IIT BOMBAY

Lab Assignment #8
Points: 5

Main TAs: *Ashish Agrawal, Govind Saju, Mayank Jain*

Course: *CS 335* – Instructor: *Preethi Jyothi*
Due date: *October 9, 2023*

General Instructions

1. Unlike previous submissions, for this assignment, you should submit .ipynb files downloaded from Colab. **Important:** Please make sure that your submissions are fully run notebooks; do not clear the outputs. This will make it easier to grade.
2. Your final submission is due on Moodle on or before **11.59 pm on Oct 9, 2023**.
3. For your final submission, you need to submit `lab8-Q1.ipynb` and `lab8-Q2.ipynb` with all the required functions fully implemented. If you attempt the extra credit part, you can also submit `lab8-EC.ipynb` that contains your enhancements.
4. You will get 2 points if you match the grader's runs, especially the test accuracies at the end of 30 epochs, for Q1 with the CNN trained from scratch. You will get 0.5 points if your test accuracies with augmentations are improved compared to using inputs without augmentations. You will get 2 points if you match the grader's runs, especially the test accuracies at the end of 30 epochs, for Q2 with the pretrained VGG-11 model. You will get 0.5 points if your test accuracies further improve with training parameters in the last VGG-11 layer.

Q1: Convolutional Neural Networks: Model from Scratch

Go to `https://colab.research.google.com`. Then, go to "File → Upload notebook" and upload the ipynb file at `https://cse.iitb.ac.in/~pjyothi/cs335/lab8-Q1.ipynb`.

Part 1. For this problem, you will implement a convolutional network using the `torch` library to discriminate between images of cats and dogs.

The network should contain the following sequence of modules:

- Convolutional Layer 1: 16 2D filters of kernel size 3 with stride factor of 1 and padding of 1.
- ReLU layer 1.
- Max pooling layer 1: Kernel size 2 and stride 2.
- Convolutional layer 2: 32 2D filters of kernel size 3 with stride factor of 1 and padding of 1.
- ReLU layer 2.
- Max pooling layer 2: Kernel size 2 and stride 2.
- Flatten into a linear layer with $32 \times 64 \times 64$ input nodes and 64 output nodes.
- ReLU layer 3.
- Linear layer mapping down 64 dimensions to 1 and a final sigmoid activation that feeds into a binary cross-entropy loss.

Part 2. With limited amounts of training data (as in this problem), it is common to adopt image transformations to augment the training data. With `torchvision` modules, it is easy to augment the input images with a variety of transformations like rotating, cropping, flipping, etc. Add this functionality in the very last cell in the notebook to augment the training data, and reevaluate performance on the test instances.

Q2: Convolutional Neural Networks: Pretrained Models

Go to `https://colab.research.google.com`. Then, go to "File → Upload notebook" and upload the ipynb file at `https://cse.iitb.ac.in/~pjyothi/cs335/lab8-Q2.ipynb`.

In limited data settings, rather than train a model from scratch, it is common to extract features using pretrained models (i.e., models trained on other image recognition tasks like ImageNet) and add new feedforward layers to train a classifier.

Pretrained model as feature extractor. In this problem, you will load a pretrained model (specifically VGG11, `https://pytorch.org/hub/pytorch_vision_vgg/`), freeze all parameters in its layers (so that no gradients are backpropagated through these layers) and add two trainable linear layers (details specified in the comments) as the final two layers. Evaluate the predictions from this model.

Selective layerwise finetuning of pretrained model. Next, you will make the last layer of the VGG-11 model trainable (and keep the rest of the layers frozen). Again, you will add two linear layers using the `Classifier`. Evaluate the model again to check how it fares compared to the fully-frozen pretrained model. Submit `submission.csv` from the Colab notebook of Q2 to the Kaggle competition at `https://www.kaggle.com/t/d9a00f88c618408db711acfb3d230c47` so that your roll number appears on the Kaggle public leaderboard.

❄ Extra Credit: Climb the Leaderboard on Kaggle

The objective for this extra credit part is to build the best possible dogs vs. cats classifier using any enhancements you like. Enhancements may include additional regularization, architectural modifications, more aggressive data augmentation, finetuning more convolutional blocks (with greater regularization), etc. Create a new notebook `lab8-EC.ipynb` for this part. Submit your predictions for the Kaggle test instances in `submission.csv` to the Kaggle competition (`https://www.kaggle.com/t/d9a00f88c618408db711acfb3d230c47`) so that your roll number appears on both the "Public Leaderboard" (and eventually the "Private Leaderboard" after the assignment concludes). Top-scoring performers on the "Private Leaderboard" (with a suitable threshold determined after the deadline passes) will be awarded up to 3 extra credit points. The exact breakdown of the three points for this question will be announced later.