MDCorr Notes

Ryan Park

December 15, 2024

Contents

1 Correlations

There are general relationships between correlations and Fourier transforms. Especially, there is the convolution theorem. Define the convolution as

$$g(t) * h(t) \equiv \int_{-\infty}^{\infty} g(\tau)\, h(t-\tau)\, \mathrm{d}\tau. \tag{1}$$

Define the Fourier transform as

$$H(f) \equiv \mathcal{F}\{h(t)\}(f) = \int_{-\infty}^{\infty} h(t)\, e^{2\pi i f t}\, \mathrm{d}t \tag{2}$$

and the inverse,

$$h(t) = \mathcal{F}^{-1}\{H(f)\} = \int_{-\infty}^{\infty} H(f)\, e^{-2\pi i f t}\, \mathrm{d}f \tag{3}$$

The convolution theorem states:

$$\mathcal{F}\{g * h\} = G(f)\, H(f) \tag{4}$$

The correlation of two functions may be written as,

$$\mathrm{Corr}(g, h) = g(t) * h(-t) \tag{5}$$

If both g and h are real, then

$$\begin{aligned} \mathrm{Corr}(g, h) &= G(f)\, H(f)^{*} \\ \mathrm{Corr}(g, h) &= \mathcal{F}^{-1}\{\mathcal{F}\{g\}\, \mathcal{F}\{h\}^{*}\} \end{aligned} \tag{6}$$

If the correlation is with the same real function [1], i.e., it is a real autocorrelation, then the solution takes a simple form

$$\mathrm{Corr}(g, g) = \mathcal{F}^{-1}\left\{|G|^{2}\right\}. \tag{7}$$

This is known as the “Wiener-Khinchin Theorem”.

2 Discretization

The Discrete Fourier Transform (DFT) of a complex series x_n with size N is computed as

$$X_k = \sum_{n}^{N-1} x_n e^{2\pi i k n / N}, \tag{8}$$

and its inverse,

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{-2\pi ikn/N}. \tag{9}$$

There is a corresponding discrete theorem called the circular convolution theorem. Because the sum is finite, the Fourier decomposition is periodc in k, and the resulting correlations of the real space array will include contributions between the beginning and end of the input array, as if it were wrapped into a ring and convolved with itself. To exclude these contributions, i.e. to compute the linear convolution as opposed to the circular one, the input array must be padded with zeros such that the wrapping does not affect length scales longer than the original input array.

3 FFT

The DFT can be denoted,

$$\begin{aligned} \boldsymbol{X} &= \mathcal{F}_N\left\{\boldsymbol{x}\right\} \\ \boldsymbol{x} &= \mathcal{F}_N^{-1}\left\{\boldsymbol{X}\right\} \end{aligned} \tag{10}$$

where N is the size of the input and output vectors of the tranform operation. Now suppose N is divisible by some integer a. The sum can be decomposed into parts,

$$\begin{aligned} X_k &= \sum_{s=0}^{a-1}\sum_{m=0}^{N/a} x_{am+s} e^{2\pi ik(am+s)/N} \\ &= \sum_{s=0}^{a-1} e^{2\pi iks/N} \sum_{m=0}^{N/a} x_{am+s} e^{2\pi ik(am)/N} \\ &= \sum_{s=0}^{a-1} e^{2\pi iks/N} \sum_{m=0}^{N/a} x_{am+s} e^{2\pi ikm/(N/a)} \\ \mathcal{F}_N\left\{\boldsymbol{x}\right\}_k &= \sum_{s=0}^{a-1} e^{2\pi iks/N} \mathcal{F}_{N/a}\left\{\boldsymbol{x}\right\}_k \end{aligned} \tag{11}$$

