Austin Ragotzy

6-11-2018

artificial neural networks

Tensorflow

So I was playing around with tensor flow and created a regular network and then I created a convolutional network. For the regular network I followed a tutorial that was on the tensorflow webpage. Making the regular network was pretty simple make some placeholders for what the network is going to be seeing as far as input, then create each layer then an out put. After that I got it to learn from its mistakes. In doing this I was able to attain upwards of 95% accuracy on the mnist dataset. I did not do a whole lot of tweaking the architecture and learning rate and stuff like that but I was able to get it up from 90% to that 95% with a few small changes.

After that I started working on the convolutional network it was pretty simple to set up all the layers and calculate loss and optimize the system, but the part I had the most trouble with was the getting the data to the right size and shape, and making the model accept data that is the size and shape that I want it to be. It took me quite a while but with the help of the python console I was able to figure out the shape of the data I was giving it and how I needed to change it. I did have to cut the size of the test data set from what they gave me because tensorflow said it took to much ram. I changed it from 10000 to 1000 and after 3000 batches of 150 I got 98.9% accuracy . I tried it with a test set if 100 and got 100% accuracy after 2000 batches of 150.

This exploration I found to be insightful because I learned about how to construct the model and that I learned a lot about how to get the data into the shape and structure it needs to be in to use in a tensor. I found the convolutional networks to be more accurate since the network was almost 4% more accurate.