



TensorFlow skill testing project: Image Recognition in TensorFlow

Objective: Tackle an open problem in image recognition: the Google Street View Housing Numbers dataset.

Architecture: Your code will consist of three modules. This document contains the specification for the second module.

Module 2: `graph_construction.py`

This module will build (but not execute) a computational graph to carry out the image classification. You'll probably have to experiment with a few different architectures before you find one that works. On this dataset, an accuracy of ~80% is reasonably good, but you may be able to aim higher than this.

You may want to begin by looking at AlexNet for some inspiration: <http://www.embedded-vision.com/sites/default/files/technical-articles/FPGAsNeuralNetworks/Figure1.jpg>. Note that the AlexNet model consists of a number of convolutional layers (with max pooling), with three fully-connected layers at the end. This is actually a fairly standard strategy. Note also that training your convolutional net with dropout can be a great way to make it more robust.

This module should contain a function with the following signature:

```
classifier(learning_rate, use_dropout) => model, train_op, accuracy, x, y
```

learning_rate	the algorithm's learning rate
use_dropout	a Boolean which, if set to True, causes the network to use dropout for training
model	output from the <code>tf.global_variables_initializer()</code> function (or from <code>tf.initialize_all_variables()</code> in earlier versions of TensorFlow).
train_op	the output of <code>tf.train.GradientDescentOptimizer(learning_rate).minimize(cost_function)</code>
accuracy	the accuracy score for the current batch
x	the input placeholder variable. You'll need this as an output of your function in order to feed in values for x during training.
y	the label placeholder variable. You'll need this as an output of your function in order to feed in values for y during training.



classifier the function that builds (but **does not** execute) the computational graph that will analyze the Google Street View Housing Numbers dataset. You may use any TensorFlow optimizer (`AdagradOptimizer`, `AdamOptimizer`, etc.), and any initialization parameters you like.

Your module can contain additional functions, but you must have **classifier()** somewhere in your script.

Once you've completed this exercise, send it to [@yazabi](#) and we'll give you feedback on your code!