Convolutional Neural Networks (CNNs)

- Sparse MLPs
  - (FFN)
  - Feed Forward Networks

- Explicit locality
  - in sequence (1D)
  - sub-image (2D)

**1D Convolution**

\[
\text{stride}=1
\]

\[
\begin{array}{c|c|c}
3 & 10 & 21 \\
\hline
1 & 5 & 10 \\
\hline
2 & 1 & 20 \\
\hline
4 & 1 & 10 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
1 & 0 & 2 \\
\hline
1 & 0 & 2 \\
\end{array}
\]

\[
k=3 \text{ (kernel size)}
\]

\[
b_i=0
\]

\[
\text{element-wise multiplication + sum (dot product)}
\]

\[
\text{learnable set of weights}
\]

\[
\{d-k+1
\]

\[
\begin{array}{c|c|c}
5 & 25 & 18 \\
\hline
6 & 6 & 18 \\
\hline
18 & 18 & 18 \\
\end{array}
\]
Padding?

Padding

Padding

$6 = L = len$

$1 \text{ channels} = 1$

$k = 3$

$m \equiv \text{out_channels}

(\text{filters} \text{, masks}, \text{kernels})$

$8 - 3 + 1 = 6$

$L = 6$

$L = 8$

$6 + 1 = 7$

$3 \text{ out_channels}$
Pooling

\[
\begin{bmatrix}
1 & 0 & 1 \\
2 & 1 & 2 \\
1 & 2 & 0 \\
2 & 2 & 1 \\
5 & 0 & 2 \\
4 & 0 & 2
\end{bmatrix}
\]

\[\text{Kx1}\]

\[\text{max} \]

\[
\begin{array}{c|c}
1 & 0 \\
2 & 0 \\
1 & 0 \\
0 & 1 \\
0 & 1 \\
4 & 0 \\
1 & 0 \\
0 & 1 \\
\end{array}
\]

\[\text{max} = 0 + \text{max \ operator} \quad (\text{instead of \ min})\]

\[k = 3 \quad \text{kernel size}\]

\[\text{kernel Channels} = 1\]

DNA

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

\[\text{in. channels} = 4\]

\[L = 8\]

\[w_1\]

\[3\]

\[4\]

\[\text{out. channels} = 3\]

\[k = 2\]

\[L = 6\]

\[m = 1 \text{d}\]

\[k = 3\]
The diagram illustrates a convolutional layer with a max pooling operation. The input data is a 2D tensor, and the convolution kernel has a size of 3x3. The max pooling operation is applied with a kernel size of 2x2, and the stride is 2. The output channels are equal to the input channels. The non-linear activation function used is ReLU.
2D Convolutions

In_channels = 1
n x n image
28 x 28

W
k = 3
h
b = 0
f = 1

n-3+1
(n-k+1) x (n-k+1)
out_channels = # output_channels = 1

Conv 2d
k = 3
out_channels?

n x n x 3
In_channels = 3

3 = # of in_channels
3
k = 3
n-k+1
2
out_channels
n = 28

k = 3
s = 1
p = 0
out_channels = 16

n - k + 1 = 28 - 3 + 1 = 26

max 2d
k = 2
s = 2
(individually for each channel)

3D Conv

3D + one-hot encoded seq
20 Amino Acids

$n \times n \times n \times 20 \quad \text{tensor in channels}$

$3 \times 3 \times 3 \times 20$

$(n-k+1)^3$