Week 7

10/7/2025

↳ Quiz

↳ Last Week: 2.6, 3.3

↳ Midterm 10/21

- practice exam(s) coming soon
- Linear Algebra → Transformers
- 6 multiple choice, three long

Context:

Supervised Learning

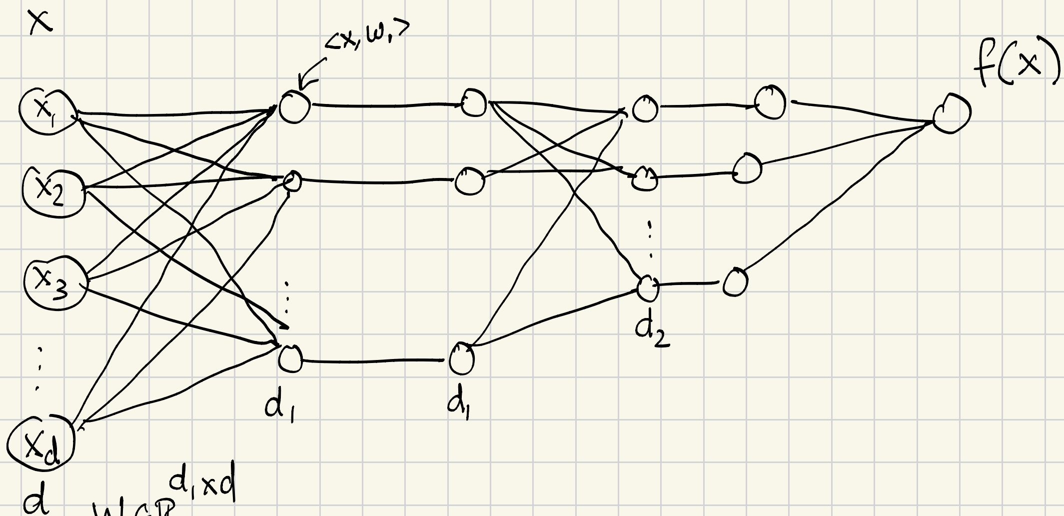
Models
- Linear Regression
- Logistic Regression
- Support Vector Machines

Feature selection
- Feature Transformation
- Kernel Trick
- Reparameterization Trick

⇒ Neural Networks!

- Convolutional (images, audio, etc)
- Transformers (text, sequential)

Review : Fully Connected NNs



$W \in \mathbb{R}^{d_1 \times d}$

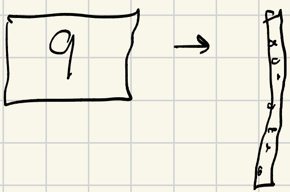
nn.Linear(d, d1) nn.ReLu() ...

$Wx = z \qquad \sigma(z) \qquad \cdots$

Issues with Linear Layers

1. Computationally expensive

2. Loss of "context"

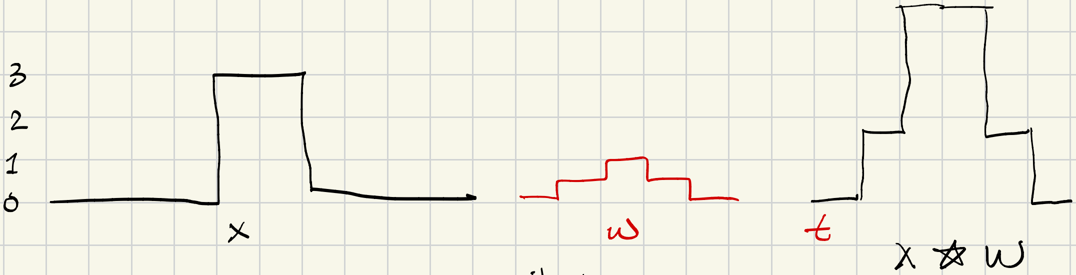


Wishlist for Convolutional Layers

1. Locality

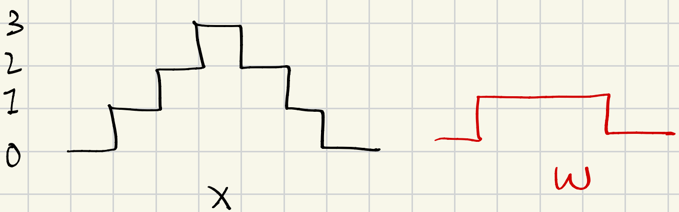
2. Shift invariance

Convolutions (1D)



Sanity :)

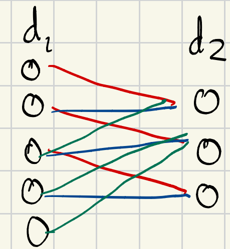
$$(x \star w)[t] = \sum_{\tau} x[t+\tau]\, w[\tau]$$



$$(x \star w)[2] = 7$$

Neural Net View

"kernel" w_1 w_2 w_3



Weights are reused!

