

```

operation =
tf.add(var, 1)update_operation =
tf.assign(var, add_operation)with tf.Session() as sess :
sess.run(tf.global_variables_initializer())for i in range(3) :
sess.run(update_operation)print(sess.run(var))
value =
sess.run(z1, feed_dict =
x1 : 1, y1 : 2)z1_value, z2_value =
sess.run([z1, z2], feed_dict =
x1 : 1, y1 : 2, x2 : [[2], [2]], y2 : [[3, 3]])print(z1_value)print(z2_value)

```

§

```

x)/sqrt(Var(x) + variable_epsilon)
x =
x *
scale +
offset
x =
1/m * sum_{i=1}^m x_i
sigma^2 =
1/m * sum_{i=1}^m (x_i -
x)^2
patch =
mpatches.Patch(color = '
red', label = '
sigmoid')blue_patch =
mpatches.Patch(color = '
blue', label = '
tanh')with tf.Session() as sess :
[x, y1, y2] =
sess.run([x, y1, y2])plt.plot(x, y1, 'r', x, y2, 'b')ax =
plt.gca()ax.annotate(r''
-2x
1/(1+e^-x)
sigmoid(x) =
1/(1+e^-x)
fun.pngactivate fun
patch =
mpatches.Patch(color = '
blue', label = '
softplus')yellow_patch =
mpatches.Patch(color = '
yellow', label = '
relu')with tf.Session() as sess :
[x, y2, y3] =
sess.run([x, y2, y3])plt.plot(x, y2, 'b', x, y3, 'y')ax =
plt.gca()plt.xlabel('x')ax.annotate(r''^x)
relu(x) =
max(x, 0)
patch, yellow_patch])plt.savefig('relu_softplus.png', dpi =
600)
softplus.png
shape =
[1, 4])print(sess.run(b))c =
tf.nn.dropout(a, 0.5, noise_shape =
[1, 1])print(sess.run(c))
data =
tf.Variable(tf.random_normal(shape =
[10, 9, 9, 3], mean =
0, stddev =
1), dtype =
tf.float32)kernel =
tf.Variable(tf.random_normal(shape =
[2, 2, 3, 2], mean =
0, stddev =
1, dtype =
tf.float32))
data, kernel, strides =
[1, 1, 1, 1], padding = '
SAME')init =
tf.global_variables_initializer()with tf.Session() as sess :
sess.run(init)print(sess.run(y).shape)

```

- ①
- ②
- ③
- ④
- ⑤
- ⑥
- ⑦
- ⑧