ML HACKATHON REPORT:-

(Classifying flowers(image))

R.Lakshmi Priya,

IMT2015037.

Lakshmi.Priya@iiitb.org

Project Description:-

In this project, i have trained an image classifier to recognise different species of flowers. You can imagine using something like this in a phone app that tells you the name of the flower your camera is looking at. In practice, you'd train this classifier, then export it for use in your application. We'll be using this dataset of 102 flower categories. You'll have an application that can be trained on any set of labelled images. Here your network will be learning about flowers and end up as a command line application.

Perquisites:-

The Code is written in Python 3.6.5 .Additional Packages that are required are: Numpy, Pandas, MatplotLib, Pytorch, PIL and json. You can use Jupiter notebook to view the files.

DATA:-

The data used for this assignment is a flower database are not provided in the repository as it's larger than lms allows.The data need to comprised of 3 folders, test, train and validate. Generally the proportions should be 70% training 10% validate and 20% test. Inside the train, test and validate folders there should be folders bearing a specific number which corresponds to a specific category,

clarified in the json file. For example if we have the image a.jpj and it is a rose it could be in a path like this /test/5/a.jpg and json file would be like this {... 5:"rose",...}. Make sure to include a lot of photos of your catagories (more than 10) with different angles and different lighting conditions in order for the network to generalize better.

Command Line Application:-

- Train a new network on a data set with train.py
    - Basic Usage : python train.py data_directory
    - Prints out current epoch, training loss, validation loss, and validation accuracy as the netowrk trains
    - Options:
        - Set directory to save checkpoints: python train.py data_dor --save_dir save_directory
        - Choose architecture (alexnet, densenet121 or vgg16 available): pytnon train.py data_dir --arch "vgg16"
        - Set hyper parameters: python train.py data_dir --learning_rate 0.001 --hidden_layer1 120 --epochs 20
        - Use GPU for training: python train.py data_dir --gpu gpu
- Predict flower name from an image with predict.py along with the probability of that name. That is you'll pass in a single image /path/to/image and return the flower name and class probability
    - Basic usage: python predict.py /path/to/image checkpoint
    - Options:

- Return top K most likely classes: `python predict.py input checkpoint ---top_k 3`
- Use a mapping of categories to real names: `python predict.py input checkpoint --category_names cat_To_name.json`
- Use GPU for inference: `python predict.py input checkpoint --gpu`

In order for the network to print out the name of the flower a .json file is required.

Hyperparameters:-

As you can see you have a wide selection of hyperparameters available and you can get even more by making small modifications to the code. Thus it may seem overly complicated to choose the right ones especially if the training needs at least 15 minutes to be completed. So here are some hints:

- By increasing the number of epochs the accuracy of the network on the training set gets better and better however be careful because if you pick a large number of epochs the network won't generalize well, that is to say it will have high accuracy on the training image and low accuracy on the test images. Eg: training for 12 epochs training accuracy: 85% Test accuracy: 82%. Training for 30 epochs training accuracy 95% test accuracy 50%.