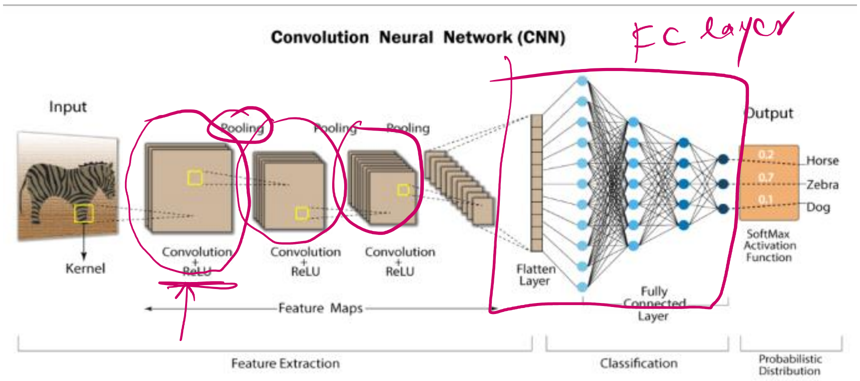
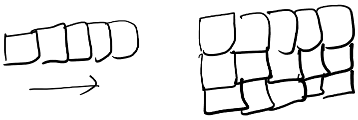
What is a CNN?

17 August 2022 06:47

Convolutional neural networks, also known as convnet, or CNNs, are a special kind of neural network for processing data that has a known grid-like topology like time series data(1D) or images(2D).

→ Convolution
→ Pooling
→ FC layer – ANN





→ ANN → Convolution

CNN → convolution ⊗

ANN → matrix mult.

Inspirations

↳ visual cortex

1998 Yann Lecun → AT&T →

↳ Microsoft → OCR hand writing recog } CNN

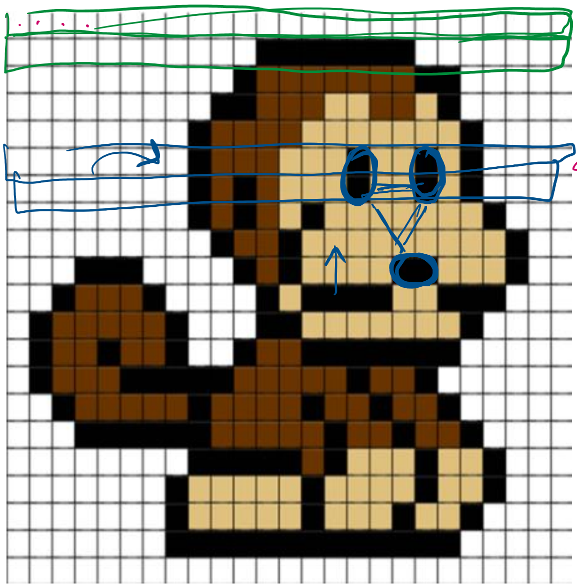
{ facial recog, self driving } CNN RNN

Why not use ANN?

17 August 2022 06:47

pixels 2D grid

CNN >> ANN



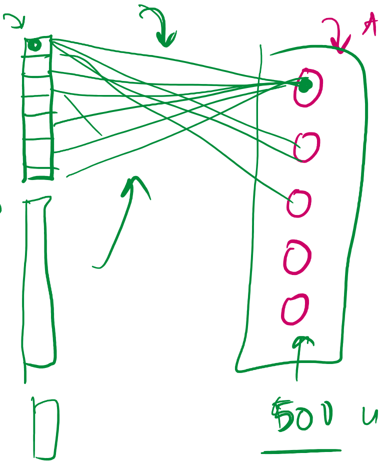
1. High Computation Cost
2. Overfitting →
3. Loss of imp info like spatial arrangement of pixels

