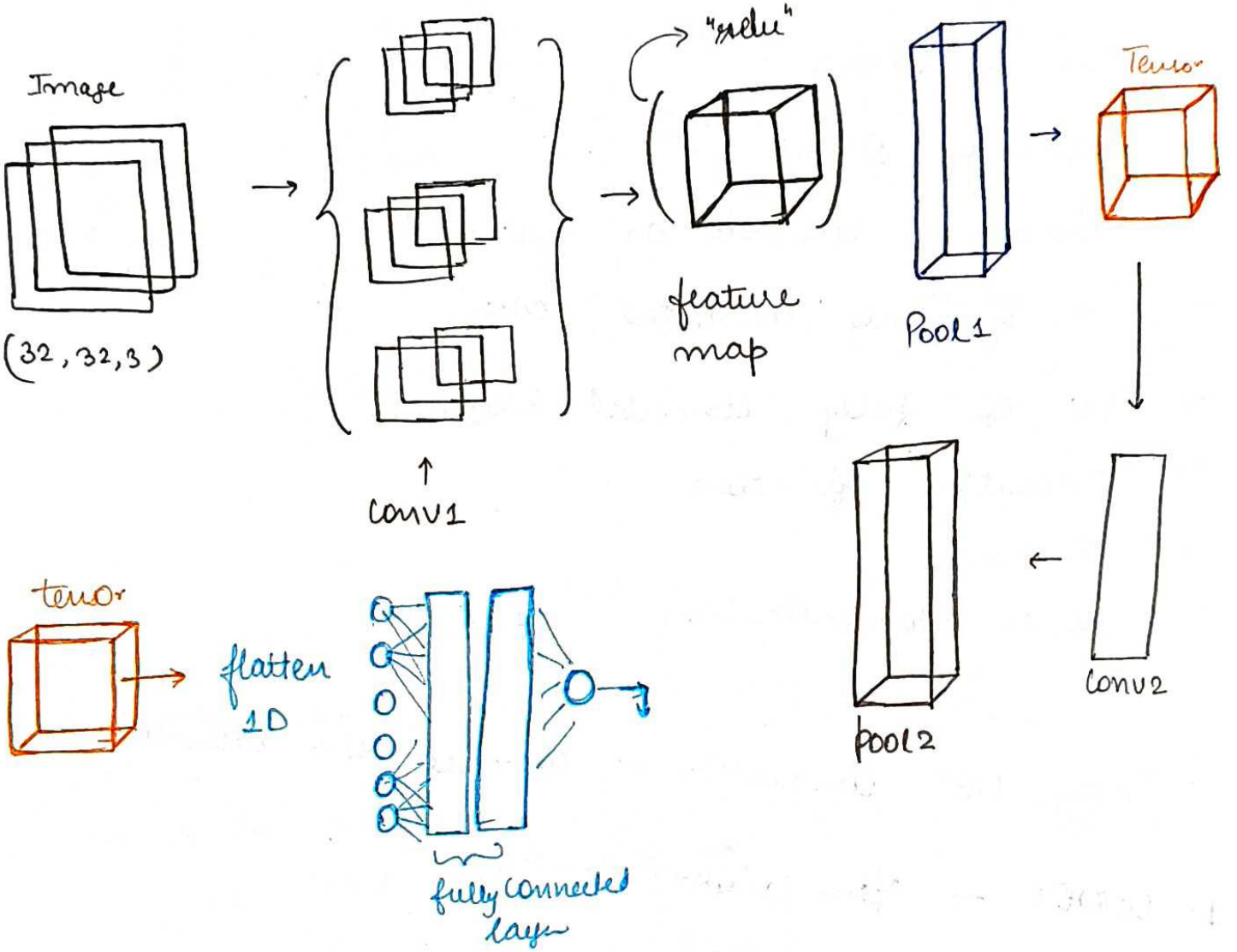
