Machine Learning: From 0 to Deep

Iris Horng

December 12, 2023

Machine Learning

The Fundamentals

Overview

data

$\mathbb{R}^{32\times 32}$

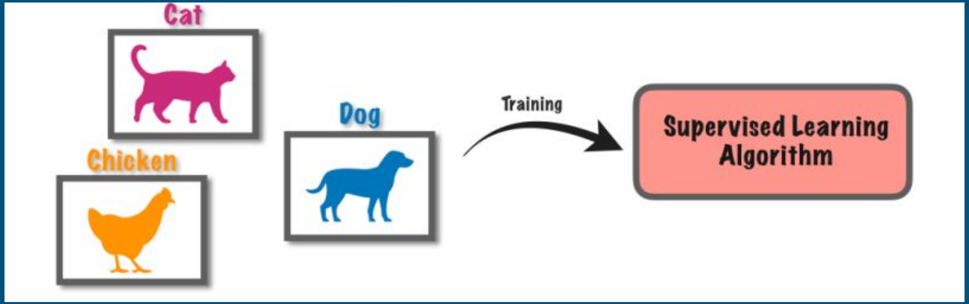
→ model →

predict a discrete set of items

dog
cat
elephant
rabbit

Supervised Learning

- Data are labeled with pre-defined classes
  - input and output (x_i , y_i)
- Goal: Given input data, predict output



[3]

Optimization Problem: Example

Linear regression:

- input: data X $\subset \mathbb{R}^n$
- output: labels Y $\subset \mathbb{R}^n$
- predictor for any x $\in$ X is

$$f(x; w, b) = w^\top x + b$$ [1]

Notes:

- different *w* and *b* give different functions *f*
- Weights: parameters

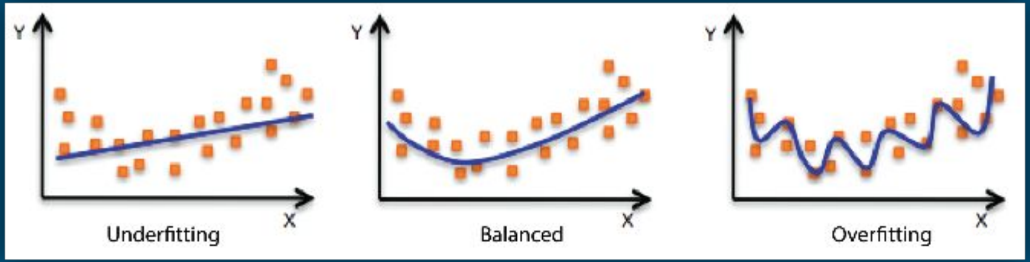
Optimization Problem: Parametric Methods

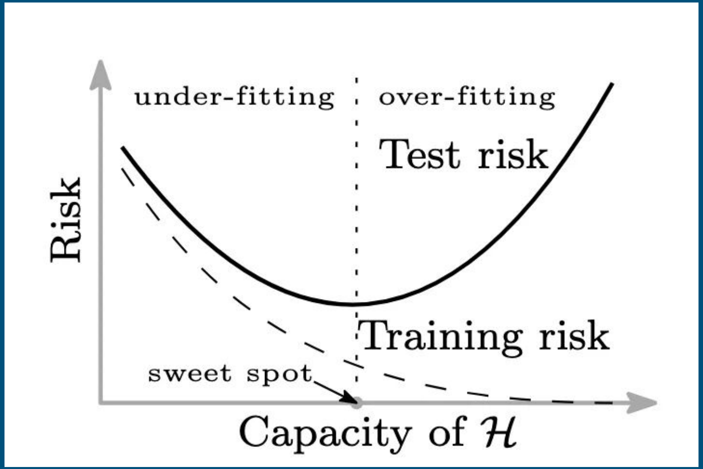
1. Select a form for the function
2. Learn the coefficients for the function from the training data.

- Ex. linear regression: parameters are w and b
- Ex. neural network: loss function is a function of parameters and data
- Note: can NOT change data. CAN change parameters

