

ML Project

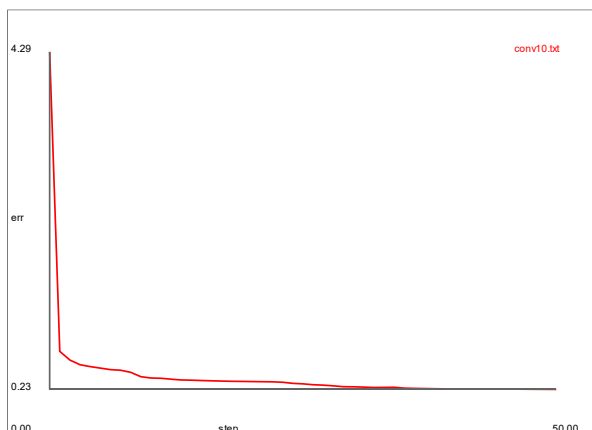
Michael Visser

300652084

The values that I started for the project were close to the outcome that I got. This was just pure luck, so the fine tuning didn't take many attempts.

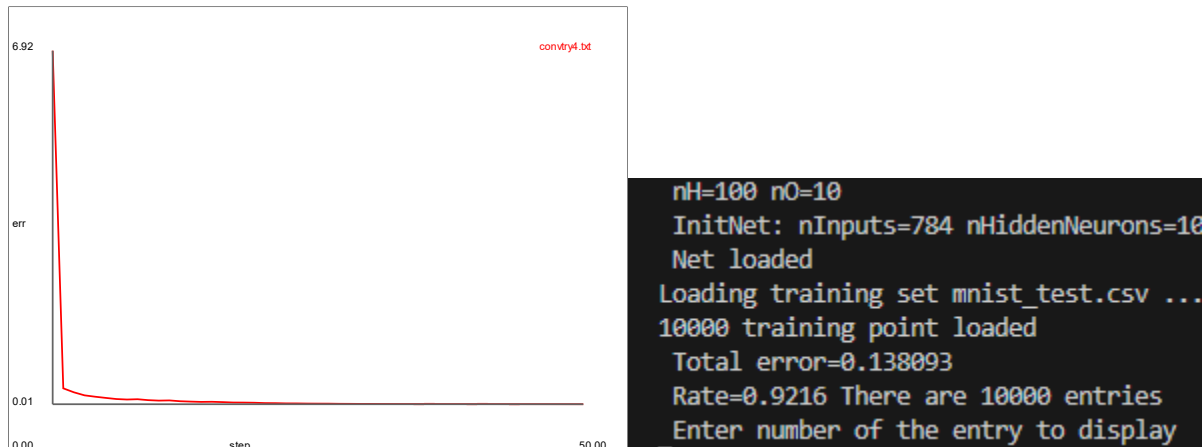
I started with a median value of 0, deviation of 1, learning rate of 1, batch of 1, hidden neurons were 50, epochs were 50, and the sample size was 5000. This gave me an error rate of around 0.3. I then put this into the test code and it gave me an accuracy of 74%.

Raising the learning rate made it worse so I decided to put it at 0.1 to see if that would be the correct direction. This gave a lower error rating but a lower accuracy, so I knew I needed to tweak the median and deviation to get a lower error rate.

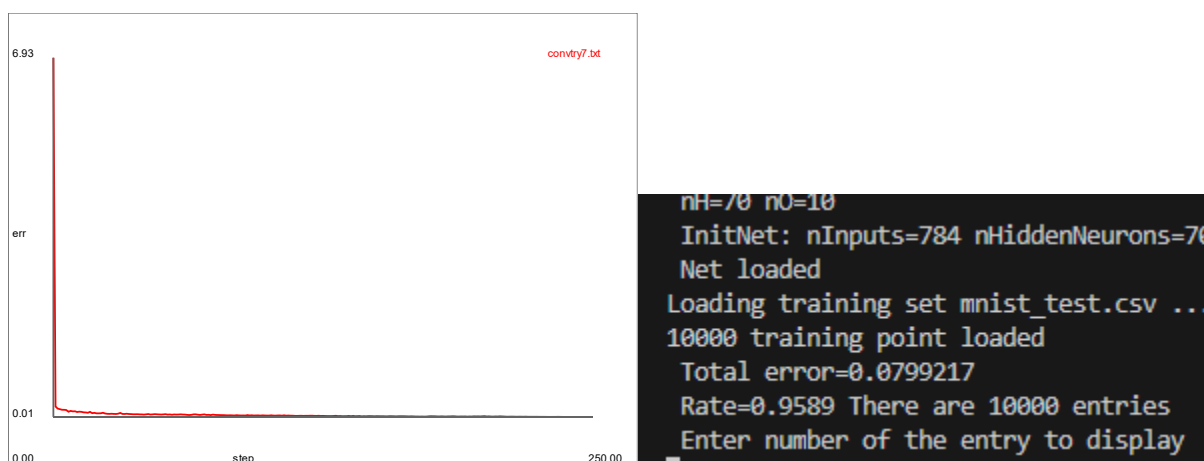


```
10000 training point loaded
Total error=0.379485
Rate=0.7208 There are 10000 entries
Enter number of the entry to display
```

I then set the median to 0.2 and the deviation to 0.4, hidden 100 epoch 50, and samples of 5000. After this the error was at 0.01 so I knew this was the right direction. This gave me an accuracy of 92% so this was a massive improvement from the original 74%



I was getting close to a value that I was acceptable with so I started tweaking the values very slightly and came to a final accuracy 95.89%. This was done by making the median 0.2, deviation 0.4, learning rate 0.3, batch of 1, hidden neurons at 100, epoch at 250, and samples at 20000. After trying to tweak the values more I wasn't getting any better results in term of accuracy so I decided to just keep the one I have at 95%.



As you can see the graph drops very suddenly down to about 0.2 which shows there is definitely improvements to be made, weather that be with the deviation, median, or the learning rate.

During this project I figured out that too many hidden neurons affected the error rate negatively. I tried two different sets of values but only changing the hidden neuron value. The difference between 70 and 100 was enough to make me stick with a value of 70. This is also the case because the value being too low.

This was also the same for the weight values. Too high or too low would negatively impact the error rate. I set with those final values because at even 0.29 or 0.31 for the learning rate, it would still be a worse outcome than 0.3. This was the same for the median and deviation as too high or low would make the error worse than the outcome I got.

While tweaking the values to try get an optimum error rate, I figured out that changing the sample value would change how the weight values affect the error rate which threw me off course a little bit, so I decided to run tests with samples at about 10000-20000 and just let them run while I did other things so that I didn't affect the other values.