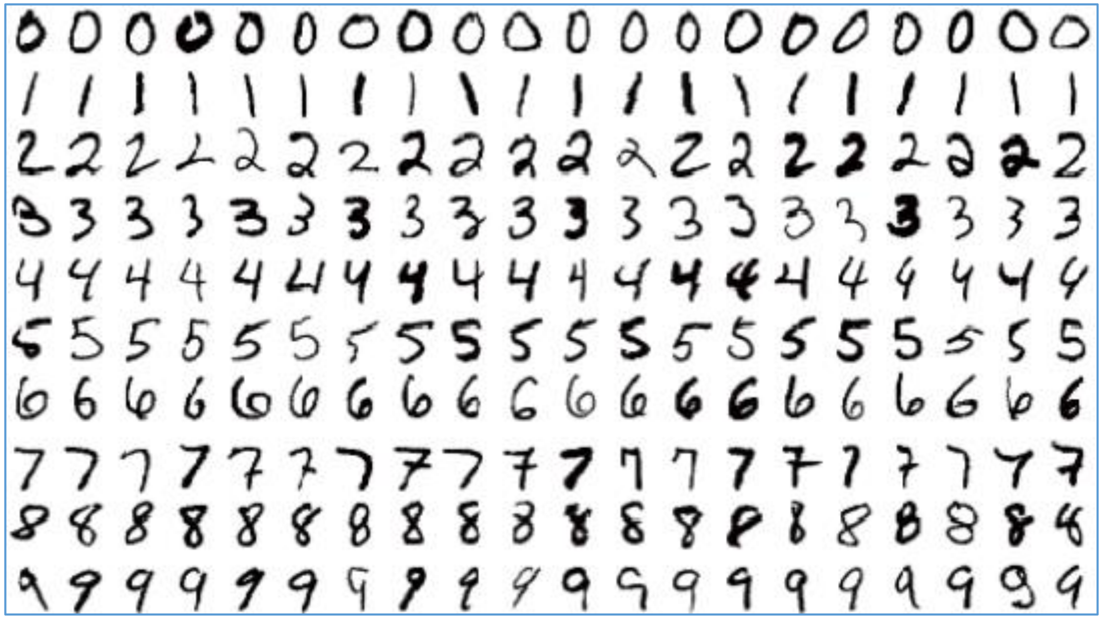
딥러닝 구현을 위한 텐서플로우 개발

김성균

5강
머신러닝 실용과 MNIST 데이터셋

01.MNIST Dataset

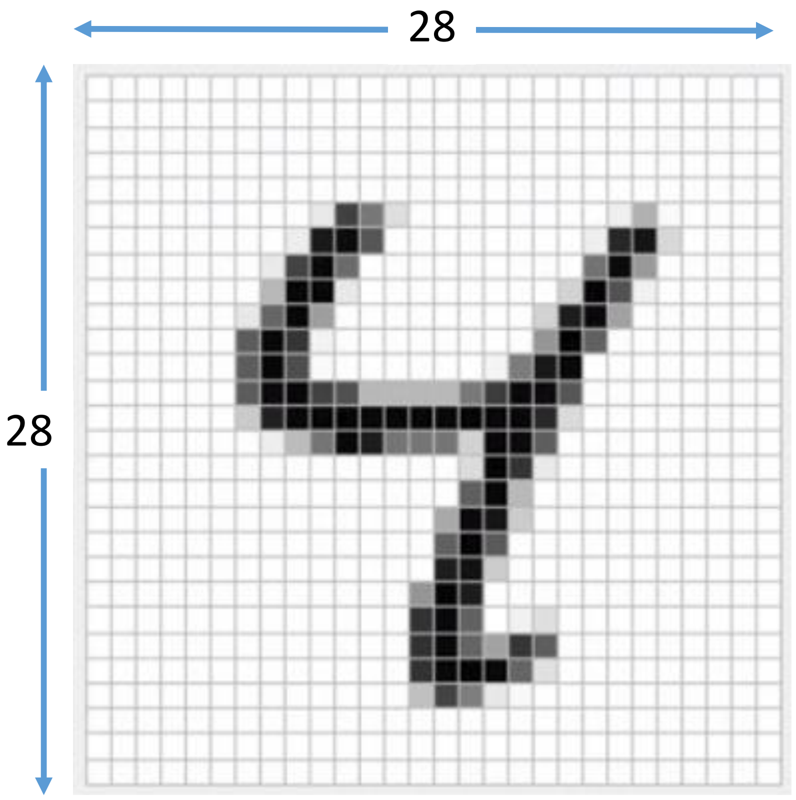
- 미국에서 우편번호를 인식해 자동분류하기 위해 만든 데이터셋



train-images-idx3-ubyte.gz: training set images (9912422 bytes)
train-labels-idx1-ubyte.gz: training set labels (28881 bytes)
t10k-images-idx3-ubyte.gz: test set images (1648877 bytes)
t10k-labels-idx1-ubyte.gz: test set labels (4542 bytes)

http://yann.lecun.com/exdb/mnist/

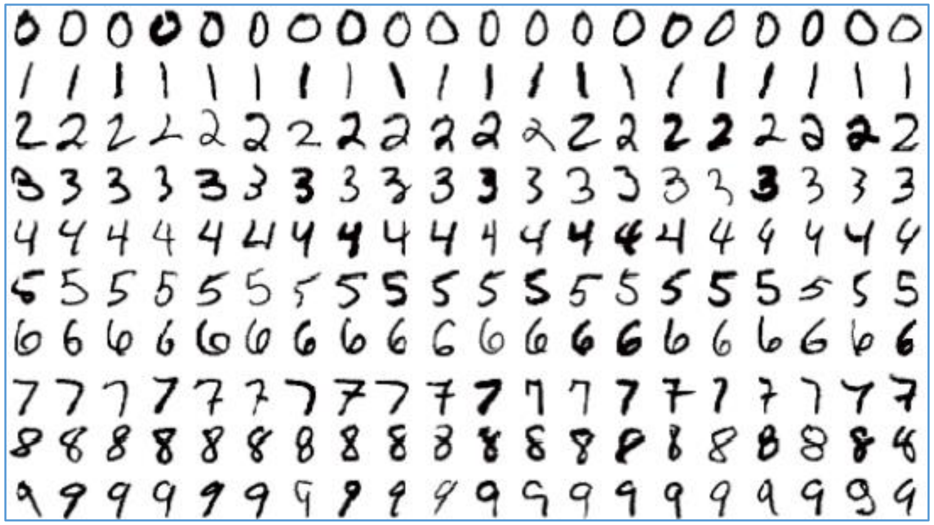
02.Image Size



```
image_size = 784 # 28 * 28
nb_classes = 10

X = tf.placeholder(tf.float32, shape=[None, image_size])
Y = tf.placeholder(tf.float32, shape=[None, nb_classes])
```

03.Import and batch



```
from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
…
for i in range(total_batch) :
    batch_xs, batch_ys = mnist.train.next_batch(batch_size)
…

# Test the model using test sets
print("Accuracy: ", accuracy.eval(session=sess, feed_dict={ X: mnist.test.images, Y:
mnist.test.labels}))
```

04. Reading data and set variables

```
from tensorflow.examples.tutorials.mnist import input_data

mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)

LEARNING_RATE = 0.01
image_size = 784 # 28 * 28
nb_classes = 10

X = tf.placeholder(tf.float32, shape=[None, image_size])
Y = tf.placeholder(tf.float32, shape=[None, nb_classes])

W = tf.Variable(tf.random_normal([image_size, nb_classes], name ="weight"))
b = tf.Variable(tf.random_normal([nb_classes], name = "bias"))
```

05. Softmax

```
hypothesis = tf.nn.softmax(tf.matmul(X, W) + b)

# Cross entropy cost/loss
cost = tf.reduce_mean(-tf.reduce_sum(Y * tf.log(hypothesis), axis=1))
optimizer = tf.train.GradientDescentOptimizer(learning_rate = LEARNING_RATE).minimize(cost)

# Test model
is_correct = tf.equal(tf.arg_max(hypothesis, 1), tf.arg_max(Y, 1))
# Calucate accuracy
accuracy = tf.reduce_mean(tf.cast(is_correct, tf.float32))
```

