

COMP4107 – Neural Networks

Assignment 4 Answers

Members: Krystian Wojcicki, Michael Kuang

Notes:

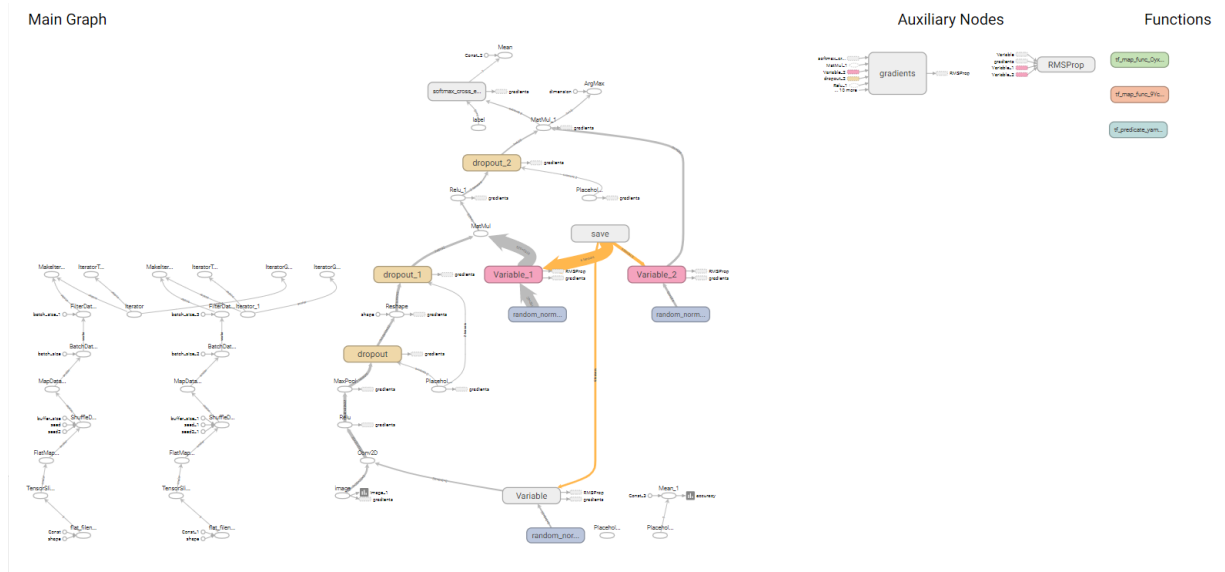
To run the data augmentation ``python preprocess_image.py``. This step is necessary before running any model.

To run a model you type ``python q1.py <model #>``, e.g: ``python q1.py 1`` to run model 1.

Running tensorboard ``tensorboard --logdir=./logs``

- 1) Data was modified by randomly rotating the image by 90 degrees times a random integer between 0 to 4, and randomly applying a salt and pepper to the image.
- 2) The 5 models that were investigated were
 - a. Model1, 1 convolutional layer with 3,3,3 kernels and 32 feature maps and max pooling of 2x2 with strides of 2x2.
 - b. Model2, 2 convolutional layers with 3,3,3 and 3,3,32 sized kernel and max pooling of 2x2 and stride of 2x2 for both layers.
 - c. Model3, 1 convolutional layer however two conv2d and relu in a row before doing pooling ie conv2d -> relu -> conv2d -> relu -> dropout. Kernels of size 3,3,3 and 3,3,32 and max pooling of 4x4 with strides of 4 by 4 were used.
 - d. Model4, Same model as model a (or model 1), but using strides of 1x1 for max pooling
 - e. Model5, Like that of model 3 but adding another convolutional layer on top of that.

