Lecture 4: Mathematical preliminaries for discrete functions

Machine Learning and Imaging

BME 548L
Roarke Horstmeyer
ML+Imaging pipeline introduction

Real World

Measurement device

Digitization

γ -> e^−

Machine Learning
ML+Imaging pipeline introduction

Real World

Continuous complex fields

(last class)  (last class, this class)

Measurement device

Digitization

γ -> e⁻

Black box transformations

• Convolution
• Fourier Transform

Machine Learning
ML+Imaging pipeline introduction

Real World

Continuous complex fields

(last class)

Measurement device

Black box transformations

• Convolution
• Fourier Transform

(last class, this class)

Digitization

γ -> e^−

Sampling Theorem

Discrete math & Linear algebra

(this class)
ML+Imaging pipeline introduction

Real World

Continuous complex fields

Black box transformations
  • Convolution
  • Fourier Transform

(last class) → (last class, this class) → (this class) → (next few weeks)

Measurement device

Digitization

γ -> e⁻

Sampling Theorem

Discrete math & Linear algebra

Machine Learning

Optimization

Linear classification

Logistic classifier

Neural networks

Convolutional NN’s

ML+Imaging pipeline introduction
ML+Imaging pipeline introduction

Real World

Continuous complex fields

Measurement device

Black box transformations
- Convolution
- Fourier Transform

Digitization

\( \gamma \rightarrow e^- \)

Sampling Theorem

Discrete math &
Linear algebra

Optimization

Linear classification
Logistic classifier
Neural networks
Convolutional NN’s

Month 2

Month 3

(last class)

(last class, this class)

(this class)

(next few weeks)
Review: black box transforms as a convolution

Knowing the point-spread function, it is direct to model any output of the black box, given an input:

\[ h(x_0, y_0) = \text{Convolve} = \text{“smear and multiply”} \]

\[ U_i(x_i, y_i) \rightarrow h(x_0, y_0) \rightarrow U_o(x_0, y_0) \]
Review: black box transforms as a convolution

Knowing the point-spread function, it is direct to model any output of the black box, given an input:

\[ U_o(x_o, y_o) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_i(x_i, y_i) h(x_o - x_i, y_o - y_i) dx_i dy_i \]

Output of linear system is a convolution of the input with its point-spread function.
Review: Convolution theorem

\[ U_1(x) \ast U_2(x) = U_1(x) \ast \text{sinc}(ax) \]
**Review: Convolution theorem**

\[ f(x) * g(x) = \mathcal{F}^{-1}[\mathcal{F}(f(x)) \cdot \mathcal{F}(g(x))] \]

Where:
- \( f(x) \) is the original function
- \( g(x) \) is the other function
- \( \mathcal{F} \) is the Fourier transform
- \( \mathcal{F}^{-1} \) is the inverse Fourier transform
- \( \cdot \) denotes the pointwise multiplication

Graphically:

1. Function \( f(x) \) is convolved with \( g(x) \) to yield \( f(x) * g(x) \).
2. Each function is transformed to the frequency domain using \( \mathcal{F} \).
3. The transformed functions are multiplied pointwise to yield \( \mathcal{F}(f(x)) \cdot \mathcal{F}(g(x)) \).
4. The result is inverted back to the time domain using \( \mathcal{F}^{-1} \).

Mathematical representation:

\[ F[f(x) * g(x)] = F[f(x)] \cdot F[g(x)] \]

Using the examples:

- \( U_1(x) \)
- \( U_2(x) = \text{sinc}(ax) \)
- \( U_1(x) * U_2(x) \)

Graphically:

- \( U_1(x) \) and \( U_2(x) \) are convolved to yield \( U_1(x) * U_2(x) \).
- Their Fourier transforms \( F[U_1(x)] \) and \( F[U_2(x)] \) are multiplied.
- The product \( F[U_1(x)] \cdot F[U_2(x)] \) is then transformed back to the time domain to obtain \( U_1(x) * U_2(x) \).
Review: black box transforms as a convolution

Knowing the point-spread function, it is direct to model any output of the black box, given an input:

\[ h(x_o, y_o) \]
Knowing the point-spread function, it is direct to model any output of the black box, given an input:

\[ H(f_{xo}, f_{yo}) \]

