Visual Recognition

Assignment 5

- Sumit Chakraborty (MT2018121)
- Shobhit Behl (IMT2016024)
- Aayush Grover (IMT2016005)

Experiment 1 : (for 20 epochs)

1. Started with initial resnet architecture code provided and got a test accuracy of 65.2%.
2. Changing the default activation of resnet layer to “tanh”, we got a test accuracy of 68.48%.
3. Now, keeping the default activation as “tanh”, but keeping the activation of every 2nd resnet layer in a res_block in the resnet model as “relu”, test accuracy increased to 68.73%.
4. This accuracy was similar to the one we got when we switched “relu” and “tanh” functions.
5. When we add “tanh” activation to the first res_block’s last resnet layer in the above model, we get accuracy of 69.03%.
6. On normalizing the data for the above model, we get accuracy of 69.37%.
7. Decreasing batch size to 16, gives us an accuracy of 73.2%.
8. Adding Dropout layer (of rate 0.2) before Dense layer in our model, we get accuracy of 72.45%.
9. Removing the above mentioned Dropout layer and adding a Dropout layer after final “y” and “x” resnet layers in a res_block with dropout rate of 0.25 and 0.35 respectively gave us a test accuracy of 71.52%.
10. Adding all 3 Dropout layers (8. And 9.), the test accuracy decreases to 71.4%.

Experiment 2 :

1. Started with initial resnet architecture code provided and got a test accuracy of 65.2%.
2. Keeping all the parameters same and changing the batch size from 32 to 64 and epochs from 20 to 30 we got a test accuracy of 67%.
3. Similarly we tried to increase the batch size up until 128 while keeping all the parameters same but the test accuracy didn’t change quite dramatically.
4. Further on decreasing the batch size to 16 with the same parameters there was a huge jump in the test accuracy.

5. Then as the batch size as 16 for further experiments we tried to tune the batch norm parameter, mainly, if batch norm is to be applied or not and if it is applied, will the batch norm be applied before the convolution layer or after it.
6. Our experiments with different combinations showed that with Batch Norm applied and after the convolution layer the test accuracy increased.
7. Then we moved on to tune our optimization functions, playing with rmsprop, SGD, Adam, Adadelta and SGD with momentum (normal).
8. While SGD with a momentum of 0.9 and Adam gave almost the same accuracy with the same learning rate, Adadelta performed the poorest give a very low accuracy.
9. We then tried to increase and decrease the learning rate with the combination of the Adam and SGD with momentum optimization functions. The results showed that with Adam as the optimization function with a learning rate of 0.001, the model performed much better than its SGD counterpart. The model finally resulted a test accuracy of 71.2% jumping into the 70% from all its initial other combinations.
10. Keeping the params of the above mentioned model same we tried to tune the loss function. We tried to use - categorical_crossentropy, sparse_categorical_crossentropy, hinge, squared_hinge and KL divergence loss functions. While the categorical_cross_entropy and sparse_cross_entropy gave the same results as above, others resulted in a drop of accuracy.

Final : (Combination of Experiments 1 and 2)

The best set of hyperparameters :

- Batch size: 16
- Epochs : 100
- Optimizer : Adam
- Learning rate : 0.001 with a decay of 1e-6
- Loss function : categorical_crossentropy
- Activation Functions: “tanh” and “relu” in the resnet layers. “Softmax” in the dense layer.
- Data Normalization
- Dropout layers after the final “y” and “x” resnet layers in a res_block with dropout rate of 0.25 and 0.35 respectively.
- Batch Norm : True and applied after convolution layer
- Depth : 44

Test Accuracy : 84.87%