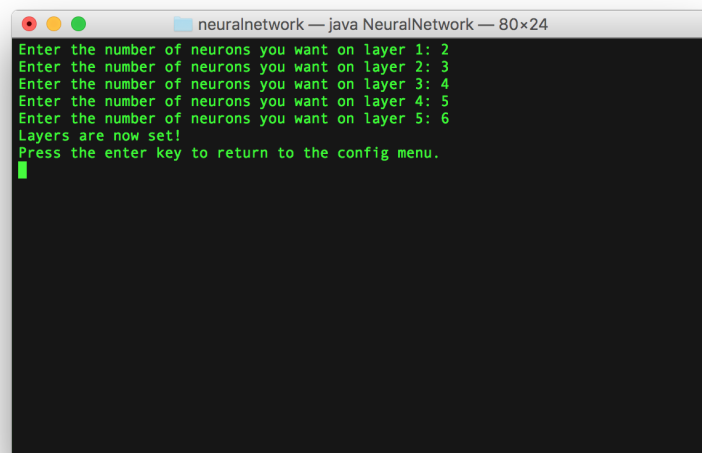
- The ability to set the threshold (so the neural network is accurate as the user wants it to be).
- The ability to set the learning rate of the network.
- Set the maximum number of training iterations the computer can take to train the network.
- Set the location of the training data
- Set the location of the testing data

Set Number of Hidden Layers

The user is prompted for the number of hidden layers.

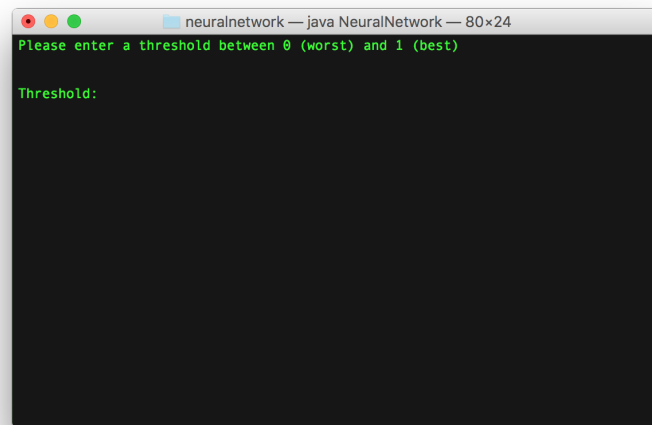


For each corresponding layer entered, the user will be prompted for the number of neurons they want on corresponding layer.

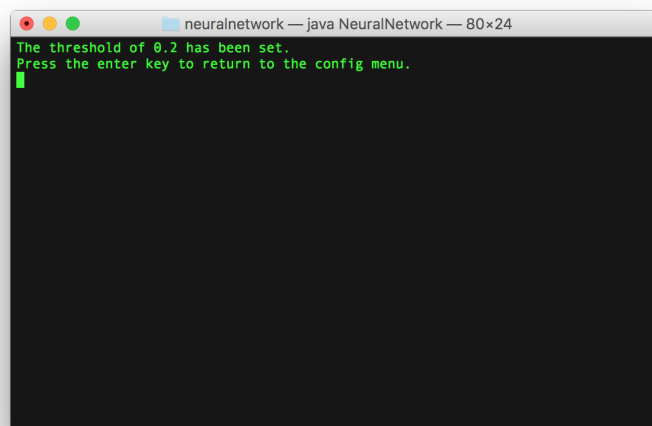


Set Threshold

The user is prompted to enter a threshold between 0 and 1. This threshold is the accuracy to which the neural network should be trained to.

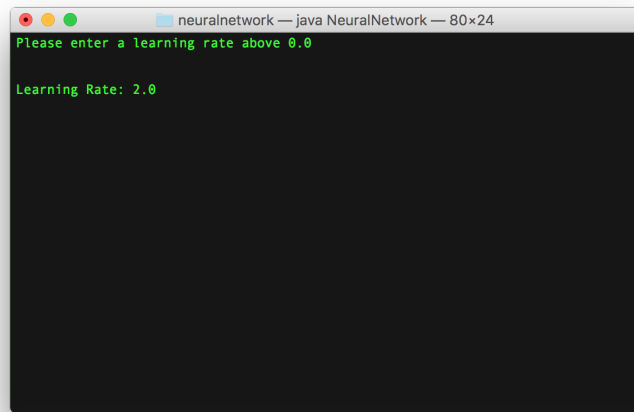


A confirmation box appears to confirm users changes to the threshold.

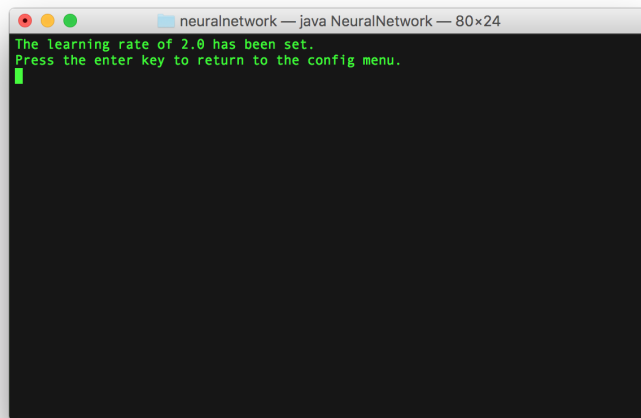


Set Learning Rate

The user is prompted to enter a learning rate. This number must be above 0.0.