1D

40x40 image

MNIST → ANN → 98%

ANN



1000 x 1000

Overfitting

1000x1000x500

40

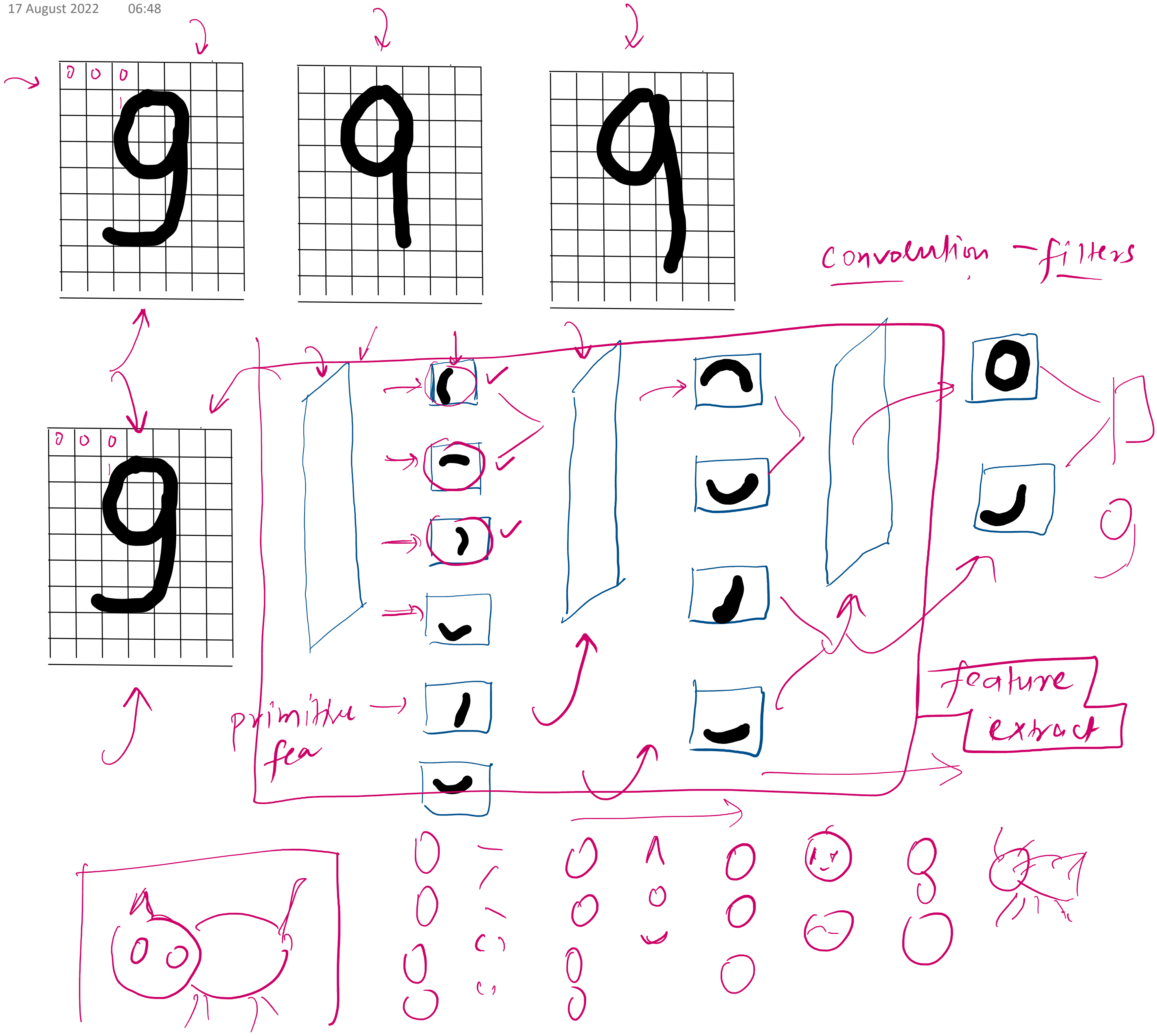
1600 x 100

16 000

1600 pixel

CNN Intuition

17 August 2022 06:48



CNN Applications

17 August 2022 06:48

Roadmap

17 August 2022 06:48

→ Biological connects

→ Convolution
  ↳ Padding + Stride

→ Pooling

→ CNN Architecture

↳ CNN Vs ANN

→ Dog Vs Cat Classifier

↳ Data Augment

↳ Popular CNN Arch
- LeNET
- AlexNET
- VGGNet
- ResNet
- Xception

→ Transfer Learning

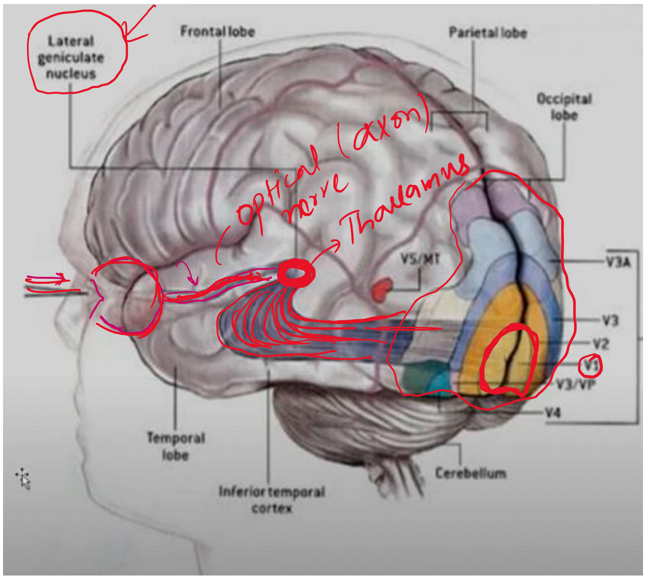
Human Visual Cortex

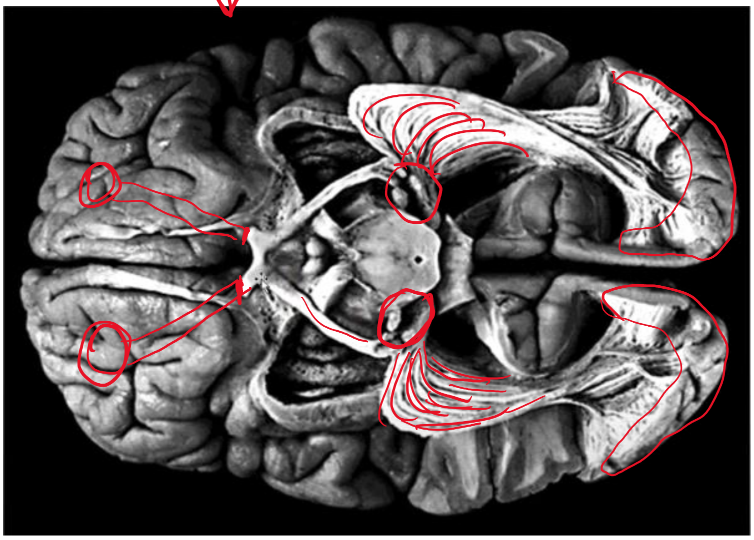
18 August 2022 16:05

2P sheet

primary V cortex

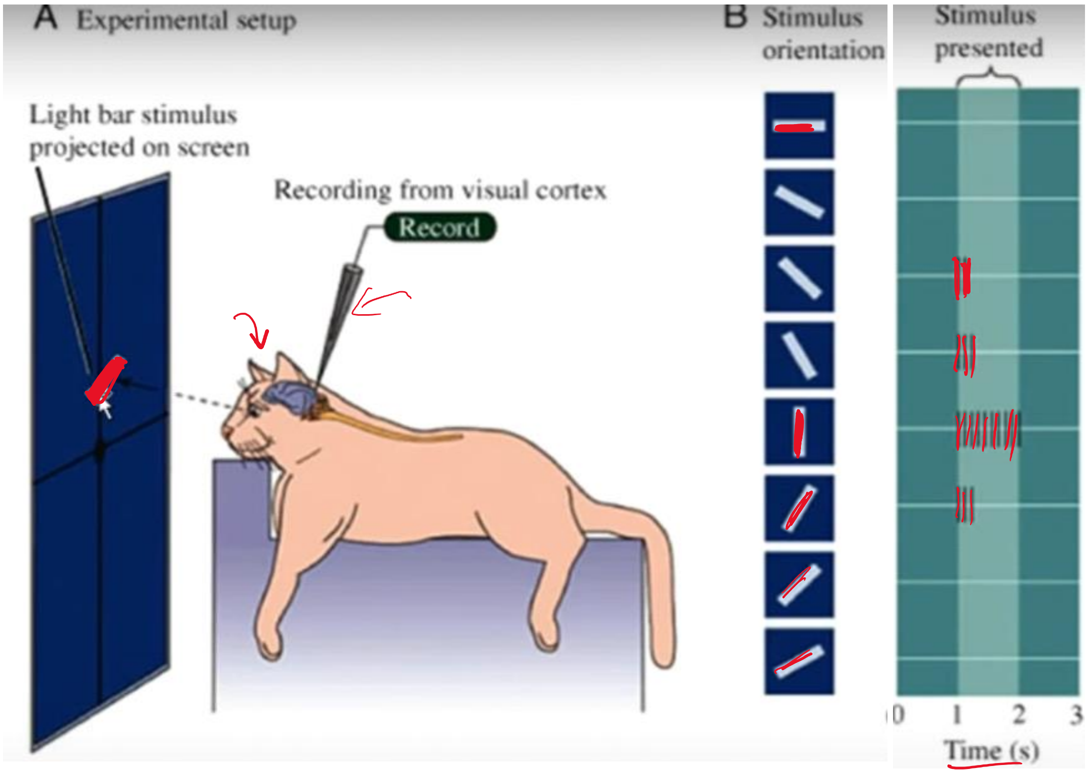
actual brain





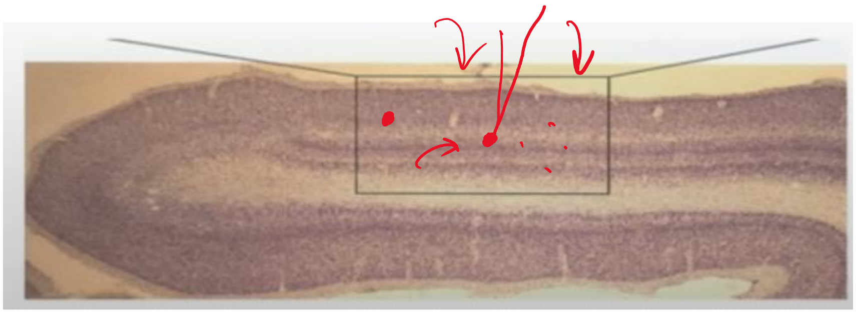
The Experiment

18 August 2022 16:09



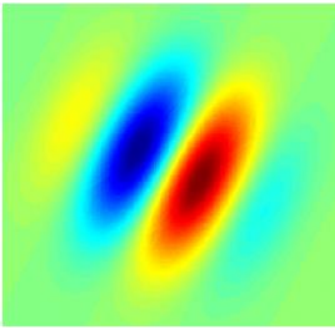
2

response



Conclusion

18 August 2022 17:57



edges

Simple cell → Orientation cell → feature detect

smaller receptive field

Complex cell

bigger receptive field

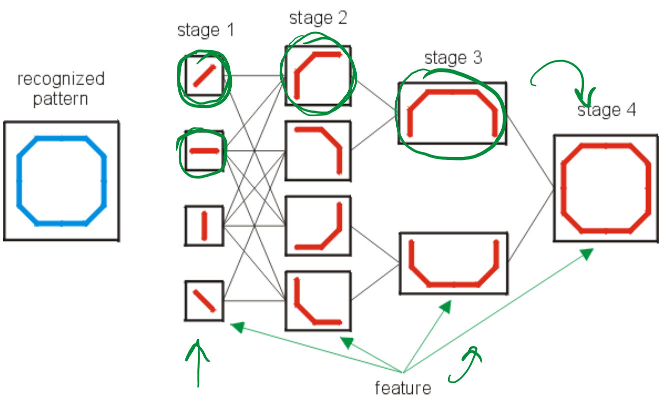
preferred stimuli

edge detection

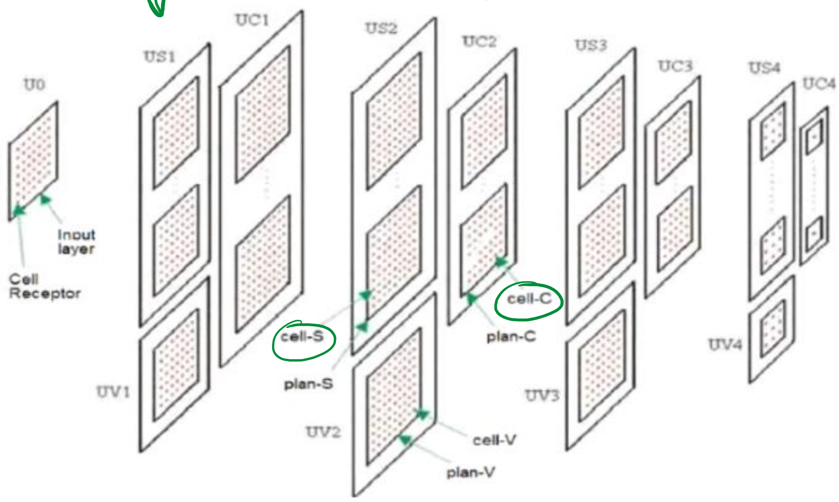
higer fea

Development

18 August 2022 16:15

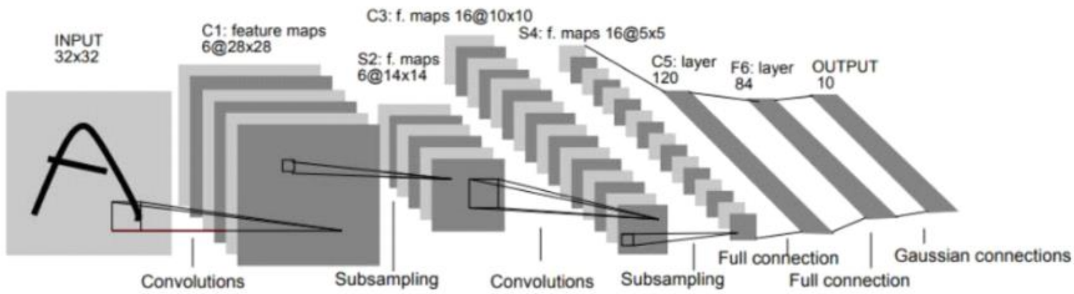


Neocognitron → Fukushima



Yann Lecun → CNN → Backprop Convolution

early CNN



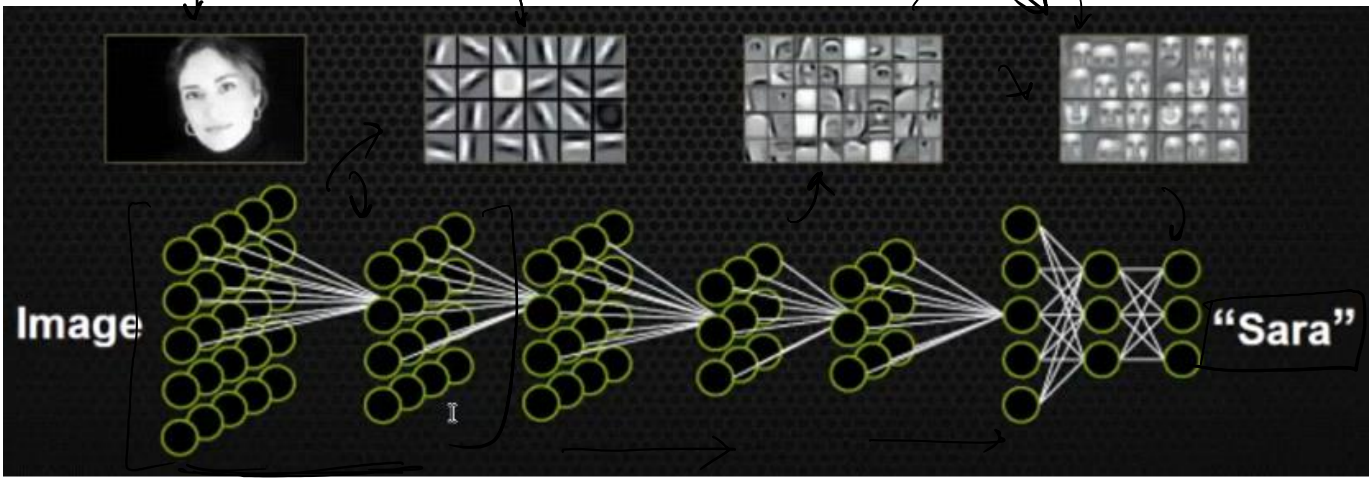
$\boxed{2012}$ → AlexNET → ImageNET → CNNs

Introduction

19 August 2022 16:45

images → Visual Cortex

edges



CNN → NN special → Convolution
Pooling
FC

CNN

Basics of Images

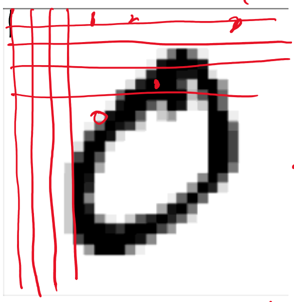
19 August 2022 16:53

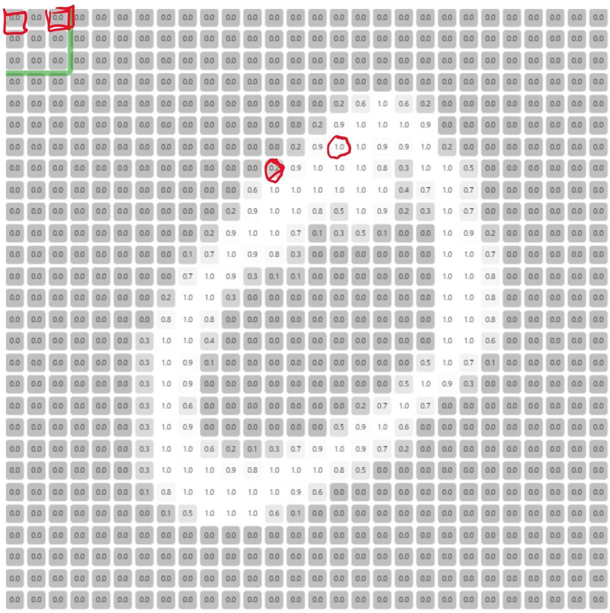
$0-255 \rightarrow \boxed{0-1}$

B/W

1 channel — greyscale → B/W

RGB → colored imgs





28x28 pixels

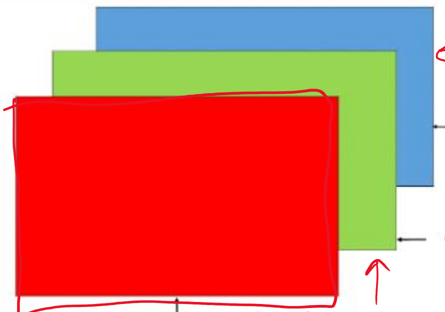
2D arrays →

2D numpy arrays

mnist → 0

(28 x 28)

3 channels → Red Green Blue (primary)



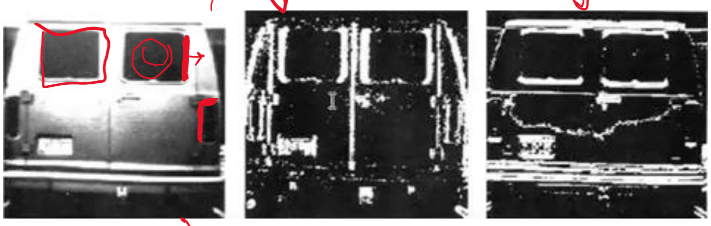
[RGB image is a three layered image]

$\boxed{228 * 228 * 3}$

3 channels

Edge Detection (Convolution Operation)

19 August 2022 16:53



255x1 + 255x1 + ...

horizontal edge detector filter

-1x0 + -1x0 + -1x0

image

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

6x6

*

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

ANN weights

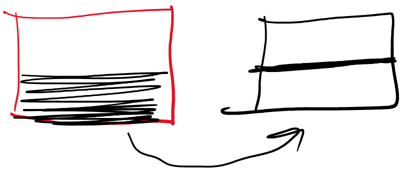
filter/ kernel

Matrix 3x3

=

0	0	0	0
255	255	255	255
255	255	255	255
0	0	0	0

feature map



0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

(6x6)

*

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

(3x3)

=

(4x4)

(28x28) → (26x26) (3x3) → ? (26x26)

nxn mxm → (n-m+1) x (n-m+1)

