Logits

Use a fully-connected layer to extract multiclass logits from the CNN.

Chapter Goals:

- Obtain the logits for each digit class

A. Multiclass logits

Since there are 10 possible digits an MNIST image can be, we use a 10 neuron fully-connected layer to obtain the logits for each digit class. The logits are the output of the `model_layers` function.

The rest of the model follows the standard format for multiclass classification:

- Softmax applied to the logits to convert them into per class probabilities
- The `labels` are one-hot vectors, where the "hot index" corresponds to the digit in the MNIST image
- Softmax cross entropy to calculate loss

Time to Code!

In this chapter, we'll create the helper function, `get_logits`, which obtains logits from the previous chapter's `dropout`.

We use a final fully-connected layer to obtain our logits, which we return as the output of our function.

Set `logits` equal to `tf.layers.dense` applied with `dropout` as the inputs, `self.output_size` as the output size, and `name` equal to `'logits'`.
Then return `logits`.

```
1  import tensorflow as tf
2
3  class MNISTModel(object):
4      # Model Initialization
5      def __init__(self, input_di
6          self.input_dim = input_
7          self.output_size = outp
```

```
    # Get logits from the dropo
    def get_logits(self, dropou
        # CODE HERE
        pass

    # CNN Layers
    def model_layers(self, inpu
        reshaped_inputs = tf.re
            inputs, [-1, self.i
        # Convolutional Layer #
        conv1 = tf.layers.conv1
            inputs=reshaped_inp
            filters=32,
            kernel_size=[5, 5],
            padding='same',
            activation=tf.nn.re
            name='conv1')
        # Pooling Layer #1
        pool1 = tf.layers.max_p
            inputs=conv1,
            pool_size=[2, 2],
            strides=2,
            name='pool1')
```