Mathematics, Pusan National University

Linear Algebra and Learning from Data

IV.5 Toeplitz Matrices and Shift Invariant Filters

Taehyeong Kim
th_kim@pusan.ac.kr

December 18, 2020

Content

Toeplitz matrix

A **Toeplitz matrix** has constant diagonals. The first row and column tell you the rest of the matrix, because they contain the first entry of every diagonal.

$$A = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & a_{-3} \\ a_1 & a_0 & a_{-1} & a_{-2} \\ a_2 & a_1 & a_0 & a_{-1} \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix}$$

Circulant matrix

Circulant matrix is Toeplitz matrix that satisfy the extra “wraparound” condition that makes them periodic.

$$C = \begin{bmatrix} c_0 & c_3 & c_2 & c_1 \\ c_1 & c_0 & c_3 & c_2 \\ c_2 & c_1 & c_0 & c_3 \\ c_3 & c_2 & c_1 & c_0 \end{bmatrix}$$

Toeplitz matrices are the matrices we use in signal processing and in convolutional neural nets (CNNs). The analysis of A is based on the two-sided polynomial $A(\theta)$ with coefficients $a_{1-n}, \ldots, a_0, \ldots, a_{n-1}$:

For analysis of A

Frequency response = symbol of A	$A(\theta) = \sum a_k e^{ik\theta}$
$\mathrm{A}(\theta)$ **is real when** A **is symmetric**	$a_k e^{ik\theta} + a_k e^{-ik\theta} = 2a_k \cos k\theta$
$\mathrm{C}(\theta)$ **is nonzero when** C **is invertible**	The symbol for C^{-1} is $1/C(\theta)$

Toeplitz matrices are noncyclic convolutions with $\mathbf{a} = (a_{1-n}, \ldots, a_{n-1})$ followed by projection:

x-space	Ax	=	convolve $a * x$,then keep components 0 to $n-1$
θ-space	$Ax(\theta)$	=	multiply $A(\theta)x(\theta)$,then project back to n coefficients

Introduction

In many problems the Toeplitz matrix is **banded**. The matrix only has w diagonals above and below the main diagonal.

Tridiagonal Toeplitz Bandwidth $\mathrm{w} = 1$

$$A = \begin{bmatrix} a_0 & a_{-1} & 0 & 0 \\ a_1 & a_0 & a_{-1} & 0 \\ 0 & a_1 & a_0 & a_{-1} \\ 0 & 0 & a_1 & a_0 \end{bmatrix}$$

We understand tridiagonal Toeplitz matrices by studying $A(\theta) = a_{-1}e^{-i\theta} + a_0 + a_1 e^{i\theta}$.

Toeplitz Matrices : Basic Ideas

Fundamental idea

In signal processing, a Toeplitz matrix is a **filter**.

$$A\mathbf{x} \quad \Longrightarrow \quad A(\theta)\mathbf{x}(\theta) \quad \textit{(but not true)}$$

Linear finite difference equations with constant coefficients produce Toeplitz matrices. The equations donąŕt change as time goes forward (**LTI** : Linear Time Invariant). They donąŕt change in space (**LSI** : Linear Shift Invariant).

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \text{ with symbol } A(\theta) = -e^{-i\theta} + 2 - e^{i\theta} = 2 - 2\cos\theta.$$

Some properties of this A

- $A(\theta) \geq 0$ tells us that A is symmetric positive semidefinite or definite.
- $A(0) = 2 - 2 = 0$ tells us that $\lambda_{\min}(A)$ will approach zero as n increases.
- The finite Toeplitz matrix A is barely positive definite.
- The infinite Toeplitz matrix is singular.

The inverse of a Toeplitz matrix A is usually not Toeplitz.

Example 2.1

$$A^{-1} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}^{-1} = \frac{1}{4}\begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 3 \end{bmatrix} \text{ is not Toeplitz}$$

Levinson Durbin recursion

Levinson found a way to use the Toeplitz pattern in a recursion, reducing the usual $O(n^3)$ solution steps for $A\mathbf{x} = \mathbf{b}$ to $O(n^2)$. It is better than some superfast algorithms proposed later for moderate n.

Application

Now we're going to expand the linear algebra to the convolution by using an example of audio data analysis. Start by representing a fully connected layer in the form of a matrix.