Review: black box transforms as a convolution

- Point spread function
  \[ h(x_o, y_o) \]
- Black box transfer function
  \[ F[\cdot] \]
- Output
  \[ \hat{U}_i(f_{xi}, f_{yi}) \]
- Input
  \[ U_i(x_i, y_i) \]
Review: black box transforms as a convolution

Knowing the point-spread function, it is direct to model any output of the black box, given an input:

\[
\hat{U}_o(f_x, f_y) = \hat{U}_i(f_x, f_y)H(f_x, f_y)
\]

Can also multiply Fourier transform of input with transfer function $H$ to obtain Fourier transform of output.
Review: black box transforms as a convolution

Knowing the point-spread function, it is direct to model any output of the black box, given an input:

\[
\hat{U}_i(f_{xi}, f_{yi}) \rightarrow \hat{U}_i(f_{xi}, f_{yi}) \cdot H(f_{xo}, f_{yo}) \rightarrow \hat{U}_o(f_{xo}, f_{yo}) \rightarrow U_o(x_o, y_o)
\]

Can also multiply Fourier transform of input with transfer function $H$ to obtain Fourier transform of output.
ML+Imaging pipeline introduction

Real World

Measurement device

Digitization

γ -> e⁻

Machine Learning
The Sampling Theorem – from Goodman Section 2.4.1

\[ U_s(x, y) = \text{comb}(x/X)\text{comb}(y/Y)U(x, y) \]

Signal sampling occurs with:

- CMOS (pixel) sensors, PMTs, SPADs
- A-to-D after antennas
- A-to-D after acoustic transducers

Sampling interval width X and Y
The Sampling Theorem – from Goodman Section 2.4.1

\[ U_s(x, y) = \text{comb}(x/X)\text{comb}(y/Y)U(x, y) \]

Signal sampling occurs with:
- CMOS (pixel) sensors, PMTs, SPADs
- A-to-D after antennas
- A-to-D after acoustic transducers

Sampling interval width X and Y

\[ \hat{U}_s(f_x, f_y) = \mathcal{F} \left[ \text{comb}(x/X)\text{comb}(y/Y) \right] * \hat{U}(f_x, f_y) \]

From Convolution Theorem
The Sampling Theorem – from Goodman Section 2.4.1

\[ \hat{U}_s(f_x, f_y) = \mathcal{F} \left[ \text{comb}(x/X)\text{comb}(y/Y) \right] \ast \hat{U}(f_x, f_y) \]
The Sampling Theorem – from Goodman Section 2.4.1

\[ \hat{U}_s(f_x, f_y) = \mathcal{F} \left[ \text{comb}(x/X)\text{comb}(y/Y) \right] * \hat{U}(f_x, f_y) \]

\[ \mathcal{F} \left[ \text{comb}(x/X)\text{comb}(y/Y) \right] = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta \left( f_x - \frac{n}{X}, f_y - \frac{m}{Y} \right) \]
The Sampling Theorem – from Goodman Section 2.4.1

\[ \hat{U}_s(f_x, f_y) = \mathcal{F} [\text{comb}(x/X)\text{comb}(y/Y)] * \hat{U}(f_x, f_y) \]

\[ \mathcal{F} [\text{comb}(x/X)\text{comb}(y/Y)] = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta \left( f_x - \frac{n}{X}, f_y - \frac{m}{Y} \right) \]

\[ \hat{U}_s(f_x, f_y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \hat{U} \left( f_x - \frac{n}{X}, f_y - \frac{m}{Y} \right) \]
The Sampling Theorem – from Goodman Section 2.4.1

\[
\mathcal{F} \left[ \text{comb}(x/X)\text{comb}(y/Y) \right] = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta \left( f_x - \frac{n}{X}, f_y - \frac{m}{Y} \right)
\]

\[
\hat{U}_s(f_x, f_y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \hat{U} \left( f_x - \frac{n}{X}, f_y - \frac{m}{Y} \right)
\]

