Requirement

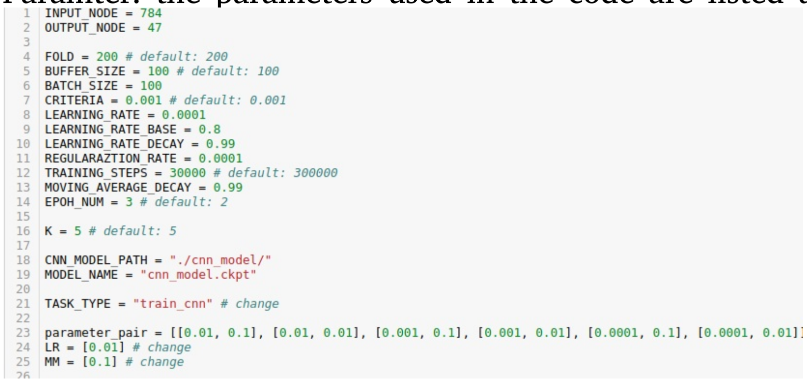
1. Anaconda
2. jupyter notebook

   http://jupyter.readthedocs.io/en/latest/install.html#install

How to use

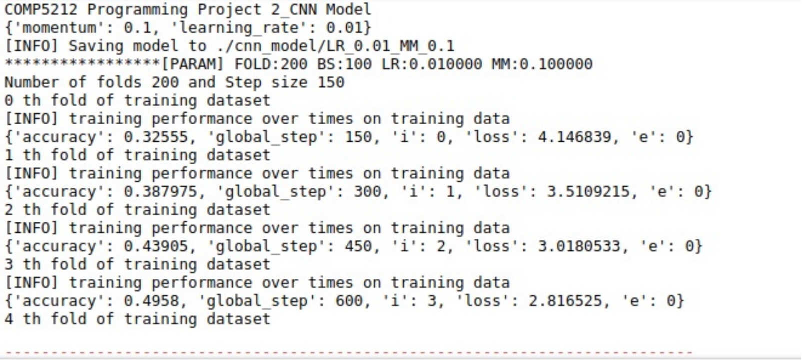
1. Code: **cnn_model_tflayer.ipynb** and **cæ_model_tflayer.ipynb** ## cnn_model_tflayer.ipynb

   1. Paramter: the parameters used in the code are listed at the top

      ```
      INPUT_NODE = 784
      OUTPUT_NODE = 47

      FOLD = 200 # default: 200
      BUFFER_SIZE = 100 # default: 100
      BATCH_SIZE = 100
      CRITERIA = 0.001 # default: 0.001
      LEARNING_RATE = 0.0001
      LEARNING_RATE_BASE = 0.8
      LEARNING_RATE_DECAY = 0.99
      REGULARAZTION_RATE = 0.0001
      TRAINING_STEPS = 30000 # default: 300000
      MOVING_AVERAGE_DECAY = 0.99
      EPOH_NUM = 3 # default: 2

      K = 5 # default: 5

      CNN_MODEL_PATH = "./cnn_model/"
      MODEL_NAME = "cnn_model.ckpt"

      TASK_TYPE = "train_cnn" # change

      parameter_pair = [[0.01, 0.1], [0.01, 0.01], [0.001, 0.1], [0.001, 0.01], [0.0001, 0.1], [0.0001, 0.01]]
      LR = [0.01] # change
      MM = [0.1] # change
      ```

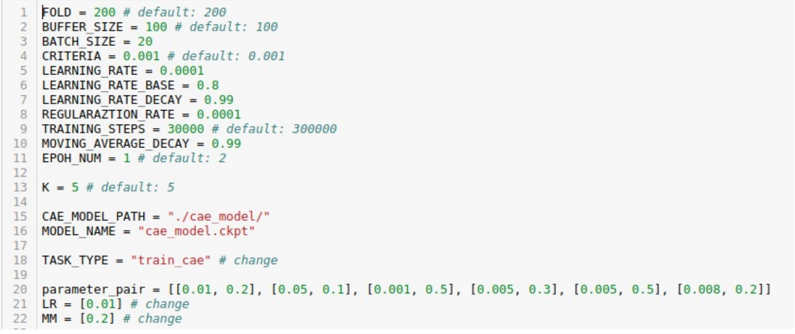
   2. Change training parameters: The LR(learning rate), MM(momentum factor) are two parameters need to be changed in paramter tuning. The variable **paramter_pair** lists the parameter selections, so you need to try each pari step by step.
   3. Change mode: There are four modes for the code(1 for training and 3 for testing): "train_cnn"(implement a cross validation technique to training the cnn model), "test cnn"(once finish training, using test set to test to model), "eval_cnn"(using evaluation set) and "test_cnn_with_train"(using training set). Please change the variable TASK_TYPE for different mode. please note that in "train_cnn", if the model was saved previously, the training process will continue the past result.
   4. Run: Ctrl+Enter to the module, you can see the result below.