Main Point: changing parameters to get the ‘best’ model

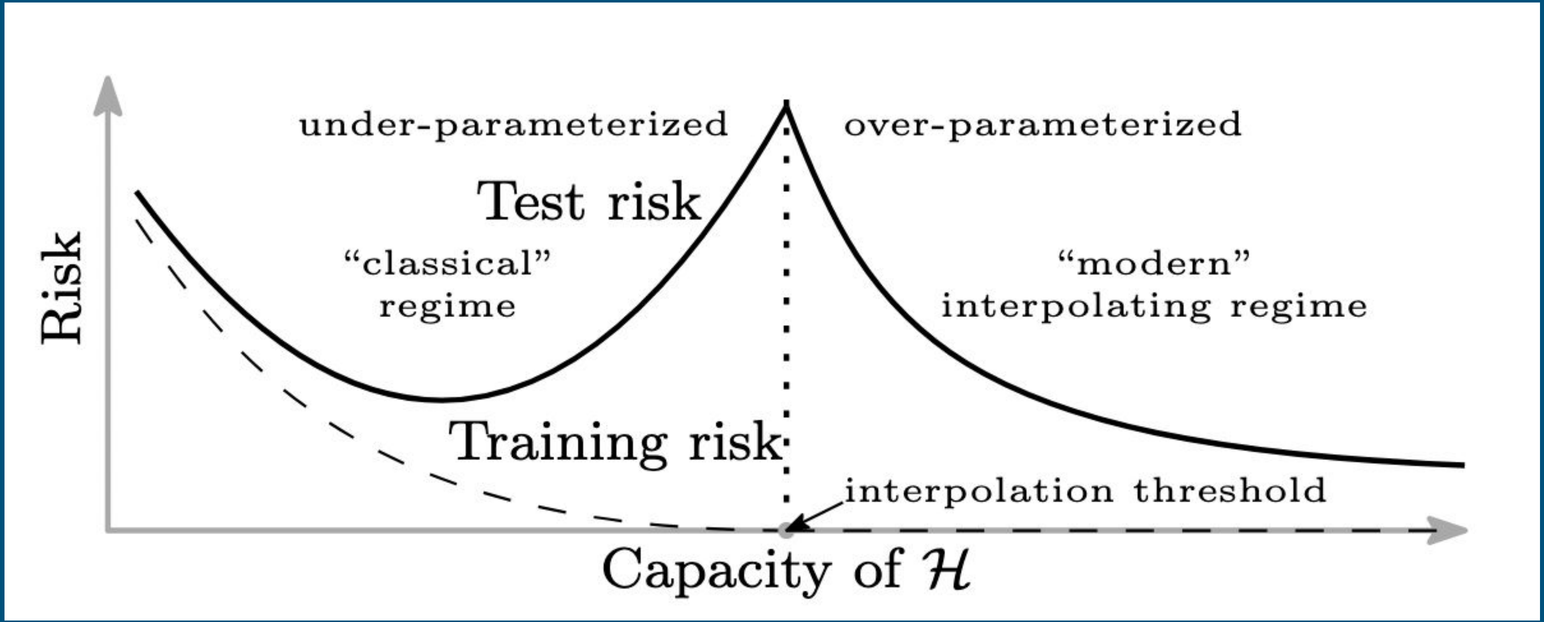
Optimization Problem: Parametric Methods





number of parameters

Optimization Problem: Extended...



Deep Learning

Neural Nets
CNNS

Overview

data

$\mathbb{R}^{32\times 32}$

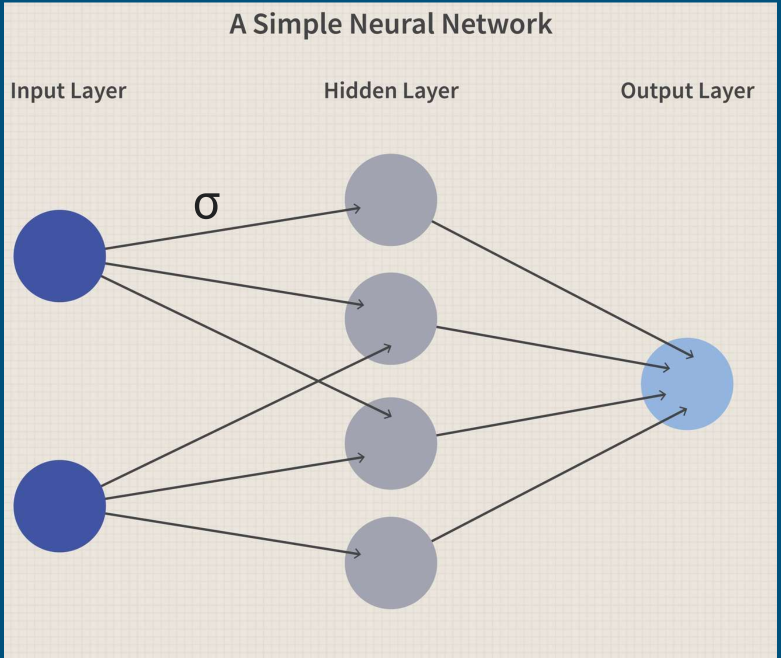
→ learned low-level features → learned mid-level features → learned high-level features → predict a discrete set of items:

dog
cat
elephant
rabbit

Neural Networks Overview

Procedure:

1. input: given data
2. linear transformation: matrix multiplication
3. apply nonlinearity σ
4. repeat …
5. output: features



[2]