Signal extends from \((-B_x, -B_y)\) to \((B_x, B_y)\) in Fourier domain.
The Sampling Theorem – from Goodman Section 2.4.1

\[
\mathcal{F} [\text{comb}(x/X)\text{comb}(y/Y)] = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta \left( f_x - \frac{n}{X}, f_y - \frac{m}{Y} \right)
\]

\[
\hat{U}_s(f_x, f_y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \hat{U} \left( f_x - \frac{n}{X}, f_y - \frac{m}{Y} \right)
\]

Signal extends from \((-B_x, -B_y)\) to \((B_x, B_y)\) in Fourier domain

Mask out copies with a rect function:

\[
\text{rect} \left( \frac{f_x}{2B_x} \right) \text{rect} \left( \frac{f_y}{2B_y} \right) \hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y)
\]

Bandwidth \((B_x, B_y)\) of signal
The Sampling Theorem – from Goodman Section 2.4.1

\[ \mathcal{F} [\text{comb}(x/X)\text{comb}(y/Y)] = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta \left( f_x - \frac{n}{X}, f_y - \frac{m}{Y} \right) \]

\[ \hat{U}_s(f_x, f_y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \hat{U} \left( f_x - \frac{n}{X}, f_y - \frac{m}{Y} \right) \]

Mask out copies with a rect function:

\[ \text{rect} \left( \frac{f_x}{2B_x} \right) \text{rect} \left( \frac{f_y}{2B_y} \right) \hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y) \]
\[
\text{rect} \left( \frac{f_x}{2B_x} \right) \text{rect} \left( \frac{f_y}{2B_y} \right) \hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y)
\]

\[
F[\bullet] \quad h(x, y) = 4B_xB_y \text{sinc}(2B_xx) \text{sinc}(2B_yy)
\]
\[
\text{rect} \left( \frac{f_x}{2B_x} \right) \text{rect} \left( \frac{f_y}{2B_y} \right) \hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y)
\]

\[
F[\bullet] \quad h(x, y) = 4B_x B_y \text{sinc}(2B_x x) \text{sinc}(2B_y y)
\]

\[
h(x, y) \ast (U(x, y) \text{comb}(x/X) \text{comb}(y/Y)) = U(x, y)
\]
\[ \text{rect} \left( \frac{f_x}{2B_x} \right) \text{rect} \left( \frac{f_y}{2B_y} \right) \hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y) \]

\[ F[\bullet] h(x, y) = 4B_x B_y \text{sinc}(2B_x x) \text{sinc}(2B_y y) \]

\[ h(x, y) \ast (U(x, y) \text{comb}(x/X) \text{comb}(y/Y)) = U(x, y) \]

\[ U(x, y) \text{comb}(x/X) \text{comb}(y/Y) = XY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \delta(x - nX, y - mY) \]
\[
\text{rect} \left( \frac{f_x}{2B_x} \right) \text{rect} \left( \frac{f_y}{2B_y} \right) \hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y)
\]

\[
F[\bullet] \quad h(x, y) = 4B_x B_y \text{sinc}(2B_x x) \text{sinc}(2B_y y)
\]

\[
h(x, y) \ast (U(x, y) \text{comb}(x/X) \text{comb}(y/Y)) = U(x, y)
\]

\[
U(x, y) \text{comb}(x/X) \text{comb}(y/Y) = XY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \delta(x - nX, y - mY)
\]

\[
U(x, y) = 4B_x B_y XY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \text{sinc} \left[ 2B_x (x - nX) \right] \text{sinc} \left[ 2B_y (y - mY) \right]
\]
The Sampling Theorem

When sampled appropriately, a discrete signal can exactly reproduce a continuous signal:

\[ U(x, y) = 4B_x B_y XY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \text{sinc}[2B_x(x - nX)] \text{sinc}[2B_y(y - mY)] \]

Continuous signal:
- EM field
- Sound wave
- MR signal

Discretized signal:
- Detected EM field
- Sampled sound wave
- Sampled MR signal
What does the Sampling Theorem mean for us?

