UDACITY

‹ Back to Deep Learning Nanodegree

Object Classification

审阅

代码审阅

HISTORY

Meets Specifications

This is a perfect first submission. You have a very good understanding of underlying concepts. Congratulations on successfully completing the project.

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.

Well done! Your range of input should now be between 0 and 1. When inputs to the neural network are normalized, neural network training is often more efficient, which leads to a better predictor.

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

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.

Well done!

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

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

Suggestion: Please explicitly set the activation to tf.reku for this layer. Different versions of tensorflow have different default settings for tf.contrib.layers.fully_connected so best to set the activation function yourself.

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

Well done! Please note that since this is the output layer we shouldn't be applying any activation functions to this layer.

Neural Network Architecture

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

Good job connecting all the layers properly!

Suggestion: You can also use multiple convolution+max pool layers to improve the ability of your network to learn complex features

Neural Network Training

The `train_neural_network` function optimizes the neural network.

The `print_stats` function prints loss and validation accuracy.

Well done!

The hyperparameters have been set to reasonable numbers.

Your choice of hyper parameter is good.
The blog below has an interesting comparison of CIFAR 10 accuracies for different architectures:
http://zybler.blogspot.ca/2011/02/table-of-results-for-cifar-10-dataset.html
You can try reading through some of the papers and understanding what has worked well for others.

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

Well done!

下载项目

返回 PATH

给这次审阅打分

学员 FAQ