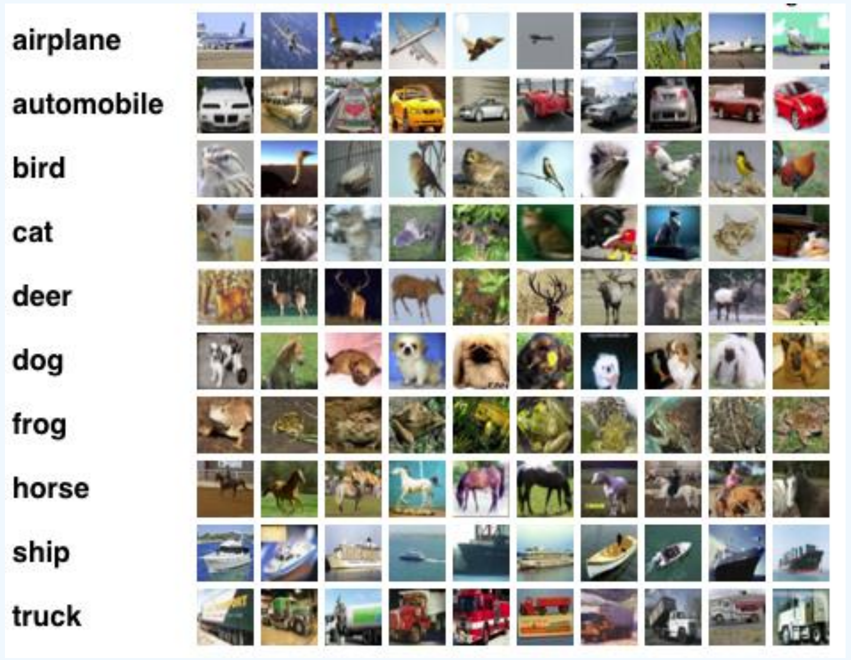
Object classification exercise

CIFAR-10 Image Classification in TensorFlow

KYUNG HEE UNIVERSITY, IMAGE COMMUNICATION LAB.

What is CIFAR 10

- The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.



- The dataset is divided into five training batches and one test batch, each with 10000 images.
- The test batch contains exactly 1000 randomly-selected images from each class.
- The training batches contain the remaining images in random order.

One-hot encode

- CIFAR-10 provides 10 different classes of the image, so you need a vector in size of 10 as well.

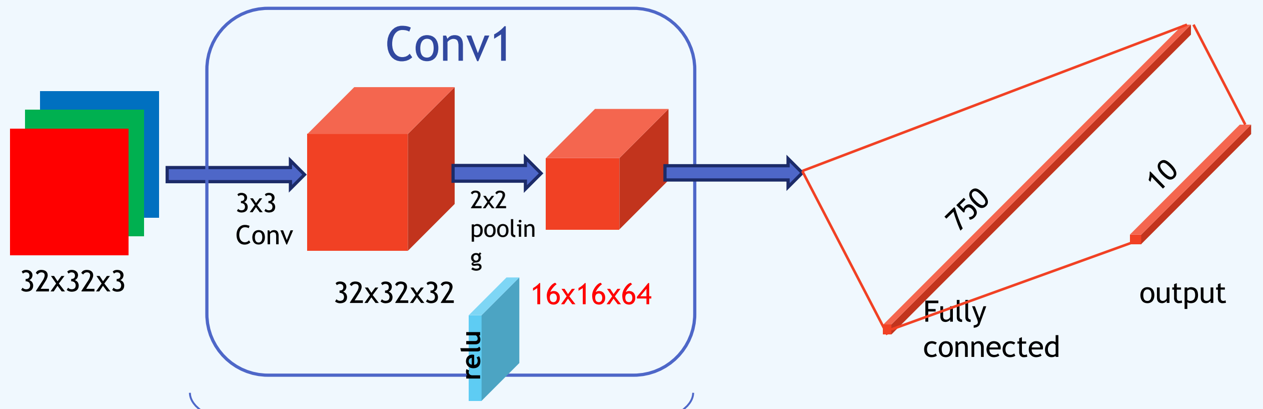
index	label
0	**airplane** (0)
1	**automobile** (1)
2	**bird** (2)
3	**cat** (3)
4	**deer** (4)
5	**dog** (5)
6	**frog** (6)
7	**horse** (7)
8	**ship** (8)
9	**truck** (9)
…	…
…	…

