Title: study of FFT Convolutions

JX-Ma

2024/10/12

This week is mainly about learning FFT, which is an acceleration algorithm for DFT (Discrete Fourier Transform).
Fourier transform mainly converts signals from the time domain to the frequency domain, studying the spectral structure and variation laws of signals
The principle of FFT application convolution operation: Convolution operation can be achieved through direct calculation in the time domain, but it can also be achieved through multiplication in the frequency domain. The basic idea of FFT convolution is to use the Fast Fourier Transform to transform the signal (input tensor) and convolution kernel into the frequency domain, multiply them in the frequency domain, and then invert them back to the time domain to achieve convolution operation.
Next, let's introduce an example. The expression for a quadratic function is assumed to be $y = x^2 + 2x + 1$. We can assume that x is time, and the value of this function changes with time. We can understand this expression as the representation of y in the time domain. We can also use a vector [1,2,1] to represent it, or use three different points on the function to represent it, as a representation of y in the frequency domain. In the time domain, y is in motion and changes with x, while in the frequency domain, y is stationary. We can infer y's motion trajectory in the time domain based on y in the frequency domain.
Assuming we have two functions, $y1 = x^2 + 2x + 1$and $y2 = x^2 - 2x + 1$, and the product of the two functions is $Y = y1 * y2$, we can obtain the final result by directly multiplying the expressions of the y1 and y2 functions. We can also take 5 points on y1 and 5 points on y2 that are the same as the horizontal axis on y1, multiply the vertical axis of these 5 points to obtain new 5 points, and use the undetermined coefficient method or other methods to express these 5 points to Y.
The second method here is to first convert the time-domain expressions of y1 and y2 into frequency-domain expressions, then multiply these points to obtain the frequency-domain expression of Y, and finally obtain the time-domain expression of Y through a method.
When applied to convolution, we can view the process as the convolution window moving in the input tensor with step size, while the convolution kernel can also be understood as moving, except that it remains stationary as the convolution window changes. For the output tensor, it is the final result obtained by

multiplying the convolution window and convolution kernel.
Similar to function multiplication, we can also use FFT to first convert the convolution window and kernel from the time domain to the frequency domain, obtaining the representation of the output tensor in the frequency domain. Then, through inverse FFT, we can convert the frequency domain representation of the output tensor to the time domain to obtain the final result.
Taking one-dimensional convolution as an example, let's introduce the process of convolution. Assuming the input tensor is [1,2,3,4,5,6,7], the convolution kernel is [1,2,3].

1. Padding filling, fill the input tensor with [1,2,3,4,5,6,7,0,0] convolution kernels [1,2,3,0,0,0,0,0], and the size after filling is the input tensor length+convolution kernel length -1; This belongs to complete FFT, which aims to consider all elements in the input tensor. There is also an effective FFT, where the size of the generated output tensor is the normal size of a standard convolution, but there is an error at this point because we need to crop the input tensor to 5.

2. Perform FFT transformation on the convolution kernel and output tensor, that is, the conversion from time domain to frequency domain.

3. Introduce some FFT transformations and the computational complexity of DFT. There is a theorem that all periodic functions can be expanded by Fourier transform. For functions without periods, they can be defined as periodic functions ranging from negative infinity to positive infinity.