\# weights = size(kernel)

vs

\# weights = $d_1 d_2$

Convolutions (2D)

$$(x \star w)[2,2] = 2$$

$$x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad w = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \qquad x \star w = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

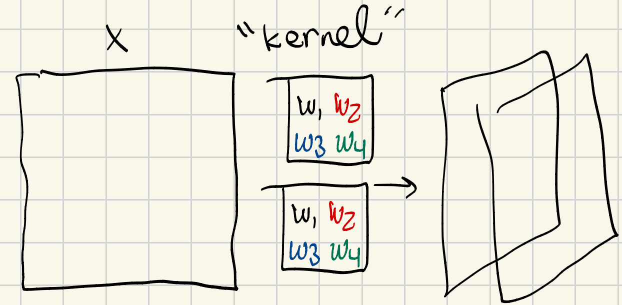
$$(x \star w)[s,t] = \sum_{\sigma} \sum_{\tau} x[s+\sigma, t+\tau]\, w[\sigma, \tau]$$

$$(x \star w)[2,2] = 3$$

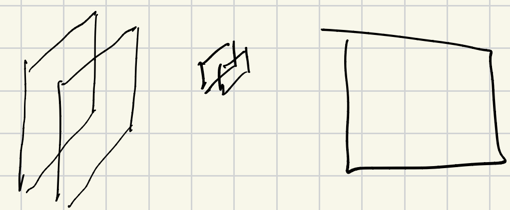
$$x = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad w = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \qquad x \star w = \begin{bmatrix} -3 & 3 \\ -3 & 3 \end{bmatrix}$$

Neural Net View

Conv(X, kernel)



Convolutions (3D)



More Considerations (ezyang)

- ↳ Padding
- ↳ Strides
- ↳ Pooling

CNN Training

↳ keep track of reused weights

Timeline of "Deep" Networks

Convolution → Residual → Transformer → Mixture of Experts

Deeper! Bigger! More data!

Residual Networks (loss landscape residual)

Skipped connections

$$h^{(l)} = \sigma(W^{(l)} h^{(l-1)})$$

vs

$$h^{(l)} = \sigma(W^{(l)} h^{(l-1)}) + h^{(l-1)}$$

Pytorch Example (:))

Reminders

↳ Pset 7 due Wednesday

↳ Office hours

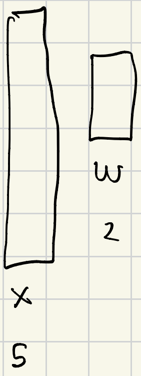
- 10-11:30 Thursday 10/16
- No office hours Monday 10/20

↳ Midterm 10/21

↳ Practice exam!

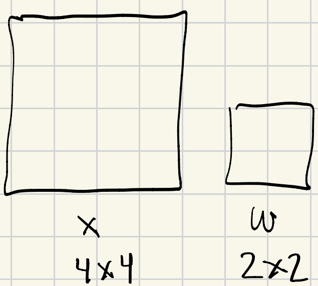
Review

1D



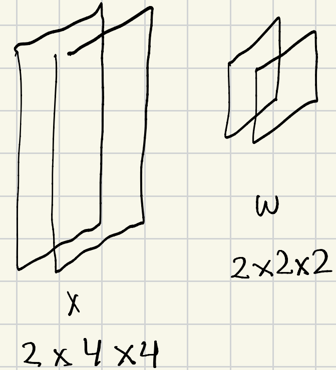
$X \star W$

2D



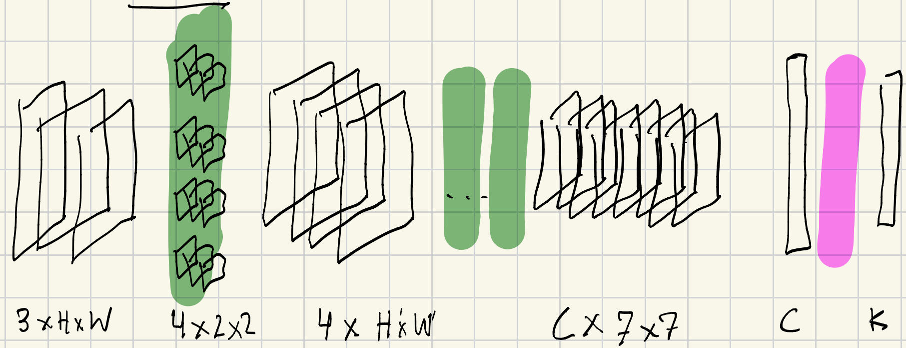
$X \star W$

3D



$X \star W$

CNN



Convolution
Linear

Motivation

Tabular data
(e.g., IRIS, diabetes)

