Week 7

10/7/2025

↳ Quiz

↳ Midterm 10/21

- practice exam(s) coming soon
- Linear Algebra → Transformers

Context:

Supervised Learning

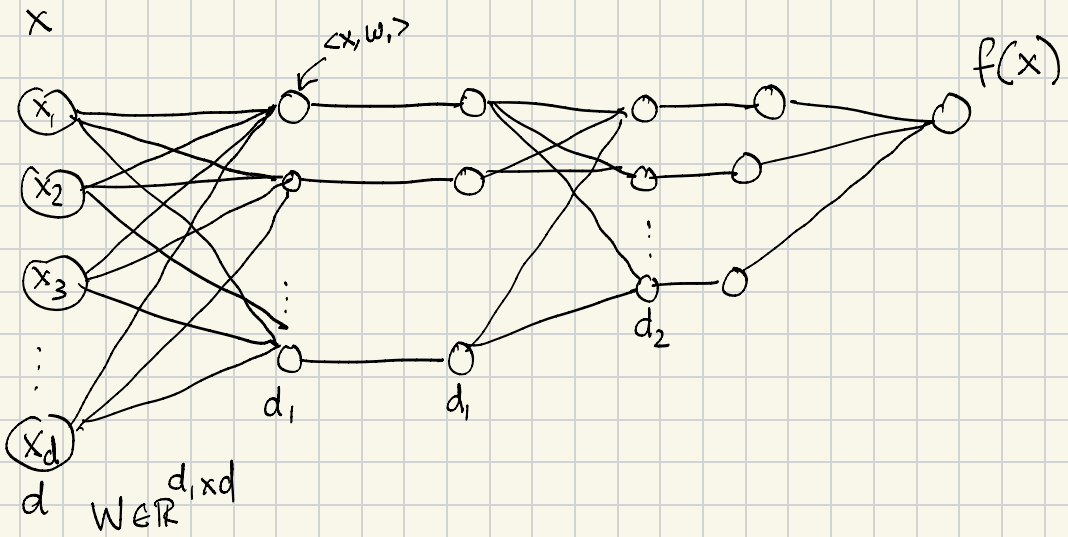
Models:
- Linear Regression
- Logistic Regression
- Support Vector Machines

Feature selection:
- Feature Transformation
- Kernel Trick
- Reparameterization Trick

=> Neural Networks!

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

Review: Fully Connected NNs



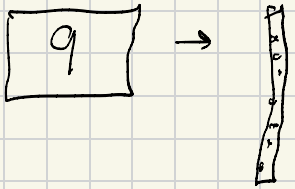
nn.Linear(d, d_1) nn.ReLU() ...

$Wx = z$ $\sigma(z)$ ...

Issues with Linear Layers

1. computationally expensive

2. Loss of "context"

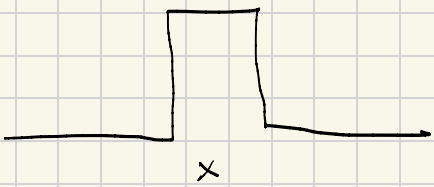


Wish list for Convolutional Layers

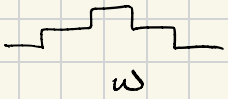
1. Locality

2. Shift invariance

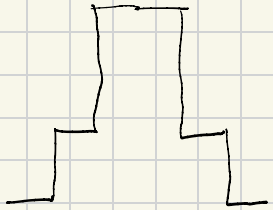
Convolutions (1D)



x



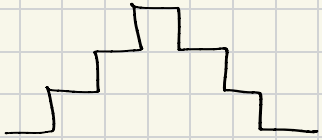
w



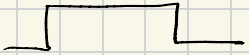
x ✡ w

Sanity ☺

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



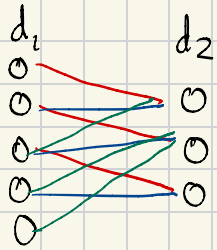
x



w

Neural Net View

"kernel" w_1 w_2 w_3



Weights are reused!

weights = size(kernel)

vs

weights = $d_1 d_2$

Convolutions (2D)

0	0	0	0
1	0	0	1
0	1	1	0
0	0	0	0

X

0	0	0
-1	-1	-1

W

1	0	1
2	2	2
1	2	1

$X \star W$

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

0	1	1	0
0	1	1	0
0	1	1	0
0	1	1	0

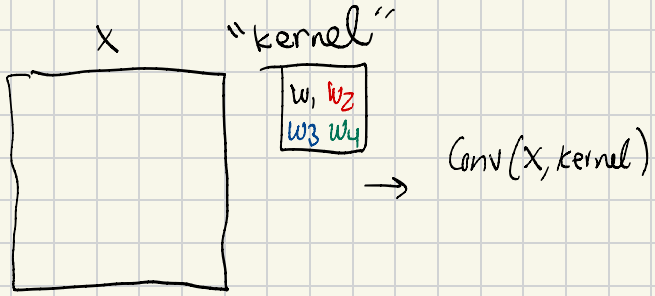
X

1	0	-1
1	0	-1
1	0	-1

W

$X \star W$

Neural Net View



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 (:)