In [1]:
```
import tensorflow as tf
from tensorflow.python.ops import rnn, rnn_cell
import numpy as np

from tensorflow.examples.tutorials.mnist import input_data

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

learningRate =.0001
trainingIters = 50
batchSize = 100
displayStep = 10

nInput = 28 #we want the input to take the 28 pixels
nSteps = 28 #every 28
nHidden = 128 #number of neurons for the RNN
nClasses = 10 #this is MNIST so you know

x = tf.placeholder('float', [None, nSteps, nInput])
y = tf.placeholder('float', [None, nClasses])

weights = {
    'out': tf.Variable(tf.random_normal([nHidden, nClasses]))
}

biases = {
    'out': tf.Variable(tf.random_normal([nClasses]))
}

def RNN(x, weights, biases):
    x = tf.transpose(x, [1,0,2])
    x = tf.reshape(x, [-1, nInput])
    x = tf.split(x, nSteps, 0) #configuring so you can get it as needed fo
r the 28 pixels

    rnnCell = rnn_cell.BasicRNNCell(nHidden)#find which RNN to use in the
documentation

    outputs, states = rnn.static_rnn(rnnCell, x, dtype=tf.float32)#for the
 rnn where to get the output and hidden state

    return tf.matmul(outputs[-1], weights['out'])+ biases['out']

pred = RNN(x, weights, biases)

#optimization
#create the cost, optimization, evaluation, and accuracy
#for the cost softmax_cross_entropy_with_logits seems really good
cost = tf.reduce_mean( tf.nn.softmax_cross_entropy_with_logits(logits=pred
, labels=y))
optimizer = tf.train.AdamOptimizer(learningRate).minimize(cost)

correctPred =tf.equal(tf.argmax(pred, 1), tf.argmax(y, 1))
accuracy = tf.reduce_mean(tf.cast(correctPred, 'float'))
```

```
init = tf.global_variables_initializer()

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

    # while step* batchSize < trainingIters:
    for step in range(trainingIters):

        for _ in range(int(mnist.train.num_examples / batchSize)):
            batchX, batchY = mnist.train.next_batch(batchSize)#mnist has a
 way to get the next batch
            batchX = batchX.reshape((batchSize, nSteps, nInput))

            sess.run(optimizer, feed_dict={x: batchX, y:batchY})

        if step % displayStep == 0:
            acc = sess.run(accuracy, feed_dict={x: batchX, y:batchY})
            loss = sess.run(cost, feed_dict={x: batchX, y:batchY})
            print("Iter " + str(step) + ", Minibatch Loss= " +
                  "{:.6f}".format(loss) + ", Training Accuracy= " +
                  "{:.5f}".format(acc))

    print('Optimization finished')

    testData = mnist.test.images.reshape((-1, nSteps, nInput))
    testLabel = mnist.test.labels
    print("Testing Accuracy:", sess.run(accuracy, feed_dict={x: testData,
y:testLabel}))
```

```
WARNING: Logging before flag parsing goes to stderr.
W1106 03:48:10.925215 10276 deprecation.py:323] From <ipython-input-1-1b96
f7fad4ca>:7: read_data_sets (from tensorflow.contrib.learn.python.learn.da
tasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/
models.
W1106 03:48:10.929205 10276 deprecation.py:323] From C:\Users\oyeoy\Anacon
da3\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist
.py:260: maybe_download (from tensorflow.contrib.learn.python.learn.datase
ts.base) is deprecated and will be removed in a future version.
Instructions for updating:
Please write your own downloading logic.
W1106 03:48:10.931200 10276 deprecation.py:323] From C:\Users\oyeoy\Anacon
da3\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist
.py:262: extract_images (from tensorflow.contrib.learn.python.learn.datase
ts.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
```

```
Extracting /tmp/data/train-images-idx3-ubyte.gz
```

```
W1106 03:48:11.380996 10276 deprecation.py:323] From C:\Users\oyeoy\Anacon
da3\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist
.py:267: extract_labels (from tensorflow.contrib.learn.python.learn.datase
ts.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
```

```
Please use tf.data to implement this functionality.
W1106 03:48:11.384986 10276 deprecation.py:323] From C:\Users\oyeoy\Anacon
da3\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist
.py:110: dense_to_one_hot (from tensorflow.contrib.learn.python.learn.data
sets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.one_hot on tensors.
W1106 03:48:11.475742 10276 deprecation.py:323] From C:\Users\oyeoy\Anacon
da3\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist
.py:290: DataSet.__init__ (from tensorflow.contrib.learn.python.learn.data
sets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/
models.
```

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

```
W1106 03:48:12.106590 10276 deprecation.py:323] From <ipython-input-1-1b96
f7fad4ca>:38: BasicRNNCell.__init__ (from tensorflow.python.ops.rnn_cell_i
mpl) is deprecated and will be removed in a future version.
Instructions for updating:
This class is equivalent as tf.keras.layers.SimpleRNNCell, and will be rep
laced by that in Tensorflow 2.0.
W1106 03:48:12.109583 10276 deprecation.py:323] From <ipython-input-1-1b96
f7fad4ca>:40: static_rnn (from tensorflow.python.ops.rnn) is deprecated an
d will be removed in a future version.
Instructions for updating:
Please use `keras.layers.RNN(cell, unroll=True)`, which is equivalent to t
his API
W1106 03:48:12.199413 10276 deprecation.py:506] From C:\Users\oyeoy\Anacon
da3\lib\site-packages\tensorflow\python\ops\init_ops.py:1251: calling Vari
anceScaling.__init__ (from tensorflow.python.ops.init_ops) with dtype is d
eprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to
 the constructor
W1106 03:48:12.304186 10276 deprecation.py:506] From C:\Users\oyeoy\Anacon
da3\lib\site-packages\tensorflow\python\ops\rnn_cell_impl.py:459: calling
Zeros.__init__ (from tensorflow.python.ops.init_ops) with dtype is depreca
ted and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to
 the constructor
W1106 03:48:12.794875 10276 deprecation.py:323] From <ipython-input-1-1b96
f7fad4ca>:49: softmax_cross_entropy_with_logits (from tensorflow.python.op
s.nn_ops) is deprecated and will be removed in a future version.
Instructions for updating:

Future major versions of TensorFlow will allow gradients to flow
into the labels input on backprop by default.

See `tf.nn.softmax_cross_entropy_with_logits_v2`.
```

```
Iter 0, Minibatch Loss= 0.512333, Training Accuracy= 0.83000
Iter 10, Minibatch Loss= 0.152185, Training Accuracy= 0.95000
Iter 20, Minibatch Loss= 0.031061, Training Accuracy= 0.99000
Iter 30, Minibatch Loss= 0.016309, Training Accuracy= 1.00000
Iter 40, Minibatch Loss= 0.051129, Training Accuracy= 0.99000
Optimization finished
Testing Accuracy: 0.9769
```

```
In [ ]:
```

```
In [ ]:
```