Chennai Mathematical Institute
Computer Vision
April 18, 2020

Due Date: Wednesday, April 29, 2020 at 10:00PM IST

Final Project – Part 2
Image Classification using Transfer Learning

Instructions:

1. Integrity and collaboration: Students are encouraged to discuss the problems with others. However, each student must submit their own work. Code should NOT be shared or copied. Please DO NOT use external code unless permitted. Plagiarism is strongly prohibited and may lead to failure of this course.

2. Start early! Training, evaluating and fine-tuning neural networks is a time-consuming task. If you start late, you will not have enough time to achieve good results.

3. If you have any question, please post your question on Piazza.

4. Write-up: Items to be included in the write-up are indicated in the individual sections. Please note that we DO NOT accept handwritten scans for your write-up in this assignment.

5. Late Policy: The assignment will not be accepted past 72 hours beyond the specified due date. You will lose 1 point for each hour you are late beyond the deadline for the first 6 hours. Beyond that, the delay will be counted in day units with a penalty of 10 points for each day you are late.

Good luck!

Overview:

In part 1 of your final project, you defined a ConvNet model, and trained it from scratch using a sizable dataset. In part 2 you will perform transfer learning by using a pre-trained network for feature extraction and by fine-tuning a pre-trained network, for a smaller dataset. Specifically, you will:

a) use the ResNet101 model pre-trained on the ImageNet database as feature extractor, add additional layers for classification, train the model, evaluate it and report the classification accuracy.
b) You will then fine-tune the ResNet101 model, unfreezing a portion of the pre-trained network, train the resulting model, evaluate it and report results.

Dataset

The dataset you are given consists of 15 scene categories, each consisting of 500 images. The categories used are airport terminal, beach, bedroom, bridge, cathedral, dining room, forest trail, highway, house, outdoor market, mountain, playground, restaurant, skyscraper, and underwater coral reef. The images for each category have been split into 375 training images and 125 test images.

As you train and tune the hyperparameters for your model, use a portion of your training set for validation, keeping 250 images for training and the remaining 125 for validation and hyperparameter selection. **Your test set should be used for evaluating your final model only**.

Detailed instructions:

1) The full dataset is located at:
   https://drive.google.com/open?id=1juDjQKzANG9Eutqbr7XN5sEkM15eBV7u
   The .zip file is 2GB.
   I have also split up the data into 3 sections with 5 categories in each section. The 3 .zip files (each under 1 GB) corresponding to the 3 sections are located at:
   https://drive.google.com/open?id=1nFtKcstCVnysI4rJRGmLdqJUBlTDdWB0

   Load the images and get them into the train, validation and test sets. The images are of different width and height. You can make use of Keras provided `ImageDataGenerator` to resize all the images to a target image size. It also has a provisions to do a train / validation split of the train dataset.
   See https://www.tensorflow.org/tutorials/load_data/images

2) For feature extraction, use the ResNet101 model pre-trained on ImageNet as a base model, add your classification layers on top, train the model using the training data, using the validation data set to tune your hyperparameters. Use the test data for evaluating your finalized model. Report on the test accuracy obtained.

3) For fine-tuning of a pre-trained network, use the ResNet101 model as in the previous step, but unfreeze a portion of it and retrain the network using the given training data. Use the validation data to tune the model parameters and evaluate the model using the test data.

4) Write a report detailing your architecture, parameter choices, and accuracy obtained for both the feature-extraction model and fine-tuned model.

Suggested Initial Parameters

Optimizer: *SGD;* Loss: *sparse_categorical_cross_entropy*
Learning rate = 0.0001 to 0.01
Momentum = 0.9
Epochs = 20 (start small)
Batch-size = 32
Try data augmentation –horizontal-flip, rotation, shift, etc.

Extra Credit:

Try the feature-extraction or the fine-tuning step with a pre-trained model that uses a different architecture (VGG or Inception). Include your experiments and findings in the report as well as a comparison with the results from the ResNet101 architecture.

What to submit:

- Link to your Google Colab notebook with relevant experiments and output.
- PDF document with your write-up component including experiments, results and interpretation.