Name and Grade: _____

1. (10 points) Using what you learned about Pytorch in the lecture, as well as your own exploration of the torch.nn and torch.optim modules, write a program that classifies objects in the CIFAR-10 dataset. You will find that you can tackle this problem, which is much harder than fitting the sine curve, in only a few lines of extra code. Attach a copy of your code as well as a short reflection. Guiding Questions: What did you learn about Pytorch through this exercise? What architecture did you use? What was your accuracy? What steps did you take to increase the accuracy? How long did you need to train? Did you perform and image preprocessing or data augmentation, and if you did, how did that affect accuracies?

   You will want to use these lines to access the dataset:

```
1   from torchvision import datasets
2   cifar10 = datasets.CIFAR10(root='data/', train=True, download=True)
3   cifar10_val = datasets.CIFAR10(root='data/', train=False, download=True)
```

2. (5 points) Replicate these two CNN architectures and train them on the CIFAR-10 Dataset (you may need to change input/output layer dimensions, do this as you like). How do the accuracies compare? Why do you think that is? You can attach your code for this as well.

LeNet

Image: 28 (height) × 28 (width) × 1 (channel)
↓
| Convolution with 5×5 kernel+2padding:28×28×6 |
↓ sigmoid
| Pool with 2×2 average kernel+2 stride:14×14×6 |
↓
| Convolution with 5×5 kernel (no pad):10×10×16 |
↓ sigmoid
| Pool with 2×2 average kernel+2 stride: 5×5×16 |
↓ flatten
| Dense: 120 fully connected neurons |
↓ sigmoid
| Dense: 84 fully connected neurons |
↓ sigmoid
| Dense: 10 fully connected neurons |
↓

Output: 1 of 10 classes

AlexNet

Image: 224 (height) × 224 (width) × 3 (channels)
↓
| Convolution with 11×11 kernel+4 stride:54×54×96 |
↓ ReLu
| Pool with 3×3 max. kernel+2 stride: 26×26×96 |
↓
| Convolution with 5×5 kernel+2 pad:26×26×256 |
↓ ReLu
| Pool with 3×3 max. kernel+2 stride:12×12×256 |
↓
| Convolution with 3×3 kernel+1 pad:12×12×384 |
↓ ReLu
| Convolution with 3×3 kernel+1 pad:12×12×384 |
↓ ReLu
| Convolution with 3×3 kernel+1 pad:12×12×256 |
↓ ReLu
| Pool with 3×3 max. kernel+2 stride:5×5×256 |
↓ flatten
| Dense: 4096 fully connected neurons |
↓ ReLu, dropout p=0.5
| Dense: 4096 fully connected neurons |
↓ ReLu, dropout p=0.5
| Dense: 1000 fully connected neurons |
↓

Output: 1 of 1000 classes