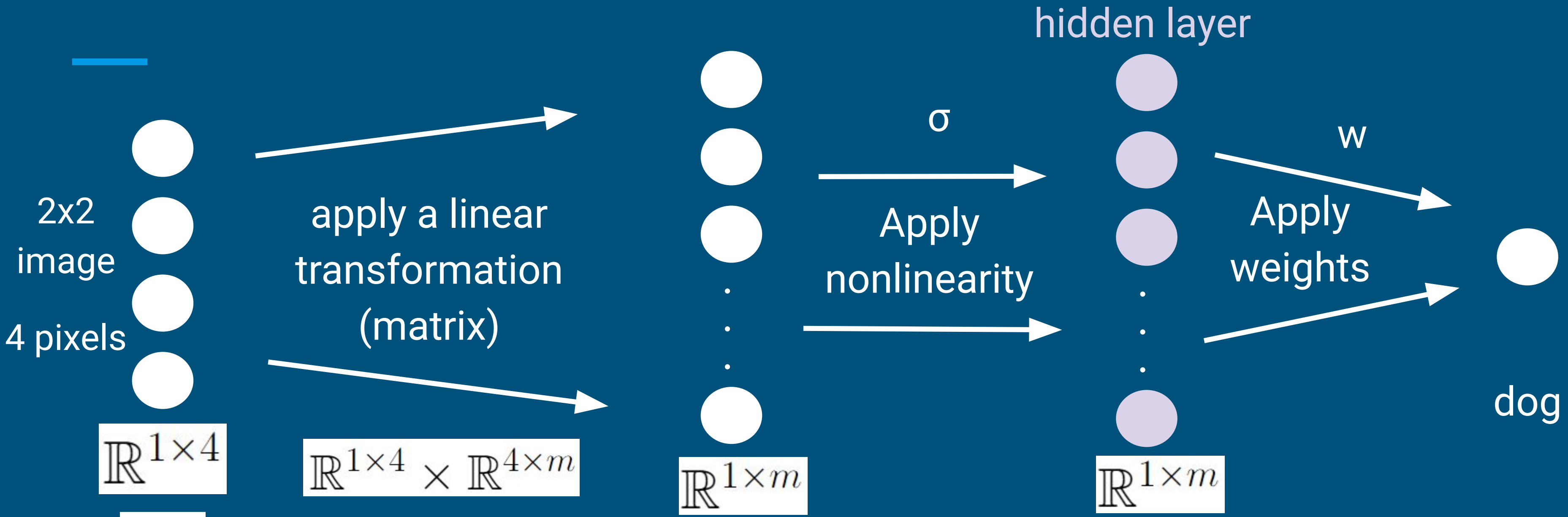
Learn feature matrix *S* and coefficients *w*

Classifier: $f(x; w, S) = \text{sign}\left(w^\top \sigma\left(S^\top x\right)\right)$ [1]

[1]

$$f(x; w, S) = \text{sign}\left(w^\top \sigma\left(S^\top x\right)\right)$$

Neural Networks Overview



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$S^T x = \begin{bmatrix} s_{1,1} & s_{1,2} & s_{1,3} & s_{1,4} \\ s_{2,1} & s_{2,2} & s_{2,3} & s_{2,4} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \end{bmatrix}$$

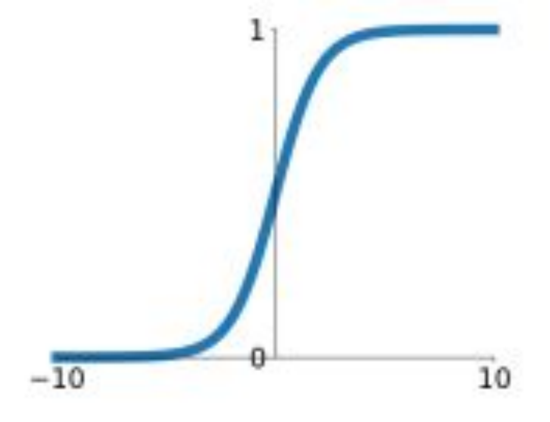
$$\sigma \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \end{bmatrix} = \begin{bmatrix} \sigma(a_1) \\ \sigma(a_2) \\ . \\ . \\ . \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \\ 0.7 \\ . \\ . \\ . \end{bmatrix}$$

Neural Networks: Activation Function

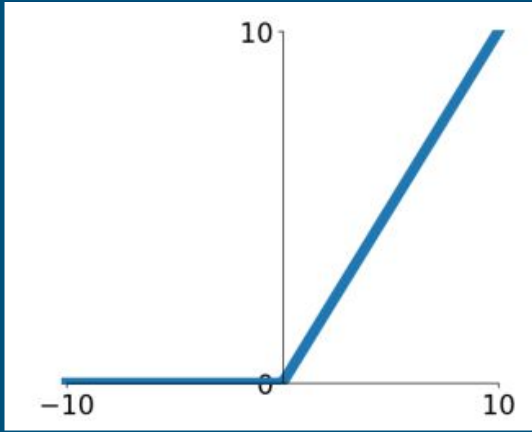
types of nonlinearities σ(·)

