

>>> README <<<

Assignment 3 - CS519W16

Author: Xi Yu

Date: 02/27/2016

TASKs:

- 1) Remove the dropout layer after the fully-connected layer (5 points). Save the model after training.
- 2) Load the model you saved at step 1 as initialization to the training. Add another fully connected layer with 512 filters at the end (10 points). Train and save the model.
- 3) Load the model you saved at step 2 as initialization. Add dropout layers to both fully-connected layers (10 points), re-train the model. (Hint: in this case you may need to manually move the weights to the correct corresponding locations in the new model, but some has mentioned that you can "pop" a layer, so it might be easier than that).
- 4) Re-train the final model (after the model changes in tunings 1-3) from scratch. Try to use an adaptive schedule to tune the learning rate, you can choose from RMSprop, Adagrad and Adam (Hint: you don't need to implement any of these, look at Keras documentation please) (5 points).
- 5) Try to tune your network in two other ways (10 points) (e.g. add/remove a layer, change the activation function, add/remove regularizer, change the number of hidden units) not described in the previous four. You can start from random initializations or previous results as you wish.
- 6) For each of the settings (1) - (5), please submit a PDF report your training loss, training error, validation loss and validation error. Draw 2 figures for each of the settings (1) - (5) (2 figures for each different tuning in (5)) with the x-axis being the epoch number, and y-axis being the loss/error, use 2 different lines in the same figure to represent training loss/validation loss, and training error/validation error.

Source Code Description:

cifar10_cnn_1.py

Functionality: Accomplish Task 1.

Outputs: records, model_weights, model_architecture --> manually saved to ./recoder_1

cifar10_cnn_2.py

Functionality: Accomplish Task 2.

Outputs: records, model_weights, model_architecture --> manually saved to ./recoder_2

cifar10_cnn_3.py

Functionality: Accomplish Task 3.

Outputs: records, model_weights, model_architecture --> manually saved to ./recoder_3

cifar10_cnn_4.py

Functionality: Accomplish Task 4.

Outputs: records, model_weights, model_architecture --> manually saved to ./recoder_4

cifar10_cnn_5_1.py

Functionality: Accomplish Task 5.1: Based on the original cifar10_cnn.py, it reduces the hidden unit number to 64.

Outputs: records, model_weights, model_architecture --> manually saved to ./recoder_5_2

cifar10_cnn_5_2.py

Functionality: Accomplish Task 5.2: Based on the original cifar10_cnn.py, it change the activation function of FCNN to sigmoid.

Outputs: records, model_weights, model_architecture --> manually saved to ./recoder_5_2

plot.py

Functionality: Plot all records and save to .png files

Outputs: .png files of plots

Record file is named by record_*.txt.

Each line of a record file == <epoch>/t<training loss>/t<training accuracy>/t<validation loss>/t<validation accuracy>/n

For setting the CUDA path, run the following commends (dependent on CUDA installation) in a Terminal .

```
echo $CUDA_PATH
export CUDA_PATH=/usr/local/cuda-7.5/
export PATH=$PATH:/usr/local/cuda-7.5/bin
export LD_LIBRARY_PATH=/usr/local/cuda-7.5/lib64
```