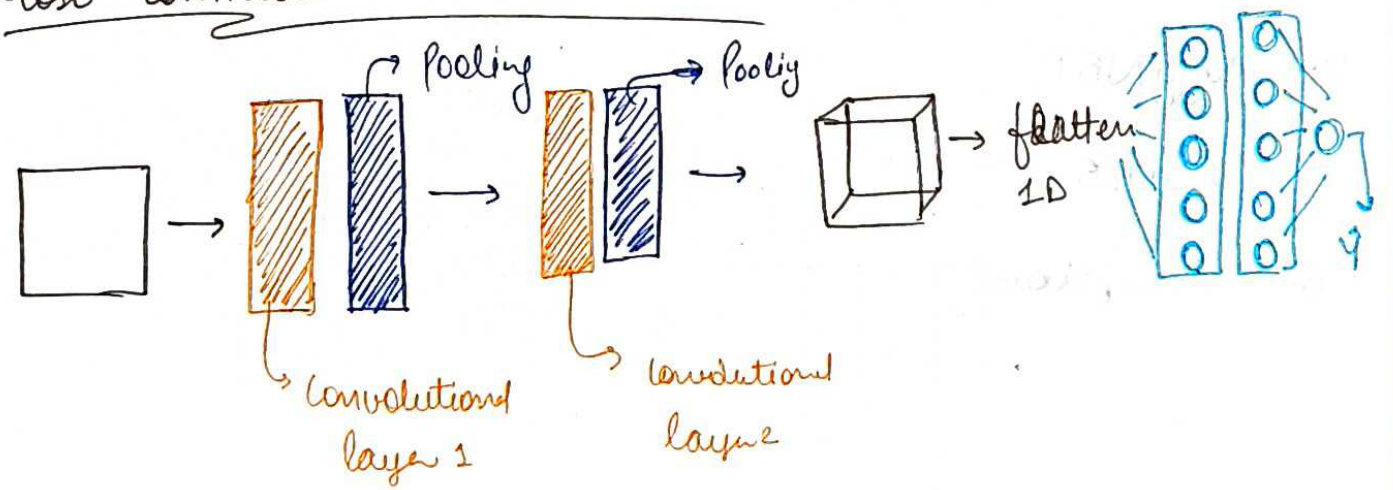
CNN Architecture

