Middle East Technical University
Electrical-Electronics Engineering Department

EE 583 Pattern Recognition

Homework 7

Due Date: 17.01.2021, 23:55 via odtuclass.metu.edu.tr

This homework is an online application of `PyTorch` library on Google Colaboratory, by the help of provided codes written in Python language. You can find more information about the syntax and functions at `https://pytorch.org/docs/stable`. Although not required, if you are interested, you can also check the tutorials at `https://pytorch.org/tutorials`.

In order to work in Google Colaboratory, you should use your Google Account. If you do not have a Google Account, you can create an account from here.

Click the following Google Colaboratory project link (prepared by Assist. Aybora Köksal aybora@metu.edu.tr). Click to `Copy to Drive` in order to save file on your own local/personal drive, to make permanent changes. You can copy the main project more than once, since in each part you will make changes to original structure. You should either keep or discard changes from previous parts, depending on their success.

Before start, check your code is running on GPU. Therefore, follow the options `Runtime` -> `Check runtime type` -> `Hardware accelerator` -> `Select GPU` -> `Save`. Also note that at every question, you should start running the code from beginning (initial random network) to make a better comparison.

In this work, you will train a Convolutional Neural Network (CNN) with CIFAR-10 dataset.

1) Click `Runtime -> Run all`. Alternatively, you can run cell by cell with clicking "Play" buttons at the left of each cell. Examine the network structure, learning rate, optimizer, loss function. Plot the learning curve for training data and find validation accuracy for the last epoch.
2) Change the optimizer to *Stochastic Gradient Descent* (SGD). Plot the learning curve for training data and find validation accuracy and loss values for the last epoch. Compare results. **Hint:** You might need to tune learning rate so that the learning curve can converge. Using momentum may produce better results.
3) Change all the activation function in the network from *ReLU* to *Sigmoid*. Plot the learning curve for training data and find validation accuracy and loss values for the last epoch. Compare results.
4) Change the kernel size of the convolutional layers to 6. Plot the learning curve for training data and find validation accuracy and loss values for the last epoch. Comment on the results. (You are required to change the input size of first fully connected layer in '`init`' function and flattened intermediate output in '`forward`' function accordingly.)
5) Remove *Max-Pooling* layers. Plot the learning curve for the training data and find validation accuracy and loss values for the last epoch. Discuss on the results. (You are required to change the input size of first fully connected layer in '`init`' function and flattened intermediate output in '`forward`' function accordingly.)
6) Add an additional convolutional layer to the model. Specify the output size and the kernel size. Plot the learning curve for training data and find validation accuracy and loss values for the last epoch. Compare results. (You should change the input size of first fully connected layer in '`init`' function and flattened intermediate output in '`forward`' function accordingly.)
7) Increase the output size of first convolutional layer to 32 and second one to 64. Accordingly, increase the output size of first FCL to 1024 and second to 256. Plot the learning curve for training data and find validation accuracy and loss values for the last epoch. Comment on the results. (You are still required to change the input size of first fully connected layer in '`init`' function and flattened intermediate output in '`forward`' function accordingly.)
8) Combine the promising methods from each of the previous parts with your own selection of number of layers, kernel sizes, output channel sizes, learning rate, optimizer, etc. to get the best result. Propose your final architecture and present your performance to get the highest grade!