Continuous fields

Discretize vectors (and matrices)

(*) Under certain conditions
Conditions to safely apply the sampling theorem

\[ U(x, y) = 4B_xB_yXY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \text{sinc}[2B_x(x - nX)] \text{sinc}[2B_y(y - mY)] \]

- Sampling must be proportional to bandwidth (2B_x and 2B_y)
- “Nyquist” sampling: X = 1/2B_x, Y = 1/2B_y
Conditions to safely apply the sampling theorem

\[ U(x, y) = 4B_x B_y X Y \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \text{sinc} [2B_x (x - nX)] \text{sinc} [2B_y (y - mY)] \]

- Sampling must be proportional to bandwidth (\(2B_x\) and \(2B_y\))
  - “Nyquist” sampling: \(X = 1/2B_x\), \(Y = 1/2B_y\)

Half-period \(\sim 1/2B_x\)

Interval width \(X\)

(Assume largest spatial freq. in signal here)

\(X < 1/2B_x\)
Conditions to safely apply the sampling theorem

\[ U(x, y) = 4B_x B_y X Y \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \text{sinc}[2B_x(x - nX)] \text{sinc}[2B_y(y - mY)] \]

- Sampling must be proportional to bandwidth (2B_x and 2B_y)
  - “Nyquist” sampling: X = 1/2B_x, Y = 1/2B_y

(Assume largest spatial freq. in signal here)

U(x)comb(x/X)

Half-period ~ 1/2B_x

Interval width X

X = 1/2B_x

Nyquist sampling – still sampling peak and trough
Conditions to safely apply the sampling theorem

\[ U(x, y) = 4B_x B_y XY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \text{sinc}[2B_x(x-nX)] \text{sinc}[2B_y(y-mY)] \]

- Sampling must be proportional to bandwidth (2B_x and 2B_y)
  - “Nyquist” sampling: \( X = \frac{1}{2B_x} \), \( Y = \frac{1}{2B_y} \)

(Assume largest spatial freq. in signal here)

Half-period ~ \( \frac{1}{2B_x} \)

Interval width \( X \)

\( X > \frac{1}{2B_x} \)

Can’t detect the frequency anymore!
Conditions to safely apply the sampling theorem

\[ U(x, y) = 4B_x B_y XY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \text{sinc}[2B_x(x - nX)] \text{sinc}[2B_y(y - mY)] \]

- Sampling must be proportional to bandwidth (2B_x and 2B_y)
  - “Nyquist” sampling: X = 1/2B_x, Y = 1/2B_y
  - Needed to avoid aliasing
Conditions to safely apply the sampling theorem

\[ U(x, y) = 4B_x B_y XY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \text{sinc}[2B_x(x - nX)] \text{sinc}[2B_y(y - mY)] \]

- Sampling must be proportional to bandwidth \((2B_x \text{ and } 2B_y)\)
  - “Nyquist” sampling: \(X = 1/2B_x, \ Y = 1/2B_y\)
  - Needed to avoid aliasing

\[ \hat{U}(f_x) \]
Conditions to safely apply the sampling theorem

\[ U(x, y) = 4B_x B_y XY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \text{sinc}[2B_x(x - nX)] \text{sinc}[2B_y(y - mY)] \]

- Sampling must be proportional to bandwidth (2B_x and 2B_y)
  - “Nyquist” sampling: \( X = 1/2B_x \), \( Y = 1/2B_y \)
  - Needed to avoid aliasing

Even longer interval \( X \)

Problem: signal “overlaps” (aliasing)
Linear Algebra – notation and basics

- We’ll (try to) write column vectors as lower case variables
- Row vectors will be denoted as the transpose
- We’ll try to write matrices as upper case variables
- We’ll try to denote if a matrix/vector is real, complex etc. and its size with a certain notation
Linear Algebra – notation and basics

Some basic vector operations you should know:

- Conjugate, transpose, conjugate transpose
- Inner product
- Hadamard (element-wise, dot-times) product
- Outer product
- Vector (matrix) addition
- Matrix-vector product
- Convolution
Matrix-vector products – two useful interpretations

1. Inner products per entry:

\[
\begin{bmatrix}
 1 & 2 & 3 \\
 10 & 11 & 12 \\
\end{bmatrix}
\begin{bmatrix}
 1 \\
 2 \\
 3 \\
\end{bmatrix}
\begin{bmatrix}
 v \\
\end{bmatrix}
\begin{bmatrix}
 1 & 2 & 3 \\
 4 & 5 & 6 \\
 7 & 8 & 9 \\
\end{bmatrix}
\begin{bmatrix}
 10 \\
 11 \\
 12 \\
\end{bmatrix}
\]
Matrix-vector products – two useful interpretations