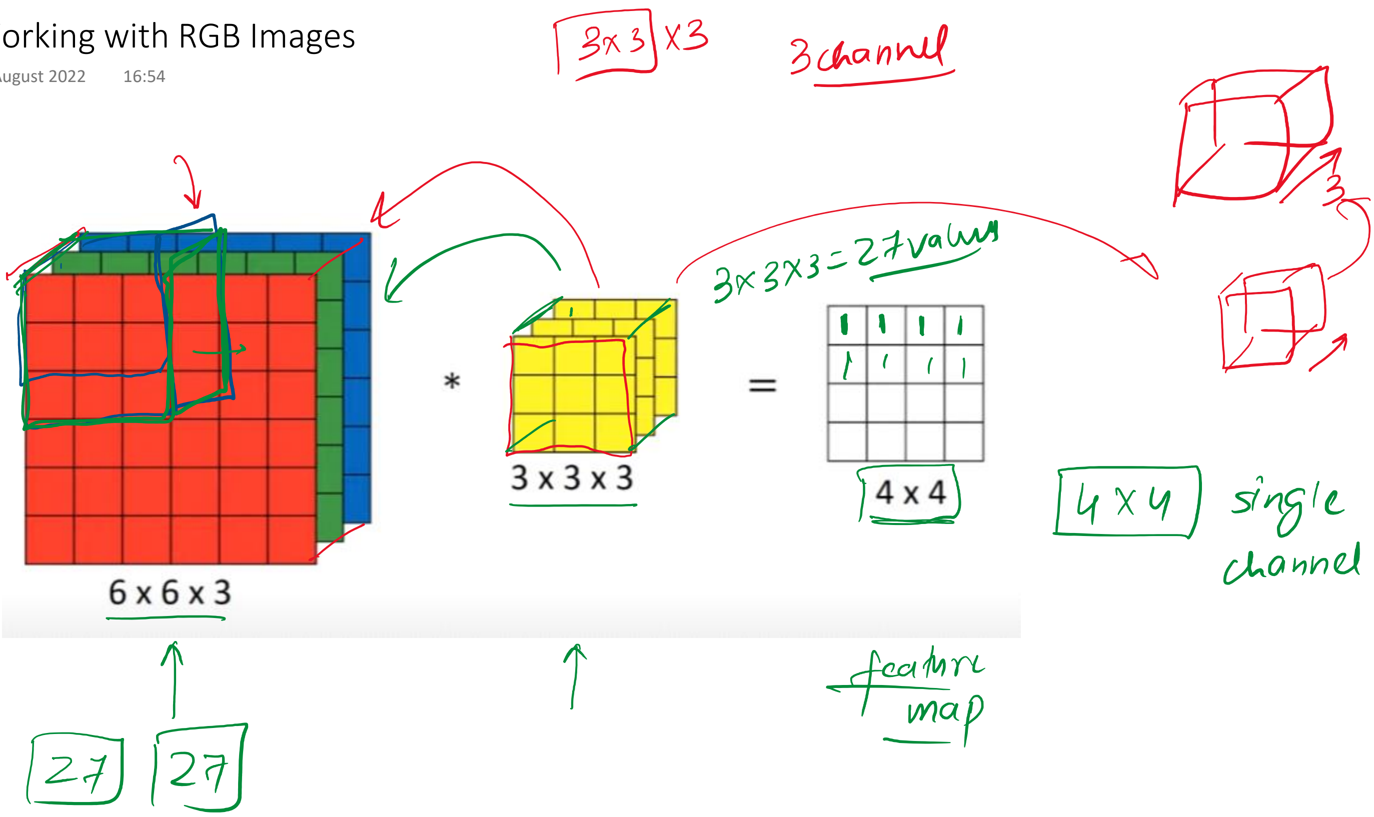
(64x64) (3x3) → (62x62)

Demo

19 August 2022 16:54

Working with RGB Images

19 August 2022 16:54



$$\boxed{m \times m \times c} \quad \boxed{n \times n \times c} \longrightarrow \underset{\text{single channel}}{\underline{(m-n+1)\,(m-n+1)}}$$

Multiple Filters

23 August 2022 08:24

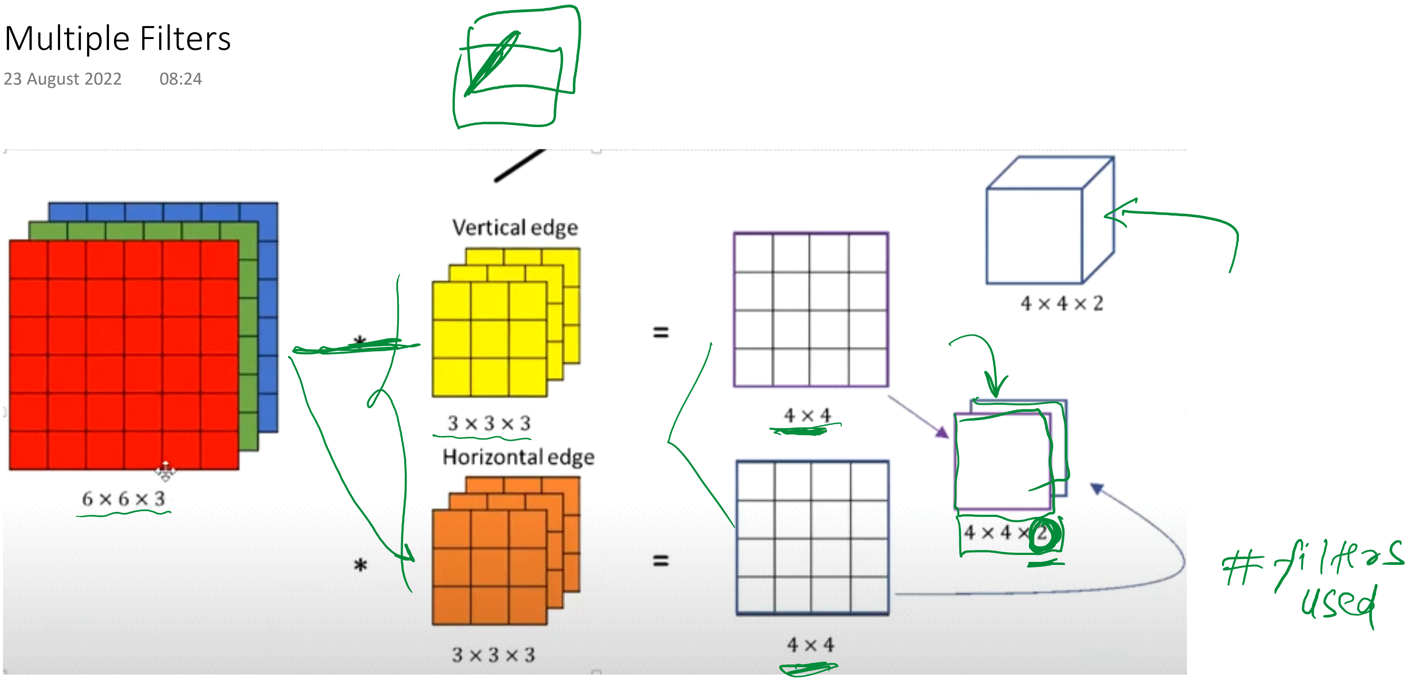


Image taken from Andrew NG's lecture

10 → $\boxed{4 \times 4 \times 10}$

Problem with Convolution

26 August 2022 14:25

What is Padding?

26 August 2022 14:26

$$\begin{bmatrix}7&2&3&3&8\\4&5&3&8&4\\3&3&2&8&4\\2&8&7&2&7\\5&4&4&5&4\end{bmatrix} * \left\{\begin{bmatrix}1&0&-1\\1&0&-1\\1&0&-1\end{bmatrix}\right\} = \begin{bmatrix}6&&\\&&\\&&\end{bmatrix}$$

7x1+4x1+3x1+
2x0+5x0+3x0+
3x-1+3x-1+2x-1
= 6

$n \times n$

5×5

$\boxed{7 \times 7}$

$f \times f$

3×3

$\rightarrow (n-f+1)\,(n-f+1)$

$(5-3+1) = 3 \times 3$

$n-f+1 = n$

$n-f+1 = 5$

$n-3+1=5 \Rightarrow n=8-1$

$n=7$

$$\begin{bmatrix}0&0&0&0&0&0&0\\0&7&2&3&3&8&0\\0&4&5&3&8&4&0\\0&3&3&2&8&4&0\\0&2&8&7&2&7&0\\0&5&4&4&5&4&0\\0&0&0&0&0&0&0\end{bmatrix}$$

7×7

convolution

Zero padding

$5 \times 5 \rightarrow 3 \times 3$

$5 \times 5 \longrightarrow 3 \times 3$

$(n - f + 1)$

$\downarrow$

$(n + 2p - f + 1)$

$5 + 2(1) - 3 + 1$

$= 7 - 3 + 1 = 5$

Keras

Valid

Same

Strides

27 August 2022 08:23

Stride = 1

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

7x7

0	0	1
1	0	0
0	1	1

3x3

3 x3

(1,1)

→ right

↓ bottom

→ result feature

Stride = (2,2)

Stride = 2 → $\frac{7-3}{2} + 1$

$2+1=3$

s

$(n-f+1) \longrightarrow \left[\frac{n-f}{s} + 1\right] \to p = p$

$\left[\frac{n+2p-f}{2} + 1\right] \longrightarrow$ strided convolution

$\frac{7+2-3}{2} + 1 = [4 \times 4]$

$\frac{n-f}{2}$ $\quad \frac{6-3}{2} \quad 1.5 = 1 + 1 = 2$

Special case

Stride=2

1	6	9	10	2	8	5
2	5	1	8	4	2	4
3	7	4	9	10	3	7
9	8	3	6	7	9	3
8	0	9	4	7	2	1

$\boxed{2 \times 3}$

$\frac{6 \times 7}{3 \times 3}$

$\left[\frac{n-f}{s} + 1\right]$

$1.9 \to 1$

$1.1 \to 1$

floor

9	8	3	6	7	9	3
8	0	9	4	7	2	1
9	10	12	6	9	8	0

7x6

$$\left\lfloor \frac{\quad}{S} + 1 \right\rfloor$$

floor

$$\left[2 \quad \frac{6-3}{2} + 1 = 1.5 + 1 \right.$$

$$1 + 1 = 2$$

$$\frac{7-3}{2} + 1 = 2 + 1 = 3$$

Why Strides are required?

27 August 2022 08:24

1) High level features

2) Computing →

Keras → stride

$$\left[\frac{n+2p-f}{s}+1\right]$$

$$\frac{28+2-3}{2}+1$$

$$13.5+1$$

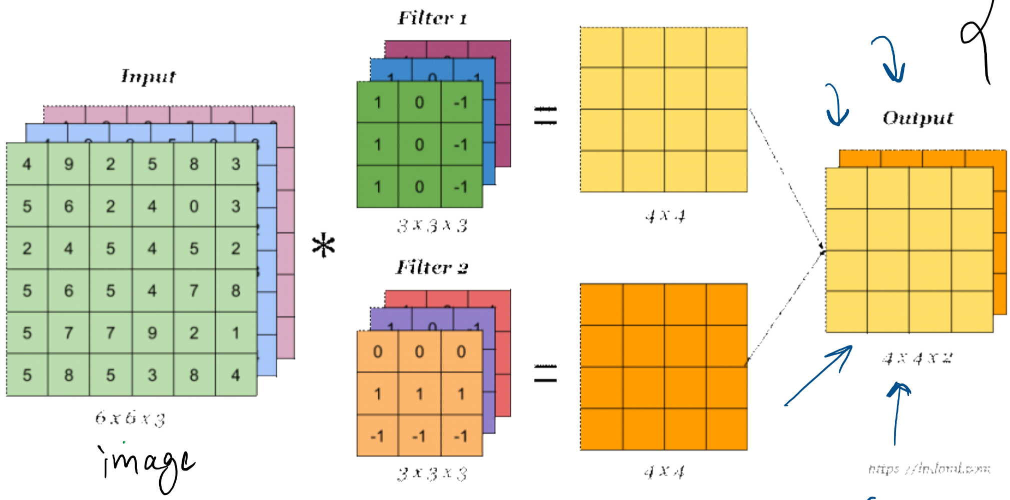
$$13+1=14$$

The Problem with Convolution

01 September 2022 09:55

stride=2

1) Memory issue
2) Tranlation variance



reduce size of feature map

pooling

image

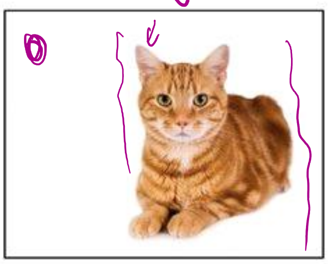
228 x 228 * 3

filters (100) → feature map → $(226 * 226) * 100 * 32$ → 19 MB

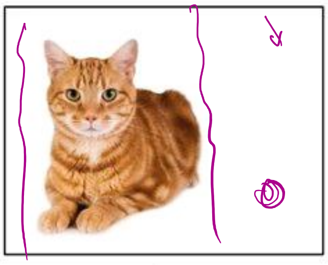
1.5 GB

1 training → 100

Translation Variance



Cat



Cat

features

{ location dependent }

location

→ pooling { down sample your feature map }

Pooling

01 September 2022 09:55

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

*

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

= relu (4x4)

-	-	-	-
-			

feature map (non-linear)

