

Codelet 6: Convolution using PyTorch

The due date for this codelet is **Wednesday, Mar 12 at 11:59PM**.

Introduction

The aim of this codelet is to expand on your PyTorch skills and to give you practice using a kernel for basic image processing. You should draw on Chapter 7 of the textbook, the PyTorch documentation, and all of your prior codelets (though you will not be training a model here).

Important: Only use basic PyTorch operations, numpy, pandas, matplotlib, PIL, and in-built Python for this codelet. The use of any other libraries, including the books d2l library results in an automatic unsatisfactory grade for this assignment. A core goal of this class is to build your competencies as a machine learning engineer. I want to minimize abstractions from other libraries so that you build these skills.

Outline

- ☐ Grading
- ☐ Image Processing using Convolution
 - ☐ Loading Image Data
 - ☐ Convolution Implementation
 - ☐ Applying Kernels

Your assignment

Your task is to:

1. Download `codelet6.zip` from the course website and open it. You will find these instructions, a folder `data` which contains an image, `utils.py`, which includes basic image processing code, and `codelet6.py`, which has scaffolding for you.
2. Complete each of the 3 broad tasks below in `codelet6.py` and include a file called `codelet6.pdf` with your answers to the written questions.

Grading

When assessing your work, satisfactory achievement is demonstrated, in part, by:

- Simple, minimal, and clear code
- Code utilizes PyTorch methods in a way that simplifies your code
- Code passes relevant tests
- Explanation extends beyond repeating the textbook/material online
- Response are concrete and clearly demonstrate an understanding of the concepts

Convolution

You will complete an implementation of a cross-correlation (convolution) using PyTorch. By the end, you should be able to apply custom kernels to images of your choosing. You have three tasks, of varying difficulty:

1. Orient yourself to image data
2. Implement convolution
3. Apply kernels to images

Loading Image Data

In this codelet, the aim is to apply kernels to images. As a first step, we need to load images to PyTorch tensors and display tensors as images. Simple functions to accomplish these tasks are provided in `utils.py`.

There is nothing for you to add here. To evidence completion of this task, and to check your own understanding, you should load and plot the `mountain.png` image in `data`. Add to the pdf accompanying your code the plotted image.

Convolution Implementation

Some scaffolding for your convolution method is provided. This is the more challenging part of this codelet, so leave time to dig into this. Make use of discord, office hours, and TA hours if you get stuck (e.g., you spend more than 1 hour on a single issue without making any progress). Your function should return the matrix that results from applying a kernel with a step size to the image (which is a matrix). Your code **should not use PyTorch's convolution implementations**. That is, it should be convolution from scratch using basic PyTorch operations. The use of other implementations or abstractions beyond basic operations will result in an automatic unsatisfactory mark for this codelet.

You should add at **least two** doctests to your method demonstrating that your code works for two different kernels with two different step sizes for the following small image matrix:

$$\begin{pmatrix} 3 & 1 & 2 & 4 \\ 2 & 2 & 4 & 0 \\ 3 & 2 & 2 & 1 \\ 3 & 4 & 4 & 0 \end{pmatrix} \quad (1)$$

You should apply your kernel to this matrix by hand, so you know your result is correct. Additionally, you should trace your code for **one** of your doctest examples. In whichever format you prefer, you should include all of the following information in your trace

- The executed lines (one by one)
- A description of what the line does (if a line is repeated, you only have to label it once)
- The result of the line (e.g., the value in a variable)

To evidence completion of this task, you should add to the pdf accompanying your code the trace of your code. Additionally, you should apply the simple edge detection kernel from class, copied below as (2), to the mountain image. Provide a copy of plot of the modified image to the pdf accompanying your code.

$$\begin{pmatrix} -1 & 1 \end{pmatrix} \quad (2)$$

Applying Kernels

The finally component of this codelet is applying kernels to images of your choosing. Minimally, you should find two kernels (that we haven't yet used in class or in this codelet) online that are commonly used in image processing and apply them to two images of your choosing.

To evidence completion of this task, you should add to the pdf accompanying your code the **two kernels** you used, the **two original images**, and the **transformed images** (with the images as plots). Additionally you should describe, at a high level, what the kernels you found do.