Sigmoid/Logistic



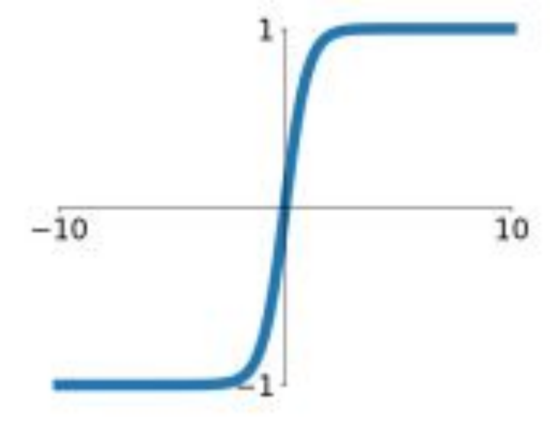
$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

ReLU



$$\begin{aligned}\text{relu}(x) &= |x|_+ \\ &= \max(0, x)\end{aligned}$$

Hyperbolic Tangent



$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Universal Approximation Theorem

Any continuous function f: $[0,1]^n \rightarrow [0,1]$ can be approximated arbitrarily well by a neural network with at least 1 hidden layer with a finite number of weights [5]

Convolutional Neural Network (CNN): Overview

Procedure:

1. input: given data
2. linear transformation: convolution
3. apply nonlinearity σ
4. repeat …
5. output: features

CNN: Convolution

Convolution x * w

x	1	2	3	4	5	6	7
w	2	4	2				

x * w	1(2) + 2(4) + 3(2) = 16				

CNN: Convolution

Convolution x * w

x

1	2	3	4	5	6	7

w

2	4	2

x * w

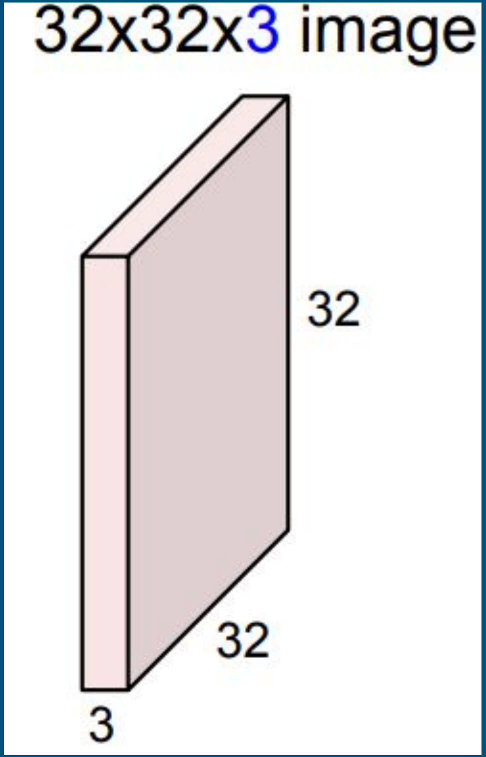
16	2(2) + 3(4) + 4(2) = 24			

CNN: Convolution

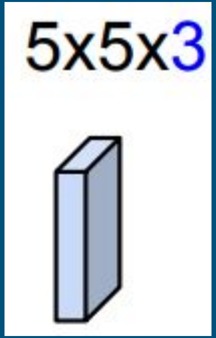
Convolution x * w

x	1	2	3	4	5	6	7
w					2	4	2
x * w		16	24	32	40	48	

CNN: Process



filter

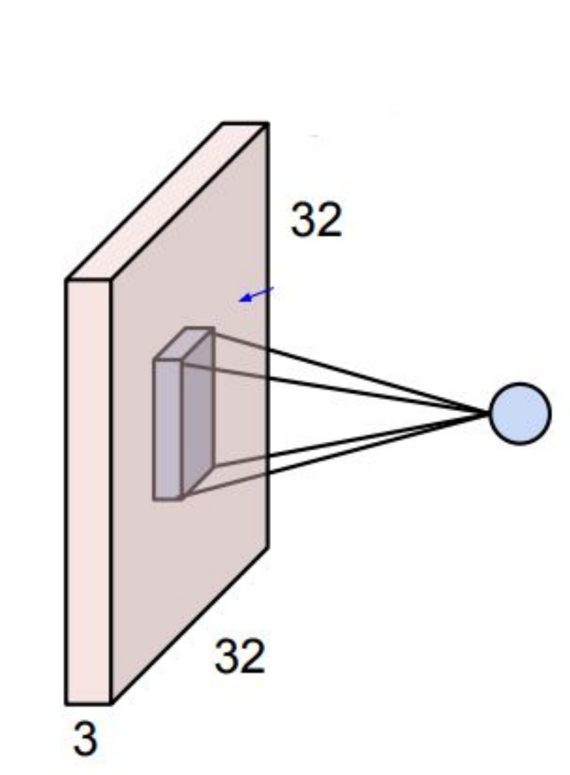


convolve the filter with the image (slide over the image spatially, computing dot products

remember to put citation

[4]