06.Training epoch/batch

```
# parameters
training_epochs = 15
batch_size = 100

sess = tf.Session()
sess.run(tf.global_variables_initializer())

for epoch in range(training_epochs):
    avg_cost = 0
    total_batch = int(mnist.train.num_examples / batch_size)

    for i in range(total_batch) :
        batch_xs, batch_ys = mnist.train.next_batch(batch_size)
        c, _ = sess.run([cost, optimizer], feed_dict={X : batch_xs, Y : batch_ys})
        avg_cost += c / total_batch
    print ( 'Epoch: ', '%04d' % (epoch* 1), 'cost =' , '{:.9f}'.format(avg_cost))
```

06.Training epoch/batch

- 하나의 Epoch = 모든 학습데이터를 한번 학습하는 것
- 배치 크기 = 한번에 처리하는 데이터의 수(사진의 장수).
- 예 : 1000 개의 학습 예제가 있고 배치 크기가 500 인 경우 1 Epoch를 완료하는 데 2 회의 반복이 필요

07. Report results on test dataset

```
# Test the model using test sets
print("Accuracy: ", accuracy.eval(session=sess, feed_dict={ X: mnist.test.images, Y: mnist.test.labels}))
```

07.Sample image show and prediction

```
# Get one and predict
r = random.randint(0, mnist.test.num_examples - 1)
print("Label      : ", sess.run(tf.argmax(mnist.test.labels[r:r+1], 1)))
print("Prediction : ", sess.run(tf.argmax(hypothesis,1), feed_dict={X : mnist.test.images[r:r +1]}))

plt.imshow(mnist.test.images[r:r+1].reshape(28,28), cmap='Greys', interpolation='nearest')
plt.show()
```

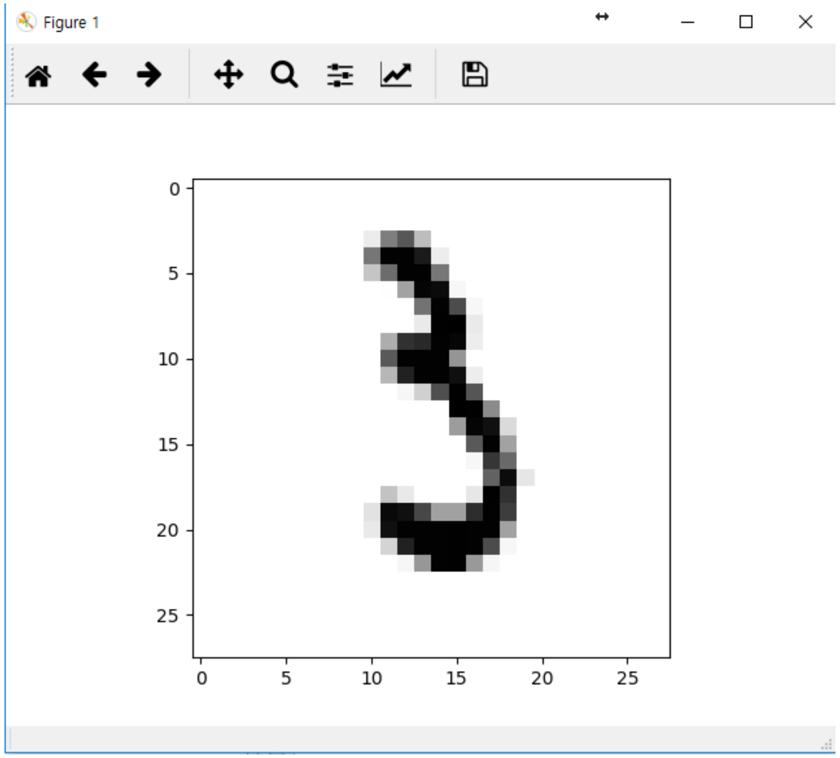


Figure 1
0
5
10
15
20
25
0
5
10
15
20
25