Deep Learning 182, HW #1

Roy Uziel	Irit Chelly
203398854	021565510

April 28, 2018

1 Network architecture

In our graph we used convolutional layers. The input image is of size 28*28.

```
new_input = tf.reshape(input_images, [-1, 28, 28, 1])
```

We defined the following hidden layers:

1. Convolutional Layer 1:
   We used 32 filters, each filter is a kernel of size 5*5. Each neuron in this layer is a result (scalar) of a convolution of each kernel centered on one neuron in the input layer. Thus, this layer consists of 28*28*32 neurons. We then compute the activation function relu on the result of each neuron:

```
    conv1 = tf.layers.conv2d(
            inputs=new_input,
            filters=32,
            kernel_size=[5, 5],
            padding="same",
            activation=tf.nn.relu)
```

2. Pooling Layer 1:
   Here we reduce the spatial size of the conv. layer by using a Max Pooling filter of size 2*2 and apply the maximum value of each 2*2 sized part of the image (Convolutional Layer 1):

```
pool1 = tf.layers.max_pooling2d(inputs=conv1, pool_size=[2,
    2], strides=2)
```

3. Convolutional Layer 2:

```
  conv2 = tf.layers.conv2d(
        inputs=pool1,
        filters=64,
        kernel_size=[5, 5],
        padding="same",
        activation=tf.nn.relu)
```

4. Pooling Layer 2:

```
      pool2 = tf.layers.max_pooling2d(inputs=conv2,
    pool_size=[2, 2], strides=2)
```

5. Reshaping

```
        pool2_flat = tf.reshape(pool2, [-1, 7 * 7 * 64])
```

6. Dense

```
        dense = tf.layers.dense(inputs=pool2_flat, units
    =1024, activation=tf.nn.relu)
```

7. Dropout

```
    dropout = tf.layers.dropout(inputs=dense, rate=0.2)
)
```

8. Output

```
    logits = tf.layers.dense(inputs=dropout, units=10)
```

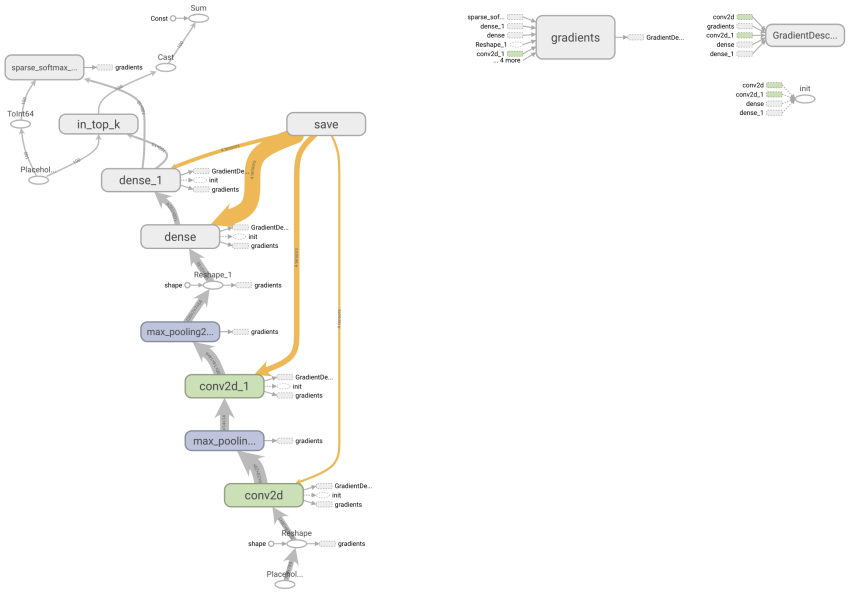

Figure 1: Tensorboard : Structure Graphs

5.2 attach a short document describing the network architecture and any other architectures that you tested

5.3 Make sure to have your full name and ID on the top of the document

5.4 It is recommended to add tensorboard screenshots that describe the results (Optional)

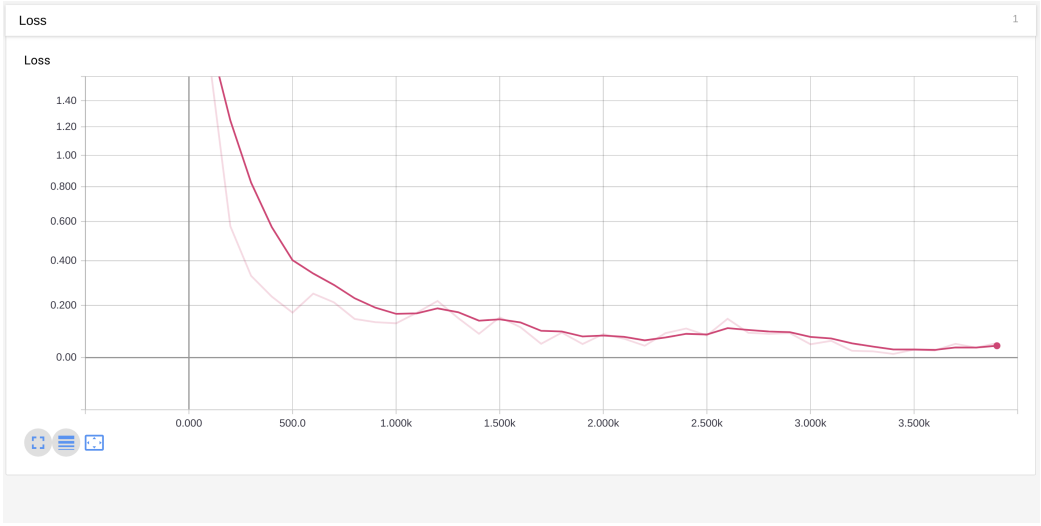

Figure 2: Tensorboard : Loss Over Time

Figure 3: Console Results