

PROJECT

Object Classification

A part of the Deep Learning Nanodegree Foundation Program

PROJECT REVIEW

CODE REVIEW

NOTES

Requires Changes

SHARE YOUR ACCOMPLISHMENT

1 SPECIFICATION REQUIRES CHANGES



Hey there, great job working on this project!

The only little things left is to add a Dropout layer to your network, please a look at the comments below.

Great progress so far, few little touches left and you will be good to go
Kudos and happy learning

I'm not sure about how the filespace relates to the training results, that actually sounds a little strange to me. I would appreciate if you could paste the link to me in the feedback section. I have given you suggestions to improve the results, look below.

Required Files and Tests



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

iPython Notebook is present.



All the unit tests in project have passed.

Your code passed the unit tests. Great job!

Rate this review



Preprocessing



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



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.



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.

Flawless implementation of all neural network layers!

Neural Network Architecture



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

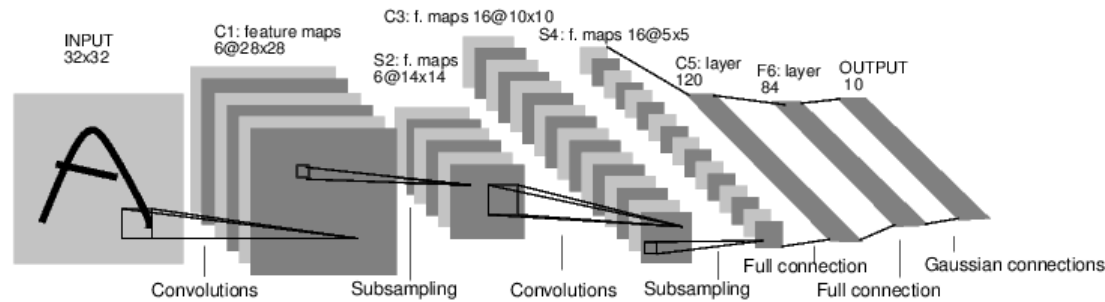
Please make sure to add a **Dropout** layer here.

To further improve the accuracy you should consider adding more of convolutional layers and have more of the fully connected ones, it will let your network be more flexible.

There is no rule of thumb in this field regarding the layers/network size, all of the great Networks are the products of countless hours of trying different combinations pretty much randomly. If you read papers there are no much explanation why one thing or another one worked, what it actually says is we took this we tried it and it worked, or it didn't work we changed this and it worked, in that manner.

You can try to implement some popular Neural Network architectures. And one of the good ones and simple ones to start with is popular LeNet, developed by Yan LeCun (director of AI Research at Facebook), and see how well it performs if you got more time try to implement AlexNet and so on.

Here is how LeNet looks like:



Do not hesitate to take ready network structures and use them for your problem, it is alright, and will save you a lot of time keeping away from reinventing the wheel.

Also, you can keep modify your own one, increase the number of Convolutional layers and fully connected ones, play with parameters, make the network a little bigger so it can learn more and see how it goes. But always keep track of the training set accuracy as well, so you understand when the network structure is too big for the task and you either need to increase the chance of dropout or decrease the NN size.

Neural Network Training



The `train_neural_network` function optimizes the neural network.



The `print_stats` function prints loss and validation accuracy.



The hyperparameters have been set to reasonable numbers.



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

Great results! Pure guessing would make it only to 10% of an accuracy, with your NN you made it to ~54%!
Suggestions given above will help you to improve it further!

 RESUBMIT PROJECT

 DOWNLOAD PROJECT

Learn the [best practices for revising and resubmitting your project](#).

RETURN TO PATH

[Student FAQ](#)