1) Convolution 2) Padding / stride 3) Pooling



Most common CNN Architecture



Make small changes in common CNN Architecture and build different architecture.

→ No. of Convolutional layer ⇒ change and make new architecture.

→ No. of filters

→ Value of strides

→ Padding is available or not

→ No. of fully connected nodes

→ No. of fully connected layers

→ activation function

→ dropout

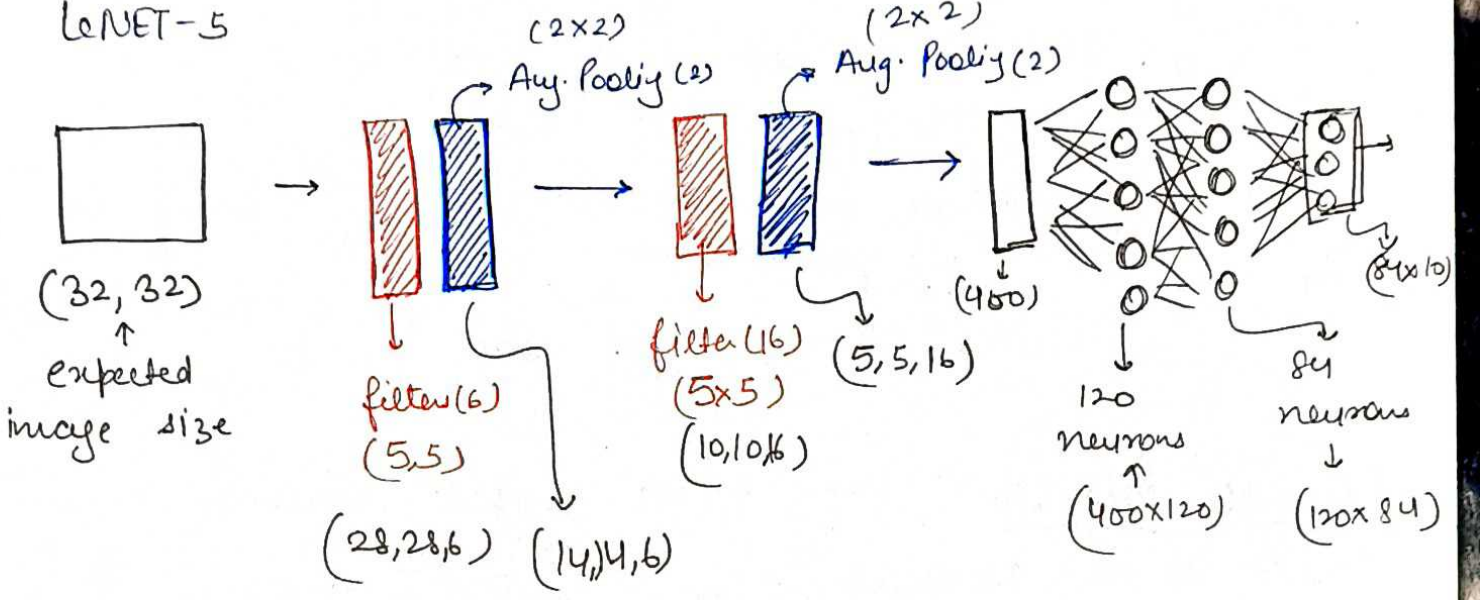
→ Batch Normalization

Image NET competition → Produce diff architecture

1. LeNET → Yann LeCunn
2. Alex NET
3. Google NET
4. VggNET
5. ResNTE
6. Inception

LeNet

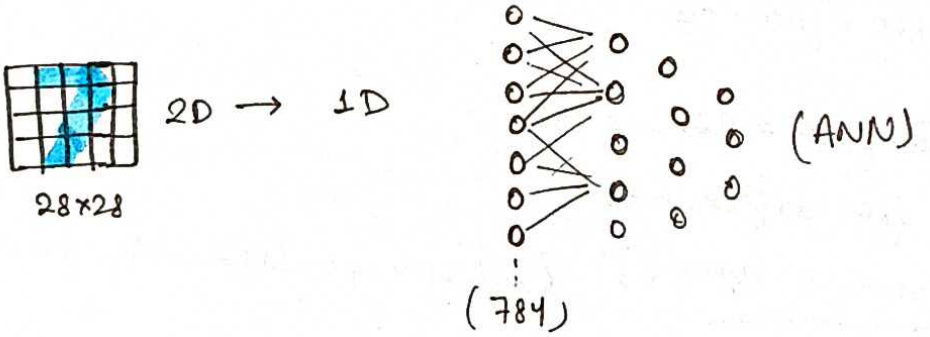
Yann Lecun → 1989-1998 → LeNET and also called LeNET-5