Max pooling
Min pooling
Avg pooling
L2 pooling
Global pooling

max pooling

3	1	1	3
2	5	0	2
1	4	2	1
4	7	2	4

4x4

(size) → (2,2)
stride → (2)
type → Max

receptive field

5	3
7	4

feature map

2x2

low level details

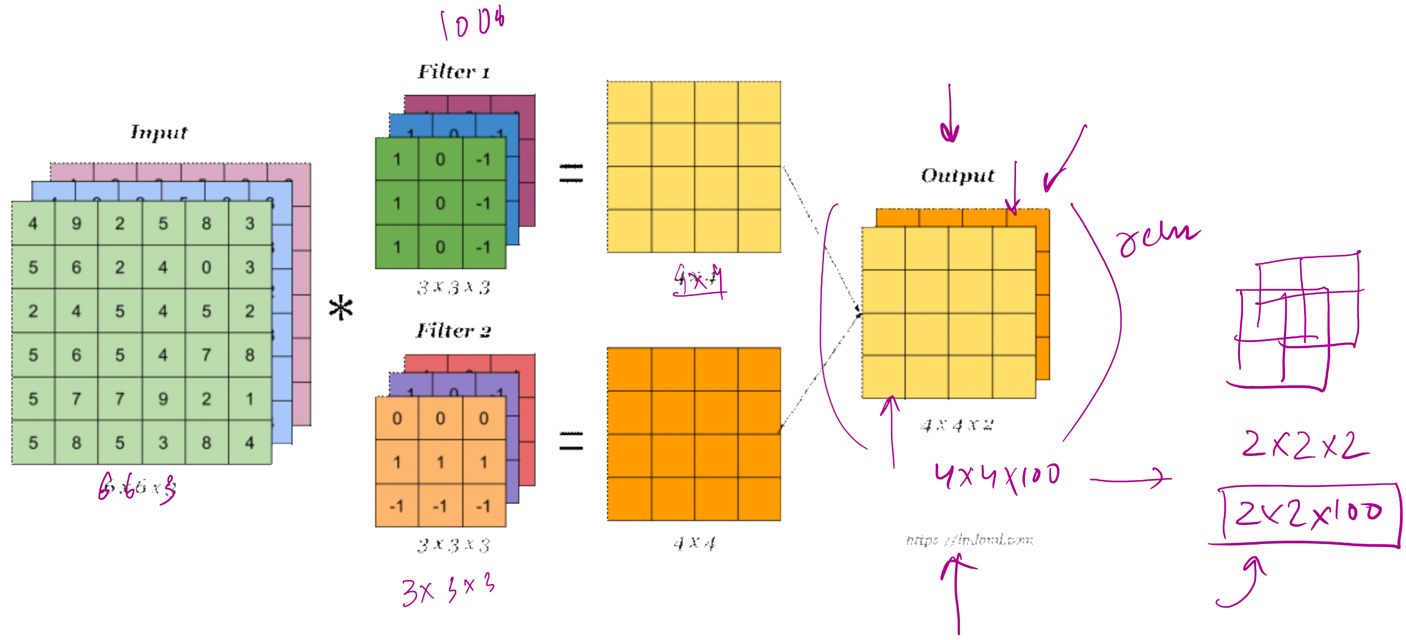
eliminat

Demo

01 September 2022 09:57

Pooling on Volumes

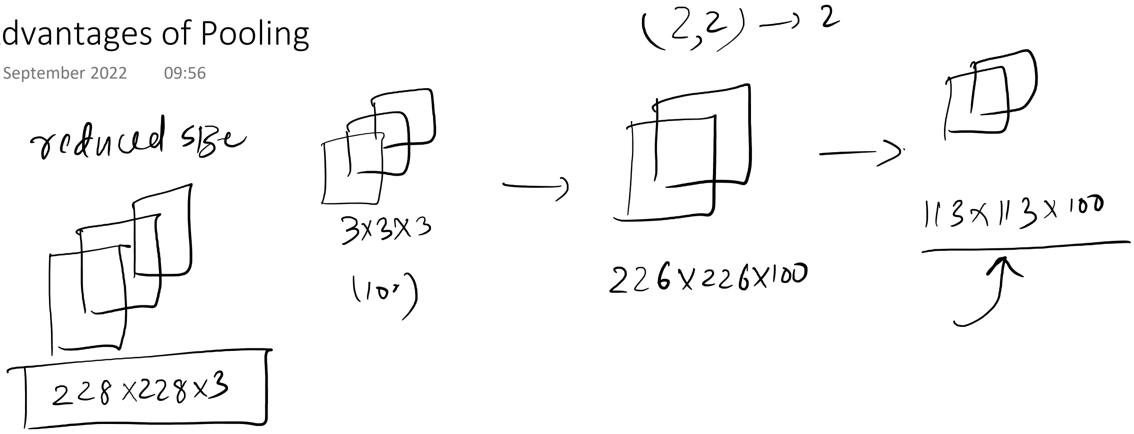
01 September 2022 09:56



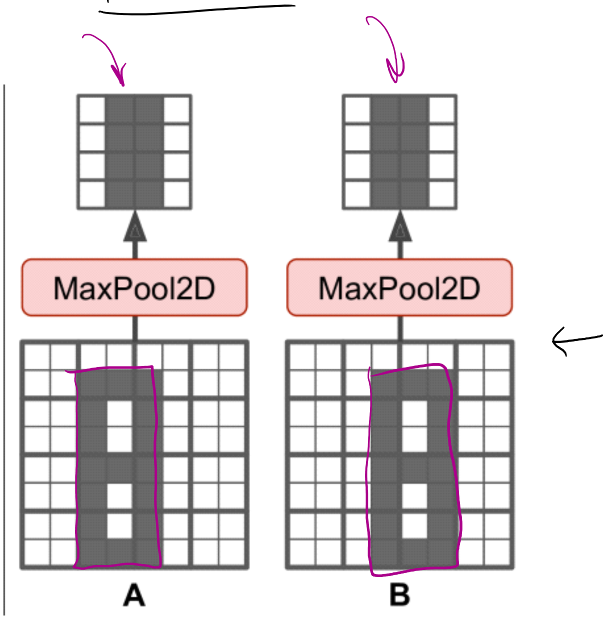
Advantages of Pooling

01 September 2022 09:56

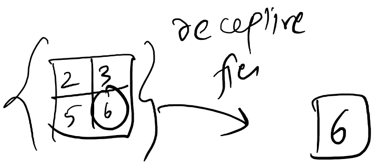
1) reduced size

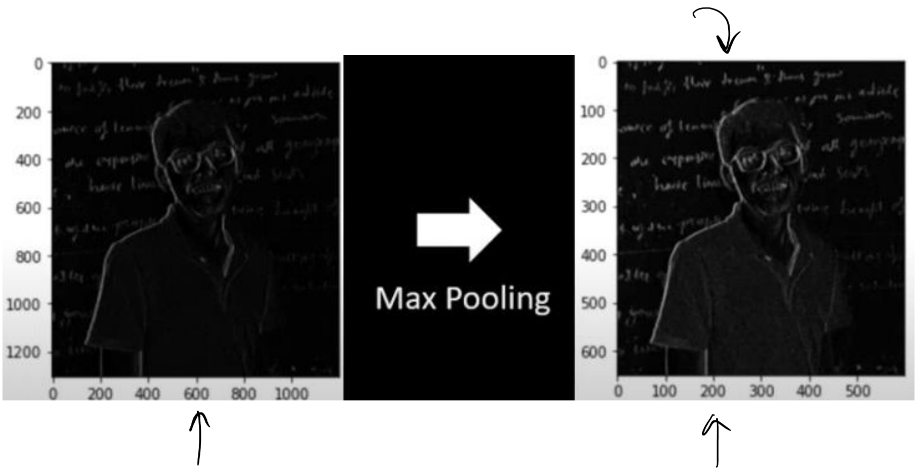


2) Translation invariance



3) Enhanced Features
(only in case of Max pooling)





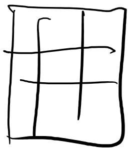
4) No need of training

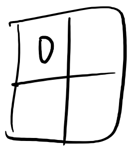
min

faster

max pooling → aggregate

avg pooling





→ (2x2)