3)



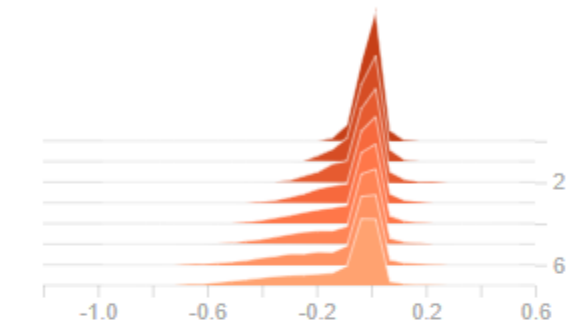
Example of computational graph of model1 generated using tensorboard. This .png is in the zip folder under model1.png

4)

weights_of_10_neuron_output_layer_model1

weights_of_10_neuron_output_layer_model1

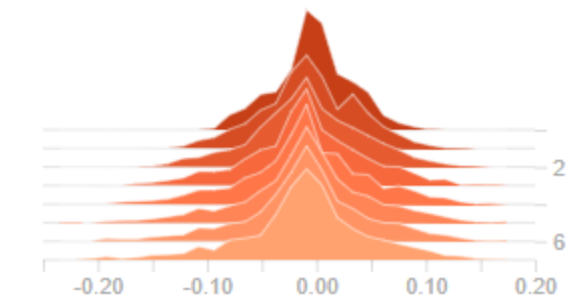
attempt_0\model 1



weights_of_first_convolution_layer_3x3x3x32_model1

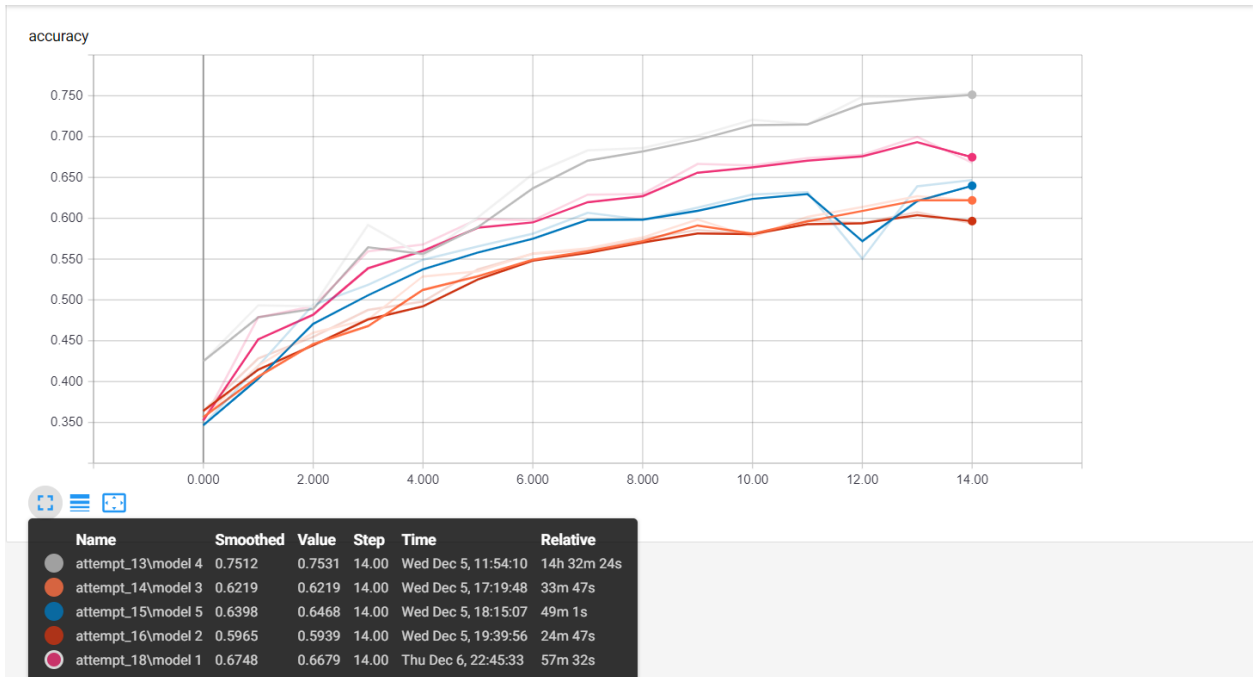
weights_of_first_convolution_layer_3x3x3x32_model1

attempt_0\model 1



Example of the weight histograms generated after each epoch. Checkpoints are done every 5 epochs and put into folder ./logs

5)



Example of accuracy of each of the 5 models being run. One small note the relative time can be inaccurate as wall time is measured not cpu time and sometimes the computer upon which these models were running on went to sleep.

