

Convolutional Neural Networks

***Ishan Sharma
4th Year, COE
TIET***

Presented By :

***Prabhjot Singh
4th Year, COE
TIET***

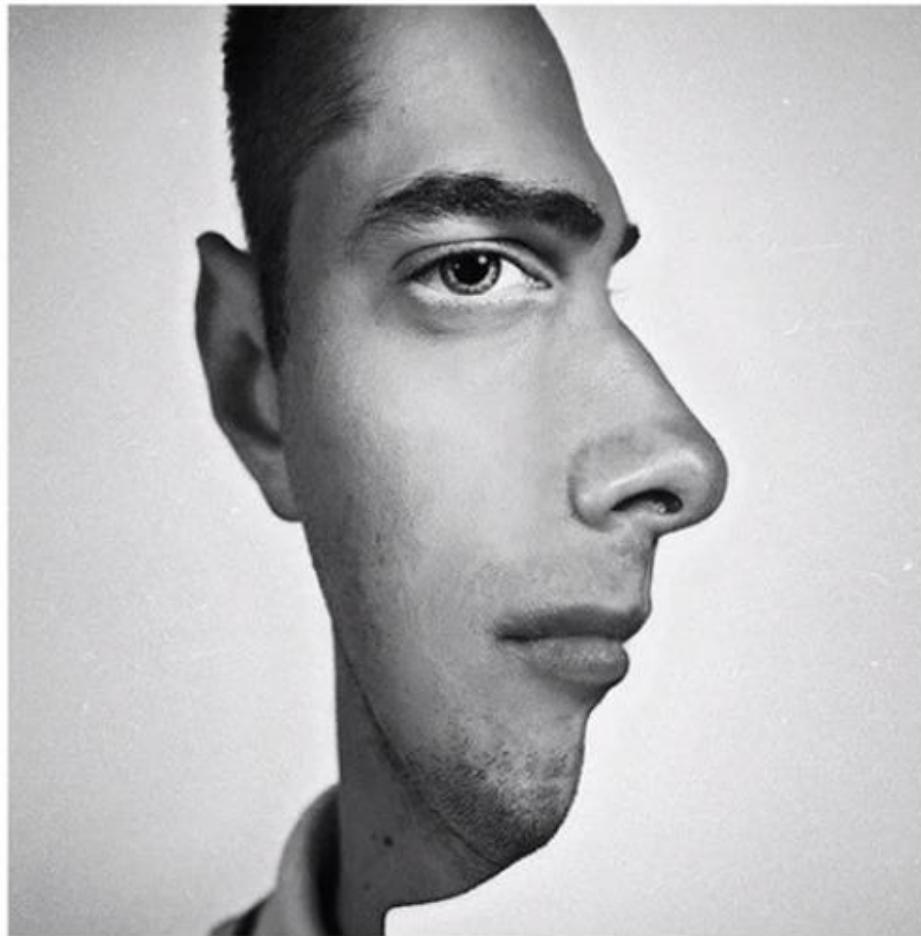
Plan of Attack

What we will learn in this section:

