Project X Writeup

Getting over the bump

It took multiple shuffling of properties before I made significant progress for both task one and two. In task one, I had added too much data augmentation for my simple CNN and it prevented me from making progress until I eventually found the culprit (gaussian noise with sigma 1 in a standardized environment). After removing it, I slowly tuned other parameters such as the optimizing method and rates until I eventually was training to 45% validation accuracy in around 25 epochs. In task two, my learning rate was way too high for any shuffle of my ideas to work, so until I lowered it to .005 down from values like .05 and .03 I was pretty stand still as far as attaining formidable accuracies.

Architecture intuitions

For task 1 we were presented with a starting model that had just one convolutional layer. Though it's kernel size of 9x9 and the following maxpooling layer implied that the perceptrons in the full connected layer were dealing with a large field of interest, the data wasn't being convolved or optimized more than once, which I understood as causing the CNN to learn a coarse understanding of the dataset. I tried to add in 2 layers of smaller convolutions in the beginning to account for another 2 layers of scales, but then after max pooling the last layer was far too large, causing errors too occur due to negative dimensions. Accordingly, I changed all of the kernel sizes to 3x3 which meant the second layer was looking at 3x3 grouping of 3x3's (same field of interest as a 9x9) and the last layer was three times convolved values in a field of interest sized at 27x27. Since the perceptrons in last layer had a large field of interest I chose not to maxpool the last layer because It would delete too much information.

For task 2 we were presented with a model that knew to recognize objects. My initial intuition was to add one more layer on top of the last object output layer, to learn to make decisions based on objects (or at least responses to objects). Accordingly, I set the gradientstop after the vgg16 model and had it only train the last layer. This didn't work too well, but in retrospect it may have been because of my learning rate. I then decided to go for something simpler, like deleting the last layer and having my 15 output fully connected layer at the end replace the layer that was learning objects. Since I was now using all of the old parameters for a new task, I turned off the gradientstop at this point. Surely, I achieved 84.6% accuracy after 25 epochs. Since I spent too much time on the assignment, this was the end.

Data augmentations

When I was faced with the data augmentation step in task 1, I initially over did it, making augmentors that made the dataset too complex for my feeble small minded CNN. They included a horizontal flipper, horizontal translations, contrast and brightness scaling, and gaussian noise. I eventually chose to stick with just the flip and translate for my model. For task 2 though I used all of them and it worked well.