UDACITY

PROJECT

Object Classification

A part of the Deep Learning Nanodegree Foundation Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

1 SPECIFICATION REQUIRES CHANGES

Great job! Keep up the good work. This tutorial has some good info on conv nets.

Required Files and Tests

The project submission contains the project notebook, called "dlnd_image_classification.ipynb".

All the unit tests in project have passed.

Preprocessing

The `normalize` function normalizes image data in the range of 0 to 1, inclusive.

Nice and simple, good job! For more real-life applications, we would typically use sklearn's StandardScaler so we don't have to hand-code everything.

George Hinton, a pioneer in neural networks, has a nice Coursera course on neural nets. This lecture explains why normalization helps.

The `one_hot_encode` function encodes labels to one-hot encodings.

Nice job! I always find the one-line solutions neat at codewars.com.
There are three one-line solutions I know of for this problem:
`np.eye(10)[x]`

or

`np.identity(10, dtype=int)[x]`

or

`sklearn.preprocessing.label_binarize(x, classes=range(10))`

Neural Network Layers

The neural net inputs functions have all returned the correct TF Placeholder.

The `conv2d_maxpool` function applies convolution and max pooling to a layer.

The convolutional layer should use a nonlinear activation.

This function shouldn't use any of the tensorflow functions in the tf.contrib or tf.layers namespace.

Good work! For ReLU, it's a good idea to set the biases to a small positive value so the neurons don't die - or use leaky relu. You can set the biases to a constant with `tf.constant(0.1)`.

To save the trouble of figuring out which stddev value is good to use for your weight initialization, there are a few xavier initializers you might want to take a look at. They initialize the weights in a more ideal way that adapts to the network shape. Here's an example of how to use them.

The `flatten` function flattens a tensor without affecting the batch size.

The `fully_conn` function creates a fully connected layer with a nonlinear activation.

The `output` function creates an output layer with a linear activation.

Neural Network Architecture

The `conv_net` function creates a convolutional model and returns the logits. Dropout should be applied to alt least one layer.

Nice model! Conventionally, 2x2 is used for the maxpool kernel and stride. The 2x2 stride will cut the image in half.

Here's more info on the architecture of conv nets. Usually we don't apply dropout to convolutional layers because they already have a lot of regularization built-in.

Every great net out there I see has the number of convolutional kernels increasing with network depth. For example, the VGG net designs. Here's a keras model, and here's a tensorflow model (the tf model code is a bit confusing) for VGG-19. They start with a number (64) for number of kernels, and double it each time they descend a layer. They are using 3x3 kernels (I've always seen it recommended to use 3x3 or 5x5 kernels, although I've seen 2x2 and 4x4 kernels work well for this project). In VGG-19, they don't use dropout until the dense layers, and then it's at 0.5.

As far as tricks for improving performance, there's batch normalization (how to use it here). I've seen dramatic improvements with that.
Another trick you could try is global average pooling.

Neural Network Training

The `train_neural_network` function optimizes the neural network.

The `print_stats` function prints loss and validation accuracy.

You're actually printing the training set accuracy. We want to see validation set to check for overfitting:
`Use the global variables valid_features and valid_labels to calculate validation accuracy.` If the validation loss is going up (and validation accuracy down), we're overfitting.

The hyperparameters have been set to reasonable numbers.

The neural network validation and test accuracy are similar. Their accuracies are greater than 50%.

Nice accuracy! You could also try using elu instead of relu for even higher accuracies: https://arxiv.org/abs/1511.07289

RESUBMIT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

RETURN TO PATH

Rate this review

Student FAQ Reviewer Agreement