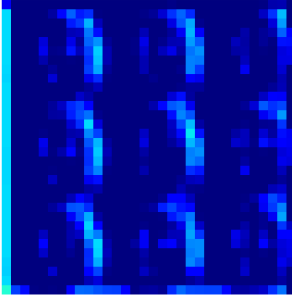
Overall between models 1,2,3,5 there are differences in accuracy, but it is tough to tell whether that is due to the limit # of epochs, due to random chance of which images were used or due to the architectures. While model4 did seem to perform the best due to above reasons it's difficult to tell if it is the best performing model.

Tensorflows example implementation of a model that performs classification for cifar-10 data, https://www.tensorflow.org/tutorials/images/deep_cnn#cifar-10_model, has 2 layers but used more features in their feature maps. Their model achieved an accuracy of ~86%, https://github.com/tensorflow/models/blob/master/tutorials/image/cifar10/cifar10_train.py, but ran over 256 epochs and used many more data-augmentation techniques. So, it seems likely that model 2/model 5 should have performed the best but given the epoch constraint there may have not been enough time for the networks to train properly.

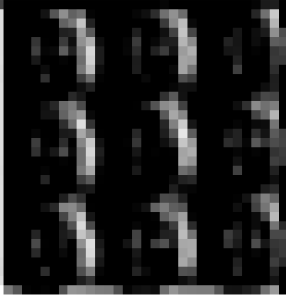
Overall given that the type of objects we are trying to classify in cifar-10 are quite complex having more than 1 layer and many feature maps seems to be optimal.

6)

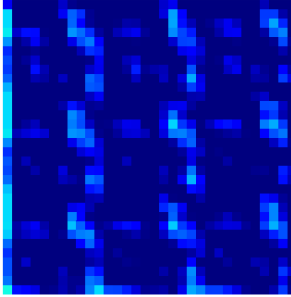
feature_map/feature_map/_1_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:28 Eastern Standard Time



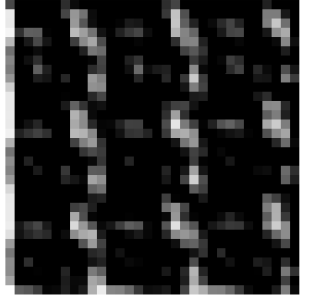
feature_map/feature_map/_1_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:28 Eastern Standard Time



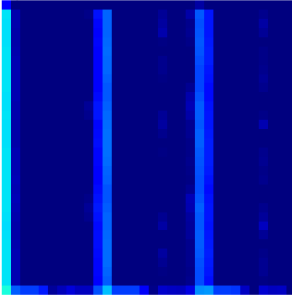
feature_map/feature_map/_2_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:28 Eastern Standard Time



feature_map/feature_map/_2_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:28 Eastern Standard Time



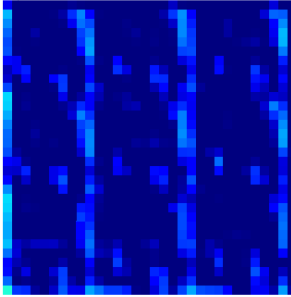
feature_map/feature_map/_3_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



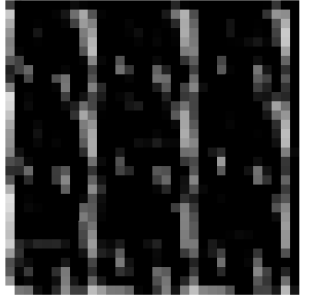
feature_map/feature_map/_3_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



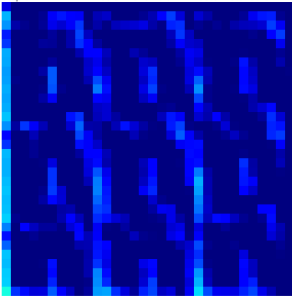
feature_map/feature_map/_4_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



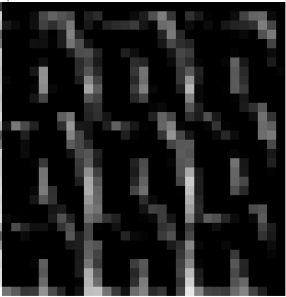
feature_map/feature_map/_4_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