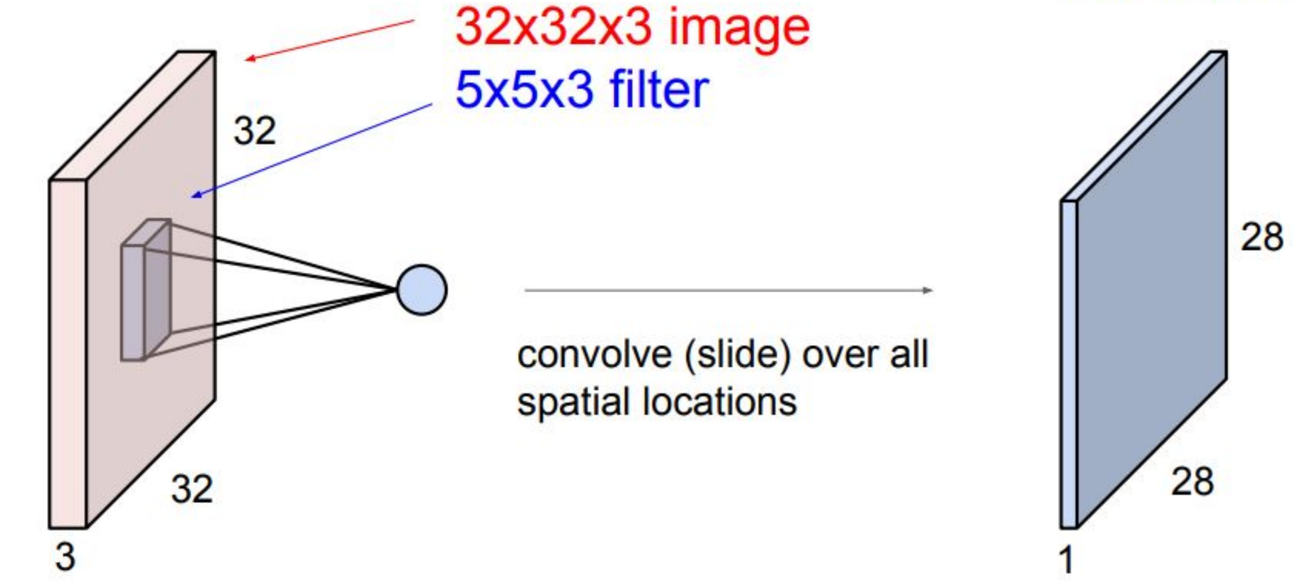
CNN: Process



convolve the filter with the image (slide over the image spatially, computing dot products

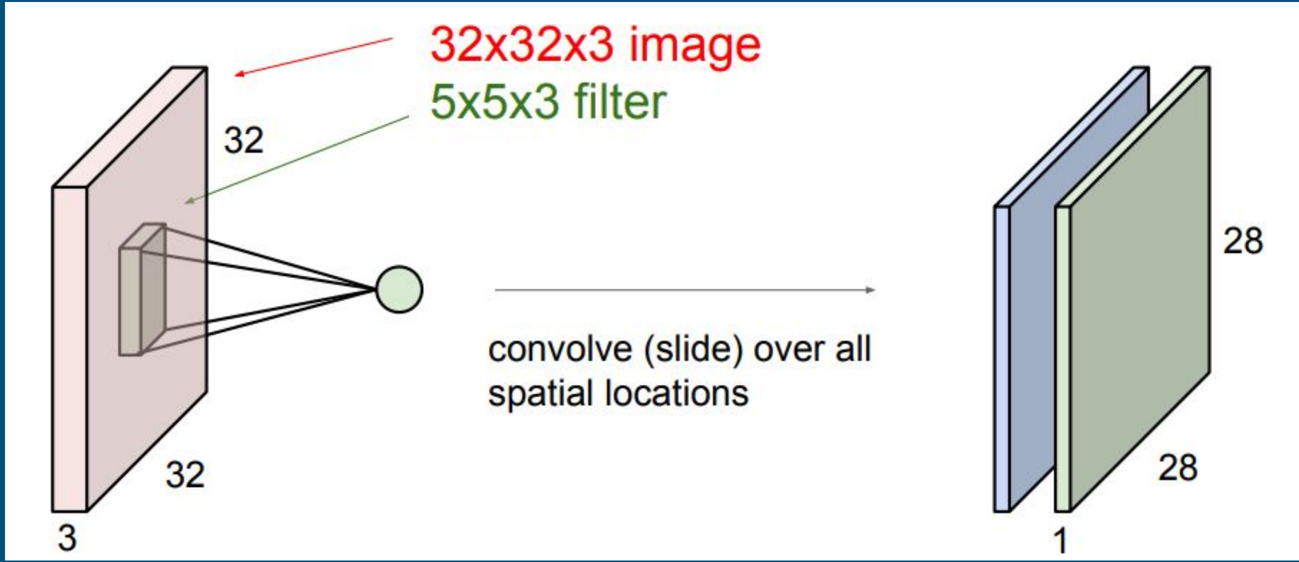
dot product between filter and small 5x5x3 chunk of the image = a number

