EE5178: Modern Computer Vision

Programming Assignment 1 - Basics of MLP and CNN

September 2, 2022

Notes:

1. Please use moodle discussion threads for posting your doubts.
2. Before posting any question, check if the same question has been asked earlier.
3. Submit a single zip file in the moodle named as `PA1_Rollno.zip` containing the report and folders containing corresponding codes.
4. Read the problem fully to understand the whole procedure.
5. Comment your code generously.
6. Put titles to all of the figures shown.
7. You are supposed to use `Python` and `Pytorch` for this assignment.

The aim of this assignment is to experiment Multilayer Feedforward Neural Network (MLP) and Convolutional Neural Networks (CNN) for image classification. Please use `Pytorch`, an open source library for implementing deep learning models, to write your code. You can use any other libraries that you may feel are necessary.

Dataset: The aim of this assignment is to implement an image classifier on the popular CIFAR-10 dataset. This dataset contains 60000 32×32 color images in 10 different classes, with 6000 images per class. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. It is a subset of a larger 80 million tiny images dataset.
Note: `Pytorch` provides CIFAR-10 dataset in the `torchvision.datasets` module, which you can use directly to load the dataset.

1 MLP

For this part of the assignment, we will implement a simple MLP baseline for image classification on CIFAR-10 dataset.

1.1 Architecture and Training

Input to the network is an image ($32 \times 32 \times 3$) flattened as 3072 dimensional vector x_i. Input is followed by layers $h_1(500)$, $h_2(250)$, $h_3(100)$. The number of units are mentioned beside the corresponding hidden layer. Use ReLU activation for all these hidden layers. The output unit consists of 10 units, one for each class. Let the output activation be linear. Since you want to do classification, use softmax to get probabilities of image belonging to 10 classes. This will be a 10 dimensional vector $\hat{y}_i$. Now use cross-entropy loss between $\hat{y}_i$ and y_i, where y_i is the one hot representation of the ground-truth label.

1.2 Deliverables

1. For the experiment above, you are required to show the plot of training error, validation error and prediction accuracy as the training progresses.

2. At the end of training, report the average prediction accuracy for the whole test set of 10000 images.

3. You should also plot randomly selected test images showing the true class label as well as predicted class label.

4. Report the confusion matrix that shows the kind of errors that your classifier makes. In this problem, your confusion matrix is a 10×10 matrix, where the rows represent the true label of a test sample and the columns represent the predicted labels of the classifier.

5. Use batch-normalization. Does it improve the test accuracy? Does it affect training time?

2 CNN

For this part of the assignment, we will implement VGG11 architecture for image classification from scratch. The architecture of VGG11 is described in the below image. Note that max-pooling layers reduce the image/feature size by half. All the experiments should be performed on CIFAR-10 datasets. Please follow the link `https://www.binarystudy.com/2021/09/how-to-load-preprocess-visualize-CIFAR-10-and-CIFAR-100.html` for the dataset exploration and visualization.

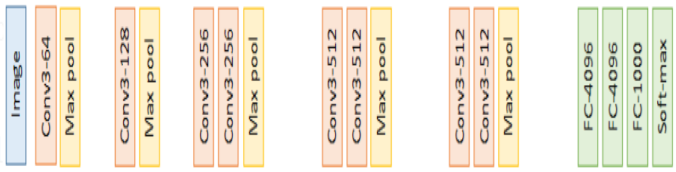

Figure 1: VGG11 architecture `https://arxiv.org/pdf/1409.1556.pdf`

2.1 Experimental Settings

Since the dataset contains 10 classes, modify (or add an extra layer) the last layer of VGG11. You are free to experiment with different settings of learning rates(such as 0.001,0.0001) and batch size for the training and report the set of hyperparameters, which resulted in the best performance. Use CrossEntropy loss and SGD optimizer for all of the experiments.
You are also free to change and play with the fully connected layers' neurons (i.e., increase or decrease the no. of neurons in a layer or increase or decrease the fully connected layers' count). For some experiments, you may also remove the last max-pooling layer (before the fully connected layers). All the experiments must be documented in the final report irrespective of the performance of a particular experiment.

Report the result with

1. Training setting (such as learning rate and training epoch used). Report this for all the different settings that you experimented with.
2. Any bottleneck/challenges you faced during training or testing.
3. Training loss/accuracy plot (loss/accuracy versus epoch)
4. Report the accuracy of the test dataset.
5. Show visual results for 5 images for each category of test classes.
6. Now download at least 5 random images (or capture images from your smartphone, camera, or any imaging device) of appropriate classes and test those images on the trained network. What is your observation?