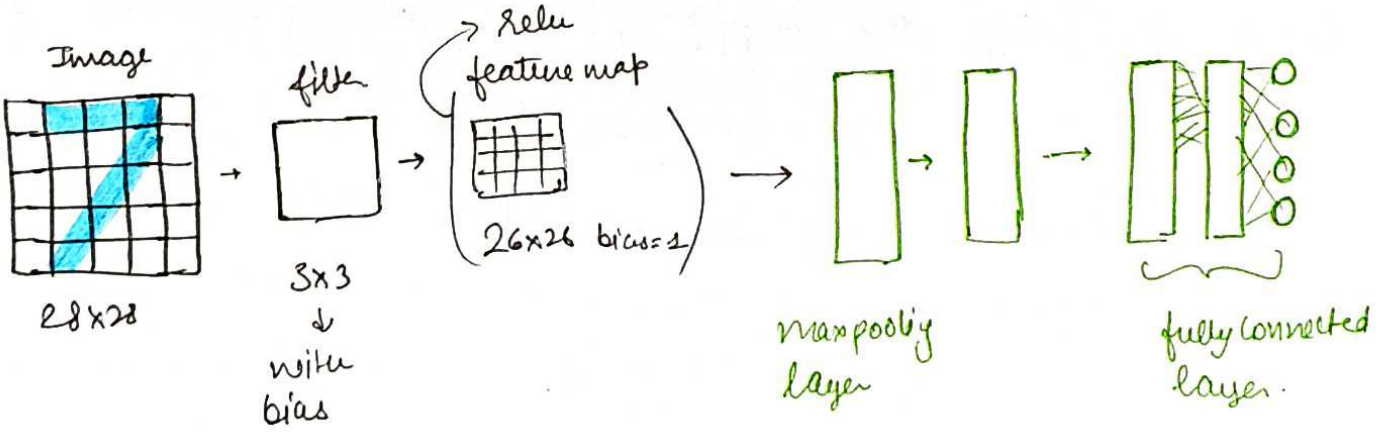


CNN Vs ANN

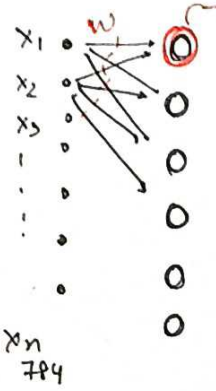
ANN Problem:

1. Computation cost
2. Overfitting
3. Loss of imp features like spatial arrangement of pixel.

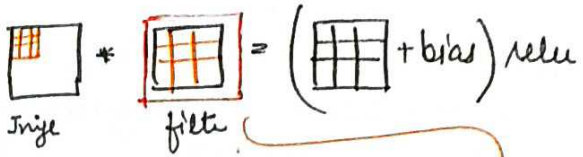




ANN → Similar → CNN



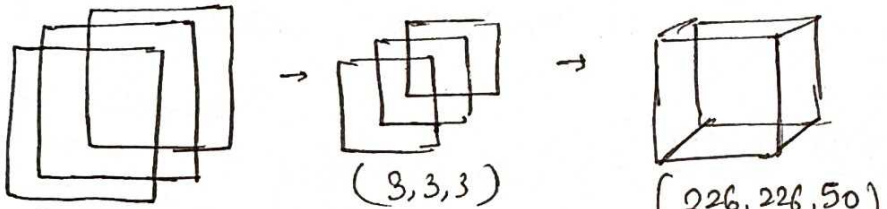
$\rightarrow \left([(w_1x_1 + w_2x_2 + \dots + w_nx_n) + b] \right)$ relu



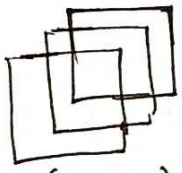
* Node and filter kind of similar
Not proper same.

relu $\left(x_1w_1 + x_2w_2 + \dots + x_qw_q + \text{bias} \right)$

Learnable Parameter



how many parameter in this CNN?



(3, 3, 3)

weigh for 1 filter = 27

for 50 filter = 27 × 50 = 1350

bias for 50 filter = 50

Total = 1400 → Trainable or Learnable parameter

What if image size is  → learning parameter?

(1080, 1080, 3)

Answer is Same Learning Parameter (1400) cause weight and bias depend on filters not on image size.

filter size ← filters → No. of filters

(image↑↑ = Parameter same)

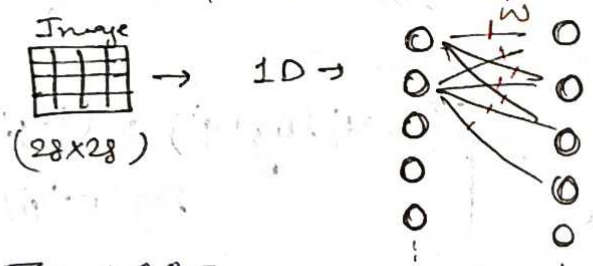
In CNN

Parameter depend on → No. of filters
→ filter size

So, if you increase the size of Image still parameter is same and computational power not increasing.

In ANN

Parameter depend on → Input (Image size)



→ If Input ↑↑ ⇒ No. of Weight ↑↑

→ computational power also increase.

→ chance to increase overfitting

Input ↑↑ ↓
No. of weight and bias ↑↑
↓
computational Power ↑↑