      ```
      COMP5212 Programming Project 2_CNN Model
      {'momentum': 0.1, 'learning_rate': 0.01}
      [INFO] Saving model to ./cnn_model/LR_0.01_MM_0.1
      ****************[PARAM] FOLD:200 BS:100 LR:0.010000 MM:0.100000
      Number of folds 200 and Step size 150
      0 th fold of training dataset
      [INFO] training performance over times on training data
      {'accuracy': 0.32555, 'global_step': 150, 'i': 0, 'loss': 4.146839, 'e': 0}
      1 th fold of training dataset
      [INFO] training performance over times on training data
      {'accuracy': 0.387975, 'global_step': 300, 'i': 1, 'loss': 3.5109215, 'e': 0}
      2 th fold of training dataset
      [INFO] training performance over times on training data
      {'accuracy': 0.43905, 'global_step': 450, 'i': 2, 'loss': 3.0180533, 'e': 0}
      3 th fold of training dataset
      [INFO] training performance over times on training data
      {'accuracy': 0.4958, 'global_step': 600, 'i': 3, 'loss': 2.816525, 'e': 0}
      4 th fold of training dataset
      ```

cæ_model_tflayer.ipynb

1. Paramter: the parameters used in the code are listed at the top



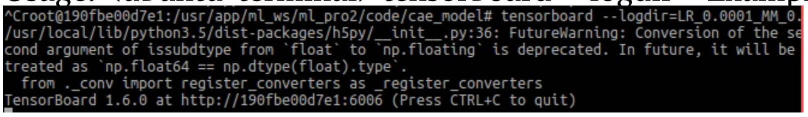
```
FOLD = 200 # default: 200
BUFFER_SIZE = 100 # default: 100
BATCH_SIZE = 20
CRITERIA = 0.001 # default: 0.001
LEARNING_RATE = 0.0001
LEARNING_RATE_BASE = 0.8
LEARNING_RATE_DECAY = 0.99
REGULARAZTION_RATE = 0.0001
TRAINING_STEPS = 30000 # default: 300000
MOVING_AVERAGE_DECAY = 0.99
EPOH_NUM = 1 # default: 2

K = 5 # default: 5

CAE_MODEL_PATH = "./cae_model/"
MODEL_NAME = "cae_model.ckpt"

TASK_TYPE = "train_cae" # change

parameter_pair = [[0.01, 0.2], [0.05, 0.1], [0.001, 0.5], [0.005, 0.3], [0.005, 0.5], [0.008, 0.2]]
LR = [0.01] # change
MM = [0.2] # change
```

2. Change training parameters: The LR(learning rate), MM(momentum factor) are two parameters need to be changed in paramter tuning. The variable **paramter_pair** lists the parameter selections, so you need to try each pari step by step.
3. Change mode: There are four modes for the code(1 for training and 3 for testing): "train_cae"(implement a cross validation technique to training the cnn model), "test cae"(once finish training, using test set to test to model), "eval_cae"(using evaluation set) and "test_cnn_with_train"(using training set). Please change the variable **TASK_TYPE** for different mode. please note that in "train_cae", if the model was saved previously, the training process will continue the past result.
4. Run: Ctrl+Enter to the module.

Note

1. If you want to visualize the result, I recommend you to use the **tensorboard** visulization tool.

   Usage: (ubuntu terminal) tensorboard --logdir= Example:



```
^Croot@190fbe00d7e1:/usr/app/ml_ws/ml_pro2/code/cae_model# tensorboard --logdir=LR_0.0001_MM_0.
/usr/local/lib/python3.5/dist-packages/h5py/__init__.py:36: FutureWarning: Conversion of the se
cond argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be
treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
TensorBoard 1.6.0 at http://190fbe00d7e1:6006 (Press CTRL+C to quit)
```

Contact

1. If you do not about jupyter, or meet some problems about running it, please feel free to contact me: jjiao@ust.hk