→ 2

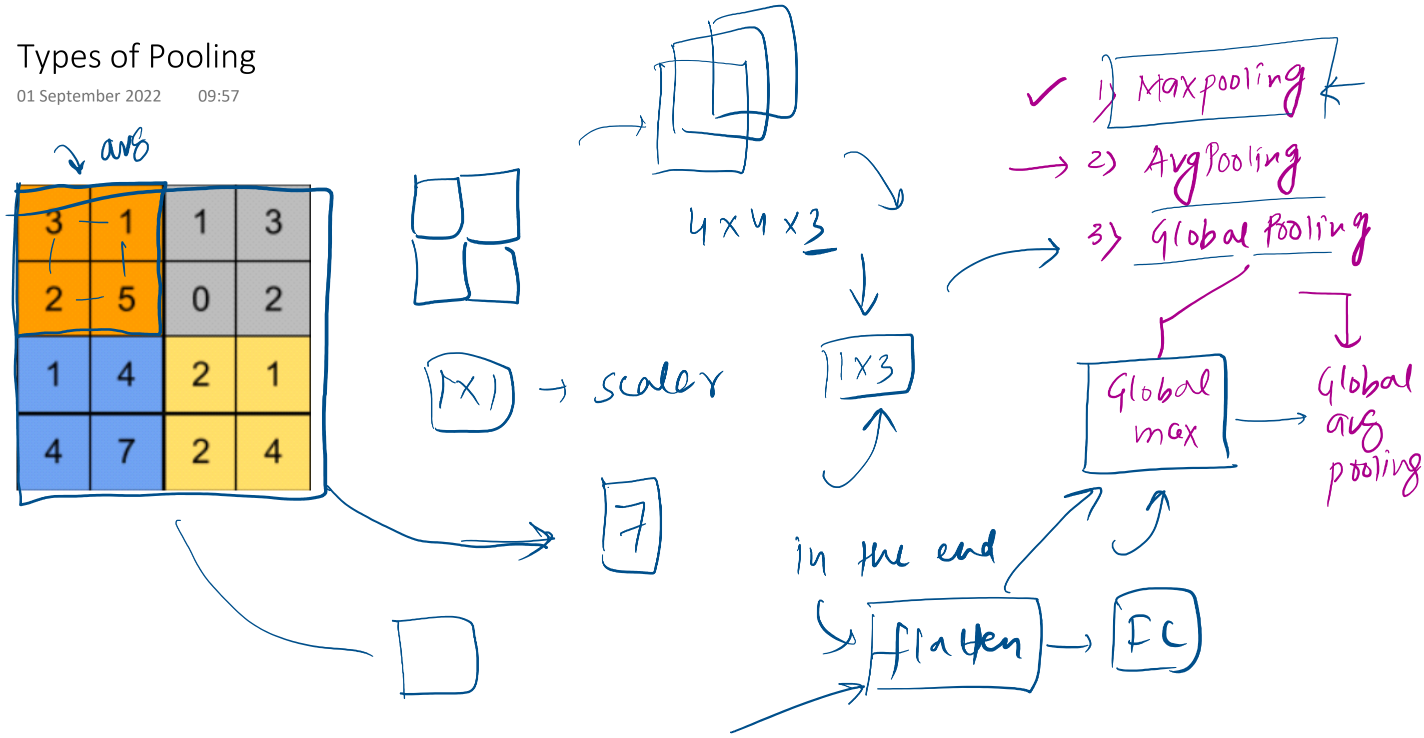
→ type

Keras Code

01 September 2022 09:56

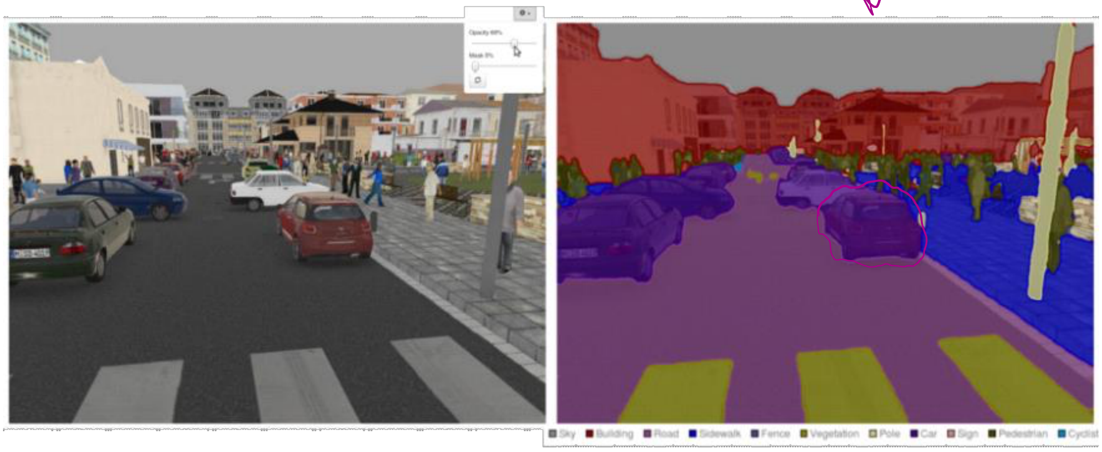
Types of Pooling

01 September 2022 09:57



Disadvantages of Pooling

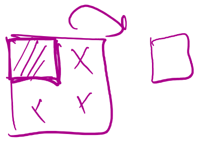
01 September 2022 09:57



lose a lot of info

4x4 → 16

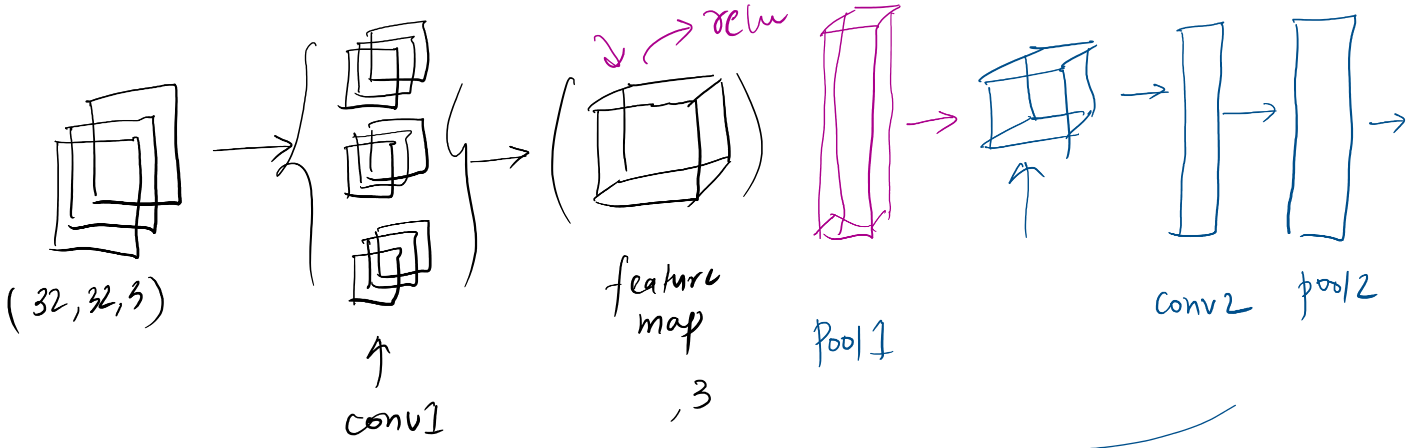
2x2 → 4

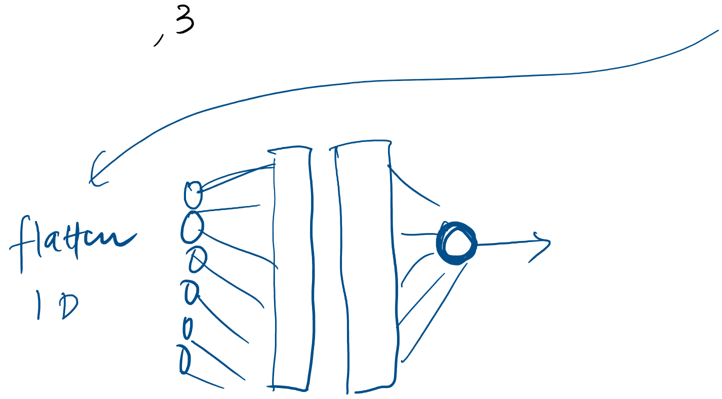


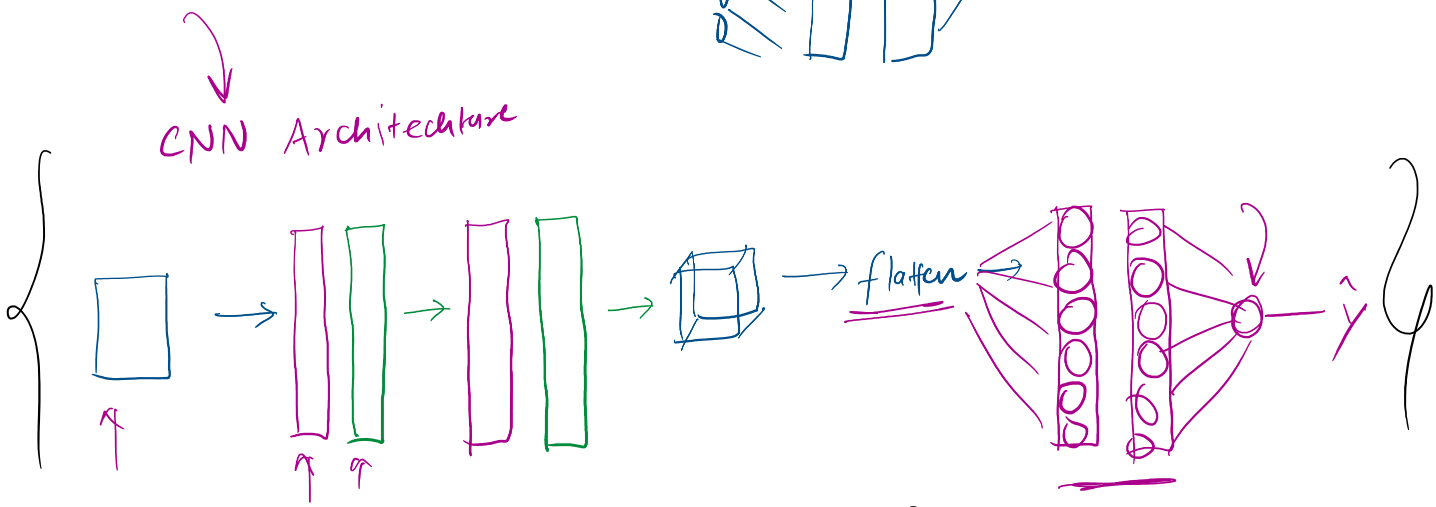
CNN Architecture

02 September 2022 10:55

1) Convolution 2) Padding /stride 3) Pooling







# conv # filters	stride padding	FC nodes # FC layer	activation dropout	Batch Norm

ImageNET

1) [LeNET] → Yann LeCuNN
2) AlexNET
3) GoogLeNET
4) VggNET
5) ResNTE
6) Inception

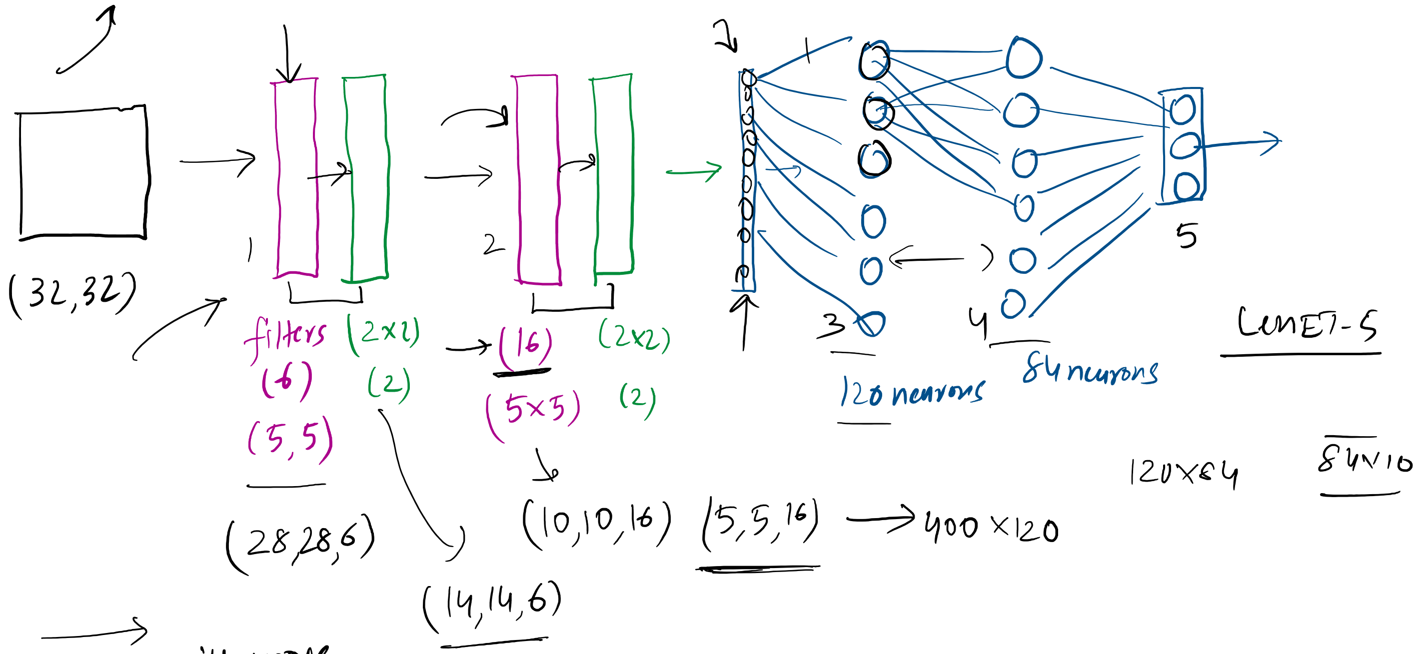
3> GoogLeNET

LeNet

02 September 2022 10:55

Yann LeCun → 1989 → 1998 → LeNET → CNN → US Navy Postal service

LeNET-5



data will move

- AlexNET
- VggNET
- ResNET
- Inception

Guidelines

02 September 2022 10:58

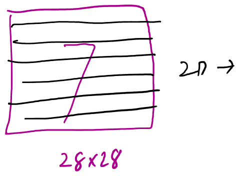
Keras Code

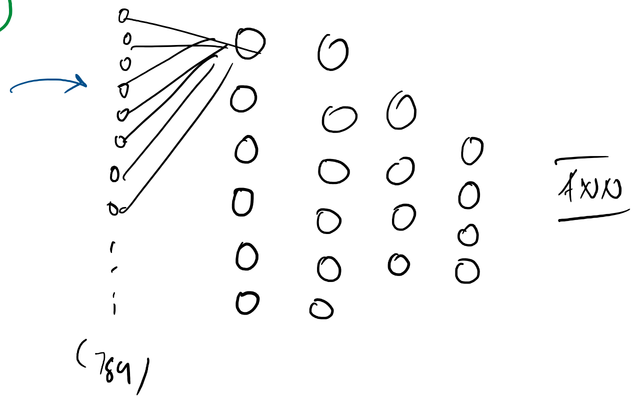
02 September 2022 14:58

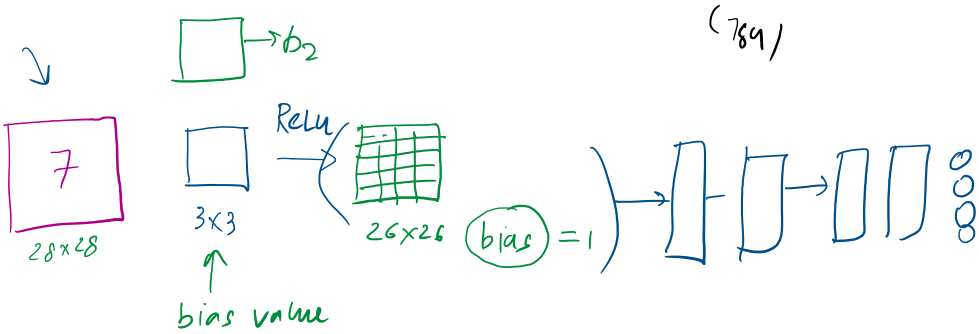
CNN Vs ANN

06 September 2022 10:00

1) Computation Cost →
2) Overfitting →
3) loss of imp features like spatial arrangement of pixels

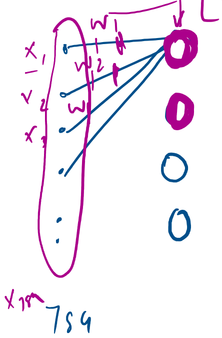




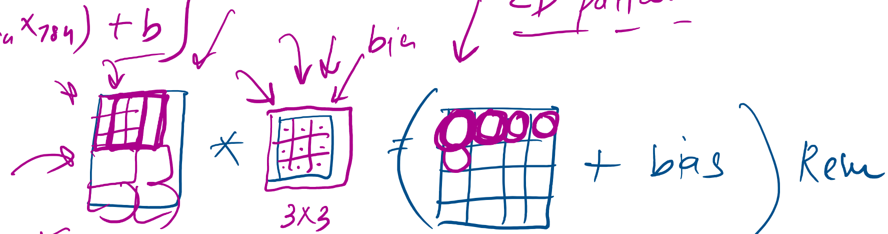


similarity ANN CNN

2D pattern

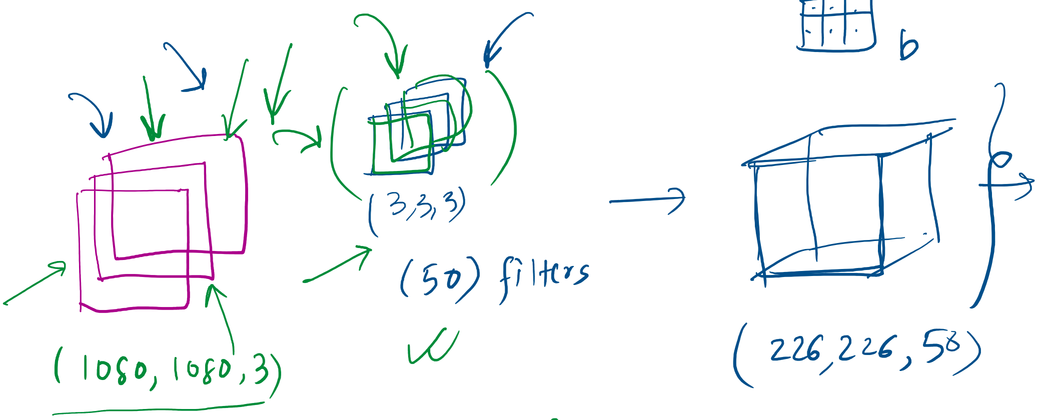


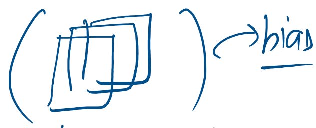
relu $[(w_1x_1 + w_2x_2 + \ldots + w_{784}x_{784}) + b]$



