Tensor flow

**Import library**

import numpy as np

import tensorflow as tf

import time

from datetime import datetime

import os.path

import data\_helpers

import func\_two\_layer\_fc

**Initialization**

import data\_helpers

data\_sets = data\_helpers.load\_data()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flags = tf.flags

FLAGS = flags.FLAGS

flags.DEFINE\_float('learning\_rate', 0.001, 'Learning rate for the training.')

flags.DEFINE\_integer('max\_steps', 2000, 'Number of steps to run trainer.')

flags.DEFINE\_integer('hidden1', 120, 'Number of units in hidden layer 1.')

flags.DEFINE\_integer('batch\_size', 400, 'Batch size. Must divide dataset sizes without remainder.')

flags.DEFINE\_string('train\_dir', 'tf\_logs', 'Directory to put the training data.')

flags.DEFINE\_float('reg\_constant', 0.1, 'Regularization constant.')

FLAGS.\_parse\_flags()

print('\nParameters:')

for attr, value in sorted(FLAGS.\_\_flags.items()):

print('{} = {}'.format(attr, value))

print()

* Define Input placeholders
* Operations

# placeholders for inputs

images\_placeholder = tf.placeholder(tf.float32, shape=[None, IMAGE\_PIXELS], name='images')

labels\_placeholder = tf.placeholder(tf.int64, shape=[None], name='image-labels')

logits = two\_layer\_fc.inference(images\_placeholder, IMAGE\_PIXELS,

FLAGS.hidden1, CLASSES, reg\_constant=FLAGS.reg\_constant)

loss = func\_two\_layer\_fc.loss(logits, labels\_placeholder)

train\_step = func\_two\_layer\_fc.training(loss, FLAGS.learning\_rate)

accuracy = func\_two\_layer\_fc.evaluation(logits, labels\_placeholder)

summary = tf.summary.merge\_all()

saver = tf.train.Saver()c