CNN: Process



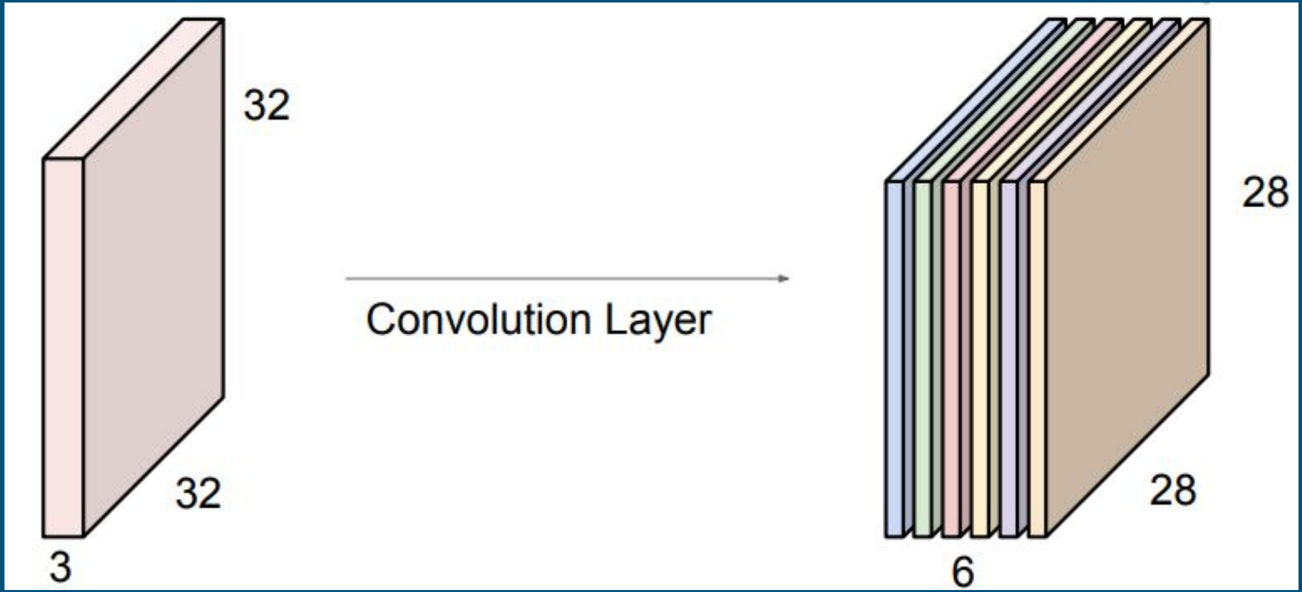
convolve the filter with the image (slide over the image spatially, computing dot products