relu $[(x_1w_1 + x_2w_2 + \ldots x_9w_9) + \text{bias}]$ ← Training Backprop

b





(3,3,3) weights

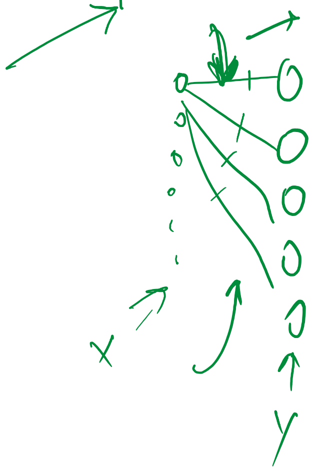
↳ 27 weights × 50

{learnable parameters}

$\boxed{1400}$

$\boxed{228, 228, 3}$ → 288 x 228 x 3   1080 x 1080 x 3

1350 wts
+ 50
1400



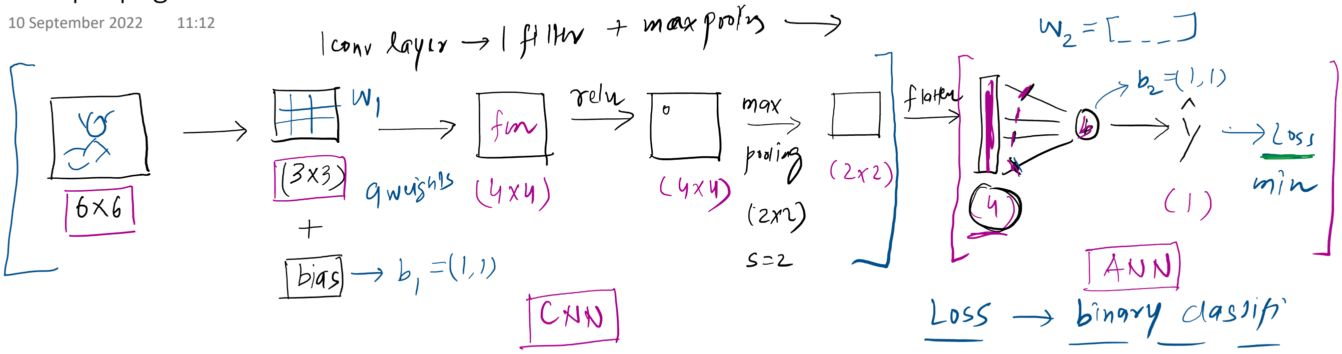
X * Y →

Overfitting AND

Backprop in CNN

Backpropagation in CNN

10 September 2022 11:12



Loss → binary classif

$$L = -y_i \log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)$$

$A_2 = \hat{y}$

Trainable Parameters

$W_1 = (3,3)$ $\quad W_2 = (1,4)$ $\quad = 15$ trainable parameters

$b_1 = (1,1)$ $\quad b_2 = (1,1)$

Logical Flow

Forward Prop

$$Z_1 = \text{conv}(X, W_1) + b_1$$
$$A_1 = \text{relu}(Z_1)$$
$$P_1 = \text{maxpool}(A_1)$$
$$F = \text{flatten}(P_1)$$
$$Z_2 = W_2 F + b_2$$
$$A_2 = \sigma(Z_2)$$

Gradient Descent

$$W_1 = W_1 - \eta \frac{\partial L}{\partial W_1}$$
$$b_1 = b_1 - \eta \frac{\partial L}{\partial b_1}$$
$$W_2 = W_2 - \eta \frac{\partial L}{\partial W_2}$$
$$b_2 = b_2 - \eta \frac{\partial L}{\partial b_2}$$

Loss is minimized

$$\frac{\partial L}{\partial W_2}$$

$$X \rightarrow \circledast \rightarrow Z_1 \xrightarrow{\text{relu}} A_1 \xrightarrow{\text{max pool}} P_1 \xrightarrow{\text{flatten}} F \xrightarrow{W_2, b_2} Z_2 \xrightarrow{\text{sigmoid}} A_2(\hat{y}) \rightarrow L$$

W_1, b_1

$$\boxed{\frac{\partial L}{\partial W_2}} = \frac{\partial L}{\partial A_2} \times \frac{\partial A_2}{\partial Z_2} \times \frac{\partial Z_2}{\partial W_2}$$

$$\boxed{\frac{\partial L}{\partial b_2}} = \frac{\partial L}{\partial A_2} \times \frac{\partial A_2}{\partial Z_2} \times \frac{\partial Z_2}{\partial b_2}$$

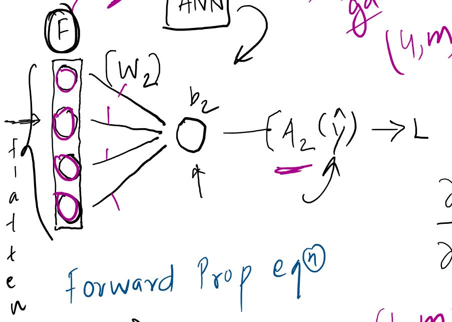
Backprop

$$\boxed{\frac{\partial L}{\partial w_1}} = \frac{\partial L}{\partial A_2} \times \frac{\partial A_2}{\partial Z_2} \times \frac{\partial Z_2}{\partial F} \times \boxed{\frac{\partial F}{\partial P_1}} \times \boxed{\frac{\partial P_1}{\partial A_1}} \times \frac{\partial A_1}{\partial Z_1} \times \boxed{\frac{\partial Z_1}{\partial w_1}}$$

$$\boxed{\frac{\partial L}{\partial b_1}} = \frac{\partial L}{\partial A_2} \times \frac{\partial A_2}{\partial Z_2} \times \frac{\partial Z_2}{\partial F} \times \frac{\partial F}{\partial P_1} \times \frac{\partial P_1}{\partial A_1} \times \frac{\partial A_1}{\partial Z_1} \times \boxed{\frac{\partial Z_1}{\partial b_1}}$$

Backprop
- → Convolution
- → Flatten
- → Max pooling

(4,1) 4,m m ANN → mb gd (4,m)



$$\boxed{\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial A_2} \, \frac{\partial A_2}{\partial Z_2} \, \frac{\partial Z_2}{\partial w_2}} \qquad \boxed{\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial A_2} \, \frac{\partial A_2}{\partial Z_2} \, \frac{\partial Z_2}{\partial b_2}}$$

Forward Prop eqn

$$Z_2 = W_2 F + b_2$$
$$A_2 = \sigma(Z_2)$$

$$\frac{\partial L}{\partial a_2} = \frac{\partial}{\partial a_2}\left[-y_i \log(a_2) - (1-y_i)\log(1-a_2)\right]$$

(1,m)

$Y = (1,m)$

$$= \frac{-y_i}{a_2} + \frac{(1-y_i)}{(1-a_2)} = \frac{-y_i(1-a_2) + a_2(1-y_i)}{a_2(1-a_2)}$$

$$\frac{\partial L}{\partial a_2} = \frac{-y_i + \cancel{y_i a_2} + a_2 - \cancel{a_2 y_i}}{a_2(1-a_2)} = \frac{(a_2 - y_i)}{a_2(1-a_2)}$$

W_2 update

Shape =

$$\frac{\partial A_2}{\partial Z_2} = \sigma(Z_2)[1-\sigma(Z_2)] = a_2[1-a_2]$$

$$\boxed{\frac{\partial Z_2}{\partial W_2} = F} \qquad \frac{\partial Z_2}{\partial b_2} = 1$$

(1,4)

$$\boxed{\frac{\partial L}{\partial w_2}} = \frac{(a_2 - y_i)}{\cancel{a_2(1-a_2)}} \times \cancel{a_2(1-a_2)} \times F = (a_2 - y_i)\,F = (A_2 - Y)F^T$$

$(A_2 - Y)$: (1,1) (1,1) → (1,1) ; F^T: (4,1)

(1,1)(1,4) → (1,4)

$$\frac{\partial L}{\partial b_2} = \frac{(a_2 - y_i)}{\cancel{a_2(1-a_2)}} \; \cancel{a_2(1-a_2)} \times 1 = (A_2 - Y)$$

m images

$$\boxed{\boxed{\frac{\partial L}{\partial w_2}} = (A_2 - Y)F^T} \qquad \boxed{\frac{\partial L}{\partial b_2} = (A_2 - Y)}$$

$$\frac{\partial L}{\partial w_2} = (A_2 - Y)F' \qquad \frac{\partial L}{\partial b_2}$$

$(1,m) - (1,m)$

$(1,m)\ (m,4) \rightarrow \boxed{(1,4)}$

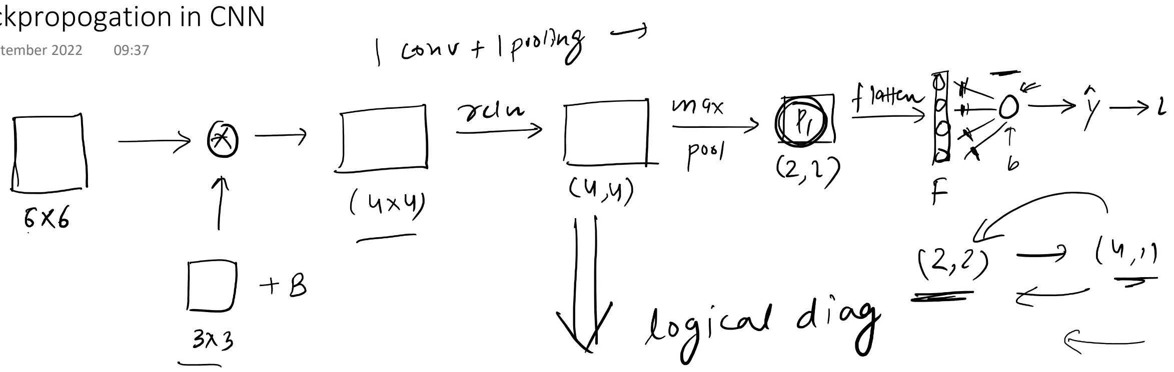
1 image

batch of image

$w_2 \rightarrow (1,4)$

Backpropogation in CNN

15 September 2022 09:37



Forward Prop

$Z_1 = conv(X, W_1) + b_1$

$A_1 = relu(Z_1)$

$P_1 = maxpool(A_1)$

$\boxed{F = flatten(P_1)}$

flatten

$Z_2 = W_2 F + b_2$

$A_2 = \sigma(Z_2)$

$L = \frac{1}{m}\sum_{i=1}^{m}\left[-y_i \log(A_2) - (1-y_i)\log(1-A_2)\right]$

$(A_2 - y)$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial A_2} \times \frac{\partial A_2}{\partial Z_2}\,\frac{\partial Z_2}{\partial F}\,\frac{\partial F}{\partial P_1}\,\frac{\partial P_1}{\partial A_1}\,\frac{\partial A_1}{\partial Z_1}\,\frac{\partial Z_1}{\partial W_1}$$

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial A_2}\,\frac{\partial A_2}{\partial Z_2}\,\frac{\partial Z_2}{\partial F}\,\frac{\partial F}{\partial P_1}\,\frac{\partial P_1}{\partial A_1}\,\frac{\partial A_1}{\partial Z_1}\,\frac{\partial Z_1}{\partial b_1}$$

1 2 3 4 5 6

reshape(P_1.shape)

6 derivatives

$\left[\frac{\partial Z_2}{\partial F}\right] = W_2 \rightarrow$

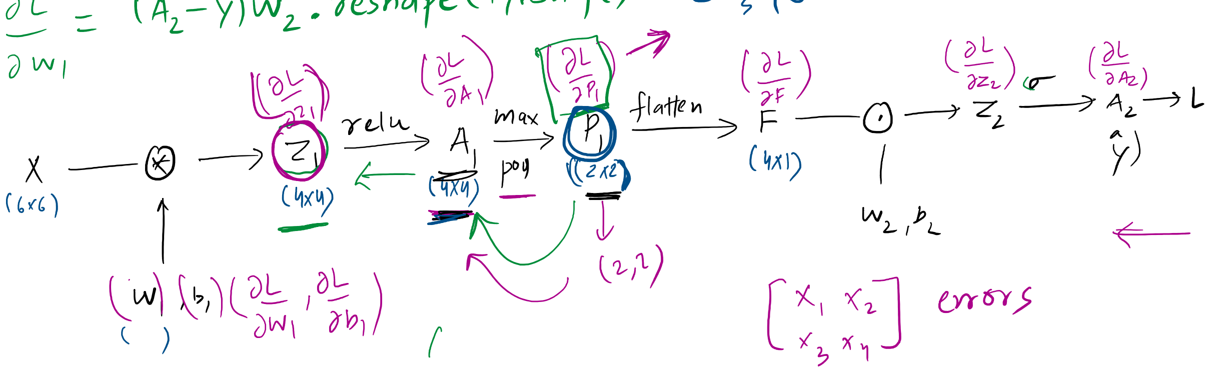
Shape ? $\rightarrow$ (F)

