

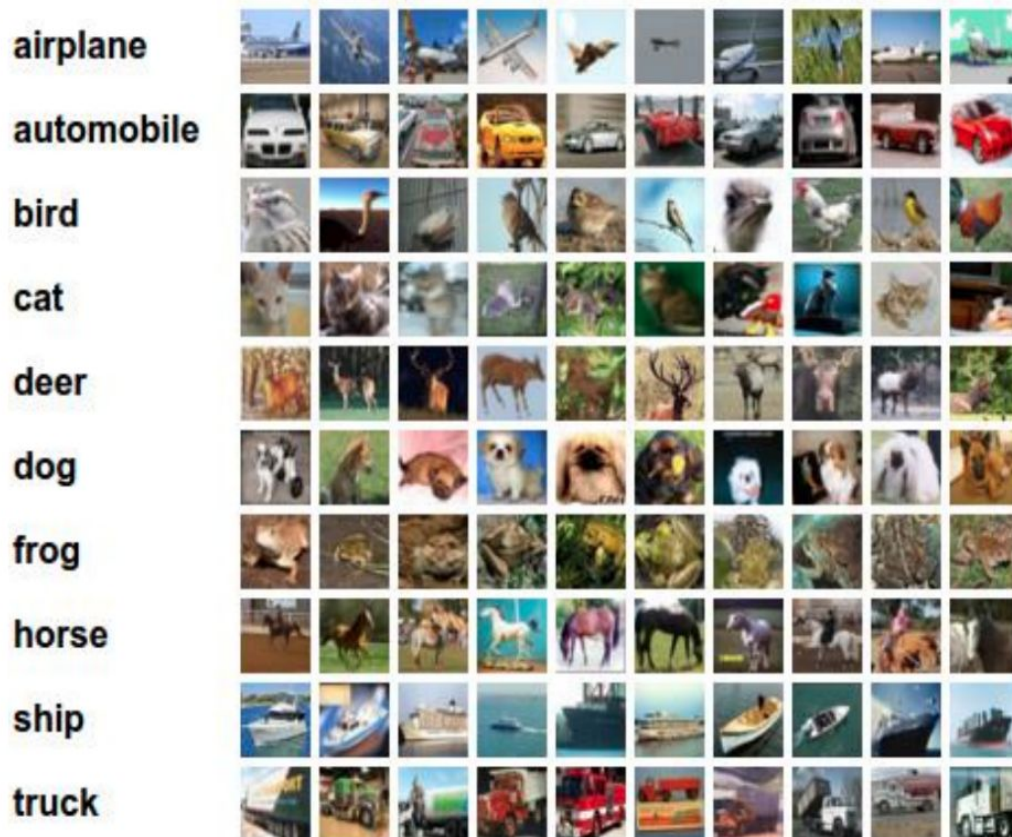
CMPUT 328 - Assignment 5

Image Classification on CIFAR-10 Dataset using CNN

Classification on CIFAR-10 dataset [5.7% of the total weight]:

Your task in this assignment is to do classification on CIFAR-10 dataset using PyTorch
CIFAR-10 dataset:

Contains 32x32x3 RGB images. There are 50000 images in the train set and 10000 images in the test set. Each image in CIFAR-10 dataset belongs to one of the ten classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.



For more information, see <https://www.cs.toronto.edu/~kriz/cifar.html>

A. Tasks:

In this assignment, you are going to implement the Convolutional Neural Network to do classification on CIFAR-10 dataset.

The network architecture in this assignment is not fixed. You will design the network architecture yourself.

However, there are some requirements that your network architecture must satisfy:

1. Your network must have at least 5 layers (a layer means convolution or fully connected layer) into an activation function like Relu, Softmax, Tanh or sigmoid..., (input layer, max pooling or

strided convolution layer used to downsample activation map doesn't count). The number of layers in your network will be very close to the number of activation functions)

2. Must have at least 1 convolution layer
3. Must have at least 1 fully connected layer at the end
4. Must have at least 1 max pooling or strided convolution layer (i.e. Convolution with stride > 1)

If your network architecture doesn't satisfy any of the above 4 requirements, you will lose marks. For every requirement that is not satisfied, you lose 10% of your mark.

Also, you have to:

- Use He/Xavier initialization (will be explained in section B)
- Use Regularization

If your network architecture satisfies the accuracy and time requirements but you didn't implement these, you will lose marks.

In addition to the above requirement, to do well in this assignment, you may have to:

- Try different activation functions (relu, tanh, sigmoid, elu...)
- Try different optimizer (For example: SGD, Adam, RMSProp, AdaDelta...)
- Try different loss function (For example: cross entropy, square error)

B. Explanation for the terms mentioned above:

1. Use He/Xavier initialization:

See:

<https://prateekvjoshi.com/2016/03/29/understanding-xavier-initialization-in-deep-neural-networks/>

You don't need to implement it manually, should be straightforward to do in PyTorch.

C. What to do:

You are given some code. There are 4 main parts in the code you need to complete:

1. load_dataset: You should load cifar-10 dataset, create the data loaders, split the dataset into training, validation and testing. **Take the last 5k of the training data to be validation data (not a random split)**
2. Network/model: you should define the network class in this part + any utilities related to weight/bias initialization
3. Train: you should have the training and validation in this part and return the final trained model
4. Test: you should test the model in this part and return predictions

D. SUBMISSION:

You need to submit 2 files:

1. Download the notebook as **.py file (not .ipynb)** and submit it.
2. Submit the saved model (**model_YOURIDNUMBER.pt**) - **(IMPORTANT!)**

DUE DATE: The due date is Friday, October 18 by 11:59 pm. The assignment is to be submitted online on eclass. For late submissions' rules please, check the course information on eclass

E. MARKING:

There is no in-lab demo for this assignment. Your final score depends on the correct implementation of algorithms and it must work on CIFAR10 dataset.

- **Base mark:**

Your algorithm will get mark based on its accuracy on Cifar-10 dataset which should **be minimum 65%** to get the maximum score. Score will scale linearly from 55-65% on the test set.

- **Time constraint:**

Your model testing **should not run for more than 30s** (With Google colab/CPU) using batch size of 1 for the test data loader.

COLLABORATION POLICY:

This must be your own work. Do not share or look at the code of other people (whether they are inside or outside the class). Do not copy the code from the Internet. You can talk to others that are in the class about solution ideas (but not so detailed that you are verbally sharing or hearing about or seeing the code). You must cite whom you talked to in the comments of your programs.