nn_example

April 12, 2017

In [1]:
```
from __future__ import division, print_function

import numpy as np
import tensorflow as tf
import utilities
```

1 Initialization

In [2]:
```
graph = tf.Graph()
sess = tf.InteractiveSession(graph=graph)
```

In [3]:
```
x0 = tf.placeholder(tf.float64, [None, 1], name='input_data')
```

In [4]:
```
W1 = tf.Variable(np.array([[1],
                           [-1]], dtype=np.float64),
                 name='W1')

W2 = tf.Variable(np.array([[1, 1],
                           [5, 2]], dtype=np.float64),
                 name='W2')
```

2 Define graph/network

In [5]:
```
def relu(x, weights, name='relu'):
    with tf.variable_scope(name):
        # Equivalent to weights.dot(x), but x can be
        # a row-vector with multiple entried
        m = tf.matmul(x, weights, transpose_b=True)
        return tf.maximum(m, 0.)
```

In [6]:
```
x1 = relu(x0, W1, name='F1')
x2 = relu(x1, W2, name='F2')

y = x2
y1, y2 = tf.split(y, 2, axis=1, name='split_y1_y2')
```

In [7]:
```
utilities.show_graph(graph)
```

```
<IPython.core.display.HTML object>
```

3 Run network

```
In [8]: x_data = np.array([[1.]])
        feed_dict = {x0: x_data}

        sess.run(tf.global_variables_initializer())
```

```
In [9]: # Evaluate output
        sess.run(y, feed_dict=feed_dict)
```

```
Out[9]: array([[ 1.,  5.]])
```

```
In [10]: # Evaluate gradient
         g1 = tf.gradients(y1, W1)
         g2 = tf.gradients(y1, W2)

         grad1, grad2 = sess.run((g1, g2), feed_dict=feed_dict)

         print(grad1, grad2, sep='\n')
```

```
[array([[ 1.],
       [ 0.]])]
[array([[ 1.,  0.],
       [ 0.,  0.]])]
```

```
In [ ]:
```