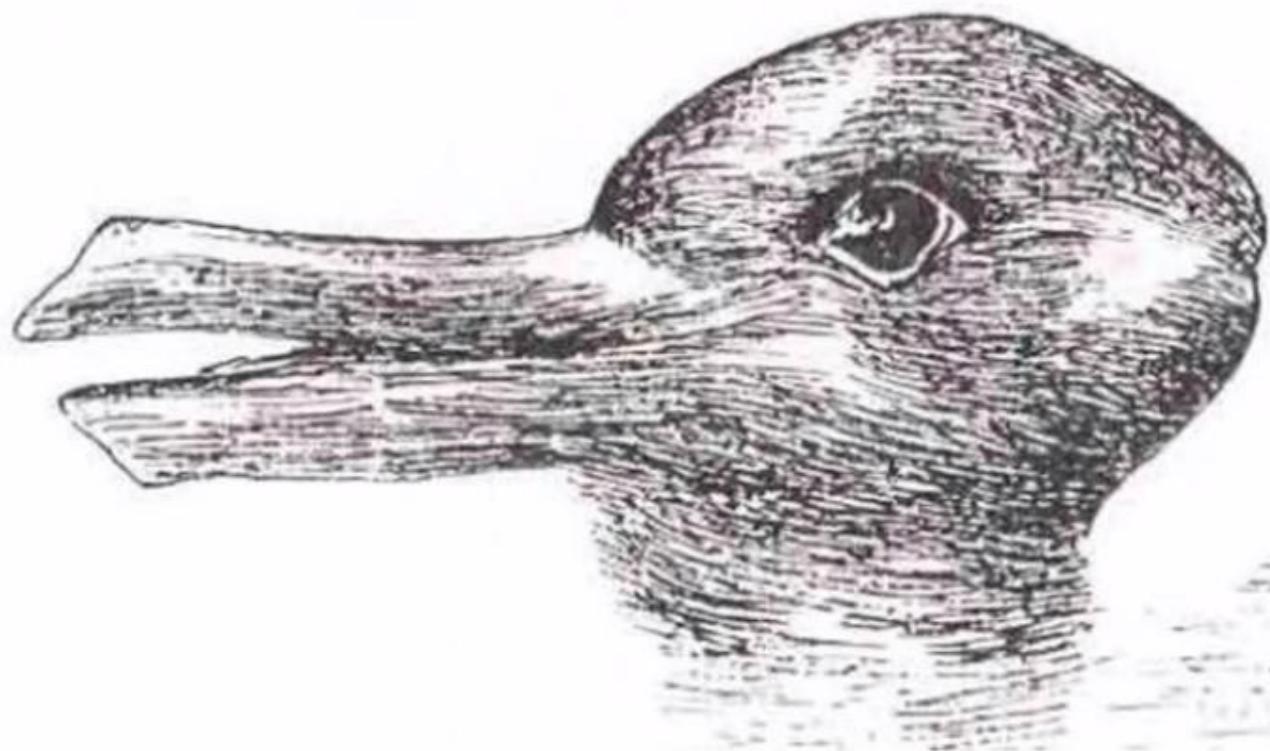
- What are Convolutional Neural Networks?
- Step 1 - Convolution Operation
- Step 1(b) - ReLU Layer
- Step 2 - Pooling
- Step 3 - Flattening
- Step 4 - Full Connection
- Summary

- EXTRA: Softmax & Cross-Entropy

Convolutional Neural Networks



Convolutional Neural Networks



Convolutional Neural Networks



Convolutional Neural Networks

Examples from the test set
(with the network's guesses)

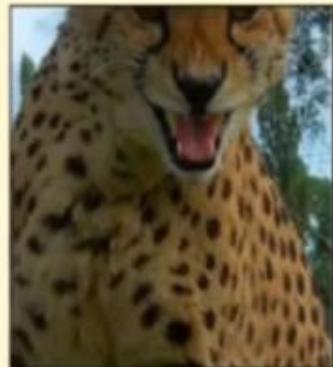
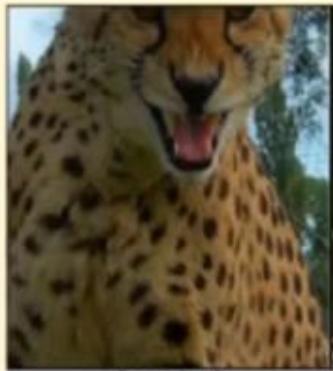


Image Source: a talk by Geoffrey Hinton

Convolutional Neural Networks

Examples from the test set
(with the network's guesses)