1. Inner products per entry:

\[ T(2,:) \begin{bmatrix} 4 & 5 & 6 \\ 10 \\ 11 \\ 12 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} u \\ 10 \\ 11 \\ 12 \end{bmatrix} \]
### Matrix-vector products – two useful interpretations

1. Inner products per entry:

\[
\begin{align*}
T(3,:) & = \\
\begin{bmatrix}
7 & 8 & 9 \\
10 & 11 & 12 \\
\end{bmatrix} & \begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
\end{bmatrix} & = & \begin{bmatrix}
v_1 \\
v_2 \\
v_3 \\
\end{bmatrix} & = & \begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{bmatrix} & \begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
\end{bmatrix} \\
= & \begin{bmatrix}
68 \\
167 \\
266 \\
\end{bmatrix} & = & \begin{bmatrix}
10 \\
11 \\
12 \\
\end{bmatrix} \\
\end{align*}
\]
Matrix-vector products – two useful interpretations

1. Inner products per entry:

\[
\begin{align*}
    v &= \begin{bmatrix} 68 \\ 167 \\ 266 \end{bmatrix} \\
    T &= \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \\
    \text{and} \\
    u &= \begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}
\end{align*}
\]

2. Weighted column sum:

\[
\begin{align*}
    v &= \begin{bmatrix} 68 \\ 167 \\ 266 \end{bmatrix} \\
    T(:,1) &= \begin{bmatrix} 1 \\ 4 \\ 7 \end{bmatrix} \\
    T(:,2) &= \begin{bmatrix} 2 \\ 5 \\ 8 \end{bmatrix} \\
    T(:,3) &= \begin{bmatrix} 3 \\ 6 \\ 9 \end{bmatrix} \\
    &= 10 + 11 + 12
\end{align*}
\]
Discrete convolution

\[
V(x_o) = \int_{-\infty}^{\infty} U(x_i) h(x_o - x_i) dx_i
\]

\[
v[x_0] = \sum_{x_i = -M}^{M} u[x_i] h[x_o - x_i]
\]
### Discrete 1D Convolution – an example

#### Steps to follow:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>List the index ‘k’ covering a sufficient range</td>
</tr>
<tr>
<td>2</td>
<td>List the input (x[k])</td>
</tr>
<tr>
<td>3</td>
<td>Obtain the reversed sequence (h[-k]), and align the rightmost element of (h[n-k]) to the leftmost element of (x[k])</td>
</tr>
<tr>
<td>4</td>
<td>Cross-multiply and sum the nonzero overlap terms to produce (y[n])</td>
</tr>
<tr>
<td>5</td>
<td>Slide (h[n-k]) to the right by one position</td>
</tr>
<tr>
<td>6</td>
<td>Repeat step 4; stop if all the output values are zero or if required.</td>
</tr>
</tbody>
</table>

[http://host.uniroma3.it/laboratori/sp4te/teaching/sp4bme/documents/LectureConvolution.pdf](http://host.uniroma3.it/laboratori/sp4te/teaching/sp4bme/documents/LectureConvolution.pdf)
Example 2: Find the convolution of the two sequences \( x[n] \) and \( h[n] \) given by,

\[
x[k] = [3 \ 1 \ 2] \quad h[k] = [3 \ 2 \ 1]
\]

<table>
<thead>
<tr>
<th>( k )</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x[k] ):</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h[-k] ):</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h[1-k] ):</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h[2-k] ):</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h[3-k] ):</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h[4-k] ):</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>( h[5-k] ):</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Hint: The value of \( k \) starts from (−length of \( h + 1 \)) and continues till (length of \( h \) + length of \( x \) − 1)

Here \( k \) starts from \(-3 + 1 = -2\) and continues till \(3 + 3 - 1 = 5\)
Example 2: Find the convolution of the two sequences $x[n]$ and $h[n]$ given by,

$$x[k] = \begin{bmatrix} 3 & 1 & 2 \end{bmatrix} \quad h[k] = \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}$$