$\frac{\partial F}{\partial P_1}$ no trainable parameters

$conv\left(X, \frac{\partial L}{\partial Z_1}\right)$

$$\frac{\partial L}{\partial W_1} = (A_2 - y)W_2 \cdot reshape(P_1.shape)$$

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



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

$\frac{\partial L}{\partial A_1} = (4,4)$

$\frac{\partial L}{\partial A_1} = (4,4)$

Flatten → no trainable parameters

A_1

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \quad \begin{bmatrix} 9 & 10 \\ 11 & 12 \end{bmatrix} \begin{bmatrix} 13 & 14 \\ 15 & 16 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 8 \\ 12 & 16 \end{bmatrix}$$

$\hat{y}$ no role

L

$$\frac{\partial L}{\partial P_1} = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix} \; 2\times 2$$

(4×4)

$\frac{\partial L}{\partial A_1}$

2x2 error → (4×4)

$$\frac{\partial L}{\partial A_1} \rightarrow \begin{bmatrix} 0 & x_1 & x_2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & x_3 & x_4 & 0 \end{bmatrix}$$

A, P

$$\frac{\partial L}{\partial W_2} = \boxed{\frac{\partial L}{\partial A_2}\frac{\partial A_2}{\partial Z_2}\frac{\partial Z_2}{\partial F}\frac{\partial F}{\partial P_1}}\boxed{\frac{\partial P_1}{\partial A_1}\frac{\partial A_1}{\partial Z_1}}\frac{\partial Z_1}{\partial W_1}$$

$(A_2 - Y)W_2.\text{reshape}(P_1.\text{shape})$ $\quad \frac{\partial L}{\partial A_1}$

$$\boxed{\frac{\partial L}{\partial A_1}} = \begin{cases} \frac{\partial L}{\partial P_{1xy}}, & \text{if } A_{mn} \text{ is the max element} \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{\partial L}{\partial b_1} = \boxed{\underbrace{\frac{\partial L}{\partial A_2}\frac{\partial A_2}{\partial Z_2}\frac{\partial Z_2}{\partial F}\frac{\partial F}{\partial P_1}}_{\frac{\partial L}{\partial P_1}}\frac{\partial P_1}{\partial A_1}}\boxed{\frac{\partial A_1}{\partial Z_1}}\frac{\partial Z_1}{\partial b_1}$$

$\boxed{\frac{\partial L}{\partial A_1}}$

$$\frac{\partial A_1}{\partial Z_1} = \begin{cases} 1 & \text{if } Z_{1xy} > 0 \\ 0 & \text{if } Z_{1xy} < 0 \end{cases}$$

$\left(\frac{\partial L}{\partial Z_1}\right)$

Convolution ↳ Backprop → max pooling, Flatten

Backprop on Convolution

$$\frac{\partial L}{\partial Z_1} = \begin{bmatrix} \frac{\partial L}{\partial Z_{11}} & \frac{\partial L}{\partial Z_{12}} \\ \frac{\partial L}{\partial Z_{21}} & \frac{\partial L}{\partial Z_{22}} \end{bmatrix}$$

$X \rightarrow \circledast \rightarrow Z_1$ $(2,2)$

$(3,3)$ $\quad W_1, b_1$

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial Z_1} \times \boxed{\frac{\partial Z_1}{\partial b_1}}$$

$\left(\frac{\partial L}{\partial Z_1}\right) \rightarrow (2,2)$

$$Z_1 = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$$

w_1, b_1

$(2,2)$

$\begin{bmatrix} z_{21} & z_{22} \end{bmatrix}$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \circledast \quad W_1 = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

b_1

$$\boxed{\frac{\partial L}{\partial b_1}} = \frac{\partial L}{\partial z_1} \times \frac{\partial z_1}{\partial b_1} = \boxed{\frac{\partial L}{\partial z_{11}} \times \frac{\partial z_{11}}{\partial b_1} + \frac{\partial L}{\partial z_{12}} \times \frac{\partial z_{12}}{\partial b_1} + \frac{\partial L}{\partial z_{21}} \times \frac{\partial z_{21}}{\partial b_1} + \frac{\partial L}{\partial z_{22}} \times \frac{\partial z_{22}}{\partial b_1}}$$

$$= \left(\frac{\partial L}{\partial z_{11}} + \frac{\partial L}{\partial z_{12}} + \frac{\partial L}{\partial z_{21}} + \frac{\partial L}{\partial z_{22}} \right) = \text{sum}\left(\frac{\partial L}{\partial z_1} \right)$$

$$\boxed{\frac{\partial L}{\partial b_1}} = \text{sum}\left(\frac{\partial L}{\partial z_1} \right) \rightarrow \text{scalar}$$

bias

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \quad W_1 = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} + b_1$$

$X \rightarrow \circledast \rightarrow z_1$

(3×3) $\quad w_1, b_1$ (2×2) $\quad \left(\frac{\partial L}{\partial z_1}\right)$

$$\frac{\partial L}{\partial W_1} = \begin{bmatrix} \frac{\partial L}{\partial W_{11}} & \frac{\partial L}{\partial W_{12}} \\ \frac{\partial L}{\partial W_{21}} & \frac{\partial L}{\partial W_{22}} \end{bmatrix} \qquad \frac{\partial L}{\partial z_1} = \begin{bmatrix} \frac{\partial L}{\partial z_{11}} & \frac{\partial L}{\partial z_{12}} \\ \frac{\partial L}{\partial z_{21}} & \frac{\partial L}{\partial z_{22}} \end{bmatrix}$$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial z_1} \times \frac{\partial z_1}{\partial W_1}$$

$$\frac{\partial L}{\partial W_{11}} = \frac{\partial L}{\partial z_{11}} \times \frac{\partial z_{11}}{\partial W_{11}} + \frac{\partial L}{\partial z_{12}} \times \frac{\partial z_{12}}{\partial W_{11}} + \frac{\partial L}{\partial z_{21}} \times \frac{\partial z_{21}}{\partial W_{11}} + \frac{\partial L}{\partial z_{22}} \times \frac{\partial z_{22}}{\partial W_{11}}$$

$$\frac{\partial L}{\partial W_{12}} = \frac{\partial L}{\partial z_{11}} \times \frac{\partial z_{11}}{\partial W_{12}} + \frac{\partial L}{\partial z_{12}} \times \frac{\partial z_{12}}{\partial W_{12}} + \frac{\partial L}{\partial z_{21}} \times \frac{\partial z_{21}}{\partial W_{12}} + \frac{\partial L}{\partial z_{22}} \times \frac{\partial z_{22}}{\partial W_{12}}$$

$$\frac{\partial L}{\partial z_{12}} \times \frac{\partial z_{12}}{} + \frac{\partial L}{} \times \frac{\partial z_{21}}{} + \frac{\partial L}{\partial z_{22}} \times \frac{\partial z_{22}}{\partial W_{21}}$$

$$\frac{\partial L}{\partial w_{21}} = \frac{\partial L}{\partial z_{11}} \times \frac{\partial z_{11}}{\partial w_{21}} + \frac{\partial L}{\partial z_{12}} \times \frac{\partial z_{12}}{\partial w_{21}} + \frac{\partial L}{\partial z_{21}} \times \frac{\partial z_{21}}{\partial w_{21}} + \frac{\partial L}{\partial z_{22}} \times \frac{\partial z_{22}}{\partial w_{21}}$$

$$\frac{\partial L}{\partial w_{22}} = \frac{\partial L}{\partial z_{11}} \times \frac{\partial z_{11}}{\partial w_{22}} + \frac{\partial L}{\partial z_{12}} \times \frac{\partial z_{12}}{\partial w_{22}} + \frac{\partial L}{\partial z_{21}} \times \frac{\partial z_{21}}{\partial w_{22}} + \frac{\partial L}{\partial z_{22}} \times \frac{\partial z_{22}}{\partial w_{22}}$$

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial L}{\partial z_{11}} x_{11} + \frac{\partial L}{\partial z_{12}} x_{12} + \frac{\partial L}{\partial z_{21}} x_{21} + \frac{\partial L}{\partial z_{22}} x_{22}$$

$$\frac{\partial L}{\partial w_{12}} = \frac{\partial L}{\partial z_{11}} x_{12} + \frac{\partial L}{\partial z_{12}} x_{13} + \frac{\partial L}{\partial z_{21}} x_{22} + \frac{\partial L}{\partial z_{22}} x_{23}$$

$$\frac{\partial L}{\partial w_{21}} = \frac{\partial L}{\partial z_{11}} x_{21} + \frac{\partial L}{\partial z_{12}} x_{22} + \frac{\partial L}{\partial z_{21}} x_{31} + \frac{\partial L}{\partial z_{22}} x_{32}$$

$$\frac{\partial L}{\partial w_{22}} = \frac{\partial L}{\partial z_{11}} x_{22} + \frac{\partial L}{\partial z_{12}} x_{23} + \frac{\partial L}{\partial z_{21}} x_{32} + \frac{\partial L}{\partial z_{22}} x_{33}$$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \qquad \frac{\partial L}{\partial z_1} = \begin{bmatrix} \frac{\partial L}{\partial z_{11}} & \frac{\partial L}{\partial z_{12}} \\ \frac{\partial L}{\partial z_{21}} & \frac{\partial L}{\partial z_{22}} \end{bmatrix}$$

$$\frac{\partial L}{\partial w_1} = \mathrm{conv}\left(X, \frac{\partial L}{\partial z_1}\right)$$

$$\frac{\partial L}{\partial w_1} = \mathrm{conv}\left(X, \frac{\partial L}{\partial z_1}\right)$$

$$\frac{\partial L}{\partial b_1} = \mathrm{sum}\left(\frac{\partial L}{\partial z_1}\right)$$

Why use Pretrained models?

03 October 2022 12:52

1) Data hungry → image data → labelled data → cat dog → google

labelling

10000 photos

manually

pinterest

dog

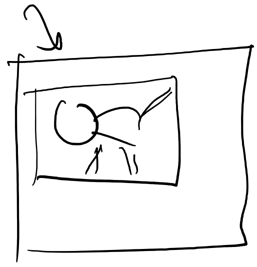
cat

2) Time → model → training

Pretrained model

ImageNET Dataset

03 October 2022 12:36



Visual Database of images (Why What and How)

Why

2006 → $\boxed{\text{Fei fei Li}}$ → $\boxed{\text{model and algorithms}}$

$\boxed{\text{1.4 million}}$ images → 1.4 crore image → 20,000 categories → labelling

1 million images → 10 lakh → bounding box

↳ object local

→ dog
→ breed
→ white

How → crowd sourcing → Amazon mechanical Turk

↘ Dataset → Deep learning

ImageNET challenge

ILSVRC

03 October 2022 12:38

Imagenet challenge → 2010 → Dataset → subset → 1 million → 10

14 crore

↳ 20,000 classes

↳ 1000 classes

Machine learning

→ feature extraction → SIFT, HOG

100 image

2010 → 28%

2011 → 25%

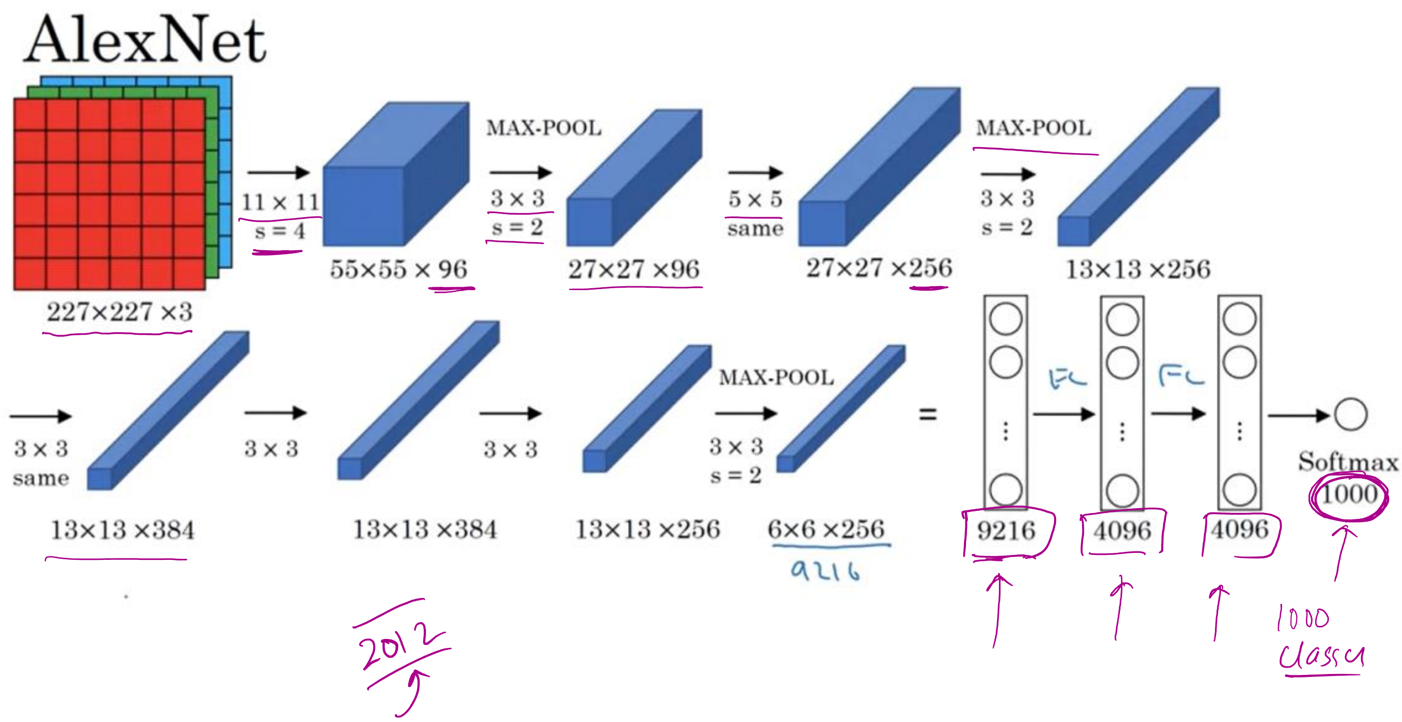
2012 → 16% → CNN

2013 →

2012 → DL

↓

AlexNET → CNN → relu, Deep, GPU



[Krizhevsky et al., 2012. ImageNet classification with deep convolutional neural networks]