cheetah

cheetah

leopard

snow leopard

Egyptian cat



Image Source: a talk by Geoffrey Hinton

Convolutional Neural Networks

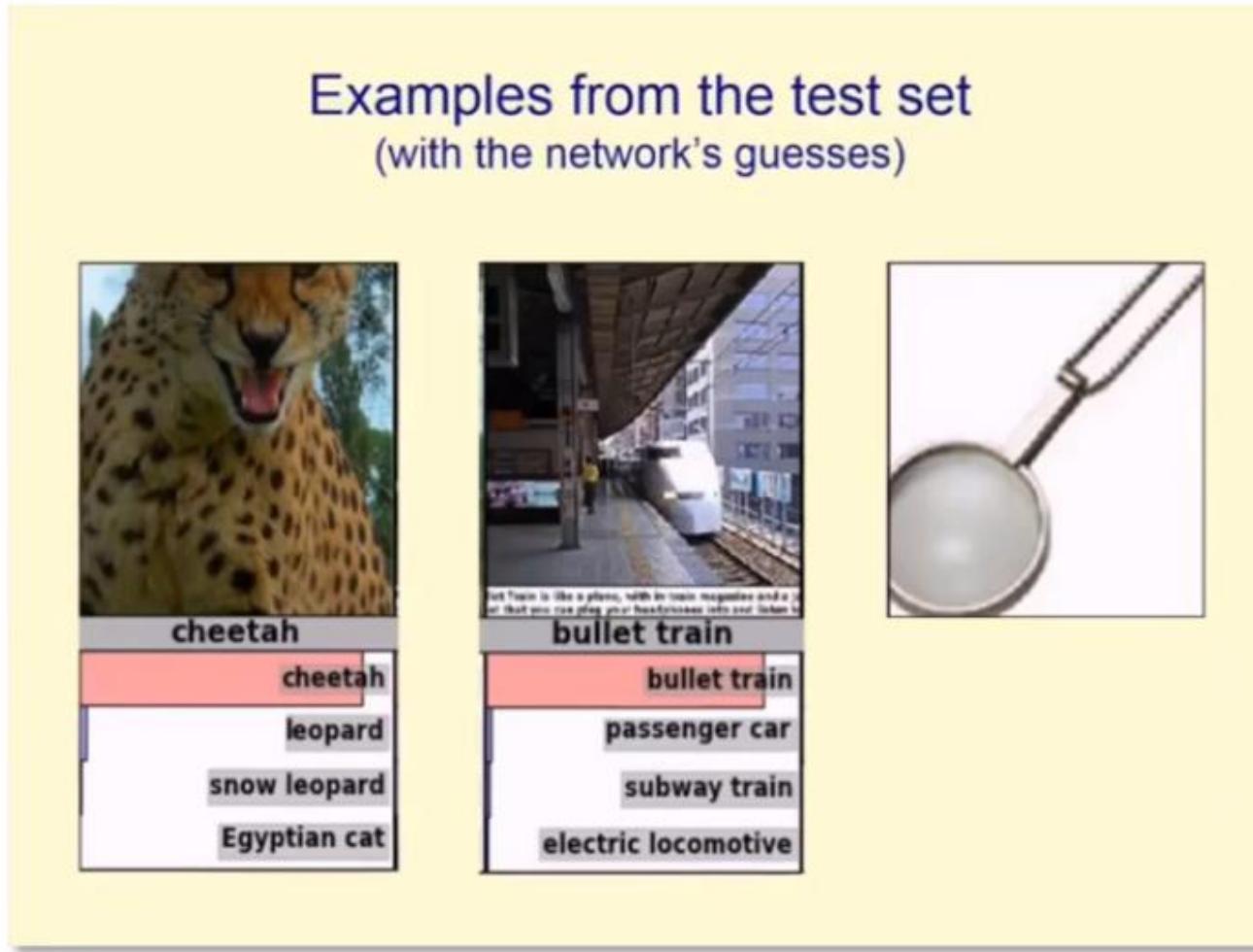


Image Source: a talk by Geoffrey Hinton

Convolutional Neural Networks

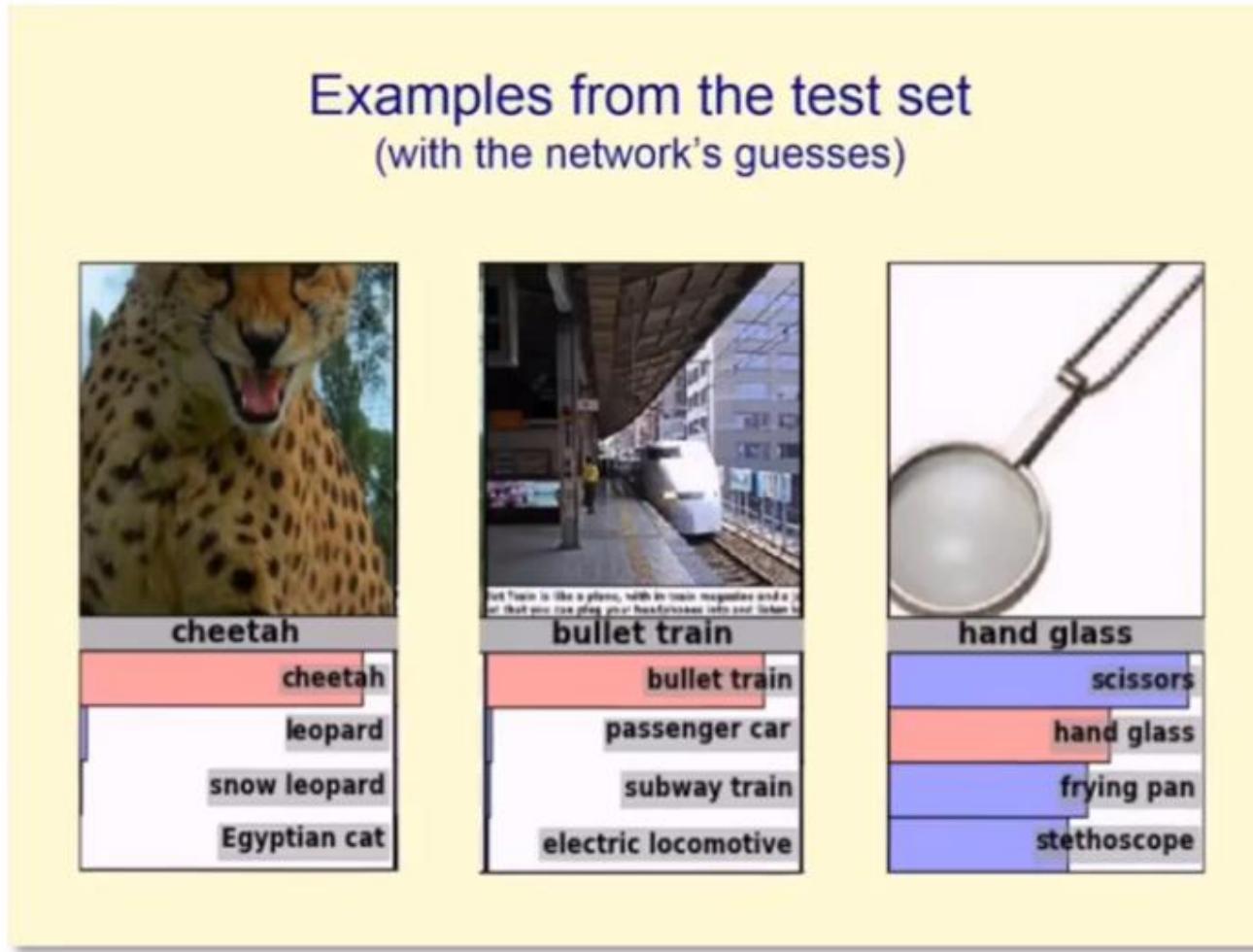


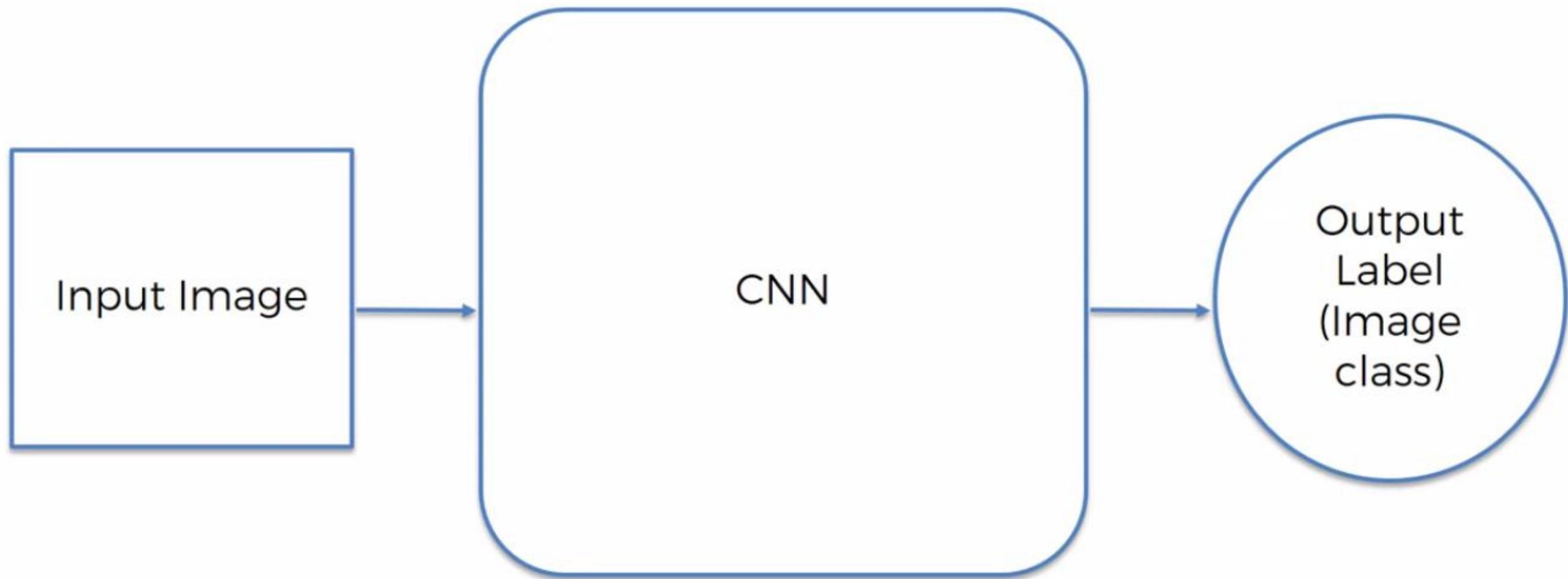
Image Source: a talk by Geoffrey Hinton

Convolutional Neural Networks

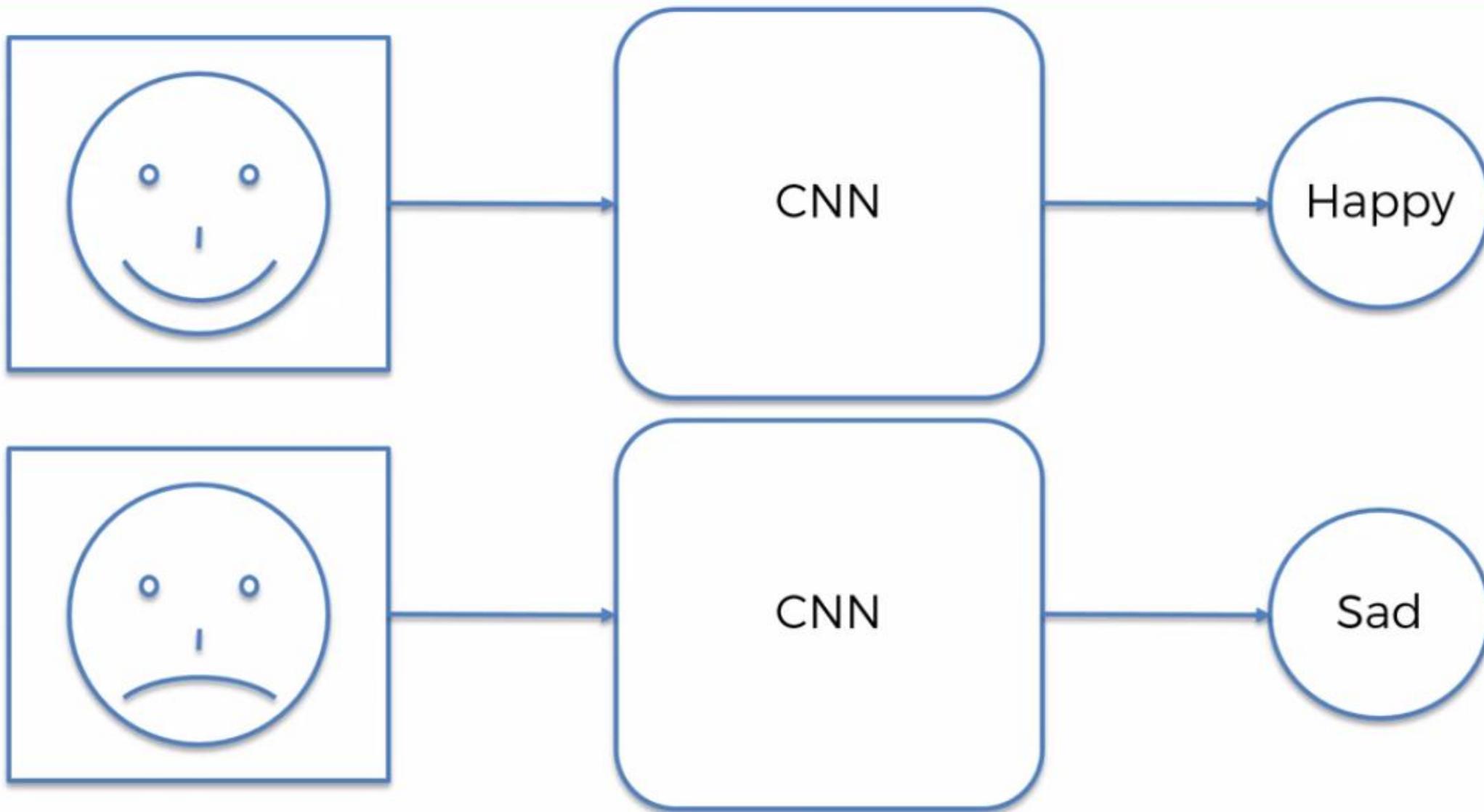


Yann LeCun

