# Convolutional Neural Network Exercises

1. **Counting Parameters**

   Consider a convolutional neural network that is used to classify images into ten classes. The following table describes the structure of the network. Complete the table with the number of trainable parameters associated with each layer. All convolutions are padded to retain the image dimensions. Assume that each convolutional filter and each dense unit have a single bias weight.

| **Layer Number** | **Layer Properties** | **Parameter Count** |
|---|---|---|
| Layer 0 | 32x32 3-channel color images | 0 |
| Layer 1 | Convolutional layer with 40 3x3 filters | |
| Layer 2 | Convolutional layer with 40 3x3 filters | |
| Layer 3 | Max Pooling layer with stride 2 and 2x2 pooling size | |
| Layer 4 | Convolutional layer with 64 3x3 filters | |
| Layer 5 | Convolutional layer with 64 3x3 filters | |
| Layer 6 | Dense layer with 128 units | |
| Layer 7 | Dense output layer with 10 units | |
| | | **TOTAL:** |

2. **Convolutional Networks in PyTorch**

   Using cifar_challenge.py as a starting point, implement a neural network classifier that conforms to the structure described above. The printout from `count_parameters` function should allow you to check your parameter counts.

   (a) What is the maximum validation performance that you see when you run this model? Is it better than your earlier results using a non-convolutional network?

   (b) After how many epochs does the model start to overfit the training data?

   (c) Try incorporating dropout, batch normalization, and/or L2 regularization to see if you can improve your validation performance.

   (d) (Optional) Try using PyTorch's image augmentation transforms to train your model using input data augmentation.