HW P2

Sunday, February 28, 2021 4:54 PM

Training Loss after 15 epochs: 0.004
Test Accuracy: 80%

Architecture:

Input: Batch size * 32 * 32, input channels=3

Conv1
- Conv2D Layer: (16 3x3 filters --> 30 * 30 * 32)
- BatchNorm2d (Normalization)
- ReLU
- MaxPool2D(2,2) -> 15*15*32

Conv2
- Conv2D Layer: (64 3x3 filters --> 13 * 13* 64)
- BatchNorm2d (Normalization)
- ReLU

Conv3
- Conv2D Layer: (128 3x3 filters --> 10 * 10 * 128)
- BatchNorm2d (Normalization)
- ReLU
- Dropout(p=0.05) (Prevent overfit)
- MaxPool2D(2,2) -> 5*5*128

FC
- Linear (3200->1024)
- ReLU
- Linear(1024->512)
- ReLU
- Linear(512->128)
- Dropout(p=0.05)
- Linear(128->10 output classes)

Parameters:
- Batch size: 64
- Epochs: 15
- Learning rate = 0.001

Criterion:
- Cross Entropy Loss

Optimizer:
- Adam Optimizer

LR_scheduler:
- ReduceLROnPlateau('min')

Reflection:

For the CNN architecture, I used three Conv layers which act as feature extractors, consisting of Conv2d, Batch Normalization (to smooth out the covariate shift that results from differing data distributions between batches), MaxPool 2d to downsample the input image, and ReLU activation. Finally, the Fully Connected layer, which acts like a "classifier", connects the flattened 3200-element vector down and uses a series of Linear layers to produce 10-element output vector.

I experimented with a number of epochs between 10 and 50, and found that with a learning

rate of 0.001 and a batch size of 64, after around 12 epochs the training loss plateaus and reaches A small value. Therefore, I used the LR scheduler and experimented with both StepLR (multiplying by a factor of 0.25 every 5 epochs), and ReduceLROnPlateau, to create an adaptive learning rate which adjusts with the number of epochs/the trianing loss.

My initial batch size of 8 is smaller, which means more frequent (and noisier) updates. I tried batch sizes of 16, 32, and 64. 64 seemed to yield smooth, relatively fast updates.

For the optimizer, Adam optimizer seemed to yield slightly higher results than SGD with momentum of 0.9.