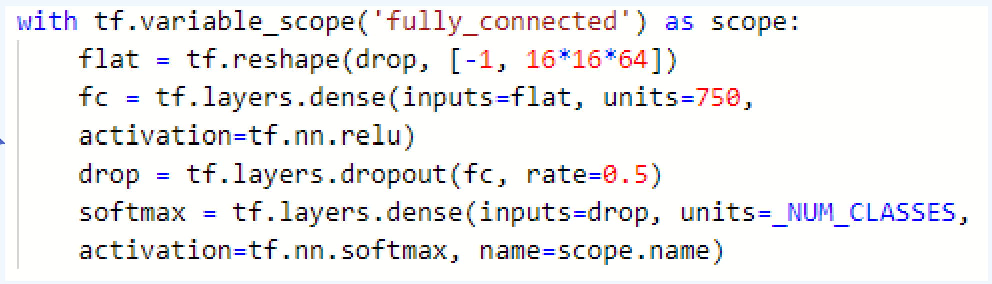
original label data

label	index											
	0	1	2	3	4	5	6	7	8	9	…	…
airplane	**1**	0	0	0	0	0	0	0	0	0	…	…
automobile	0	**1**	0	0	0	0	0	0	0	0	…	…
bird	0	0	**1**	0	0	0	0	0	0	0	…	…
cat	0	0	0	**1**	0	0	0	0	0	0	…	…
deer	0	0	0	0	**1**	0	0	0	0	0	…	…
dog	0	0	0	0	0	**1**	0	0	0	0	…	…
frog	0	0	0	0	0	0	**1**	0	0	0	…	…
horse	0	0	0	0	0	0	0	**1**	0	0	…	…
ship	0	0	0	0	0	0	0	0	**1**	0	…	…
truck	0	0	0	0	0	0	0	0	0	**1**	…	…

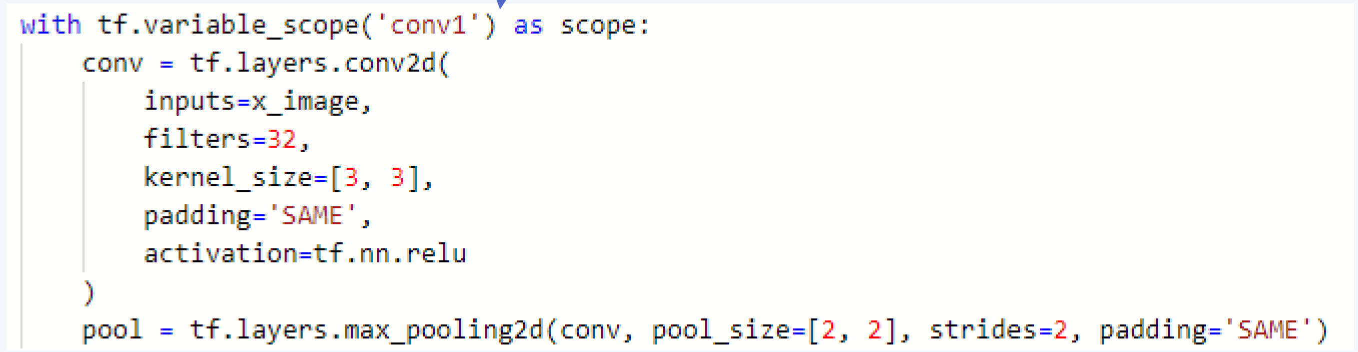
one-hot-encoded label data

Model example





```
with tf.variable_scope('fully_connected') as scope:
    flat = tf.reshape(drop, [-1, 16*16*64])
    fc = tf.layers.dense(inputs=flat, units=750,
    activation=tf.nn.relu)
    drop = tf.layers.dropout(fc, rate=0.5)
    softmax = tf.layers.dense(inputs=drop, units=_NUM_CLASSES,
    activation=tf.nn.softmax, name=scope.name)
```