Thus the Fourier transform can be decomposed into the sum of smaller Fourier transforms. This results in a computational speedup because $\mathcal{F}_{N/a}\left\{\boldsymbol{x}\right\}_k$ is periodic in k with periodicity N/a. Thus

$$\mathcal{F}_{N/a}\left\{\boldsymbol{x}\right\}_k = \mathcal{F}_{N/a}\left\{\boldsymbol{x}\right\}_{k \bmod N/a}. \tag{12}$$

Therefore

$$\mathcal{F}_N\left\{\boldsymbol{x}\right\}_k = \sum_{s=0}^{a-1} e^{2\pi iks/N} \mathcal{F}_{N/a}\left\{\boldsymbol{x}\right\}_{k \bmod N/a}. \tag{13}$$

Similarly,

$$\mathcal{F}_N^{-1}\left\{\boldsymbol{x}\right\}_n = \frac{1}{N}\sum_{s=0}^{a-1} e^{-2\pi ins/N} \mathcal{F}_{N/a}^{-1}\left\{\boldsymbol{X}\right\}_{n \bmod N/a}. \tag{14}$$

The single thread runtime of this substep l with size n_l is

$$\begin{aligned} R_l\left(n_l\right) &= a_l n_l + a_l R_{l-1}\left(n_l/a_l\right) \\ &= a_l\left(n_l + R_{l-1}\left(n_l/a_l\right)\right) \end{aligned} \tag{15}$$

This is because each $k \in 0, ..., n_s - 1$ must be evaluated. Assuming a_s R_{s+1} subproblems have already been performed, that leaves an $O\left(1\right)$ read operation from the child calculation and a butterfly operations for each k. If N has the prime factorization a_p of length P, then there will be P layers. Enumerate these layers starting from a base case $a_1, a_2, ..., a_P$.

$$\begin{aligned}
R_2 &= a_2\left(n_2 + R_1\right) \\
R_3 &= a_3\left(n_3 + a_2\left(n_2 + R_1\right)\right) \\
R_4 &= a_4\left(n_4 + a_3\left(n_3 + a_2\left(n_2 + R_1\right)\right)\right) \\
&= a_4 n_4 + a_4 a_3 n_3 + a_4 a_3 a_2 n_2 + a_4 a_3 a_2 R_1 \\
&\vdots \\
R_P = R\left(N\right) &= \sum_{l=2}^{P} n_l \prod_{p=2}^{P} a_p + R_1 \prod_{p=1}^{P} a_p \\
&= \sum_{l=2}^{P} n_l \prod_{p=l}^{P} a_p + R_1 N
\end{aligned} \tag{16}$$

The size of each layer is given by

$$n_l = \prod_{p=1}^{l-1} a_p \tag{17}$$

Thus

$$R = \sum_{l=1}^{P} a_l \prod_{p=1}^{P} a_p = N \sum_{l=1}^{P} a_l \tag{18}$$

The indices respect a similar recursive relationship. Adding level indices to indicate location of data,

$$\mathscr{F}_{s,n_l} \tag{19}$$

$$x_{n,l} = a_l x_{n,l} + s_l \tag{20}$$

In explicit form,

$$\begin{aligned}
x_{n,l-1} &= a_l x_{n,l} + s_l \\
x_{n+1,l} &= \frac{x_{n_l} - s_l}{a_l}
\end{aligned} \tag{21}$$

The remapping for the last layer index should be

$$x_j^{|a|} \leftarrow N\left(\cancel{x_i}^0\right) + \sum_{p=|a|-1}^{0} \left(\prod_{r=0}^{p-1} a_r\right) s_p\left(j\right) \tag{22}$$

But

$$s_p\left(j\right) = j \mod \prod_{r=0}^{p-1} a_r \tag{23}$$

- explicit index mapping
- complex number respresentation.
- The RDF cutoff must coordinate with the ghost mapping when the realspace cutoff is small.
- This can be fixed with the command

```
comm_modify cutoff {RDF cutoff + skin}.
```

References

[1] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing.* Cambridge University Press, USA, 1992.