CSCI 390 Machine Learning – HW4

DUE: Sunday, December 1 - 11:59PM

Part 1 (50 points)

For part 1, you will experiment with a simple convolutional neural network in order to determine the effect of various hyperparameters on model performance. First, train a simple CNN on the MNIST dataset (you may copy the code provided in the class slides). Then, experiment with the network hyperparameters, including layer count, kernel count, kernel size, and stride. Keep all other model parameters the same, including limiting training to only 5 epochs. Document your findings; you should evaluate at least 10 different configurations. Which was the best model? Consider the model size (parameter count in `model.summary`) for each of your model configurations. What is the smallest model you can design that still has acceptable performance (>99% test accuracy)?

Part 2 (50 points)

In part 2, you will dive deeper into the workings of CNNs. Read this guide on how to visualize feature maps inside convolutional nets. Apply the methods discussed in the guide to your smallest CNN from part 1. Use the first digit in the MNIST dataset as your input image. Record the responses from the first nine kernels as your first layer feature maps. Repeat this process to record the first nine feature maps from the second layer as well. Analyze the different maps. Do the different feature maps in the first layer capture different features? Are the second layer maps more complex/abstract than those from the first layer?

Part 3 (50 points)

There have been many successful attempts to create highly accurate CNN models for research purposes. In part 3, you will compare the performance of a few of them. TensorFlow allows the user to instantiate pre-trained models based on many of these architectures in `keras.applications` (read more here). Compare the accuracy of four different models on the test set of CIFAR-100 (also streamable from keras), InceptionV3, VCG19, MobileNetV2, and ResNet101V2. Make sure to change the `input_shape` parameter on each model to 32x32x3 in order to match the images from CIFAR-100. Record the accuracy and the size of each model and calculate the ratio of test accuracy to model size (MB).

Let's pretend that you are the engineer for a team developing a computer-vision powered robotic system for a factory. Fast prediction times are prized, so all ML-related computations will be processed locally on the robot's hardware. However, resources are rather limited, so smaller models are preferred for this task. Using the data that you have collected and the data from the summary page of these models on the keras documentation, decide which of the four pre-trained models you would choose as your starting point. Explain why you think this is the best choice.

Submission

Your submission will consist of a Word or pdf file, containing the items below, and your source code for all problems. Your Word file should contain:

- Responses to parts 1-3
- Feature map images for part 2
- Complete Python code for parts 1-3