CNN: Process



consider a **second filter**

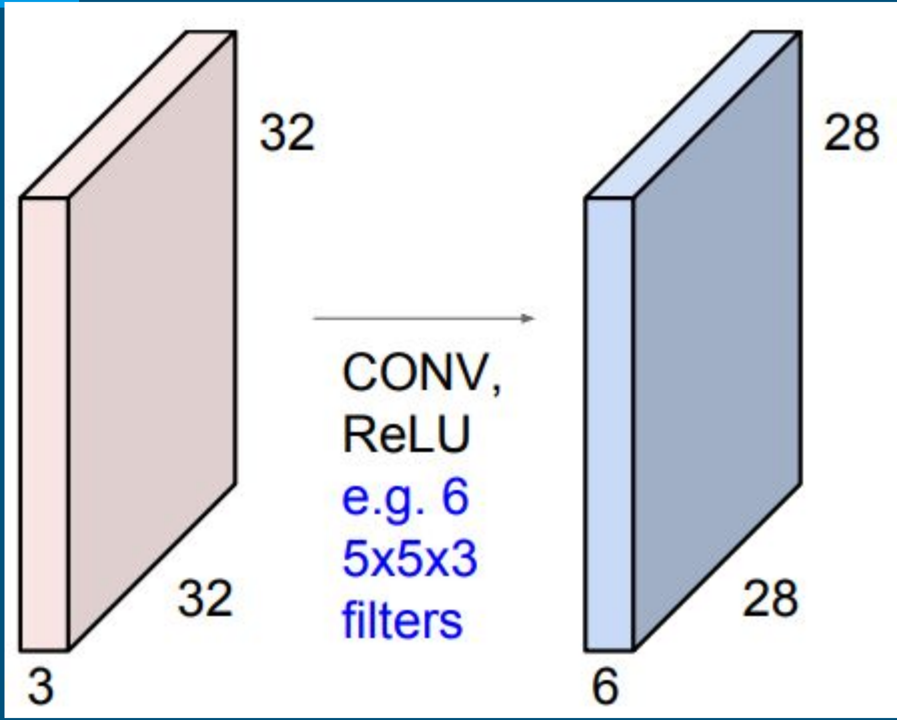
CNN: Process



Ex: say we have 6 of these 5x5 filters

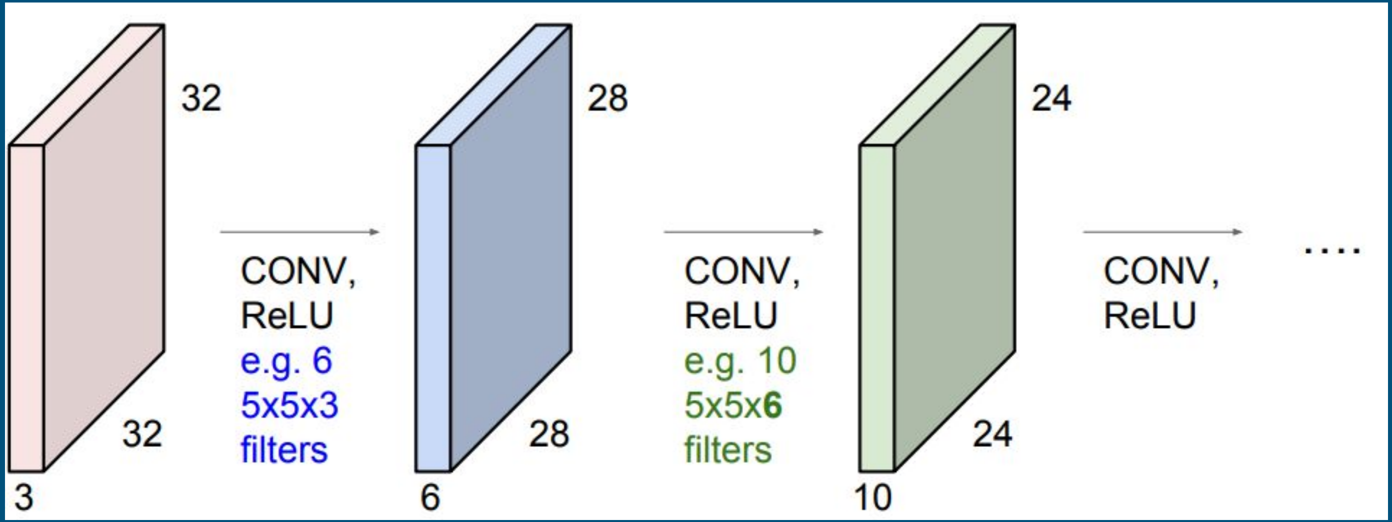
get “new image” of size 28x28x6

CNN: Process



Apply an activation function/nonlinearity eg. ReLU

CNN: Process



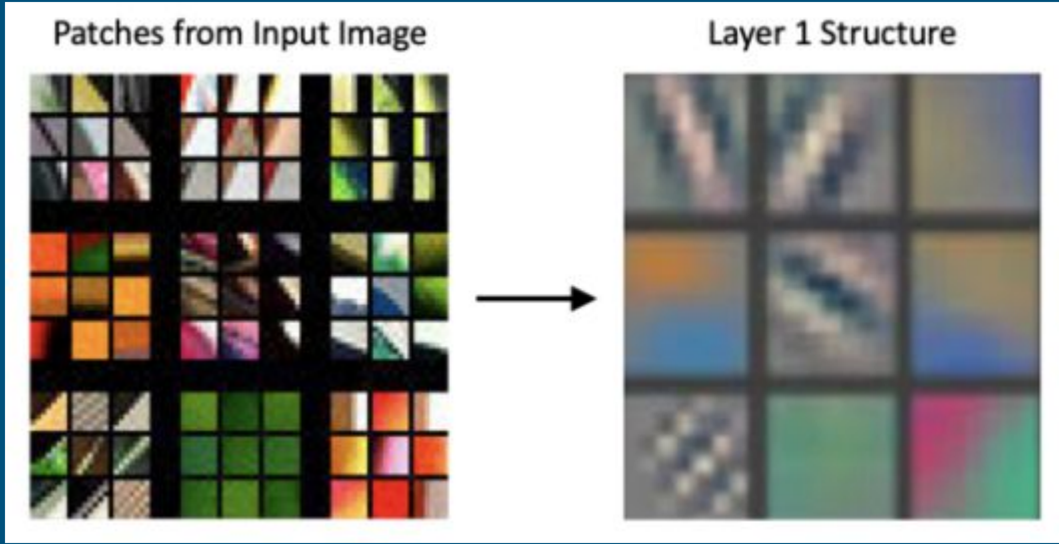
Repeat

Hidden Layer

Hidden Layer

CNN: Learn Features

Each layer learns higher dimensional features

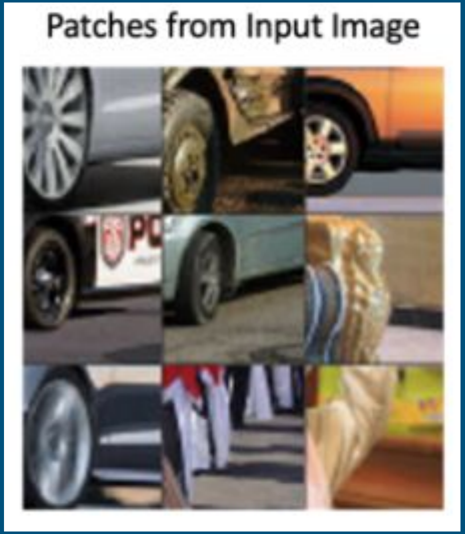


[8]