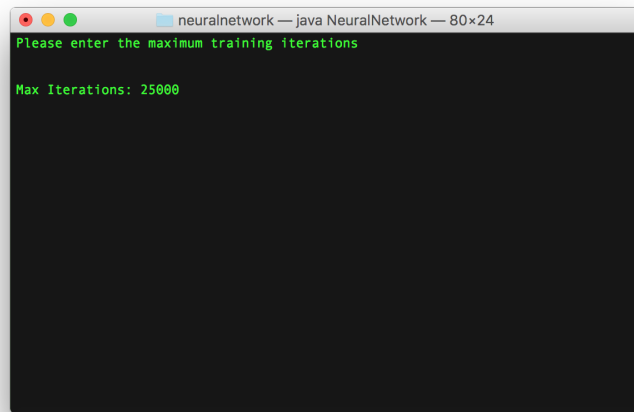


A confirmation box appears to confirm the users change to the learning rate.

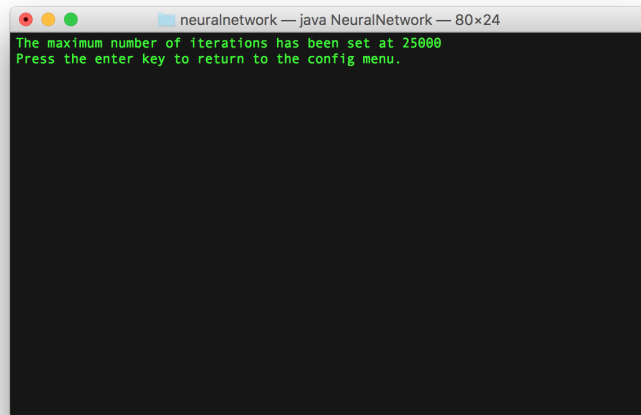


Set Maximum Training Iterations

The user is prompted to enter the number of training iterations. This number must be above 0.



A confirmation box appears to confirm the users change to the max number of iterations.



Set Training Data

The user is prompted to enter the new location of where they want their training data loaded from. This is the directory to a folder. Either a fully qualified path can be used or you can use a relative path from the current running directory of the application. This path cannot include the tilde “~” character.

```
neuralnetwork — java NeuralNetwork — 80x24
Please enter the directory of the training data

/Users/Brenton/Downloads/neuralnetwork/ $ Training_Data
```

```
neuralnetwork — java NeuralNetwork — 80x24
Please enter the directory of the training data

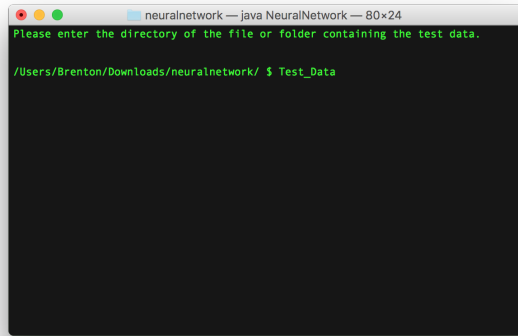
/Users/Brenton/Downloads/neuralnetwork/ $ /users/brenton/downloads/neuralnetwork
/training_data/
```

A confirmation box appears to confirm the users change to the training data files.

```
neuralnetwork — java NeuralNetwork — 80x24
The directory "Training_Data" has been set as the location for training data.
Press the enter key to return to the config menu.
```

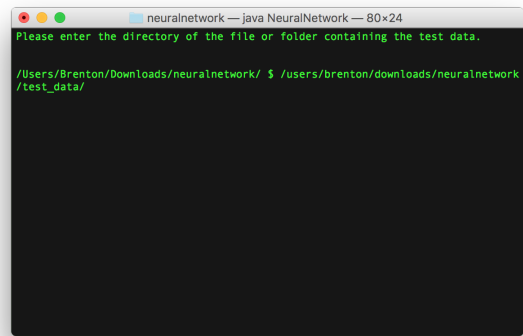
Set Test Data

The user is prompted to enter the new location of where they want their test data loaded from. This can be a folder or an individual file. Either a fully qualified path can be used or you can use a relative path from the current running directory of the application.



```
neuralnetwork — java NeuralNetwork — 80x24
Please enter the directory of the file or folder containing the test data.

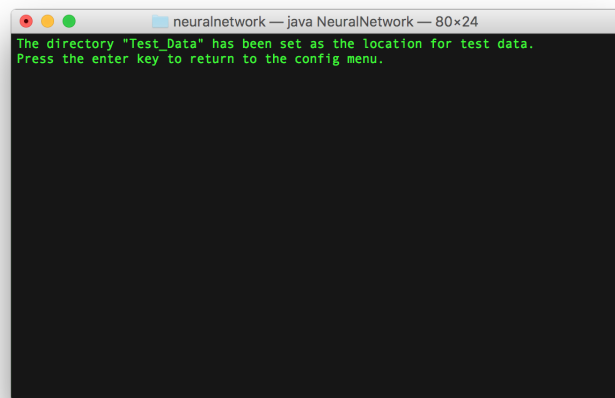
/Users/Brenton/Downloads/neuralnetwork/ $ Test_Data
```



```
neuralnetwork — java NeuralNetwork — 80x24
Please enter the directory of the file or folder containing the test data.

/Users/Brenton/Downloads/neuralnetwork/ $ /users/brenton/downloads/neuralnetwork
/test_data/
```

A confirmation box appears to confirm the users change to the training data files.



```
neuralnetwork — java NeuralNetwork — 80x24
The directory "Test_Data" has been set as the location for test data.
Press the enter key to return to the config menu.
```