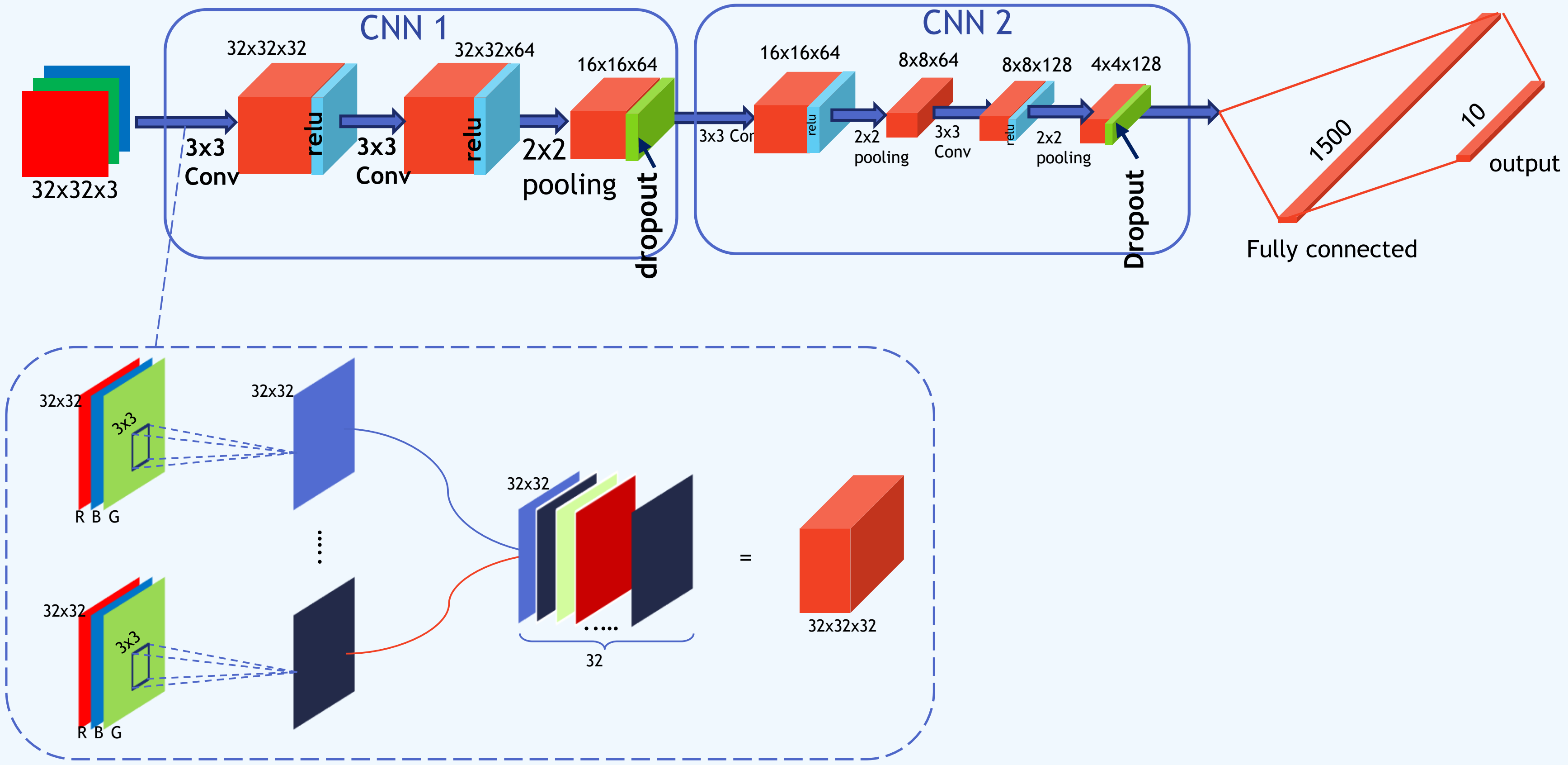


```
with tf.variable_scope('conv1') as scope:
    conv = tf.layers.conv2d(
        inputs=x_image,
        filters=32,
        kernel_size=[3, 3],
        padding='SAME',
        activation=tf.nn.relu
    )
    pool = tf.layers.max_pooling2d(conv, pool_size=[2, 2], strides=2, padding='SAME')
```

- 16*16*64 is calculated from the size of 16x16x64 after pooling operation
- _NUM_CLASSES is number of class in cifar-10 dataset (10 classes)

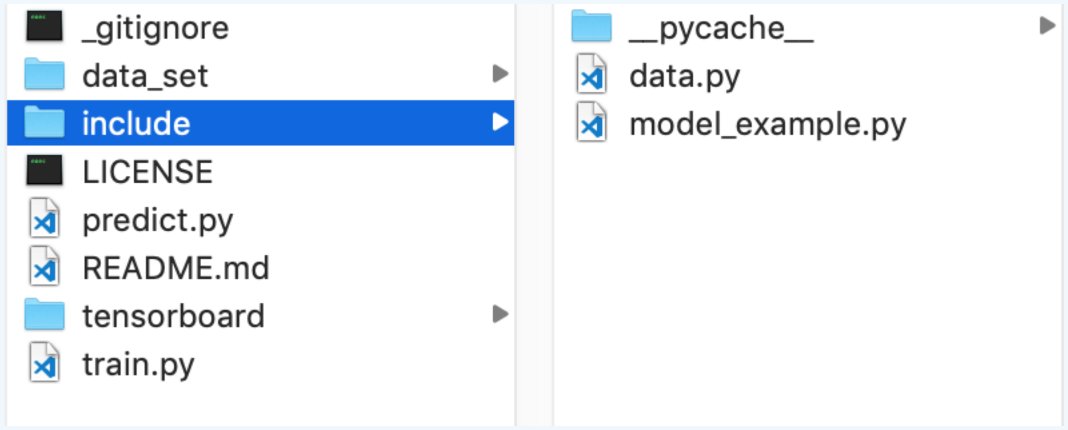
Exercise requirement:

- Goal:
  - Build the Convolutional neural network model to classify 10 class on cifar-10 dataset using TensorFlow.
  - Evaluate the performance of network model
  - Modify the example model provided in model.py using the following model.
  - Run code, and evaluation.



Overview of Program

- Structure of example project



- *train.py*: main function that need to be run for training model
- *predict.py* is needed to test trained model

- *data_set* folder contain CIFAR-10 data
- *Include* folder: data.py has functions of data processing, model_example.py is the design of CNN model
- *tensorboard* folder: to save trained model after training phase. NOTE: you have to delete this folder if you –re-trained model with new designed, otherwise error occurs.

- *YOUR TASK is to build the model by modify the model_example.py file in include folder*

- *Turn in the code folder that includes the compete model_example.py.*

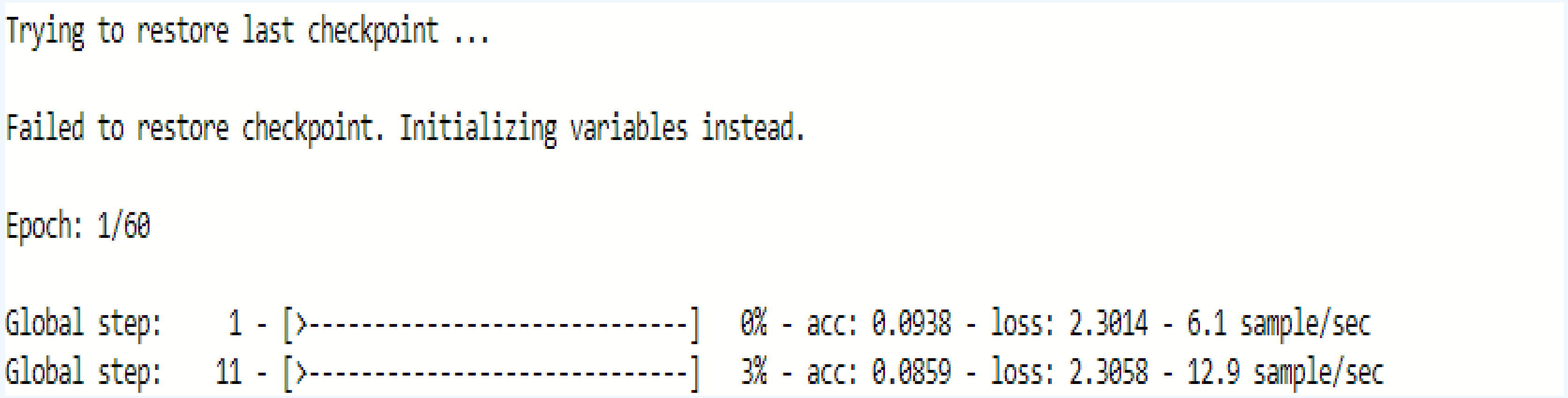
Unit of Data : Epochs, Batch Size , Iterations

Epoch	Batch Size	Iterations
One Epoch is when an ENTIRE dataset is passed forward and backward through the neural network only ONCE.	• Total number of training examples in a single batch. • We cannot pass entire dataset into neural network at once. So we have to divide the dataset into batches, each batch have Batch Size. • Batch Size is chosen based on the memory size of training computer. Big Batch Size can help training faster, but requires more memory and computation power.	• Iterations is the number of batches needed to complete one epoch. • If dataset have 100,000 images, and batch size is 100, then, the dataset is divided to 1000 batches, and number of iteration is 1000 to pass entire data. (Thatis, one iteration is for one batch.) When 1000 iters is passed, it completes 1 epoch. Number of epochs depends on the repeation of training decided by user.)

What you have to submit

1. Submit model_example.py file and tensorboard folder only.
2. We include the model_example.py and copy tensorboard folder and evaluate the submitted program by running predict.py.
3. Whenever the program runs, the log file appear on screem/. You have to copy all of log screen and save it to log.txt. Submit the log.txt.

Log screen looks like this



```
Trying to restore last checkpoint ...

Failed to restore checkpoint. Initializing variables instead.

Epoch: 1/60

Global step:     1 - [>-----------------------------]   0% - acc: 0.0938 - loss: 2.3014 - 6.1 sample/sec
Global step:    11 - [>-----------------------------]   3% - acc: 0.0859 - loss: 2.3058 - 12.9 sample/sec
```