11785 HW 2 Write-up

Jagjeet Singh (jagjeets)

1. **Strided CNN Architecture**

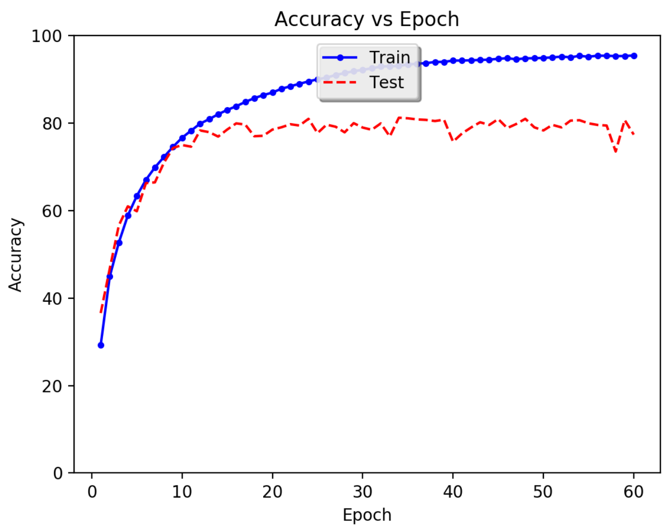

Figure 1: Training and Test accuracy vs Epochs

Hurdles faced and how they were overcome:

1. Gradient descent getting stuck at local minima: The training error was consitently 2.303 because the gradient descent was getting stuck at local minima. To overcome this problem, I tried different learning rates to come up with the one which was overcoming the problem of local minima. Surprisingly, decaying learning rate was not helping. As such, I kept a fixed learning rate of 0.01 which proved to be a good enough estimate. Another precaution that I adopted was to use momentum in gradient descent

2. Large data size: As the datasize was huge, the training was remarkably slow. I tried different batchsizes and finally settled to the batchsize of 4

3. Parameter Initialization: Parameter initialization was another factor which helped in overcoming the problem of local minima. In PyTorch, the default initialization of conv2D parameters was not an appropriate one for this architecture. As such, I read about more initialization methods and finally implemented Xavier initialization desctibed in Understanding the difficulty of training deep feedforward neural networks - Glorot, X. Bengio, Y. (2010). PyTorch provided a straightforward implementation of this.

4. Low Accuracy: The accuracy was initially very low (about 56%). So I tried different data prerocessing methods and gradient descent optimizers and finally achieve an accuracy of 80% after 80 epochs.

5. Weight Decay: A weight decay of 0.001 helped in avoiding overfitting the training data

2. **All CNN Architecture**

Changes made as compared to Strided CNN:

1. More convolution layers: This is an ALL-CNN architecture as suggested in the reference paper. Compared to the previous architecture which had 7 convolution layers, this has 9 convolution layers.

2. Data Augmentation: The input data is heavily augmented in this architecture. The input size of 32x32 is first 80%cropped and then re-scaled to 126x126. Then a random horizontal flip is applied followed by normalization.

3. Final layer size: Unlike the previous case where global averaging was done over 6x6 spatial dimensions, here averaging is done over 30x30 spatial dimensions.

4. Accuracy: Due to 4 times larger image size, each epoch was taking too long to give an idea about achievable test accuracy. However, since the architecture is similar to the one suggested in the reference paper (with 4.41% test error), I expect the accuracy to be more than 92%.