**What external sources did you make use of, and how did you adapt these, go further or experiment in order to achieve the best possible result?**
I mainly adapted the PyTorch beginner blitz CIFAR-10 tutorial. I tweaked it to make it easier to knob tune, as well as condensed some of the code and output. I added a validation set and validation step using this geeks-for-geeks tutorial. To decide on the class design for my FCNN, I drew inspiration from this github repo and reused the same training and testing method used for my CNN. I then modified each of the hyperparameters as I explain in a later section.

**Between fully connected and convolutional neural networks, which performed better? Why do you think this is the case?**
Convolutional performed better. This is due to the filters that are applied during the steps which help the CNN with pattern recognition. It also leaves more knob tuning to be behind the scenes rather than with the programmer.

**How did you decide on your learning rate, optimization, any regularization or other hyperparameters?**

I simply used knob tuning. I started with the optimizer, going through each one in the PyTorch documentation and viewing the accuracy given when using the default settings. I found Adam and SGD to be at the top and I looked online to verify my results. I then played with the learning rate, starting at the recommended upper bound of 0.001 and lower bound 0.0001, and slowly centered the value until I found the area 0.0004-0.0007 to give a similar accuracy of around 55%. Then I set the epoch count to 10 and watched until I saw the loss improvement fall off and set that as the cut-off. Then I added more hidden layers to the network (only for CNN, adding more layers did not improve the FCNN), drastically improving my results. I added layers until the performance dropped off. At some point I played with the kernel size and the batch size, but did not find that they drastically changed the performance of the NNs so I focused my attention to other hyperparameters.

**State your best performance for fully connected and convolutional neural networks.?**
FCNN: acc - 49%, loss ~ 1.6279712915420532

CNN: acc – 66%, loss ~ 1.836150884628296

**If you had more time or could repeat the assignment, what might you do differently?**

- Watched more videos to build a greater understanding of neural networks. More specifically CNNs and FCNNs
- Play with more permutations of hyperparameters
- Finish the basic structure earlier so that I could spend more time playing with hyperparameters
- I didn't know that you could use GPU until the end for Google Colab, which was lots of time wasted