Convolutional Neural Networks

Sergei Sotnikov

July 2022

1 Introduction

Image classification is one of the central challenges in computer vision. There have been numerous algorithms enabling computers to distinguish between objects based on visual data. The applications range from medical diagnosis to self-driving vehicles.

The state of the art in image classification are Convolutional Neural Networks (CNNs) which, when tested on datasets such as CIFAR-10, achieve precision higher than humans do on average (Source 3). This obviously doesn't mean that computer vision is anywhere near the abilities of human eye and brain, however, it does mean that CNNs can be utilized to deal with certain tasks where the price of human error is too high, or the nature of the task is of little intellectual/creative value. Thus, there have been examples of AI algorithms identifying brain tumors on scans where doctors initially missed it (Source 3).

Given the effectiveness of CNNs and the fact that they are related to the material we covered in class, I chose to explore how CNNs work for my final project. In the project I will give an overview of what computers actually "see" when we have visual data as an input; what an image classification pipeline looks like in general; what is convolution from the mathematical standpoint; how does CNN work.

2 Image encoding

What computer "sees" when we are talking about images is a matrix with numerical values. For a grayscale image, we would have a two-dimensional matrix with numerical values representing color. For example, 0 can be black, 256 white, and 125 would be grey. While grayscale images work in some cases, we might need more color information. In such case, the image is encoded in RGB format, which is represented by a three-dimensional matrix. The first two dimensions work like grey scale matrix, where the coordinates of each cell represent the location of the pixel on the image. The third dimension, also known as channel, has depth of three and corresponds to the values of red, green or blue.

3 Pipeline

When the input in a form of matrix is received by the machine, there are several steps that are associated with a typical image classification algorithm. Pooling allows to reduce the size of the image, and therefore the amount of computation, although some information is lost in the process. There are different ways of pooling - for example, max pooling replaces the values of four adjacent pixels with the maximum value thus reducing the size of the image. Feature extraction is another important step in image classification pipeline. Usually there are lots of layers, each responsible for a specific type of feature. For example, at the beginning the features extracted would be very basic, like curves or angles. Later, more complex features can be detected. The extracted information is then passed to the neural network where the actual classification occurs.

4 Convolution

Convolution is an operation that expresses the amount of overlap between two functions as one of these functions is shifted over the other. Below is the mathematical definition of convolution:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau.$$

This formula is helpful computation-wise, however, it is hard to gain intuitive understanding of convolution based on the integral alone. An example I found helpful in visualizing and interpreting convolution is from signal processing. Let's say we have two square wave signals as on **Figure 1**. We want to see how much overlap occurs between two signals. In order to do this we first reflect signal g around the y-axis, which leads to τ being negative in the integral for g. Then we shift the signal by some time t to the left. Therefore, we have $g(t-\tau)$.

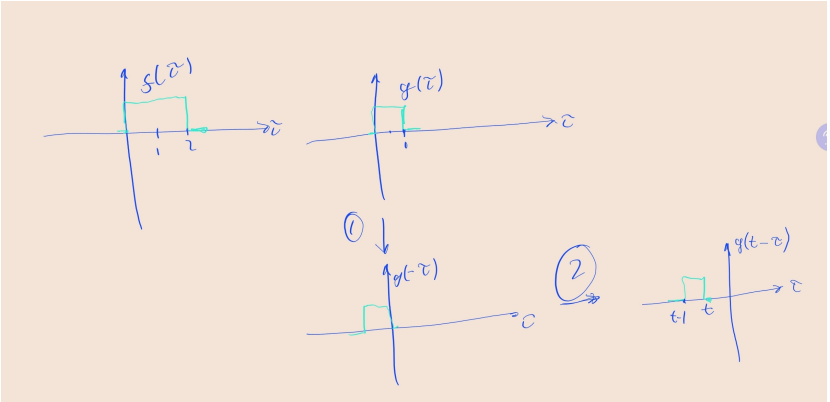

Figure 1:

Now, on **Figure 2**, we have two signals and we start moving g towards f. Initially, the product is 0. For $t > 0$ the product begins to increase as there is more overlap achieved. Then the curve has a plateau as g is traveling through f and then starts decreasing until it reaches zero again.

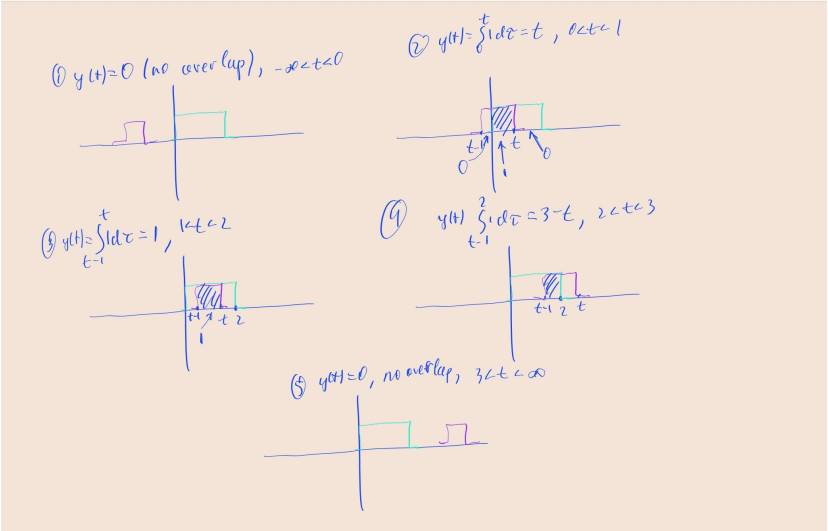

Figure 2:

