

In []:

```
import tensorflow as tf
sess = tf.Session()
```

In []:

```
data = tf.placeholder("float", shape=[1, 7, 7, 1])
conv = tf.nn.conv2d(data, filter=tf.Variable(tf.constant(0.1, shape=[4, 4, 1, 1])), strides=[1, 2, 2, 1],
                    padding='SAME')
print(conv.get_shape()) # prints (1, 4, 4, 1), but should be (1, 5, 5, 1)
```

In []:

```
# one_hot
oh0 = tf.one_hot([0, 1, 2, 3, 4], depth=10)
oh1 = tf.one_hot([0, 1, 2, 3, 4], depth=3)
oh2 = tf.one_hot([[0, 1, 1], [1, 0, 2], [1, 0, 0]], depth=3)

print('-----one hot-----')
print('-----\n oh0 \n')
print(sess.run(oh0))

print('-----\n oh1 \n')
print(sess.run(oh1))

print('-----\n oh2 \n')
print(sess.run(oh2))
```

In []:

```

# reshape
x = [0., 1., 2., 3., 4., 5., 6., 7.]
y = tf.reshape(x, [2, 2, 2])
# the order is
print(sess.run(y))

# unstack
# unstack is rolling the index to the first: e.g when axis =1 : [1, 2, 3] -> [2, 1, 3]
y0 = tf.unstack(y, axis=0)
y1 = tf.unstack(y, axis=1)
y2 = tf.unstack(y, axis=2)

# transpose
# transpose is more powerful than unstack
y0_2 = tf.transpose(y, [0,1,2])
y1_2 = tf.transpose(y, [1,0,2])
y2_2 = tf.transpose(y, [2,0,1])

print('-----unstack-----')
print('-----\n y \n')
print(sess.run(y))
print('-----\n y0 \n')
print(sess.run(y0))
print('-----\n y1 \n')
print(sess.run(y1))
print('-----\n y2 \n')
print(sess.run(y2))

print('-----transpose-----')
print('-----\n y0_2 \n')
print(sess.run(y0_2))
print('-----\n y1_2 \n')
print(sess.run(y1_2))
print('-----\n y2_2 \n')
print(sess.run(y2_2))

```

In []:

```

# exercise
# the data is like (pictures, pixel, channel) change it to (picture, channel, pixel_row, pixel_column)
pic1 = [[11,225,13],[255,43,23], [42,34,200], [255,255,0]]
pic2 = [[255,52,255],[0,0,223], [0,0,0], [255,255,0]]
raw_data = [pic1, pic2]

nd = tf.transpose(raw_data, [0,2,1])
shape = nd.get_shape()
shape
nd = tf.reshape(nd, [nd.get_shape()[0], nd.get_shape()[1], 2, 2])
new_data = sess.run(nd)
print(new_data)

```

In []:

```
#-----  
# part 3  
# squeeze: remove the redundant dimension  
import numpy as np  
a = np.array([[1, 2, 3, 4, 5]])  
a2 = tf.squeeze(a)  
  
b = np.array([[[1], [2]], [[3], [4]]])  
b2 = tf.squeeze(b)  
  
print('-----squeeze-----')  
print('-----\n a2 \n')  
print(a)  
print(a.shape)  
a2_ = sess.run(a2)  
print('\nsqueezed\n:')  
print(a2_)  
print(a2_.shape)  
  
print('-----\n b2 \n')  
print(b)  
print(b.shape)  
b2_ = sess.run(b2)  
print('\nsqueezed\n:')  
print(b2_)  
print(b2_.shape)
```