CELL0027 Week 9 Assignment

March 6, 2024

We will consider some basic operations in (bio)image analysis, and the use of machine learning (ML) for a simple regression task. You have been given a dataset of 250 single channel microscopy images of cell nuclei, corresponding to four stages of mitosis plus apoptosis.

In this assignment you will analyse the image data and create a rudimentary image segmentation method. Then, using the PYTORCH ML framework, you will implement a convolutional operator to perform edge detection in the images (see Fig 1). Using this as a basis, you will then implement an ML training loop that attempts to learn the edge detection features directly from the image data in a supervised learning approach. We will make extensive use of the 2D convolution operation implemented in PYTORCH[1]:

$$\text{out}(N_i, C_{out_j}) = \text{bias}(C_{out_j}) + \sum_{k=0}^{C_{in}-1} \text{weight}(C_{out_j}, k) \star \text{input}(N_i, k) \tag{1}$$

Where the input size of the images is (N, C_{in}, H, W) and the output size is (N, C_{out}, H, W). The dimensions of the image are (H, W), which is $(80, 80)$ for this dataset. N is the batch size, or number of images analysed in parallel. The $\star$ operator denotes a valid 2D cross-correlation. Since we only have one channel for our input data, and we only want to calculate the convolution with a single filter $C_{in} = C_{out} = 1$. For the purposes of today, the bias term is ignored, *i.e.* $\text{bias} : x \to 0$.

You will need to use and adapt the Python code in the Jupyter notebook that we worked on during the tutorial[2].

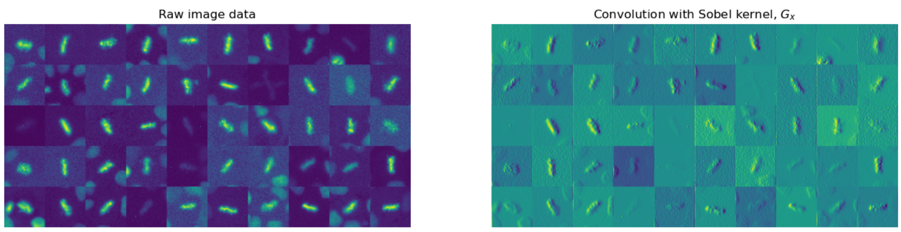

Figure 1: Performing edge detection using the PYTORCH framework.

Exercises

1. Calculate intensity histograms for each class within the dataset. Are there systematic differences between the classes?

2. Use the intensity histograms to calculate a threshold which could be used to näively segment the images into foreground and background.

3. Write a simple Python function to implement the segmentation:

$$y = \begin{cases} 0, x \leq \text{threshold} \\ 1, x > \text{threshold} \end{cases} \tag{2}$$

   How well does this segmentation function work, in practice? Can you think of a simple way to improve it?

[1] https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html
[2] https://github.com/lowe-lab-ucl/CELL0027

4. In the lecture, we defined two convolutional kernels that could be used to calculate image gradients (also known as Sobel filters). Together these can be used to implement an edge detection algorithm. Using the Python code provided, implement the G_x and/or G_y kernels in PYTORCH.

   Extra: try to compute both the magnitude and direction of the gradients in the image using the results obtained by applying both kernels separately to the image.

5. Using the Jupyter notebook provided, try building and training a simple model that learns how to perform the Sobel filtering. The concept is to use pairs of images (the raw image, and the Sobel filtered image) as input and ground truth for a regression model with a single convolutional layer.

   Explore the architecture of the model, the role of the various parameters and the evolution of the loss during training. Plot the loss as a function of the training iteration. Can you think of a way to measure how well the learned kernel approximates the real kernel?

   Extra: can you modify the code and model to learn to compute the full magnitude and direction of the gradients in the image?