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

But in general, audio data is much longer.

Sample Rate

A commonly seen unit of sampling rate is Hz, which stands for Hertz and means "samples per second". As an example, 48 kHz is 48,000 samples per second. Simply put, sample rate is the digital conversion of the sound we usually hear.

The number of samples in the audio data equals the length of the audio and the multiplication of the sample rate. Therefore, as such, the weighted matrix will grow.

Application

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & \cdots & w_{1n} \\ w_{21} & w_{22} & w_{23} & w_{24} & \cdots & w_{2n} \\ w_{31} & w_{32} & w_{33} & w_{34} & \cdots & w_{3n} \\ w_{41} & w_{42} & w_{43} & w_{44} & \cdots & w_{4n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

The above equation will be difficult to learn. But fortunately, there's a way to make it simple.

Data locality

An object is directly influenced only by its immediate surroundings.

Data stationarity

The mean and variance of the time series data are constant.

Application

By data locality, teh first row of our matrix becomes a kernel of size 3 denote as $\mathbf{a}^{(1)} = \begin{bmatrix} a_1^{(1)} & a_2^{(1)} & a_3^{(1)} \end{bmatrix}$.

$$\begin{bmatrix} a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & 0 & \cdots & 0 \\ w_{21} & w_{22} & w_{23} & w_{24} & \cdots & w_{2n} \\ w_{31} & w_{32} & w_{33} & w_{34} & \cdots & w_{3n} \\ w_{41} & w_{42} & w_{43} & w_{44} & \cdots & w_{4n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Application

And by data stationarity, use the kernel $\mathbf{a}^{(1)}$ that we previously defined. We use this kernel by placing it at each step each time, and at the same time, we get the following results:

$$\begin{bmatrix} a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & 0 & \cdots & 0 \\ \vdots & & \vdots & & \ddots & \ddots & \ddots & & \vdots \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{bmatrix}$$

Repeated reuse of definite kernels is called weight sharing.

Application

After this change, the remaining parameters are 3. Compared to a weighted matrix with a parameter value of 12 earlier, the current number of parameters is very limited and we would like to extend this.

By increasing the parameters, we can make convolution multiple layers using different kernels, such as $\mathbf{a}^{(2)}$ and $\mathbf{a}^{(3)}$.

Given the first kernel $\mathbf{a}^{(1)}$ and input vector $\mathbf{x}$, the first entry in the output given by this layer is $a_1^{(1)}x_1 + a_2^{(1)}x_2 + a_3^{(1)}x_3$. Therefore, the whole output vector looks like the following:

$$\begin{bmatrix} \mathbf{a}^{(1)}x[1:3] \\ \mathbf{a}^{(1)}x[2:3] \\ \mathbf{a}^{(1)}x[3:5] \\ \vdots \end{bmatrix}$$

The same matrix multiplication method can be applied on following convolutional layers with other kernels to get similar results.

Application

Now, we are going to explore Convolution as a ‘running scalar product’.

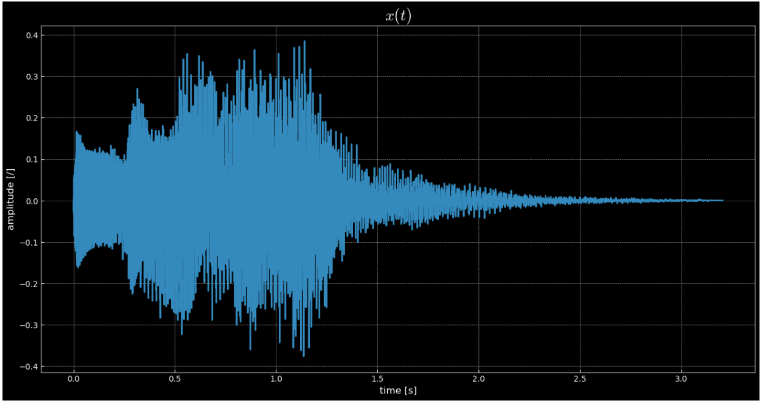

Figure 1: A visualization of the audio signal.

The audio signal $x(t)$ is actually the sound played when turning off the Windows system.

Figure 2: Notes for the above audio signal.

