

Read Me

This .ipynb is for the Kaggle in-class competition: "Quick Draw".

The codes is designed to run on google Colab with GPU computation platform, yet it also works on local machine with only CPU supported.

Before running this ipynb file, you should download the training and testing data. Please go to <https://www.kaggle.com/c/ift3395-6390-f2018> (<https://www.kaggle.com/c/ift3395-6390-f2018>), download 4 data files: `train_images.npy` , `train_labels.csv` , `test_images.npy` , and `sample_submission.csv` , create a new folder named "input" in the directory where this .ipynb file is, and put the 4 files into the folder created.

`pytorch 0.4.1` is selected as our machine learning modules; besides, codes are also relied on the following basic python libraries:

- `python 3.5.2`,
- `numpy 1.14.2`
- `matplotlib 2.2.0`,
- `torchvision 0.2.1`

Some other modules are also used yet not fully listed here.

The whole ipynb is divided into the following 11 sections:

- Support for Codes Running on Colab with GPU
- Necessary Modules & Basic Tool Functions
- Loading Original Data
- Baseline model - Linear SVM
- Data Pre-Processing
- Data Augmentation
- Models
- Preparation for Training Competitive Models
- Training Models
- Performance Analysis
- Testing Models on Test Dataset

The first section gives us an access to use the Colab platform with high speed computation. If you encounter errors when running codes in this section, the following link could help you:

- <https://colab.research.google.com/> (<https://colab.research.google.com/>)
- <https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d> (<https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d>)

The second section imports basic modules required and use GPU resources if available. If error occurs in this section, it is probably because some modules are not installed. If GPU is not available, CPU will be used.

From the thrid section until to the end of the file, we built our codes by first implementing some methods, then use some codes to test and show the outputs of the methods to make sure these methods are well functioned. A global bool variant `SHOW_DEMO` is used to open or close these code blocks which is somehow like:

```
if USE_DEMO:
    # codes here
    pass
```

We recommend you set `SHOW_DEMO = True` for the first time running this ipynb, such that you have a good understanding of the role of each method we implemented. After you have a clear idea of all the methods, you may use `SHOW_DEMO = False` to skip all check codes, focus on the first cell in the section: **Training Models**, select `run` before in your notebook menus, and start one of the most exciting parts of the ipynb: training the best model we built for the competition.

Without further due, let's start by executing the first code cell(line):

```
SHOW_DEMO = True
```

We hope you enjoy these codes.

If you have any suggestions, contact: qqiangye@gmail.com (<mailto:qqiangye@gmail.com>)

License: BSD

In []:

1	
---	--