<table>
<thead>
<tr>
<th>$k$</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x[k]$</td>
<td></td>
<td></td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[1-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[2-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[3-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[4-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[5-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$y$: 9
**Example 2:** Find the convolution of the two sequences \( x[n] \) and \( h[n] \) given by,

\[
x[k] = \begin{bmatrix} 3 & 1 & 2 \end{bmatrix} \quad h[k] = \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}
\]

<table>
<thead>
<tr>
<th>( k )</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x[k] ):</td>
<td></td>
<td></td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h[-k] ):</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h[1-k] ):</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h[2-k] ):</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h[3-k] ):</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h[4-k] ):</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h[5-k] ):</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( y \): 9 6+3
Example 2: Find the convolution of the two sequences $x[n]$ and $h[n]$ given by,

$$x[k] = [3 \ 1 \ 2] \quad h[k] = [3 \ 2 \ 1]$$

<table>
<thead>
<tr>
<th>$k$</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x[k]$</td>
<td></td>
<td></td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[1-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[2-k]$</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[3-k]$</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[4-k]$</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[5-k]$</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$y$: $9 \ 6+3 \ 3+2+6$
Example 2: Find the convolution of the two sequences $x[n]$ and $h[n]$ given by,

$$x[k] = [3 \ 1 \ 2] \quad h[k] = [3 \ 2 \ 1]$$

<table>
<thead>
<tr>
<th>$k$</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x[k]$</td>
<td></td>
<td></td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[1-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[2-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[3-k]$</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[4-k]$</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[5-k]$</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$$y: \quad 9 \quad 6+3 \quad 3+2+6 \quad 1+4+0$$
Example 2: Find the convolution of the two sequences $x[n]$ and $h[n]$ given by,

$$x[k] = [3 \ 1 \ 2], \quad h[k] = [3 \ 2 \ 1]$$

<table>
<thead>
<tr>
<th>$k$</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x[k]$</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[1-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[2-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[3-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[4-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h[5-k]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$$y: \quad 9 \quad 6+3 \quad 3+2+6 \quad 1+4+0$$

$$y: \quad [9 \quad 9 \quad 11 \quad 5 \quad 2 \quad 0]$$
Discrete convolution

\[ V(x_o) = \int_{-\infty}^{\infty} U(x_i) h(x_o - x_i) dx_i \]

\[ v[x_0] = \sum_{x_i=-M}^{M} u[x_i] h[x_o - x_i] \]

Discrete 2D convolution

\[ V(x_o, y_o) = \int \int_{-\infty}^{\infty} U(x_i, y_i) h(x_o - x_i, y_o - y_i) dx_i dy_i \]

\[ v[x_0, y_0] = \sum_{y_i=-L}^{L} \sum_{x_i=-M}^{M} u[x_i, y_i] h[x_o - x_i, y_o - y_i] \]
Discrete 2D convolution

https://www.psi.toronto.edu/~jimmy/ece521/Tut1.pdf
Discrete 2D convolution: edge conditions and even kernels

From MATLAB definition of conv2:

\[
C(j, k) = \sum_p \sum_q A(p, q)B(j - p + 1, k - q + 1)
\]

\(p\) and \(q\) run over all values that lead to legal subscripts of \(A(p, q)\) and \(B(j - p + 1, k - q + 1)\).

\(i=1, j=1\): Start in the upper left corner at \(A(1,1)\) with **lower right** of flipped version of \(B\) \([B(1,1)]\):

- For corner-to-corner alignment, doesn’t matter if matrix size is even or odd
- Output matrix will be larger than input matrices

Output: 8 x 8

<table>
<thead>
<tr>
<th></th>
<th>25</th>
<th>100</th>
<th>149</th>
<th>205</th>
<th>49</th>
<th>130</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>105</td>
<td>150</td>
<td>225</td>
<td>149</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>130</td>
<td>140</td>
<td>165</td>
<td>179</td>
<td>130</td>
</tr>
<tr>
<td>37</td>
<td>113</td>
<td>147</td>
<td>96</td>
<td>189</td>
<td>83</td>
<td>90</td>
</tr>
<tr>
<td>140</td>
<td>54</td>
<td>253</td>
<td>145</td>
<td>255</td>
<td>137</td>
<td>254</td>
</tr>
<tr>
<td>0</td>
<td>140</td>
<td>54</td>
<td>53</td>
<td>78</td>
<td>243</td>
<td>90</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>140</td>
<td>17</td>
<td>0</td>
<td>23</td>
<td>222</td>
</tr>
</tbody>
</table>
Discrete 2D convolution: edge conditions and even kernels

