Lab 0

0 Practical

This task is meant to give you a refresh PyTorch or Tensorflow or Keras framework, which you will be using for the course exercises and also the project. So, the time you spend on this task is an investment for the project. You are free to choose any framework that you feel comfortable working with. You also are going to observe the effect of transfer learning on a (once) very popular architecture. Some parts are marked as optional since you have done it before; you can skip them, but it is a good exercise to reinforce previous knowledge..

Useful Links :

. PyTorch

. Tensorflow

. Keras

NOTE - Before getting started make sure you have the required packages installed along with a suitable IDE you want to work on. Also if your computer doesn't supports these installations, you can also work on Google Colab. Here Google provides computational capacity (to some extent) for running deep learning codes. These are the same as Jupyter Notebooks.

0.1 Task

The following steps will guide you in creating your own experiment.

- Download and prepare CIFAR-10 dataset (it is already available in the above mentioned libraries)
  - PyTorch
  - Keras and Tensorflow
- Write a simple CNN network for classifying images
  - use LeakyReLU as the activation function (optional)
  - use SGD as the optimizer and 0.0001 as the learning rate, and keep all default parameters (optional)
  - Report the accuracy on the test set (optional)
- Change the optimiser to Adam and run again the experiment. Report accuracy on test set.
- Swap the LeakyReLUs for Tanh. Then run again the experiment and report accuracy on test set. Make a separate file for this experiment.
- Visualize the results of these runs on a Tensorboard. Just put any screenshot of the web interface with the experiments to prove you got it working is enough. (for example- show the training loss on tensorboard)

0.2 Task

0.2.1 Transfer Learning from ImageNet

- Download and prepare CIFAR-10 dataset (it is already available in the above mentioned libraries)
- Use AlexNet as the model (Pytorch AlexNet)
- You have to perform two separate experiments-
  - Train the model for CIFAR-10 data, Report the test test accuracy. (also referred as fine tuning the model)
  - Use the pretarined weights of AlexNet, in other words use AlexNet as a pretrained network for image classification on CIFAR-10 data (also referred as Feature Extraction), Report the test test accuracy. (optional)
- In both the above cases remember to add an extra fully connected layer to the classifier with number of neurons = 10, because there are 10 classes in CIFAR-10 dataset. This layer will be trainable in both cases.
- Explain (briefly!) what is the difference between the two runs and why there is a difference in performance. (optional)

0.2.2 Transfer Learning from MNIST

- Prepare a CNN of your choice and train it on the MNIST data. Report the accuracy
- Use the above model as a pre-trained CNN for the SVHN dataset. Report the accuracy
- In the third step you are performing transfer learning from MNIST to SVHN (optional).