

Homework 3: CNN for CIFAR10

Alex Faustino - afausti2

September 30, 2019

The attached file, `faustino_hw3.py`, contains a script which creates a CNN using PyTorch's `Sequential` class. The CNN is trained on a GPU, using `Adam` as the optimizer, to correctly classify images from the CIFAR10 data set. The input, X , is augmented using the `torchvision.transforms` class. The CNN has three convolution blocks and two fully-connected hidden layers. Each convolution block contains two convolution layers; a batch normalization after each convolution; and a max pool and dropout layer at the end of each block. The output from the last fully-connected hidden layer is passed through `Softmax` to determine the CNN's prediction.

Additionally, there are two utility functions and three utility classes in `faustino_hw3.py`. The functions, `save_checkpoint` and `load_checkpoint`, allow for the saving and loading of the model's state after each epoch. The classes, `AverageBase`; `RunningAverage`; and `MovingAverage`, are used to smooth the training and validation results' plots.

The CNN parameters are:

$$X = 3 \times 32 \times 32$$

$$K_1 = 32 \times 3 \times 3$$

$$K_2 = 64 \times 3 \times 3$$

$$K_3 = 128 \times 3 \times 3$$

$$\text{dropout}_1 = 0.2$$

$$\text{dropout}_2 = 0.3$$

$$\text{dropout}_3 = 0.4$$

$$d_{h1} = 576$$

$$d_{h2} = 288$$

$$\alpha_{init} = 0.001$$

$$\text{training batch size} = 128$$

$$\text{testing batch size} = 64$$

The results from training for 120 epochs can be seen in Figure 1 and Figure 2 with the final epoch's test accuracy being **84.0%**.

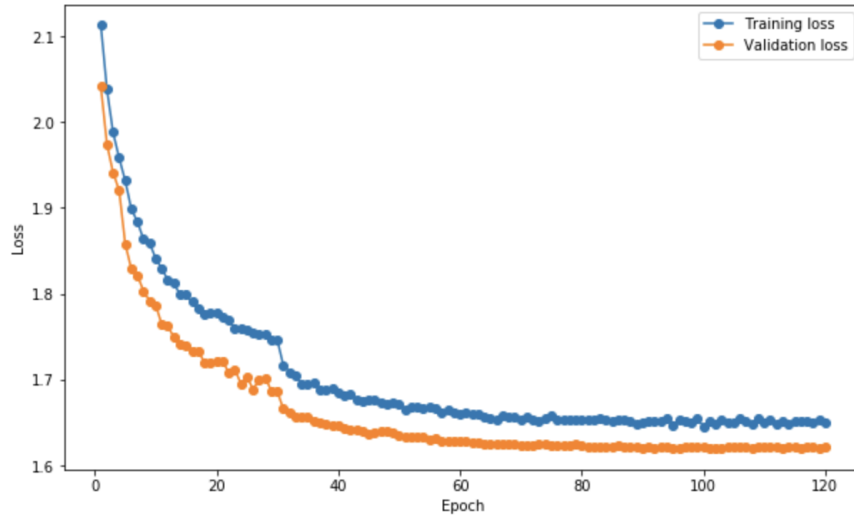


Figure 1: Training and validation loss of the CNN model over 120 epochs.

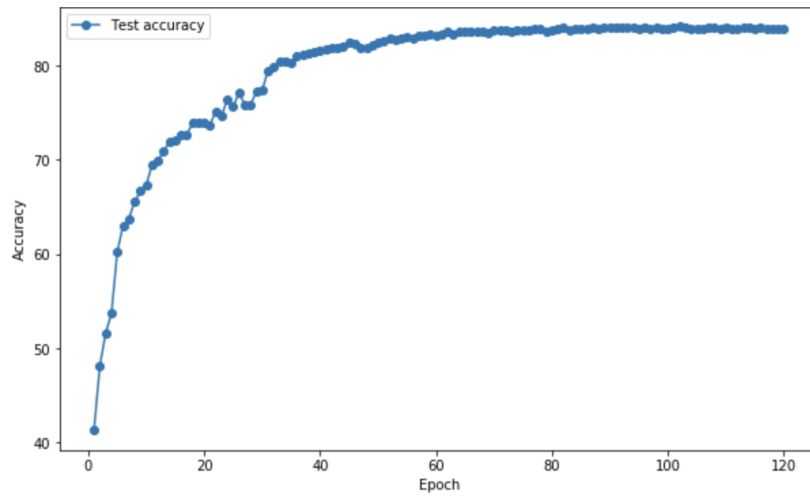


Figure 2: Test accuracy of the CNN model over 120 epochs. We can see that the model achieves the required 80% accuracy around epoch 40.