Lab 3: Custom Data Loader and Transfer Learning

ENGN8536, 2018

August 28, 2018

1 Custom Data Loader

In this lab we will be working with a custom dataset, a binary dataset containing dog and cat images. The main goal for this lab is to learn to write the data loader for custom datasets and also fine-tuning a pre-trained model.

PyTorch data loading tutorial: https://pytorch.org/tutorials/beginner/data_loading_tutorial.html

- First download the dataset from Wattle, and separate the data into training set and validation set.
- To load your own custom data, you will create your own class which inherits Dataset class (e.g. DogCatDataset(Dataset)).
- You can use **glob** to read all filenames and generate a list of image paths.
- Generate a list of labels according to filenames.
- Use the same network architecture from last week to train on the dogs and cats dataset. Since this is a binary dataset, you need to slightly modify the architecture and pick a suitable loss function. Draw training plots same as last week. Remember to separate training set and validation set. What accuracy do you get for your validation set?

2 Transfer Learning

The idea of transfer learning is to take a CNN that is pretrained on some other dataset (e.g. ImageNet) and then finetune the network on our own dataset, in this case, dog and cat images. This saves us the effort from training a deep network from scratch, which often takes a long time to converge and requires large computational resources.

- Load a pretrained network model using torchvision (e.g. ResNet, DenseNet, AlexNet or VGGNet).
- Create your network model (e.g. ResNet, DenseNet, AlexNet or VGGNet), but modify the final fully-connected layer and also the loss function.
- Load the parameters from the pre-trained network to your network model. If not sure how to do this, try to use Google.
- Train this model after loading parameters from the pre-trained model. If you want to train faster, one thing you can try doing is freeze all previous layer except the final fully-connected layer, i.e. only fine-tune the classifier. We can do this because all previous layers basically perform feature extraction and they are trained on ImageNet, which is a big dataset contains millions of images.
- Draw plots and compare with previous networks. What accuracy do you get for your validation set?

3 Submission

Please only submit a report in a single pdf format. In the report, you are required to demonstrate what you have done and also all your plots and results. Attach your important code at the end of the report if you want, though I will not carefully read it.