CS 577 s21 – Assignment 4

Due by 4/27/2021

- In this assignment you will explore the topic of convolutional neural networks . You will need to implement several Convolutional Neural Networks on GPU using a singularity container to solve classification problems. This assignment will use the Xsede GPU cluster. The implementation should generally be done in Keras but you may use TensorFlow or PyTorch If you are already familiar with them and would like to use them. Note that we may not be able to offer help with frameworks other than Keras. You are not required to use the Xsede cluster.
- Follow the submission instructions of the first assignment.

Theoretical questions

1. Let I be a 4×4 RGB image where the R channel is all 1-s and G channel is all 2-s. The B channel has a value of 1 in its first row, a value of 2 in its second row, a value of 3 in its third row, and a value of 4 in its 4th row. Compute the convolution of this image with a 3×3filter having all ones without zero padding.
2. Repeat the previous question with zero padding.
3. Repeat the previous question when using dilated (atrous) convolution with a dilation rate of 2.
4. Explain the template matching interpretation of convolution.
5. Explain how multiple scale analysis can be achieved with a fixed window size (using a pyramid).
6. Explain how to compensate for spatial resolution decrease using depth (number of channels) and the purpose for doing so.
7. Given a $128 \times 128 \times 32$ tensor and 16 convolution filters of size $3 \times 3 \times 32$, what with be the size of the resulting tensor when convolving without zero padding.
8. Repeat the previous question when using a stride of 2.
9. Explain how the number of channels can be reduced using a 1×1 convolution.
10. Explain the interpretation of convolution layers and the difference between early and deeper convolution layers.

11. Let I be an image as in question 1. Write the result obtained using max pooling with a 2×2 filter with a stride of 2.
12. Explain the purpose of pooling.
13. Explain the purpose of data augmentation and when it is most useful.
14. Explain the purpose of transfer learning and when it is most useful.
15. Explain the need for freezing the coefficients of the pre-trained network.
16. Explain how he coefficients of a pre-trained network can be fine-tuned.
17. Explain the purpose of inception blocks.
18. Explain the advantage of residual blocks.
19. Explain how intermediate activations of convolution layers can be visualized given an input. What is the purpose for doing so?
20. Explain how the filter weight of the trained convolution layers can be visualized (using gradient ascent to find the input with maximal response). What is the purpose for doing so?
21. Explain how the heatmap of class activation can be visualized for a specific image and class. explain how pooled gradients can be used to weight channels in this visualization. Explain the purpose of this visualization.

Programming questions

1. Binary classification:
   (a) Download the Kaggle's Cats and Dogs [1] dataset and select a subset of 2000 dogs and 2000 cat images to be used for training, validation, and testing. The use of subsets is intended to simulate the condition of limited data.
   (b) Define a training, validation, and testing data generators.
   (c) Build a convolutional neural network using several convolution, pooling, and normalization layers, followed by one or more dense layers. Flatten the data between the convolution and dense layers.
   (d) Evaluate performance and tune hyperparameters as needed.
   (e) Visualize the activation of some of the convolution layers and draw a conclusion.
   (f) Visualize the filters learned in training by finding the input that will maximize their response and draw a conclusion.
   (g) Replace your colvolution layers with the pre-trained convolution base of VGG16. Train with the convolution base frozen and evaluate the results.

[1] https://download.microsoft.com/download/3/E/1/3E1C3F21-ECDB-4869-8368-6DEBA77B919F/kagglecatsanddogs_3367a.zip

(h) Once the previous training step is complete fine tune the convolution base: unfreeze the convolution base and continue to train. Evaluate the results you get and draw a conclusion.

(i) Modify the data generator to perform data augmentation. Retrain with frozen convolution base and evaluate the results.

2. Multiclass classification:

(a) Download the CIFAR-10 [2] and load the pickled data into your program.

(b) Build a basic convolutional neural networks with several convolution, pooling, and normalization layers. Flatten the output of the convolution layers and pass it to a single dense layer that will produce the output using softmax activation.

(c) Test the performance of the model you built and tune hyper parameters as needed.

(d) Add one or two inception blocks and test performance.

(e) Remove the inception blocks and add one or two residual blocks instead. Test performance and compare to previous results.

[2] https://www.cs.toronto.edu/~kriz/cifar.html