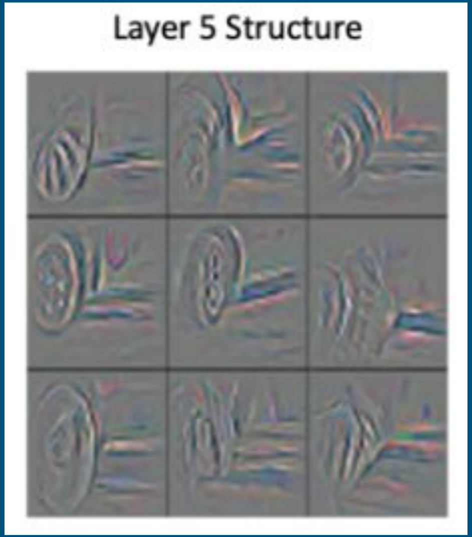
1st layer learns to identify basic structural elements (eg. edges, color blobs)

CNN: Learn Features

Each layer learns higher dimensional features



[8]



[8]

after many layers, it can learn hierarchical structure

Summary

dog

References

[1] Chaudhari, P. (2022). ESE 546: Principles of Deep Learning. Pratik Chaudhari.

[2] Chen, J. (n.d.). What is a neural network?. Investopedia. https://www.investopedia.com/terms/n/neuralnetwork.asp

[3] Jeffares, A. (2020, September 15). Supervised vs unsupervised learning in 2 minutes. Medium. https://towardsdatascience.com/supervised-vs-unsupervised-learning-in-2-minutes-72dad148f242

[4] Lecture 7: Convolutional Neural Networks - Stanford University. (n.d.). http://cs231n.stanford.edu/slides/2016/winter1516_lecture7.pdf

[5] Mitliagkas, Ioannis. IFT 6085 - Lecture 10 Expressivity and Universal Approximation Theorems Part 1, mitliagkas.github.io/ift6085-2020/ift-6085-lecture-10-notes.pdf. Accessed 11 Dec. 2023.

[6] Morimoto, Juliano & Ponton, Fleur. (2021). Virtual reality in biology: could we become virtual naturalists?. Evolution: Education and Outreach. 14. 10.1186/s12052-021-00147-x.

[7] Murphy, K. P. (2023). Probabilistic Machine Learning: Advanced Topics. MIT press.

[8] Understanding convolutional neural network: A complete guide. LearnOpenCV. (2023, November 6). https://learnopencv.com/understanding-convolutional-neural-networks-cnn/

Special thanks to

My DRP mentor:

Léonardo Ferreira Guilhoto

Thank you!

Questions?

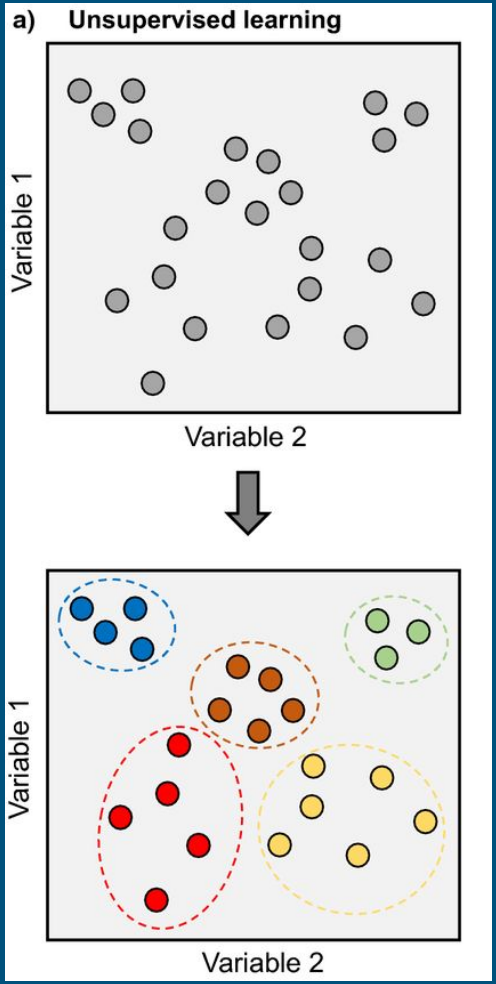
Neural Networks: Additional Terminology

- **Features** at each layer can be studied
- **Hidden layers**: intermediate layers that create features
    - wide: lots of features/neurons
- Weights (ie. parameters): matrices S

Unsupervised Learning

- class labels are unknown
- Goal: Given input data, establish the existence of classes

trying to find picture for example



[6]

Convolutional Neural Network Overview

Feature extraction

Convolutional Neural Network Overview

what is convolution (explain convolution of x with w) - sticky note example

then do nonlinearity (sigma (x * w) thing)

basic idea: linear transformation (convlution) then activation function and keep repeating detecting edges eg in the book pg 47 (kinda like intermediate features)

CNN example with two vectors (like pg 44)

then the bird example: https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/