From Tensorflow definition of conv2:

\[
\text{output}[b, i, j, k] = \\
\text{sum}_{di, dj, q} \text{input}[b, strides[1] \times i + di, strides[2] \times j + dj, q] \\
\times \text{filter}[di, dj, q, k]
\]

Start convolution kernel inside image: align upper-left of image A with upper right of kernel B

- Output matrix will be smaller than input image and filter
- We will work through these numbers carefully!
Linear Algebra – notation and basics

Some basic types of matrices & terms that you should know about:

• Symmetric (Hermitian) matrix: $A = A^T$ if $A$ is real, $A = A^H$ if $A$ is complex

• Square, hot-dog and hamburger matrices

• Invertible matrix

• Diagonal matrix

• Toeplitz matrix

• Banded matrix
Discrete Fourier Transforms

\[ \hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp(-2\pi i f_x x) \, dx \]

\[ \hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp(-2\pi i f_x x / M) \]

Inner product of \( u \) with different complex expon.

- np.fft(u), np.fftshift(np.fft(np.fftshift(u)))
- \( \text{fft} = \text{fast Fourier transform}, \text{much more comp. efficient than matrix multiplication!} \)
Discrete Fourier Transforms

\[
\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp(-2\pi i f_x x) \, dx
\]

\[
\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp(-2\pi i f_x x / M)
\]

Inner product of \( u \) with different complex exponent.

- np.fft(u), np.fftshift(np.fft(np.ifftshift(u)))
- fft = fast Fourier transform, much more comp. efficient than matrix multiplication!
Discrete Fourier Transforms

\[ \hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp(-2\pi i f_x x) \, dx \]

\[ \hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp(-2\pi i f_x x / M) \]

*Inner product of u with different complex expon.*

- np.fft(u), np.fftshift(np.fft(np.iftshift(u)))
- fft = fast Fourier transform, much more comp. efficient than matrix multiplication!
Discrete Fourier Transforms

\[ \hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp(-2\pi i f_x x) \, dx \]

\[ \hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp(-2\pi i f_x x / M) \]

\[ \hat{u} = \text{FT Matrix, } \theta \]

\[ \text{np.fft(np.eye(10))} \]

Treats 1\textsuperscript{st} entry of \( \hat{u} \) as \( f_x=0 \)
Discrete Fourier Transforms

\[
\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp(-2\pi i (f_x x)) \, dx
\]

\[
\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp(-2\pi i f_x x / M)
\]

Treats middle entry of \( \hat{u} \) as \( f_x = 0 \)

\[
\text{np.fftshift(np.fft(np.ifftshift(np.eye(10))))}
\]
**Discrete convolution theorem**

Convolution theorem. If $\mathcal{F}\{g(x, y)\} = G(f_x, f_y)$ and $\mathcal{F}\{h(x, y)\} = H(f_x, f_y)$, then

$$
\mathcal{F}\left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(\xi, \eta) \ h(x - \xi, y - \eta) \ d\xi \ d\eta \right\} = G(f_x, f_y) \ H(f_x, f_y).
$$

(2-15)
Discrete convolution theorem

Convolution theorem. If $\mathcal{F}\{g(x, y)\} = G(f_x, f_y)$ and $\mathcal{F}\{h(x, y)\} = H(f_x, f_y)$, then

$$\mathcal{F}\left\{ \int\!\!\int_{-\infty}^{\infty} g(\xi, \eta) h(x - \xi, y - \eta) \, d\xi \, d\eta \right\} = G(f_x, f_y) H(f_x, f_y). \quad (2-15)$$

