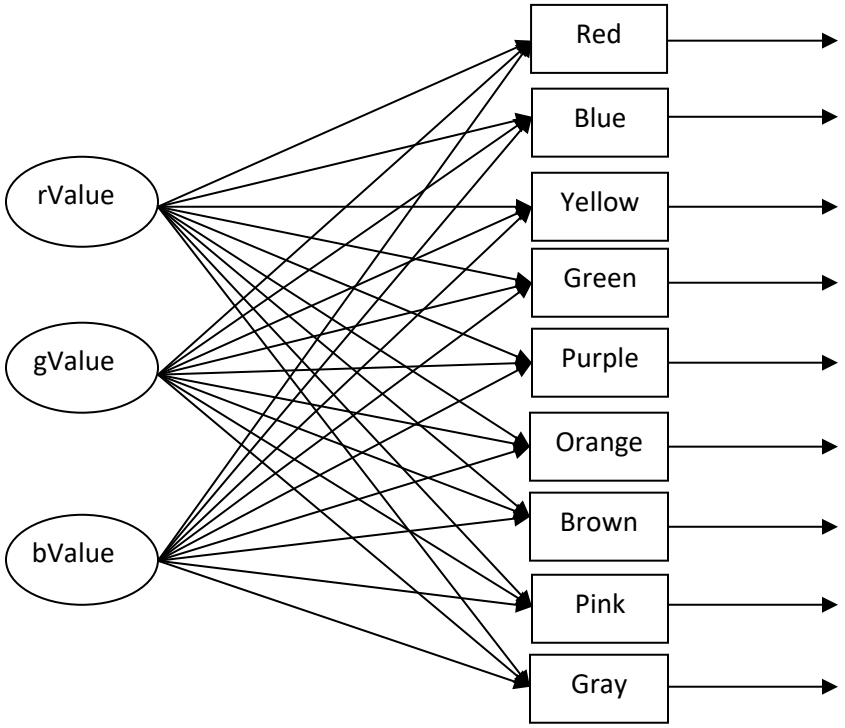
Project 4: Neural Nets

Goal: To gain experience with implementing an artificial neural network and with the design choices that accompany using one.

Description: You are going to create a neural network to learn to classify colors. At a minimum, you should implement perceptron learning to train your network. (If you wish to create a multi-layer neural network and implement backpropagation to train it, then you are certainly welcome to do so – and you will likely get better accuracy results.)

The input to each node will specify the quantity of red, blue and green in a particular color (0 - 255). The network will have 9 output nodes to classify the color as "Red", "Blue", "Yellow", "Green", "Purple", "Orange", "Brown", "Pink" or "Gray". For example, if the color is purple, the "Purple" node should output 1 and all others should output 0.



Training Data: You have been given a set of data files: nn_debug.txt, training.txt and test.txt. You should use nn_debug as a starting point for creating your own debugging file and you should use your debug file for the development and debugging of your neural net code. Only once you are confident that it is debugged should you turn to the real data, which has been split into a training set and a testing set. You are welcome to break the training data into a training and validation set, if you would like. I have an additional held out set of data

that the grader will use for testing that is different from either of your sets. The data was generated by past and present students, so there is some noise in it.

It is your choice whether you wish to pre-process your inputs to put them in the range 0 - 1, rather than 0 -255.

Parameters: Your code should provide a convenient interface for setting the various parameters that go into your neural network. Some options for the interface would be: a set of constants at the top of your source code file, a GUI form, or a config file. Whichever interface you use, your README file should clearly explain how to use each parameter and what values are appropriate for it. Your code should validate that the parameter values are appropriate. The following parameters should be controllable:

- The filename containing the training data.
- The initial learning rate.
- The number of epochs.
- Whether to load in the initial weights for the neural network from a file or initialize them randomly.
- Etc.

You can choose whatever threshold value you like for the neurons as well as whether to include a bias input.

Your code should provide the option to store the currently learned weights into a file, using whatever format you choose.

Testing:

Your program should be able to open a set of testing data and run the samples through the network. Compute the following statistics and display them at the end of the test set:

- What percentage of the examples in the testing data set were perfectly classified: i.e. only the correct neuron and none of the others fired.
- What percentage of the examples in the testing data set caused multiple neurons to fire.
- What percentage of the examples in the testing data set caused zero neurons to fire.
- For each individual color neuron, calculate the percentage of times that it (a) fired correctly, (b) fired when it shouldn't have (false positive), and (c) failed to fire when it should have (false negative).

Coding Requirements:

You are free to choose which language you would like to use in implementing the project. The grader already has C++, Java and Python compilers installed. If you wish to use a different language, you must include a README file with your project submission in which you give step-by-step instructions on where to download the compiler, how to install it (and any libraries that are needed) and how to run your project once the compiler is installed.

Your code is expected to follow all good programming practices, i.e. well-commented, broken into functions, thoroughly tested, etc.

Submission:

Please zip up the entire contents of your project and submit the zip file via Canvas. At a minimum, the zip file should contain:

- Your source code
- Your README file