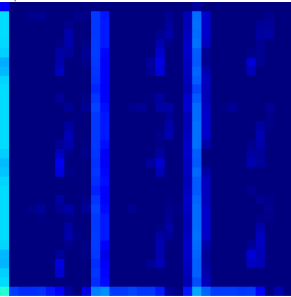
feature_map/feature_map/_5_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



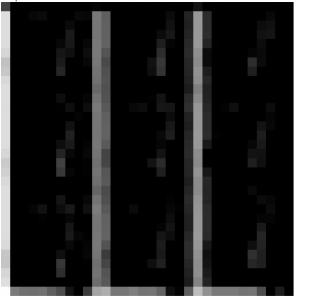
feature_map/feature_map/_5_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



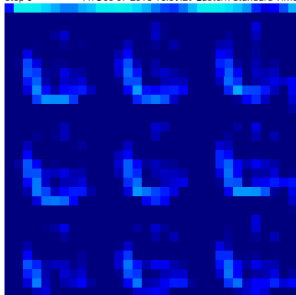
feature_map/feature_map/_6_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



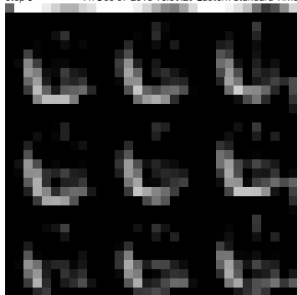
feature_map/feature_map/_6_highest_activation_for_feature_heatmap_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



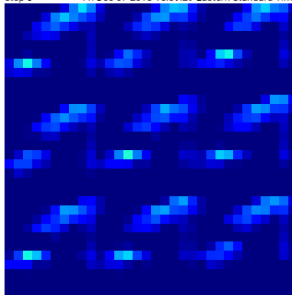
feature_map/feature_map/_7_highest_activation_for_feature_map_28/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



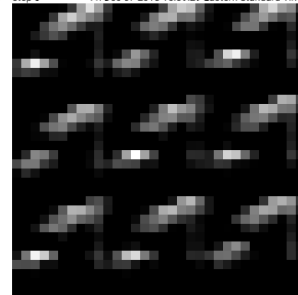
feature_map/feature_map/_7_highest_activation_for_feature_map_28/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



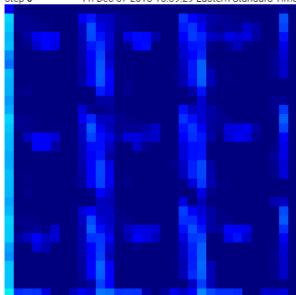
feature_map/feature_map/_8_highest_activation_for_feature_map_20/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



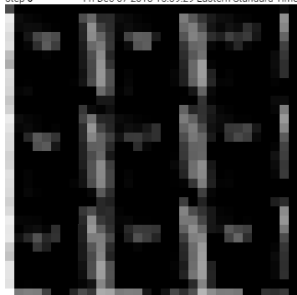
feature_map/feature_map/_8_highest_activation_for_feature_map_20/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



feature_map/feature_map/_9_highest_activation_for_feature_map_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



feature_map/feature_map/_9_highest_activation_for_feature_map_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



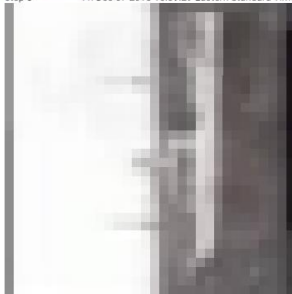
image/image/image_causing_1_highest_activation_for_feature_map_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:28 Eastern Standard Time



image/image/image_causing_2_highest_activation_for_feature_map_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:28 Eastern Standard Time



image/image/image_causing_3_highest_activation_for_feature_map_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



image/image/image_causing_4_highest_activation_for_feature_map_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



image/image/image_causing_5_highest_activation_for_feature_map_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



image/image/image_causing_6_highest_activation_for_feature_map_23/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



image/image/image_causing_7_highest_activation_for_feature_map_28/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



image/image/image_causing_8_highest_activation_for_feature_map_20/image
attempt_0/model_1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



image/image/image_causing__9_highest_activation_for
_feature_map_23/image
attempt_0\model 1
step 0 Fri Dec 07 2018 15:39:29 Eastern Standard Time



Example of the subset of the top 9 patches for layer 1 of model 1. For each epoch tensor board generates the top 9 patches for 9 random feature maps.