CAP 4410: Computer Vision Programming Assignments 3:
Hyperparameter optimization
Due Date: Thursday, November 18, 11:59 pm
Submission Type: Upload onto Canvas

Problem Statement: This problem is related to the notebook entitled, CAP_4410_Lecture_16_ Convolutional Neural Networks.ipynb.

Given the LeNet5 as defined in this notebook, try the following variations. For each variation, start from the choices used in the code in the notebook and find the best option among the options explored. Decide on the best choice as per test accuracy.

Hyperparameters to vary:

1. Different optimization functions (https://pytorch.org/docs/stable/optim.html):
    a. SGD, Adam, RMSProp
2. Batch_size = 16, 32, 64, 256
3. Convolution kernel sizes: 3, 5, 7
4. Number of output channels: (4, 12), (6, 16), (8, 20)
5. Pooling: average pooling, maximum pooling.
6. Activation function: softmax, ReLU, LeakyReLU
7. Number of convolution layer: 1, 2, 3
8. Number of fully connected layers: 1, 2, 3
9. (Learning rate, #epochs): (0.00001, 1000), (0.0001, 500), (0.001, 250), (0.01, 100), (0.1, 50)
10. Initializations: (i) default, (ii) xavier_*normal*_, and (iii) kaiming_*normal*_

Submission Requirements:

Please upload a ZIP file containing the following files:

1. All code files to reproduce your results, including plots.
2. A maximum 5-page technical report containing the following sections:
    - Pseudo-code for any code you have written

      (See https://link.springer.com/content/pdf/bbm%3A978-1-4471-5173-9%2F1.pdf for pseudo code conventions to use.)
    - Show the learning process by plotting the train and test accuracy using at most 11 figures. Create one figure for each of the above variations. In the 11-th figure, plot the best case for each of the nine cases above. Give some thought on how to plot the curves so that they are distinguishable. Label the axes and use legends.
    - Report the hyperparameter that results in the best test performance.
    - Show one plot of a learning process that results in overfitting. One way to spot overfitting is that the final test and train accuracies differ by a large amount.

- 4. Display the activations (outputs) of the first and second layer of LeNet for different inputs (e.g., sweaters and coats)..

3. Your code will be tested to see and compared for plagiarism. A demo session could be required if there is problems with the code you submit.

Grading:

Assignment will be graded out of 100:

1. Code (out of 20): quality of coding, readability, understandability (comments, variable names, etc.)

2. Correctness (out of 20): Will compare function outputs with different setting and parameters and test against instructor's code outputs.

3. Submission Requirements Met (out of 10): Can you follow the submission guidelines? Did you turn in your deliverables as instructed with proper naming conventions and packaging requirements as the instructor asked?

4. Report (out of 50): To get top grades on the reports, you must have professional IEEE or CVPR formatted reports which are dual column, include good example images, mathematical formulas, etc. You can write them in any word processor but for the best results, check out Overleaf.com which uses LaTeX (lay-tech) but you do not need to use it. Your report must be in the zip submission as a PDF, report.pdf. Make sure to spend time on this, it is worth half your grade and should be taken seriously.

Solutions to your **programming assignments** must be self-sufficient and **not dependent on other computer vision code** other than image read and write and helper functions from pytorch. You may use packages for display graphics or mathematics packages, such as for linear algebra (numpy or scikit).

All reuse of code must be clearly acknowledged in the source code, any README files, and also in the report. Failure to do so will be considered plagiarism.