If $\mathcal{F}\{g[x, y]\} = G[f_x, f_y]$ and $\mathcal{F}\{h[x, y]\} = H[f_x, f_y]$, and if we know that

$$g[x, y] * h[x, y] = \sum_{l=-L}^{L} \sum_{m=-M}^{M} g[m, l] h[x - m, y - l],$$

then from the Convolution Theorem we have,

$$\mathcal{F}[g[x, y] * h[x, y]] = G[f_x, f_y] H[f_x, f_y]$$
Discrete convolution theorem example – same thing as continuous case

\[ U_1(x,y) \] 
\[ * \] 
\[ F[U_1] \] 
\[ = \] 
\[ F[h] \] 
\[ = \] 
\[ F^{-1}[H\hat{U}_1] \]
Convolutions as a big matrix multiplication

\[(u * h)[x] = \sum_{m=0}^{N+M-2} u[m]h[x-m] \quad \Rightarrow \quad y = u * h = \begin{bmatrix}
    h_1 & 0 & \cdots & 0 & 0 \\
    h_2 & h_1 & \cdots & \vdots & \vdots \\
    h_3 & h_2 & \cdots & 0 & 0 \\
    \vdots & h_3 & \cdots & h_1 & 0 \\
    \vdots & \vdots & \ddots & \vdots & \vdots \\
    h_{m-1} & \cdots & h_2 & h_1 \\
    h_m & h_{m-1} & \cdots & \vdots & \vdots \\
    0 & h_m & \cdots & h_{m-2} & \vdots \\
    0 & 0 & \cdots & h_{m-1} & h_{m-2} \\
    \vdots & \vdots & \ddots & \vdots & \vdots \\
    0 & 0 & \cdots & h_m & h_{m-1} \\
    0 & 0 & \cdots & 0 & \cdots & h_m
\end{bmatrix} \begin{bmatrix}
    u_1 \\
    u_2 \\
    u_3 \\
    \vdots \\
    u_n
\end{bmatrix}\]
Last thing – matrix and vector derivatives

\[ u = Wv \]

\[ \frac{du}{dv} = \]
Last thing – matrix and vector derivatives

$$u = Wv$$

$$\frac{du}{dv} =$$

$$u_3 = W_{3,1}v_1 + W_{3,2}v_2 + \ldots + W_{3,M}v_M$$

$$\frac{\partial u_3}{\partial v_2} = \frac{\partial}{\partial v_2}(W_{3,1}v_1 + W_{3,2}v_2 + \ldots + W_{3,M}v_M) = \frac{\partial}{\partial v_2}W_{3,2}v_2 = W_{3,2}$$
Last thing – matrix and vector derivatives

\[ u = Wv \]

\[ \frac{du}{dv} = \]

\[ u_3 = W_{3,1}v_1 + W_{3,2}v_2 + \ldots + W_{3,M}v_M \]

\[ \frac{\partial u_3}{\partial v_2} = \frac{\partial}{\partial v_2}(W_{3,1}v_1 + W_{3,2}v_2 + \ldots + W_{3,M}v_M) = \frac{\partial}{\partial v_2}W_{3,2}v_2 = W_{3,2} \]

\[ \frac{\partial u_i}{\partial v_j} = W_{i,j} \]

\[ \frac{du}{dv} = W \]
Last thing – matrix and vector derivatives

\[ u = Wv \]

\[ \frac{du}{dv} = \]

\[ u_3 = W_{3,1}v_1 + W_{3,2}v_2 + ... + W_{3,M}v_M \]

\[ \frac{\partial u_3}{\partial v_2} = \frac{\partial}{\partial v_2}(W_{3,1}v_1 + W_{3,2}v_2 + ... + W_{3,M}v_M) = \frac{\partial}{\partial v_2}W_{3,2}v_2 = W_{3,2} \]

\[ \frac{\partial u_i}{\partial v_j} = W_{i,j} \]

\[ \frac{du}{dv} = W \]

• When confused, write out one entry, solve derivative and generalize
• Use dimensionality to help (if \( \mathbf{x} \) has \( N \) elements, and \( \mathbf{y} \) has \( M \), then \( d\mathbf{y}/d\mathbf{x} \) must be \( NxM \)
• Take advantage of *The Matrix Cookbook*:
  • https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf