

ML-MOOC-NPTEL (/github/Santara/ML-MOOC-NPTEL/tree/master)
/ lecture5 (/github/Santara/ML-MOOC-NPTEL/tree/master/lecture5)

1. Load dependencies

```
In [1]: import numpy as np
import tflearn
from tflearn.datasets import mnist
```

2. Load data

```
In [2]: MNIST_data = mnist.read_data_sets(one_hot=True)

Extracting mnist/train-images-idx3-ubyte.gz
/Users/Anirban/anaconda/lib/python2.7/gzip.py:275: VisibleDeprecationWarning:
  chunk = self.extrabuf[offset: offset + size]
/Users/Anirban/anaconda/lib/python2.7/site-packages/tflearn/datasets/mnist.py
  data = data.reshape(num_images, rows, cols, 1)

Extracting mnist/train-labels-idx1-ubyte.gz
Extracting mnist/t10k-images-idx3-ubyte.gz
Extracting mnist/t10k-labels-idx1-ubyte.gz
```

```
In [3]: data_train = MNIST_data.train
data_validation = MNIST_data.validation
data_test = MNIST_data.test
```

```
In [4]: X, y = data_train._images, data_train._labels
```

3. Make the model

In [5]:

```
tflearn.init_graph(num_cores=4)

net = tflearn.input_data(shape=[None, 784])
net = tflearn.fully_connected(net, 100, activation='relu')
net = tflearn.fully_connected(net, 100, activation='relu')
net = tflearn.fully_connected(net, 10, activation='softmax')

net = tflearn.regression(net, loss='categorical_crossentropy', optimizer='adagrad',
                          metrics=['accuracy'])

model = tflearn.DNN(net)
```

4. Train the model

In [6]:

```
model.fit(X, y, n_epoch=1, batch_size=10, show_metric=True)
```

```
Training Step: 5500 | total loss: 0.13695
| Adam | epoch: 001 | loss: 0.13695 - acc: 0.9644 -- iter: 55000/55000
Training Step: 5500 | total loss: 0.13695
| Adam | epoch: 001 | loss: 0.13695 - acc: 0.9644 -- iter: 55000/55000
--
```