Assignment 5

COS 470/570: Image Processing and Computer Vision

Objective:

The purpose of this assignment is to enhance your understanding of Convolutional Neural Networks (CNNs) through the practical implementation and analysis of the AlexNet architecture using PyTorch. By building, training, and evaluating this model, you will gain deeper insights into its structure and performance on image classification tasks.

Background:

AlexNet is a pioneering convolutional neural network (CNN) that significantly impacted the field of deep learning, particularly in the domain of computer vision. Developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, AlexNet was introduced in 2012 at the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It outperformed the second-place competitor by a substantial margin, reducing the top-5 error rate from 25.8% to 16.4%, a groundbreaking achievement at the time.

References:
https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Tasks:

Task 1: Understand CIFAR-10.

1.1 Document your understanding of the CIFAR-10 dataset. Refer to this link (https://www.cs.toronto.edu/~kriz/cifar.html) and outline what CIFAR-10 is, including the number of images, how they are divided into training and test sets, and the categories available.

1.2 Implement a data loader for the CIFAR-10 using PyTorch's pre-built data classes. You can find more information and guidelines from this link:

https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

Task 2: Implementing AlexNet using PyTorch. Study the structure of AlexNet provided in the table below and implement it as a Python class using PyTorch. Your class should represent the AlexNet neural network's architecture.

Layer	Type	Filter/Stride	Output Size	Description
Input	-	-	227x227x3	RGB image input
1	Convolution	11x11/4	55x55x96	96 filters, stride of 4, padding 0
2	Activation	-	55x55x96	ReLU
3	Max Pooling	3x3/2	27x27x96	Pooling with a 3x3 window, stride of 2
4	Convolution	5x5/1	27x27x256	256 filters, stride of 1, padding of 2
5	Activation	-	27x27x256	ReLU
6	Max Pooling	3x3/2	13x13x256	Pooling with a 3x3 window, stride of 2
7	Convolution	3x3/1	13x13x384	384 filters, stride of 1, padding of 1
8	Activation	-	13x13x384	ReLU
9	Convolution	3x3/1	13x13x384	384 filters, stride of 1, padding of 1
10	Activation	-	13x13x384	ReLU
11	Convolution	3x3/1	13x13x256	256 filters, stride of 1, padding of 1
12	Activation	-	13x13x256	ReLU
13	Max Pooling	3x3/2	6x6x256	Pooling with a 3x3 window, stride of 2
14	Fully Connected	-	4096	Flatten and connect to 4096 neurons
15	Activation	-	4096	ReLU
16	Dropout	-	4096	Dropout applied during training to reduce overfitting
17	Fully Connected	-	4096	Connect to another 4096 neurons
18	Activation	-	4096	ReLU
19	Dropout	-	4096	Dropout applied during training to reduce overfitting
20	Fully Connected	-	Number of classes	Output layer, 10 for cifar-10 classification

In AlexNet, the 'dropout' technique is utilized. You can find more details at:

https://d2l.ai/chapter_multilayer-perceptrons/dropout.html

https://pytorch.org/docs/stable/generated/torch.nn.Dropout.html

Task 3: Model training and evaluation.

3.1 Train your AlexNet model on the CIFAR-10 classification task.

3.2 Experiment with different hyperparameters and document your findings. Use the table provided below as a baseline for your experiments (I have provided my initial configuration, which I did not fine-tune. Please use it as a baseline and attempt to improve upon these settings). Aim to improve on the baseline settings and try at least three different configurations.

Learning rate	Optimizer	Batch size	Epoch	Best_test_accuracy	Best_training_accuracy
0.001	Adam	64	10	66%	71%

Submission:

Submit your Python scripts along with a report detailing your code and approach. Ensure your report provides enough evidence (screenshots) to prove that your code is functional. The deadline of this assignment is Nov 25 at 11:59 PM.