

COL780 : Computer Vision

Assignment 4

Gaurav Jain : 2019CS10349

1 Implementation Details

- I have modified the VGG architecture by removing the last FC layer and added a new FC layer according to the number of classes. I do not freeze the convolutional layers' weights to get better fine-tuning of the model.
- In the final hyperparameter setting, I have performed training for 15 epochs with a learning rate of 0.001 and Nesterov+Momentum optimizer with weight decay of 0.001. I tracked the training loss and validation accuracy for each epoch.
- For the best model, I have selected the model with best validation accuracy. I have obtained a model that shows 99.2% validation accuracy and 99.2% test accuracy. This link contains the best model trained by me.

2 Experiments

2.1 Optimizer

I tried Adam, SGD+Momentum and Nesterov+Momentum as optimizers. I observed that Nesterov+Momentum gives the best validation accuracy and converges faster compared to other optimizers.

2.2 Learning Rate

- I experimented with learning rates of 1e-1, 1e-2, 1e-3 and 1e-4. I observed that 1e-1 and 1e-2 are very high learning rates and the training loss is not converging with these learning rates.
- The training loss with learning rate of 1e-3 seems to converge around 15-20 epochs. This is the learning rate at which I was able to obtain best validation accuracy.
- The training process with learning rate of 1e-4 seems to converge around 25-30 epochs with identical training loss for learning rate 1e-3. Thus, 1e-3 is the learning rate selected for the final model.

2.3 Regularization

All experiments are done using Nesterov+Momentum=0.9 and training is performed for 15 epochs with L2 regularization.

| Weight Decay | Training Loss | Best Validation Accuracy | Test Accuracy on Best model |
|--------------|---------------|--------------------------|-----------------------------|
| 1 | 3.203 | 4% | 4% |
| 0.1 | 2.556 | 26.4% | 28.2% |
| 0.01 | 0.344 | 97.6% | 92.8% |
| 0.001 | 0.047 | 99.2% | 99.2% |
| 0.0001 | 0.003 | 97.6% | 98.4% |

3 How to run the code?

Run the command: `python assign4.py arg1`. This will train the model for 15 epochs with learning rate 0.001 and weight decay present in arg1. The path to the dataset folder is hard-coded inside the file.