David Schulz
CS 3450
Lab 8: CNNs
5/14/2021

Training Runs After Changing Hyperparameters

Motivations for Changes

1. No change (first run)
2. The only thing I changed was putting 3 filters into the first convolutional layer instead of one. I just wanted to see not only if it improved the accuracy at all, but also if it still worked without breaking.
3. Since the network was already working very well with the Fashion-MNIST dataset, I moved on to training with the CIFAR-10 dataset. To start, I also put the number of filters back down to 1. To fit the dataset, the number of channels in the first convolutional layer must be set to 3.
4. I tried changing the first convolutional layer to have 3 filters again just to compare its performance to how it was with the Fashion-MNIST dataset.
5. I changed the filter size of the first convolutional layer to 5x5 to see if a larger filter would affect the accuracy. I also changed the number of filters back to 1 because making it 3 only seemed to make it worse.
6. After changing the filter size of the first layer to 5x5 showed some improvement, I changed the number of filters back to 3 to see if it would assist the larger filter size better.
7. After several more unrecorded runs, I was able to narrow down to a network that got over 40% testing accuracy with the CIFAR-10 dataset. The biggest change is that I added another Linear layer with 128 nodes since it was something I hadn't tried yet. In addition, I lowered the learning rate to 0.001, set the first convolutional layer to have 8 filters, and only did the training for 15 epochs. I set the learning rate to 0.001 because 0.01 seemed to train it too fast and make it unstable. I set the first convolutional layer to have 8 filters because it seemed to be a good estimate based on values I previously attempted. I only did it for 15 epochs because similar previous runs showed that 20 was unnecessary.

Reflection

In this lab, after adding the convolutional and flatten layers, it seems safe to say that the performance of the new network is a significant improvement over the previous lab's network with only fully-connected layers. The network got over 80% testing accuracy after the first try without the need for 100 hidden nodes in a linear layer. If we continued to spend time finding better network structures and hyperparameters, the performance would be an even larger improvement. Also, from the runs I tried, it appears that larger or more filters in the convolutional layers doesn't necessarily mean that the performance will improve.

Values During Each Run

Run	GPU Estimate	Layers, Channels, Filters*	Learning Rate	Reg Constant	Epochs	Test Accuracy	GPU Usage
1	**Mem:** 500MiB **Time:** 20mins	**Layers:** 2 Conv, 1 Linear **Channels:** 1,1 **Filter Count:** 1,1 **Filter Size:** Both 3x3	0.01	1/60,000	20	0.8113	909MiB
2	**Mem:** 909MiB **Time:** 25mins	**Layers:** 2 Conv, 1 Linear **Channels:** 1,3 **Filter Count:** 3,1 **Filter Size:** Both 3x3	0.01	1/60,000	20	0.8424	909MiB
3	**Mem:** 1200MiB **Time:** 35mins	**Layers:** 2 Conv, 1 Linear **Channels:** 3,1 **Filter Count:** 1,1 **Filter Size:** Both 3x3	0.01	1/50,000	20	0.2787	1315 MiB
4	**Mem:** 1315MiB **Time:** 35mins	**Layers:** 2 Conv, 1 Linear **Channels:** 3,3 **Filter Count:** 3,1 **Filter Size:** Both 3x3	0.01	1/50,000	20	0.1	1315 MiB
5	**Mem:** 1500MiB **Time:** 40mins	**Layers:** 2 Conv, 1 Linear **Channels:** 3,1 **Filter Count:** 1,1 **Filter Size:** 5x5, 3x3	0.01	1/50,000	20	0.2853	1315 MiB
6	**Mem:** 1315MiB **Time:** 35mins	**Layers:** 2 Conv, 1 Linear **Channels:** 3,3 **Filter Count:** 3,1 **Filter Size:** 5x5, 3x3	0.01	1/50,000	20	0.289	1315 MiB
7	**Mem:** 1315MiB **Time:** 25mins	**Layers:** 2 Conv, 2 Linear (128 nodes) **Channels:** 3,8 **Filter Count:** 8,1 **Filter Size:** Both 3x3	0.001	1/50,000	15	0.4462	1315 MiB

*Padding is always 0