The result of the convolution operation can be expressed as:

$$y(t) = \begin{cases} 0 & t \leq 0 \\ t & 0 \leq t \leq 1 \\ 1 & 1 \leq t \leq 2 \\ 3-t & 2 \leq t \leq 3 \\ 0 & 3 \leq t \leq \infty \end{cases}$$

Lastly, **Figure 3** contains a side by side representation of the signal and the convolution (Source 2).

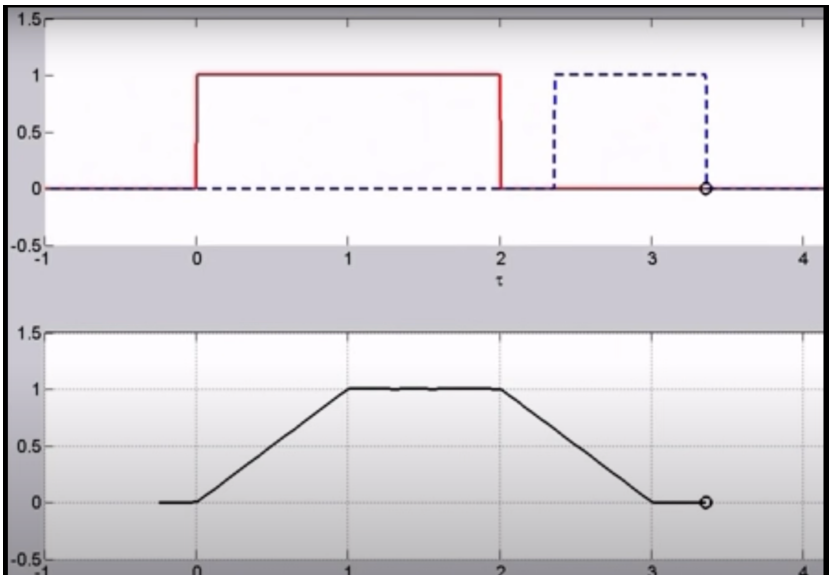

Figure 3:

5 CNN

It turns out that computing a product of two functions as one is shifted over the other lies at the core of CNNs. The way CNN works is it slides a filter over an image, iterating through the entire image, step by step. The filter is a matrix, values of which determine what kind of feature we are looking for. The element-wise product of the filter matrix and the part of the image matrix "under" the filter is taken and the value is passed to the next layer where more complex features can be extracted. **Figure 4** contains an example (Source 3).

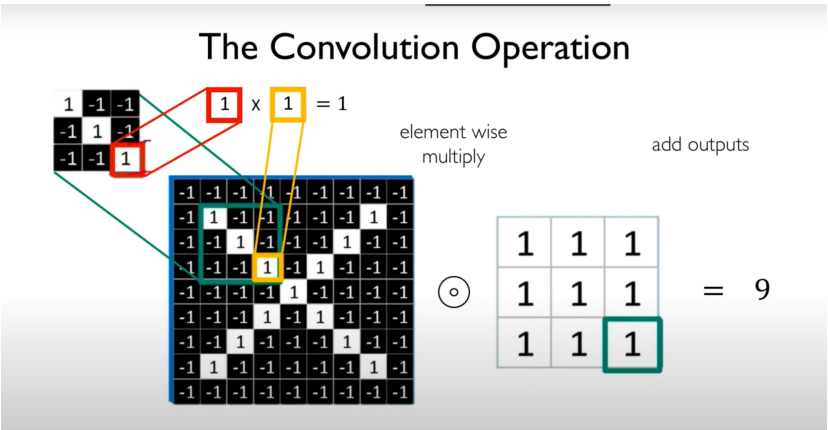

Figure 4:

On the Figure, the overall task is to detect "X" feature on an image. This task is split into sub-tasks, such as the one on the Figure. The goal of the sub-task is to find the left upper "arm" of the X. For that, we create a 3 by 3 matrix filter and slide it over the image computing the element-wise products, an instance of which is shown on the Figure. Greater value would correspond to a more pronounced feature, in this case 9 is when the filter and the image coincide precisely. As the filter slides through the image, the element-wise product reveals where the filter was activated the most, which, in turn, shows the location of a given feature on the image. The obtained data is passed to the next layer, in which the operation is repeated for a higher order feature.

6 References

- Source 1 - article on image classification.
- Source 2 - lesson on convolution and signal processing.
- Source 3 - MIT 6.S191 lecture on CNNs
- Source 4 - MIT 6.S094 lecture on CNNs