Epoch-Example

October 10, 2018

In [5]:
```
from tensorflow.examples.tutorials.mnist import input_data
import tensorflow as tf
import numpy as np
from batchhelper import batches  # Helper function created in Mini-batching section


def print_epoch_stats(epoch_i, sess, last_features, last_labels):
    """
    Print cost and validation accuracy of an epoch
    """
    current_cost = sess.run(
        cost,
        feed_dict={features: last_features, labels: last_labels})
    valid_accuracy = sess.run(
        accuracy,
        feed_dict={features: valid_features, labels: valid_labels})
    print('Epoch: {:<4} - Cost: {:<8.3} Valid Accuracy: {:<5.3}'.format(
        epoch_i,
        current_cost,
        valid_accuracy))

n_input = 784  # MNIST data input (img shape: 28*28)
n_classes = 10  # MNIST total classes (0-9 digits)

# Import MNIST data
mnist = input_data.read_data_sets('MNIST data', one_hot=True)

# The features are already scaled and the data is shuffled
train_features = mnist.train.images
valid_features = mnist.validation.images
test_features = mnist.test.images

train_labels = mnist.train.labels.astype(np.float32)
valid_labels = mnist.validation.labels.astype(np.float32)
test_labels = mnist.test.labels.astype(np.float32)

# Features and Labels
```

```
features = tf.placeholder(tf.float32, [None, n_input])
labels = tf.placeholder(tf.float32, [None, n_classes])

# Weights & bias
weights = tf.Variable(tf.random_normal([n_input, n_classes]))
bias = tf.Variable(tf.random_normal([n_classes]))

# Logits - xW + b
logits = tf.add(tf.matmul(features, weights), bias)

# Define loss and optimizer
learning_rate = tf.placeholder(tf.float32)
cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits=logits, labels=lab
optimizer = tf.train.GradientDescentOptimizer(learning_rate=learning_rate).minimize(cos

# Calculate accuracy
correct_prediction = tf.equal(tf.argmax(logits, 1), tf.argmax(labels, 1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

init = tf.global_variables_initializer()

batch_size = 128
epochs = 80
learn_rate = 0.1

train_batches = batches(batch_size, train_features, train_labels)

with tf.Session() as sess:
    sess.run(init)

    # Training cycle
    for epoch_i in range(epochs):

        # Loop over all batches
        for batch_features, batch_labels in train_batches:
            train_feed_dict = {
                features: batch_features,
                labels: batch_labels,
                learning_rate: learn_rate}
            sess.run(optimizer, feed_dict=train_feed_dict)

        # Print cost and validation accuracy of an epoch
        print_epoch_stats(epoch_i, sess, batch_features, batch_labels)

    # Calculate accuracy for test dataset
    test_accuracy = sess.run(
        accuracy,
        feed_dict={features: test_features, labels: test_labels})
```

```
        print('Test Accuracy: {}'.format(test_accuracy))
```

```
Extracting MNIST data/train-images-idx3-ubyte.gz
Extracting MNIST data/train-labels-idx1-ubyte.gz
Extracting MNIST data/t10k-images-idx3-ubyte.gz
Extracting MNIST data/t10k-labels-idx1-ubyte.gz
Epoch: 0    - Cost: 2.06     Valid Accuracy: 0.71
Epoch: 1    - Cost: 1.49     Valid Accuracy: 0.782
Epoch: 2    - Cost: 1.28     Valid Accuracy: 0.815
Epoch: 3    - Cost: 1.14     Valid Accuracy: 0.836
Epoch: 4    - Cost: 1.03     Valid Accuracy: 0.849
Epoch: 5    - Cost: 0.956    Valid Accuracy: 0.857
Epoch: 6    - Cost: 0.895    Valid Accuracy: 0.863
Epoch: 7    - Cost: 0.845    Valid Accuracy: 0.868
Epoch: 8    - Cost: 0.804    Valid Accuracy: 0.871
Epoch: 9    - Cost: 0.769    Valid Accuracy: 0.873
