UDACITY

< Back to AI Programming with Python Nanodegree

Image Classifier Application

REVIEW

CODE REVIEW 8

HISTORY

▸ train.py 4

▸ predict.py 3

▾ README.md 1

```
1  # AI Programming with Python Project
2
3  Project code for Udacity's AI Programming with Python Nanodegree program. In tl
4
5  Deliverables
6      Part 1
7      Image Classifier Project_AudreyTan_gpu.html
8          Completed Jupyter Notebook with run results from GPU environment
9
10     Image Classifier Project_AudreyTan_cpu.html
11         Completed Jupyter Notebook with run results from CPU environment
12
13    Part 2
14     train.py
15         Python command line app for training a new network on a dataset and sa
16
17     predict.py
18         Python command line app for using a saved checkpoint to predict the cl
19
20 Supporting files
21     cat_to_name.json
```

```
    workspace_utils.py

Installation Instruction
    unzip the zip archive into a folder structured like the aipnd-project folde

Run Instruction
    to run Part 1 deliverable as .ipynb, copy its cell contents into Image Clas

    run train.py and predict.py in a terminal session in the aip-project worksp
    predict.py can run in either CPU or GPU mode, it will verify a valid checkp
    N.B : though it's feasible to run train.py in CPU mode, it is more practica

    python train.py -h
```

AWESOME

Well done writing a clear and detailed `README` file along with instructions on how to run the project!

If you'd like to learn more about writing an effective `README` file, consider checking out this course.

```
      usage: train.py [-h] [--save_dir SAVE_DIR]
                      [--arch {densenet121,densenet161,vgg16}] [-lr LEARNING_RA
                      [-hu HIDDEN_UNITS] [-e EPOCHS] [--gpu]
                      [data_dir]

      positional arguments:
        data_dir              path to datasets

      optional arguments:
        -h, --help            show this help message and exit
        --save_dir SAVE_DIR   path to checkpoint directory
        --arch {densenet121,densenet161,vgg16}
                              model architecture: densenet121 | densenet161 | v
                              (default: densenet121)
        -lr LEARNING_RATE, --learning_rate LEARNING_RATE
                              learning rate (default: 0.001)
        -hu HIDDEN_UNITS, --hidden_units HIDDEN_UNITS
                              hidden units, one or multiple values (comma separ
                              enclosed in single quotes. Ex1. one value: '500'
                              multiple values: '1000, 500'
        -e EPOCHS, --epochs EPOCHS
                              total no. of epochs to run (default: 3)
        --gpu                 train in gpu mode

      Example calls:
      Ex 1, use data_dir 'flowers': python train.py flowers
      Ex 2, use save_dir 'chksav' to save checkpoint: python train.py --save_di
      Ex 3, use densenet161 and hidden_units '1000, 500': python train.py --arc
      Ex 4, set epochs to 10: python train.py -e 10
      Ex 5, set learning rate to 0.002: python train.py -lr 0.002
      Ex 6, train in GPU mode (subject to device capability): python train.py -


    python predict.py checkpt -h
      usage: predict.py [-h] [-img IMG_PTH] [-cat CATEGORY_NAMES] [-k TOP_K] [-
                        [checkpoint]
```

```
72        positional arguments:
73          checkpoint            path to saved checkpoint
74
75        optional arguments:
76          -h, --help            show this help message and exit
77          -img IMG_PTH, --img_pth IMG_PTH
78                                path to an image file
79          -cat CATEGORY_NAMES, --category_names CATEGORY_NAMES
80                                path to JSON file for mapping class values to cat
81                                names
82          -k TOP_K, --top_k TOP_K
83                                no. of top k classes to print
84          --gpu                 predict in gpu mode
85
86        Example Calls
87        Ex 1, use checkpoint 'chkpt.pth' in 'chksav': python predict.py chksav/ch
88        Ex 2, use top_k 4 and GPU : python predict.py --top_k 4 --gpu
89        Ex 3, use img_pth 'flowers/test/91/image_08061.jpg' and cat name mapper
90            python predict.py --img_pth flowers/test/91/image_08061.jpg --categor
91
```

▸ workspace_utils.py

RETURN TO PATH

Rate this review

Student FAQ