$\Rightarrow$ Linear

Image and audio data
(e.g., MNIST, imagenet)

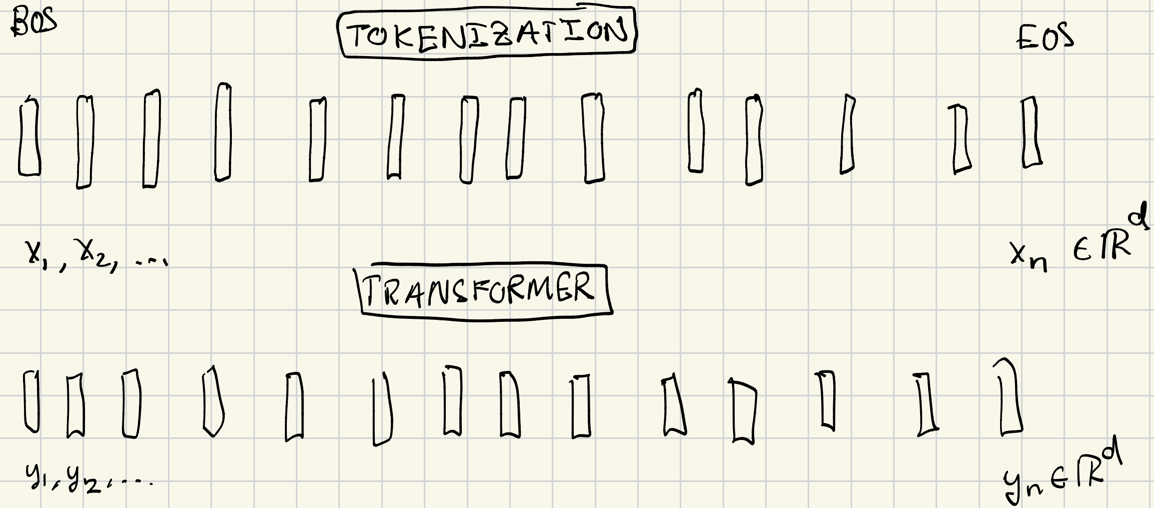
$\Rightarrow$ Convolution

Sequential data
(e.g., text, time series)

$\Rightarrow$ Transformer

Example

I live in Claremont. As usual, I will dress for ___



Transformer Wishlist

1. handle variable length input
2. numeric representation of words
3. process context (e.g., I $\Leftrightarrow$ Claremont $\Leftrightarrow$ dress)

Self Attention

Input: $x_1, \dots, x_n$

Output: $y_1, \dots, y_n$

Idea: Build y_i out of $x_1, \dots, x_n$

$\boxed{\text{Attempt \#1}}$

$$y_i = \sum_{j=1}^{n} W_{ij}\, x_j$$

group similar words

$$\begin{array}{c|c} & x_1 \; x_2 \; \dots \; x_n \\ \hline \begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{matrix} & \left[\begin{matrix} \; & \; & \; \\ & & \\ & & \\ & & \end{matrix} \right] \end{array}$$

W

"similarity" between x_i and x_j

$$W_{ij} = \frac{e^{\langle x_i, x_j \rangle}}{\sum_{k} e^{\langle x_i, x_k \rangle}}$$

softmax

But x_i appears three different places

1. query $\qquad q_i = W^{(q)} x_i$
2. key $\qquad k_i = W^{(k)} x_i$
3. value $\qquad v_i = W^{(v)} x_i$

$$\begin{array}{cc} & j \\ i & \left[\begin{matrix} \; & \; & \; \\ & & \\ \multicolumn{3}{c}{\text{softmax}} \\ & & \end{matrix} \right] \end{array} \qquad \left[\begin{matrix} \; \\ \; \\ \; \\ \; \end{matrix} \right]$$