Epoch: 10   - Cost: 0.739    Valid Accuracy: 0.876
Epoch: 11   - Cost: 0.713    Valid Accuracy: 0.879
Epoch: 12   - Cost: 0.689    Valid Accuracy: 0.88
Epoch: 13   - Cost: 0.668    Valid Accuracy: 0.881
Epoch: 14   - Cost: 0.649    Valid Accuracy: 0.883
Epoch: 15   - Cost: 0.631    Valid Accuracy: 0.885
Epoch: 16   - Cost: 0.614    Valid Accuracy: 0.885
Epoch: 17   - Cost: 0.599    Valid Accuracy: 0.886
Epoch: 18   - Cost: 0.585    Valid Accuracy: 0.889
Epoch: 19   - Cost: 0.571    Valid Accuracy: 0.89
Epoch: 20   - Cost: 0.559    Valid Accuracy: 0.891
Epoch: 21   - Cost: 0.547    Valid Accuracy: 0.893
Epoch: 22   - Cost: 0.536    Valid Accuracy: 0.893
Epoch: 23   - Cost: 0.525    Valid Accuracy: 0.892
Epoch: 24   - Cost: 0.515    Valid Accuracy: 0.892
Epoch: 25   - Cost: 0.505    Valid Accuracy: 0.892
Epoch: 26   - Cost: 0.496    Valid Accuracy: 0.893
Epoch: 27   - Cost: 0.488    Valid Accuracy: 0.893
Epoch: 28   - Cost: 0.48     Valid Accuracy: 0.893
Epoch: 29   - Cost: 0.472    Valid Accuracy: 0.894
Epoch: 30   - Cost: 0.464    Valid Accuracy: 0.895
Epoch: 31   - Cost: 0.457    Valid Accuracy: 0.895
Epoch: 32   - Cost: 0.45     Valid Accuracy: 0.895
Epoch: 33   - Cost: 0.444    Valid Accuracy: 0.896
Epoch: 34   - Cost: 0.438    Valid Accuracy: 0.896
Epoch: 35   - Cost: 0.432    Valid Accuracy: 0.896
Epoch: 36   - Cost: 0.426    Valid Accuracy: 0.898
Epoch: 37   - Cost: 0.42     Valid Accuracy: 0.898
Epoch: 38   - Cost: 0.415    Valid Accuracy: 0.899
Epoch: 39   - Cost: 0.41     Valid Accuracy: 0.899
Epoch: 40   - Cost: 0.405    Valid Accuracy: 0.9
```

```
Epoch: 41   - Cost: 0.4      Valid Accuracy: 0.901
Epoch: 42   - Cost: 0.396    Valid Accuracy: 0.901
Epoch: 43   - Cost: 0.392    Valid Accuracy: 0.901
Epoch: 44   - Cost: 0.387    Valid Accuracy: 0.901
Epoch: 45   - Cost: 0.383    Valid Accuracy: 0.902
Epoch: 46   - Cost: 0.379    Valid Accuracy: 0.902
Epoch: 47   - Cost: 0.376    Valid Accuracy: 0.903
Epoch: 48   - Cost: 0.372    Valid Accuracy: 0.903
Epoch: 49   - Cost: 0.368    Valid Accuracy: 0.904
Epoch: 50   - Cost: 0.365    Valid Accuracy: 0.904
Epoch: 51   - Cost: 0.361    Valid Accuracy: 0.904
Epoch: 52   - Cost: 0.358    Valid Accuracy: 0.905
Epoch: 53   - Cost: 0.355    Valid Accuracy: 0.905
Epoch: 54   - Cost: 0.352    Valid Accuracy: 0.906
Epoch: 55   - Cost: 0.349    Valid Accuracy: 0.907
Epoch: 56   - Cost: 0.346    Valid Accuracy: 0.907
Epoch: 57   - Cost: 0.343    Valid Accuracy: 0.907
Epoch: 58   - Cost: 0.341    Valid Accuracy: 0.908
Epoch: 59   - Cost: 0.338    Valid Accuracy: 0.908
Epoch: 60   - Cost: 0.336    Valid Accuracy: 0.908
Epoch: 61   - Cost: 0.333    Valid Accuracy: 0.908
Epoch: 62   - Cost: 0.331    Valid Accuracy: 0.908
Epoch: 63   - Cost: 0.328    Valid Accuracy: 0.909
Epoch: 64   - Cost: 0.326    Valid Accuracy: 0.909
Epoch: 65   - Cost: 0.324    Valid Accuracy: 0.909
Epoch: 66   - Cost: 0.322    Valid Accuracy: 0.909
Epoch: 67   - Cost: 0.32     Valid Accuracy: 0.91
Epoch: 68   - Cost: 0.317    Valid Accuracy: 0.91
Epoch: 69   - Cost: 0.315    Valid Accuracy: 0.91
Epoch: 70   - Cost: 0.314    Valid Accuracy: 0.91
Epoch: 71   - Cost: 0.312    Valid Accuracy: 0.911
Epoch: 72   - Cost: 0.31     Valid Accuracy: 0.911
Epoch: 73   - Cost: 0.308    Valid Accuracy: 0.911
Epoch: 74   - Cost: 0.306    Valid Accuracy: 0.911
Epoch: 75   - Cost: 0.304    Valid Accuracy: 0.911
Epoch: 76   - Cost: 0.303    Valid Accuracy: 0.911
Epoch: 77   - Cost: 0.301    Valid Accuracy: 0.911
Epoch: 78   - Cost: 0.299    Valid Accuracy: 0.911
Epoch: 79   - Cost: 0.298    Valid Accuracy: 0.912
Test Accuracy: 0.909200012684
```

In [ ]: