February-June 2023 Semester
CS671: Deep Learning and Application
Programming Assignment 5

Date: April 18, 2023
Deadline for submission of code and report: Sunday, April 30, 2023, 10:00 PM

The objective of this assignment is to deepen your understanding of the convolutional neural network. You are given the subset of Caltech-101 dataset for the image classification. Each group is given 5 classes. Given dataset is train-validation-test separated. The Caltech-101 dataset consists of color images with varying sizes. **You must resize the images to a size of 224 x 224**. The tasks for this assignment are as follows:

1. Develop a convolutional neural network where the input layer should take 224 x 224 3 channel image. Consider following different architectures:
   i. **Architecture-1 (4-layer architecture)**: First convolutional layer contains 8 11x11 filters with stride 4 and padding 0 followed by 3x3 max pooling operation with stride 2. The second convolutional layer contains 16 5x5 filters with stride 1 and padding 0 followed by 3x3 max pooling operation with stride 2. Then flatten it and add two fully connected layers. Consider 128 hidden nodes with rectified linear activation function in the first hidden layer and 5 neurons with a softmax activation function in the second layer (output layer).
   ii. **Architecture-2 (5-layer architecture)**: The first two convolutional layers are the same as Architecture-1. The third convolutional layer contains 32 3x3 filters with stride 1 and padding 0 followed by 3x3 max pooling operation with stride 2. Then flatten it and add two fully connected layers. Consider 128 hidden nodes with rectified linear activation function in the first hidden layer and 5 neurons with a softmax activation function in the second layer (output layer).
   iii. **Architecture-3 (6-layer architecture)**: The first two convolutional layers are the same as Architecture-1. The third convolutional layer contains 32 3x3 filters with stride 1 and padding 0 (no max pooling). The fourth convolutional layer contains 64 3x3 filters with stride 1 and padding 0 followed by 3x3 max pooling operation with stride 2. Then flatten it and add two fully connected layers. Consider 128 hidden nodes with rectified linear activation function in the first hidden layer and 5 neurons with a softmax activation function in the second layer (output layer).
   a. Observe the classification accuracy for training and validation data for each of the architectures. Present the accuracy with confusion matrix. Clearly describe the architectures about the size of the resulting feature maps at each layer etc.
   b. Choose the best architecture based on the validation accuracy. Present the accuracy and confusion matrix for that best architecture on test data.
   c. Consider one image from the training set (from any class). Plot all 8 feature maps from the first convolutional layers and a selected 8 feature maps from the remaining convolutional layers of the best architecture. Write your observations clearly.
   d. Consider one image from the training set of each of the classes. Pass each image to CNN with the best architecture. Find out a neuron in the last convolutional layer (for each image) that is maximally activated. Trace back to the patch in the image which

causes these neurons to fire. Visualize the patches in each of the images which maximally activate that neuron.

2. Leverage Tensorflow Keras API, use VGG19 pretrained on ImageNet. Modify the classification layer of VGG19 to 5 output nodes. Retrain only the classification layer.
    a. Observe the classification accuracy for training, validation, and test data. Present the accuracy with confusion matrix and compare with that of the best performance obtained in Qn. 1b.
    b. Consider one image from the training set of each of the classes (same images as in Qn 1d). Pass each image to CNN. Find out a neuron in the last convolutional layer (for each image) that is maximally activated. Trace back to the patch in the image which causes these neurons to fire. Visualize the patches in each of the images which maximally activate that neuron.
    c. Leverage Tensorflow Keras API and visualize the influence of input pixels on any of the 5 neurons in the last convolutional layer of a pretrained VGG19 network after passing the same image considered in Qn. 1c. Consider guided-backpropagation algorithm to find the influence. In the report give the selected image, position of each of the 5 neurons in last convolutional layer and resultant gradient images. Write your observations clearly.
    d. Leverage Tensorflow Keras API and visualize localization map (heat map) using the GradCAM (https://keras.io/examples/vision/grad_cam/). Consider one image from each of the classes (same images as in Qn 1d). For each image visualize the localization map (heat map) that highlights the important regions in that image for predicting its class and other classes (Total 25 images i.e., 5 images for each of the images)

Each group of students must use the dataset identified for that group only.

You can use deep learning APIs (Tensorflow, PyTorch, Keras, etc.).

Report should be in PDF form and report by a team should also include the observations about the results of studies.

Instruction:

Upload in Moodle all your codes in a single zip file. Note that code(s) should be in a .py file, if you are coding in Python.

- **Give the name of the code folder as Group<*number*>_Assignment5_code**
  Example: Group01_Assignment5_code.
- **Give the name of the zip file as Group<*number*>_Assignment5_code.zip**
  Example: Group01_Assignment5_code.zip

Upload the report as PDF file.

- **Give the name to the report file as Group<*number*>_Assignment5_report.pdf**
  Example: Group01_Assignment5_report.pdf

We will not accept the submission if you don't follow the above instructions.