Andrew Ng

Famous Architectures

03 October 2022 12:39

2010 → ML model → 28%

2011 → ML Model → 25%

2012 → Alex NET → 16.4%

2013 → ZFNET → 11.7%

2014 → VGG → 7.3% → famous →

2015 → Google NET → 6.7%

2016 → ResNET → 3.5% → humans → 5% → 100

more layers

Idea of Pretrained Models

03 October 2022 12:39

VggNET
ResNET

→ ImageNET → $\boxed{1000}$

Keras → pretrained → use

use them

Keras Demo

03 October 2022 12:40

Problem with training your own model

10 October 2022 10:48

1) Data hungry → labelled → 10,000 → google

10,000 → cat/dog → manual labour

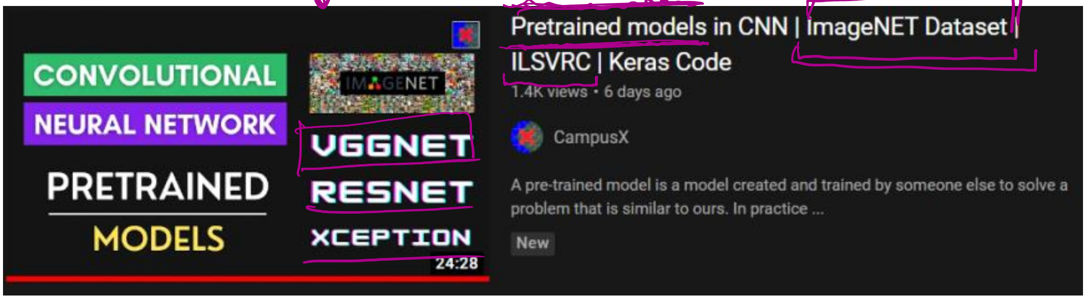
2) lot of time → days/weeks/

Using Pretrained Models

10 October 2022 10:48

2010 2012

images → daily objects
animal



1.4 million

1000 classes

chair
bread
dog breed

VGG ResNET

Deep learning

chair
bread
dog
tomato

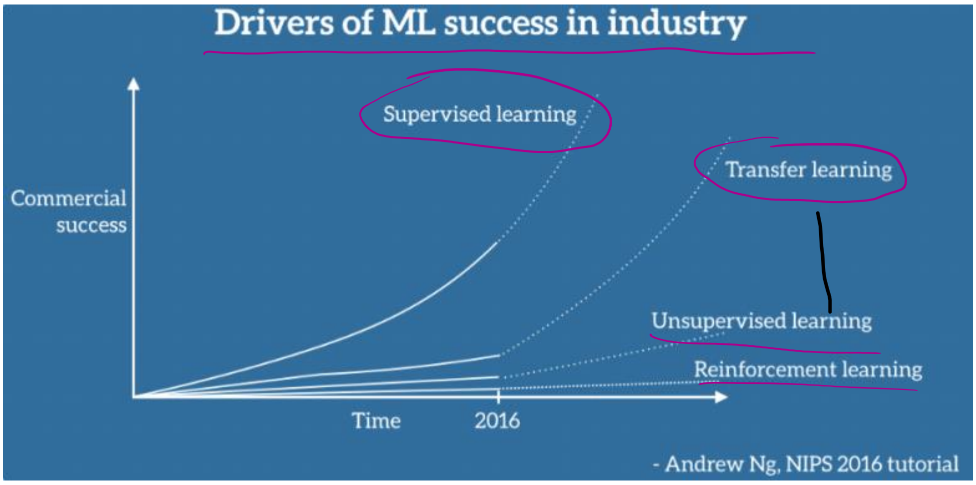
→ Transfer learning

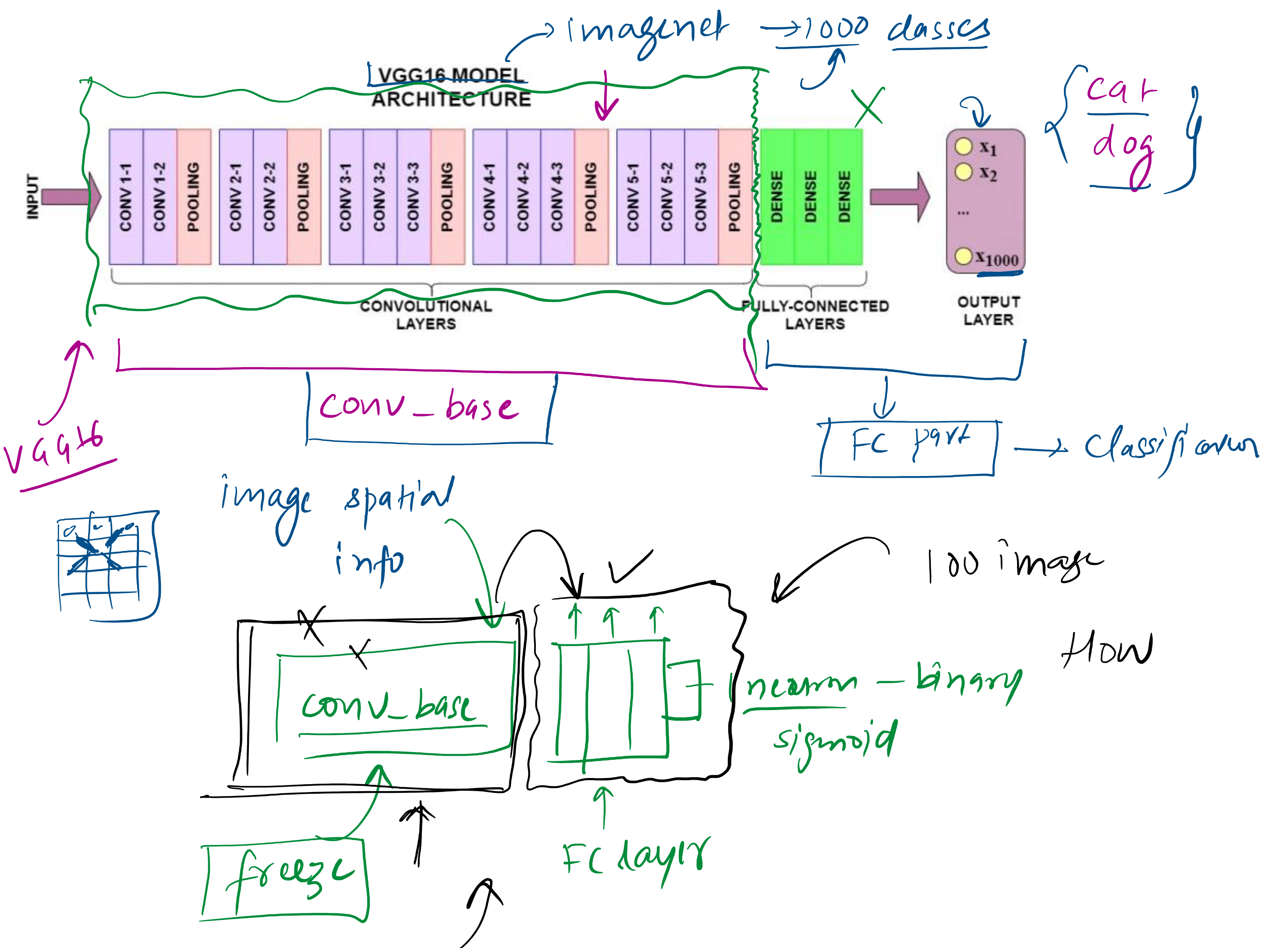
phone/tablet

Transfer Learning

10 October 2022 10:49

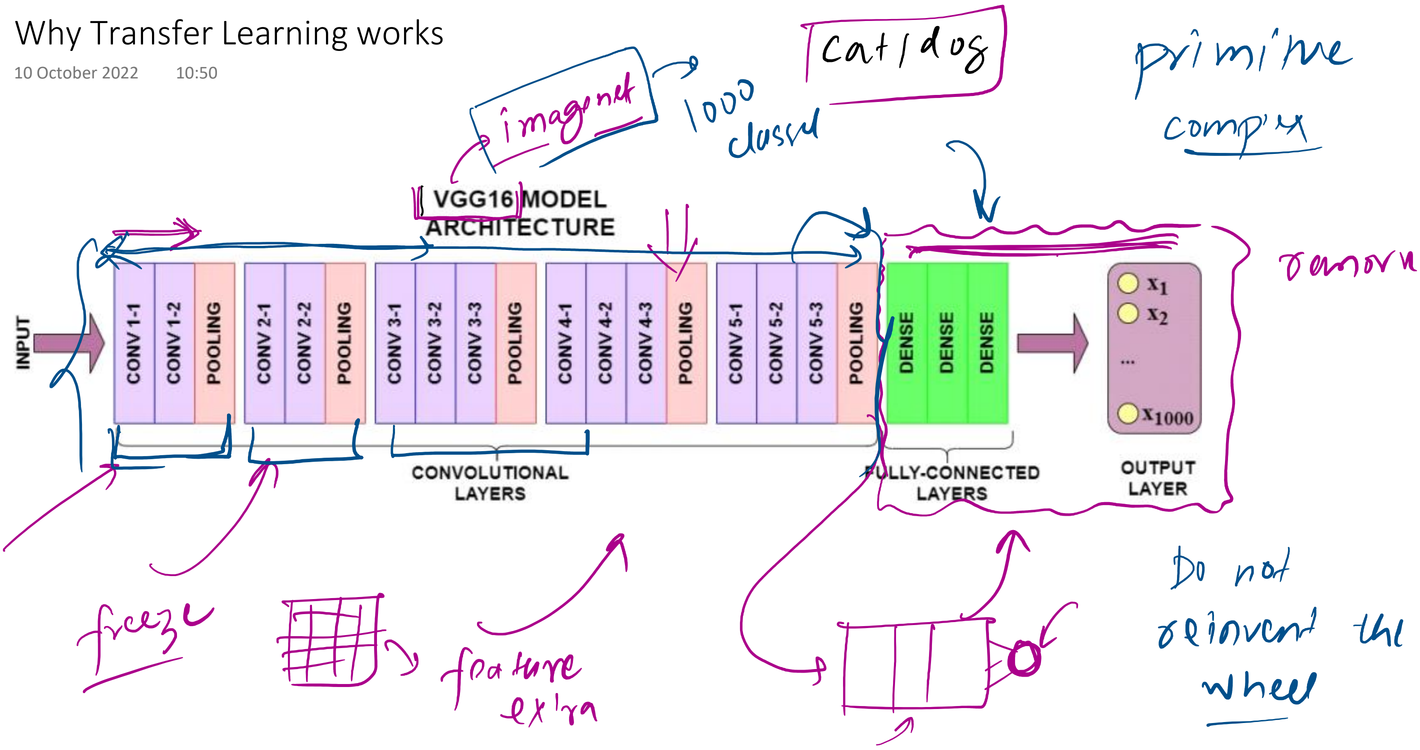
Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.





Why Transfer Learning works

10 October 2022 10:50



Ways of doing Transfer Learning

10 October 2022 10:50

Feature extraction

Fine tuning

phone
tablet

image classif → dog / cat

simlr

similar

imageNET

animals