Assignment 5: Deep Learning

Overview

This project is an introduction to deep learning tools for computer vision. You will design and train deep convolutional networks for scene recognition using **PyTorch**, an open source deep learnig platform.

Remember Assignment 3: Scene recognition with bag of words? You worked hard to design a bag of features representations that achieved 60 to 70% accuracy (most likely) on 15-way scene classification. You might have done the spatial pyramid extra credit and gotten up to nearly 80% accuracy. We're going to attack the same task with deep learning and get higher accuracy. Sort of -- training from scratch won't work quite as well as assignment 3, fine-tuning an existing network will work much better than assignment 3.

In Part 0 of the project you will train a simple network from scratch. You will run the code that is already provided (you do not need to write any code for this part) to train a simple network. In your report, report the performance of this simple network. This section will train a simple network that achieves only 30% to 35% accuracy (just slightly better than the tiny images baseline in assignment 3).

In Part 1, you will modify the dataloader used in Part 0 to include a few extra pre-processing steps. Pre-processing data allows the network to be more robust against small variations in the data. Examples of pre-processing transforms are jittering, flipping, and normalization (some of which are already called for you in the starter code). In Part 1 you will also modify the simple network by adding dropout, batch normalizations and more layers. These modifications might increase recognition accuracy to around 55%. Unfortunately, we only have 1,500 training examples so it doesn't seem possible to train a network from scratch which outperforms hand-crafted features (extra credit to anyone who proves us wrong).

For Part 2, you will instead *fine-tune* a pre-trained deep network to achieve more than 80% accuracy on the task. We will use the pretrained AlexNet network which was not trained to recognize scenes at all.

These different approaches (starting the training from scratch or fine-tuning) represent the most common approach to recognition problems in computer vision today -- train a deep network from scratch if you have enough data (it's not

always obvious whether or not you do), and if you cannot then fine-tune a pre-trained network instead.

Starter Code Outline

The following is an outline of the student code:

- `create_datasets(...)`: Creates training and testing data loaders for 15 scene database. The data loaders include pre-processing steps.
- `create_part2_model(...)`: Modifies the passed in network for fine-tuning.
- `SimpleNet`: The simple network.
  - `__init__()` : Sets up and initializes the layers of the simple netowrk.
  - `forward()`: One forward pass of the network. You do not need to modify this function.

Rubric

- +70 pts: Part 1: Build a convolutional network with pre-processing on the input data (jittering, normalization). Also add dropout regularization, batch normalization, and at least one additional convolutional layer which achieves at least 50% test accuracy (for any training epoch) on the 15 scene database.
- +30 pts: Part 2: Fine-tune AlexNet to achieve at least 80% test accuracy on the 15 scene database.
- -10 pts: Bad code style.