If we use Fourier transform (FT) all the notes would come out together and it will be hard to figure out the exact time and location of each pitch. Therefore, a localized FT is needed. It is also known as spectrogram.

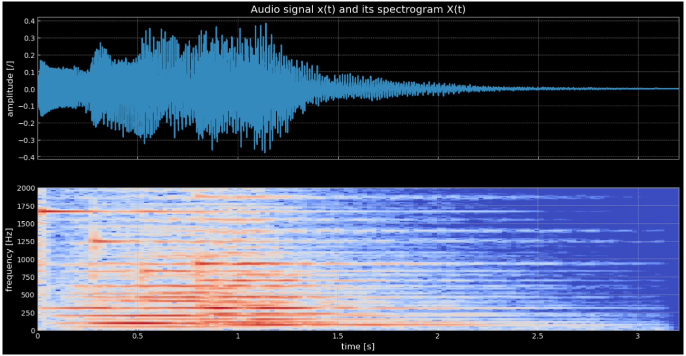

Figure 3: Audio signal and its spectrogram.

Application

Convolution of the input signal with all the pitches can help extract all notes in the input piece. The spectrograms of the original signal and the signal of the concatenated pitches is shown in Figure 4 while the frequencies of the original signal. It is also known as spectrogram.

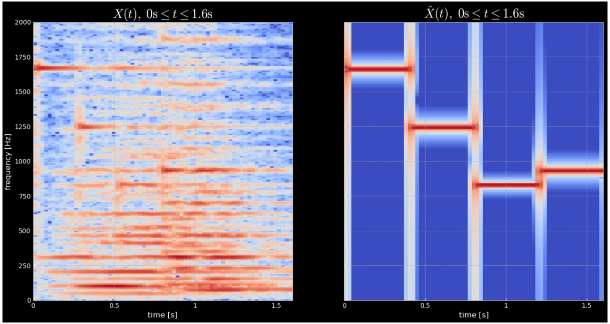

Figure 4: Spectrogram of original signal (left) and Sepctrogram of the concatenation of pitches (right).

And the four pitches is shown in Figure 5.

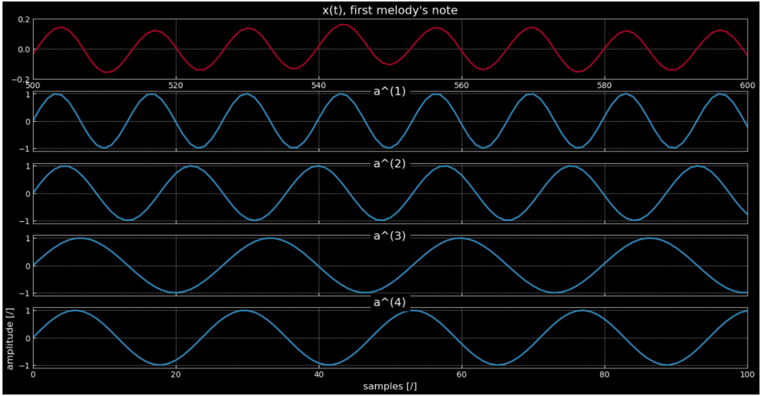

Figure 5: First note of the melody.

Application

Fig 6 along with the audio clips of the convolutions prove the effectiveness of the convolutions in extracting the notes.

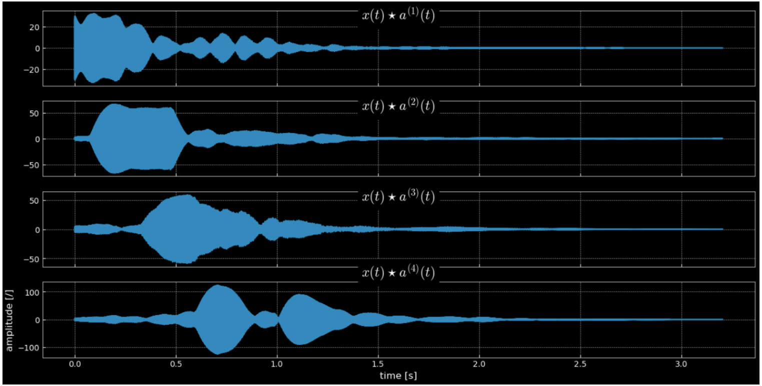

Figure 6: Convolution of four kernels.

Thank you!