Assignment 2

AIL721: Deep Learning. Semester II, 2024-2025.
Due Date: March 17th, 2025, 9:00 AM

7 February 2025

Instructions:
a) Please ensure that all code submitted is your original work. You are allowed to discuss in a group; however Copying code or using AI tools such as ChatGPT for assistance in coding assignments is strictly forbidden. We will employ tools such as MOSS to detect plagiarism and other tools to identify AI-generated content.
b) Unless mentioned otherwise, only opencv, pillow, numpy, pytorch and matplotlib libraries may be used. Direct commands for algorithms to be implemented are not allowed.
c) Submit the code in the form of Python files only. Submission should be zipped with the file name ENTRY NUMBER.zip (e.g., 2023AIB2080.zip) and uploaded to Moodle/given link.

1 Implementing ResNet 2.5 Marks

Residual Networks (ResNet) present a very simple idea to introduce identity mappings via residual connections. They are shown to significantly improve the quality of training (and generalization) in deeper networks. Before starting this part of the assignment, you should thoroughly read the ResNet paper link. We will experiment with a dataset on Butterfly & Moths species classification.

1.1 Image Classification using Residual Network

This sub-part will implement ResNet for Image Classification.
Dataset : Link, the data is divided into train/val folders and a dummy test folder to help you create your submission files. The *MASKS* directory contains ground-truth segmentation masks for val and dummy test set. We have 12K training images, 500 validation and 100 dummy test images.

1. You will implement a ResNet architecture from scratch in PyTorch. Assume that the total number of layers in the network is given by 6n+2. This includes the first hidden (convolution) layer processing the input of

size 224×224. This is followed by n layers with feature map size 224×224, followed by n layers with feature map size 112×112, n layers with feature map size given by 56×56, and finally a fully connected output layer with r units, r being number of classes. The number of filters in the 3 sets of n hidden layers (after the first convolutional layer) are 32,64,128, respectively. There are residual connections between each block of 2 layers, except for the first convolutional layer and the output layer. All the convolutions use a filter size of 3×3 inspired by the VGG net architecture [2]. Whenever down-sampling, we use the convolutional layer with stride of 2. Appropriate zero padding is done at each layer so that there is no change in size due to boundary effects. The final hidden layer does a mean pool over all the features before feeding into the output layer. Refer to Section 4.2 in the ResNet paper for more details of the architecture. Your program should take n as input. It should also take r as input denoting the total number of classes.

2. Train a ResNet architecture with n=2 as described above on the given dataset. For dataset, r = 100. Decay or schedule the learning rate as appropriate. Feel free to experiment with different your choice of optimizer, lr-scheduler, data-augmentation strategies.

2 Free lunch for Image Segmentation 2.5 Marks

In this part, we'll utilize the model trained for image-classification to perform image-segmentation without additional training. We start with the hypothesis that in order to classify different kinds of Butterfly and Moth species the model need to clearly distinguish between foreground and background of the image. We'll base our experiment on this hypothesis.

Read about gradient based image feature importance methods (GRAD-CAM, Guided-Backprop) [1, 3]. Specifically use the gradient of an image w.r.t to it's predicted logit to compute an importance map over an image. You're free to experiment with different thresholding methods to binarize your image into a segmentation mask. You'll be judged on the IOU(Intersection-over-union) score between your predicted masks and ground-truth masks. You can verify your implementation with the gt-masks provided in the dataset.

3 Submission Guidelines

Please follow the submission guidelines carefully to ensure your work is evaluated correctly:

- Submit a `train.py` script that trains the ResNet model from scratch and saves the trained model in the current directory.
- Submit an `evaluate.py` script that:

    - Loads the trained model.
    - Takes the path to test images as input.
    - Outputs the classification accuracy.
    - Generates and saves segmentation maps for the test images.

- We will run the IOU computation separately using the provided ground-truth masks and your predicted masks.

References

[1] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[3] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net, 2015.