

## CS 547 / IE 534 Deep Learning, Fall 2019

### Homework 3

Author: Vardhan Dongre (vdongre2)

Implemented and trained a deep neural network Python using pytorch using **Google colab GPU hardware accelerator** for the CIFAR10 dataset. The neural network was trained on the Training Set using **ADAM**.

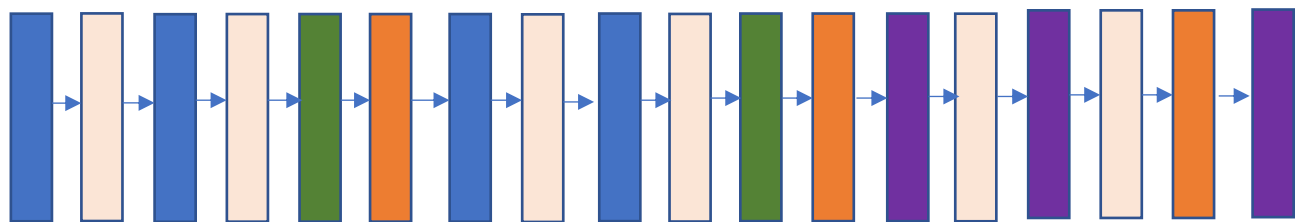
This implementation achieved an accuracy of **83.83%** on the Test Set.

#### Implementation:

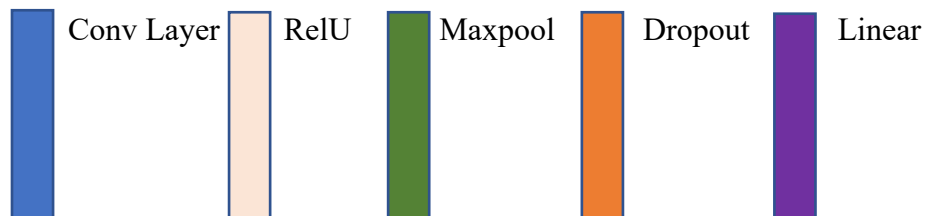
The code can be understood in the following subheads:

**1. Data:** CIFAR10 database is a database of rgb images. It has 60,000 32x32 color images in 10 different classes. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class

**2. Model:** The model/Learning Algorithm adopted is a Deep Neural Network. The model architecture includes layers of Convolution, ReLU, Maxpooling, Dropout and linear function. In this implementation extensive use of pytorch's library for model development and layers has been done. The data is first prepared by standard augmentation processes which include giving random rotations of 5 degrees or 10 degrees. A horizontal and vertical flip is also applied. The images are converted to a tensor and normalized. The model architecture can be understood by the following dig.



where,



In this implementation we have used kernels of size (3,3) and have provided a padding of (1,1). The initial channels for the first convolutional layer are taken to be 48 and the channels are doubled on each convolutional operation.

**3. Training:** The model is trained using the ADAM algorithms and we employ the negative log-likelihood error. For the training purposes Google Colab's GPU hardware accelerator has been used. The code includes torch.cuda that sets up and runs the CUDA operations. We set up a device in the beginning that makes sure the tensors are allocated to this device. We train the model for 100 epochs and using a batch size of 100.

**4. Testing:** The model is tested on the test data and we receive a 83.83% accuracy on it.

#### References:

<https://pytorch.org/docs/stable/>  
[https://computersciencewiki.org/index.php/Max-pooling/\\_Pooling](https://computersciencewiki.org/index.php/Max-pooling/_Pooling)  
<https://github.com/Ziyu0>  
<https://github.com/xinyanyang>