Enea Dushaj

The code is written in C++ and is compiled and run in visual studio. It trains and tests the neural network all at once. When run, the code prompts the user for the following:

* A file specifying the initial neural network to be trained
* A training file for the given neural network
* A test set file to test the neural network
* The number of epochs to train the neural network
* The learning rate
* A name for the output file representing the trained neural network to output to the current working directory
* A name for the results file when the trained neural net is applied to the test set

The terminal displays the neural net before and after training, as well as the output of the neural network, and the metrics of the neural network.

When “AIProject2\_Dushaj.cpp” is compiled and run on visual studio, you will be able to train and test a neural network, while also outputting files for the trained neural net and results of testing.

The data set I am working with is the Iris data set from the UC Irvine repo. It is a very famous and popular dataset used for pattern recognition. It classifies iris plants into one of 3 types of iris plant. Each plant may only belong to one class at a time. The output is the class of plant, which is either Iris Setosa, Iris Versicolour, or Iris Virginica. Every example represents a plant, for which measurements for the sepal length, sepal width, petal length, and petal width have been take, and the classification of the plant is given. I have modified the data to output in the following way:

Whereas the provided data set is, Iris Versicolour, or Iris Virginica as the output, I

* 1 0 0 for Iris Setosa
* 0 1 0 for Iris Versicolour
* 0 0 1 for Iris Virginica

There are 4 input values, and 3 output values. The 3 output values are outlined above. The 4 input values are as follows:

1. sepal length in cm  
2. sepal width in cm  
3. petal length in cm  
4. petal width in cm

Where the sepal is the green part underneath the petal that supports the petal.

The dataset contains 150 points. 100 are used for training, 50 are used in the test set. They were randomly split up and put into each file.

Povided are the following files:

* iris.init – the initial neural network with 10 nides in the hidden layer, with randomly generated weights. The file was created using the file named, “initial\_nn\_generator.cpp” This code takes in the number of inputs, number of hidden layers, and number of output layers, and generates a neural network with random weights between 0.0 and 1.0
* iris.train – contains 100 random data points from the dataset for training
* iris.test - contains 50 random data points from the dataset for testing
* iris.trained – the trained neural net after 100 epochs with learning rate of 0.2
* iris.results – the metrics of the output of the neural network as compared to the test set
* iris.data – contains the original data downloaded from UC Irvine’s repository
* iris.names – description of the data set from the creator

The data was put into an excel spreadsheet, the maximum value of the inputs was found, and every input was divided by the maximum value in the respective file. This was done to “normalize” the data.

Included in the email is the original data set.

The neural trains well with 100 epochs and a learning rate of 0.2, with 10 hidden layers.