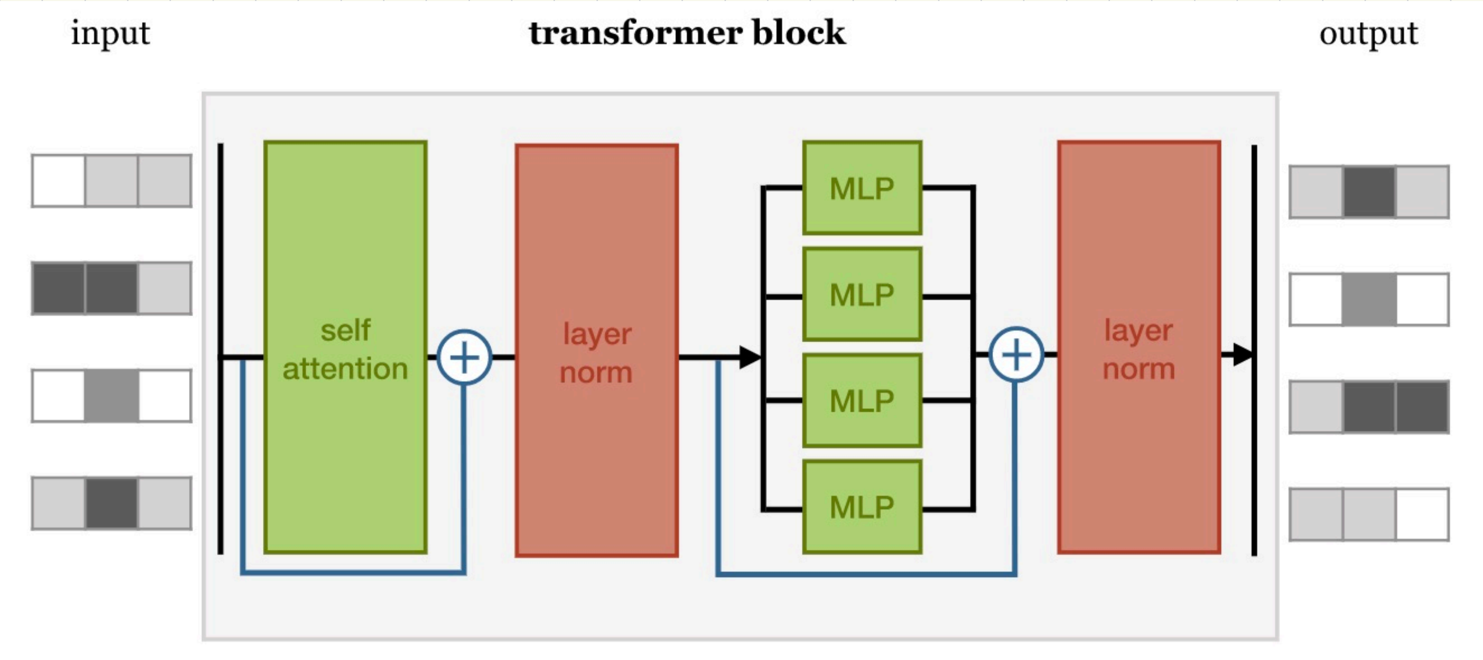
$\langle q_i, k_j \rangle$

softmax

$n \times n \qquad\qquad n \times d$

$$\underset{n\times d}{X}\;\underset{d\times d}{W^{(q)}}\;\underset{d\times d}{W^{(k)}}\;\underset{d\times n}{X^T} \qquad\qquad \underset{n\times d}{X}\;\underset{d\times d}{W^{(v)}}$$

Transformer



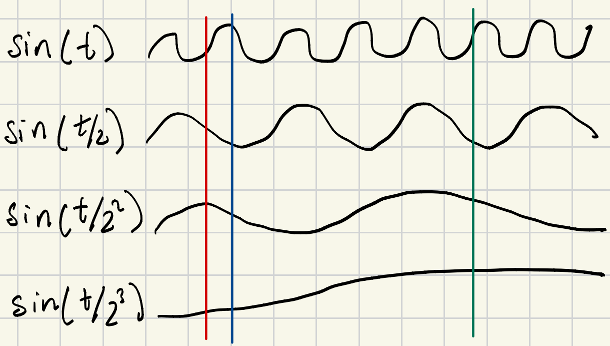
Positional Encoding

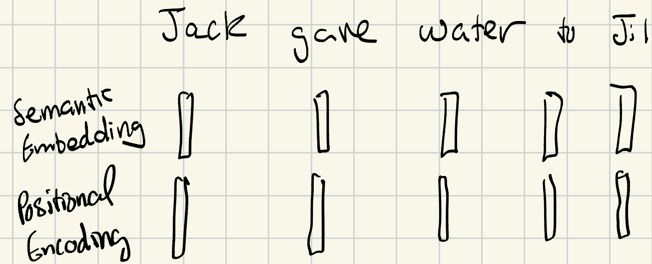
"Jack gave water to Jill"

vs

"Jill gave water to Jack"

Idea: Add positional encoding to tokens





Autoregressive Caching

$$\begin{array}{cc} & \begin{array}{ccccc} x_1 & x_2 & \cdots & \cdots & x_n \end{array} \\ \begin{array}{c} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_n \end{array} & \begin{bmatrix} & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 \\ & & & 0 & 0 \\ & & & & 0 \\ & & & & \end{bmatrix} \end{array}$$

"causal mask"

y_i built only from prior tokens

Benefit: Adding x_n ONLY changes y_n

↳ $O(n)$ updates vs $O(n^2)$