Convolutional Neural Networks

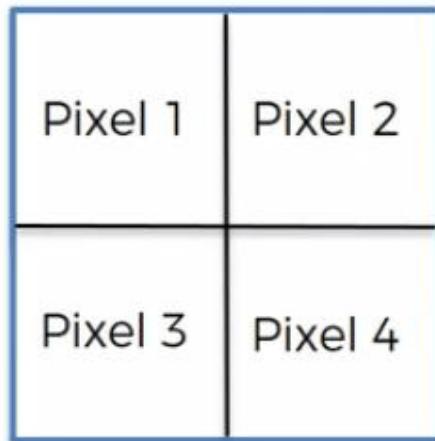


Convolutional Neural Networks

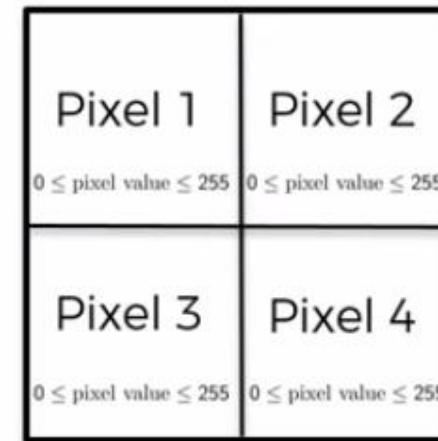


Convolutional Neural Networks

B / W Image 2x2px



2d array



Colored Image 2x2px

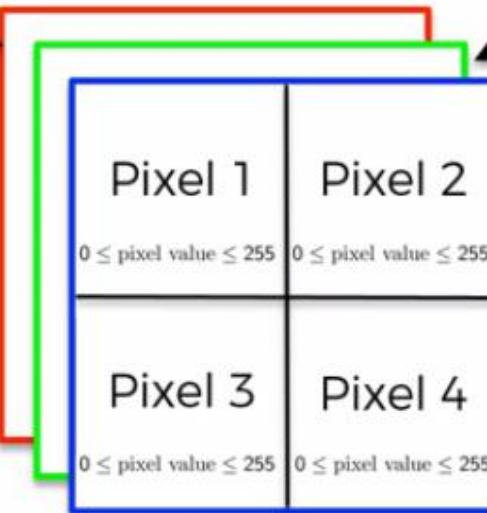


3d array

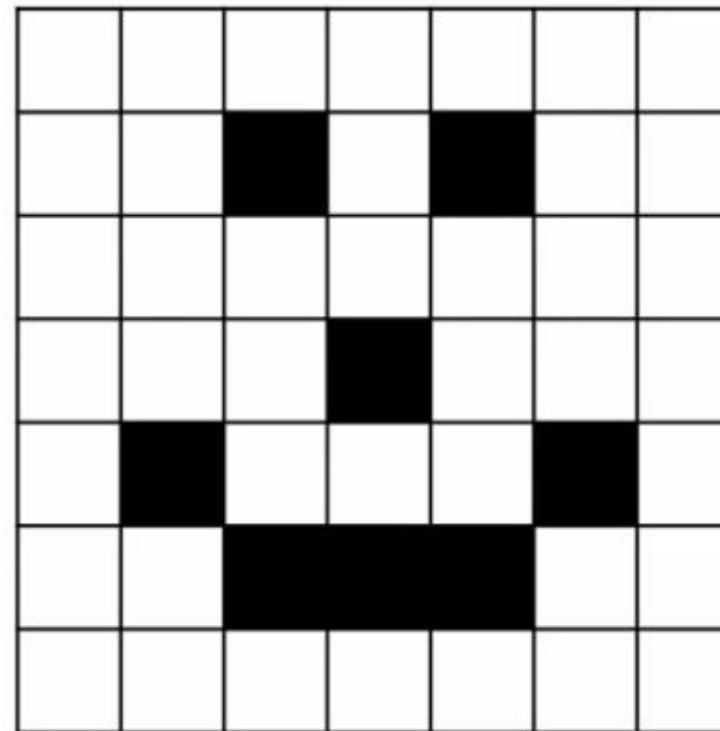
Red channel

Green channel

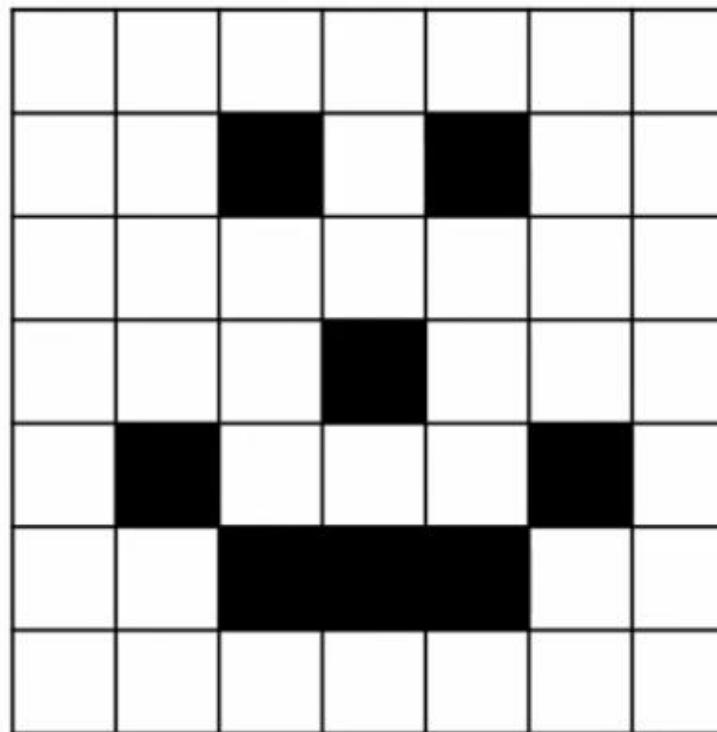
Blue channel



Convolutional Neural Networks



Convolutional Neural Networks



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Convolutional Neural Networks

STEP 1: Convolution



STEP 2: Max Pooling



STEP 3: Flattening



STEP 4: Full Connection

Step 1 - Convolution

Step 1 - Convolution

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |

Input Image

| | | |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |

Feature
Detector

Step 1 - Convolution

Note: Here we will use
stride of one.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Input Image



| | | |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |

Feature
Detector



| | | | | |
|---|--|--|--|--|
| 0 | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Feature Map

Step 1 - Convolution

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



Input Image

| | | |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |

Feature
Detector



| | | | | |
|---|---|--|--|--|
| 0 | 1 | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Feature Map

Step 1 - Convolution

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



Input Image

| | | |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |

Feature
Detector



| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 2 | 1 |
| 1 | 4 | 2 | 1 | 0 |
| 0 | 0 | 1 | 2 | 1 |

Feature Map

Note: Maximum value of feature map signifies the feature detected.

e.g Eyes,Nose

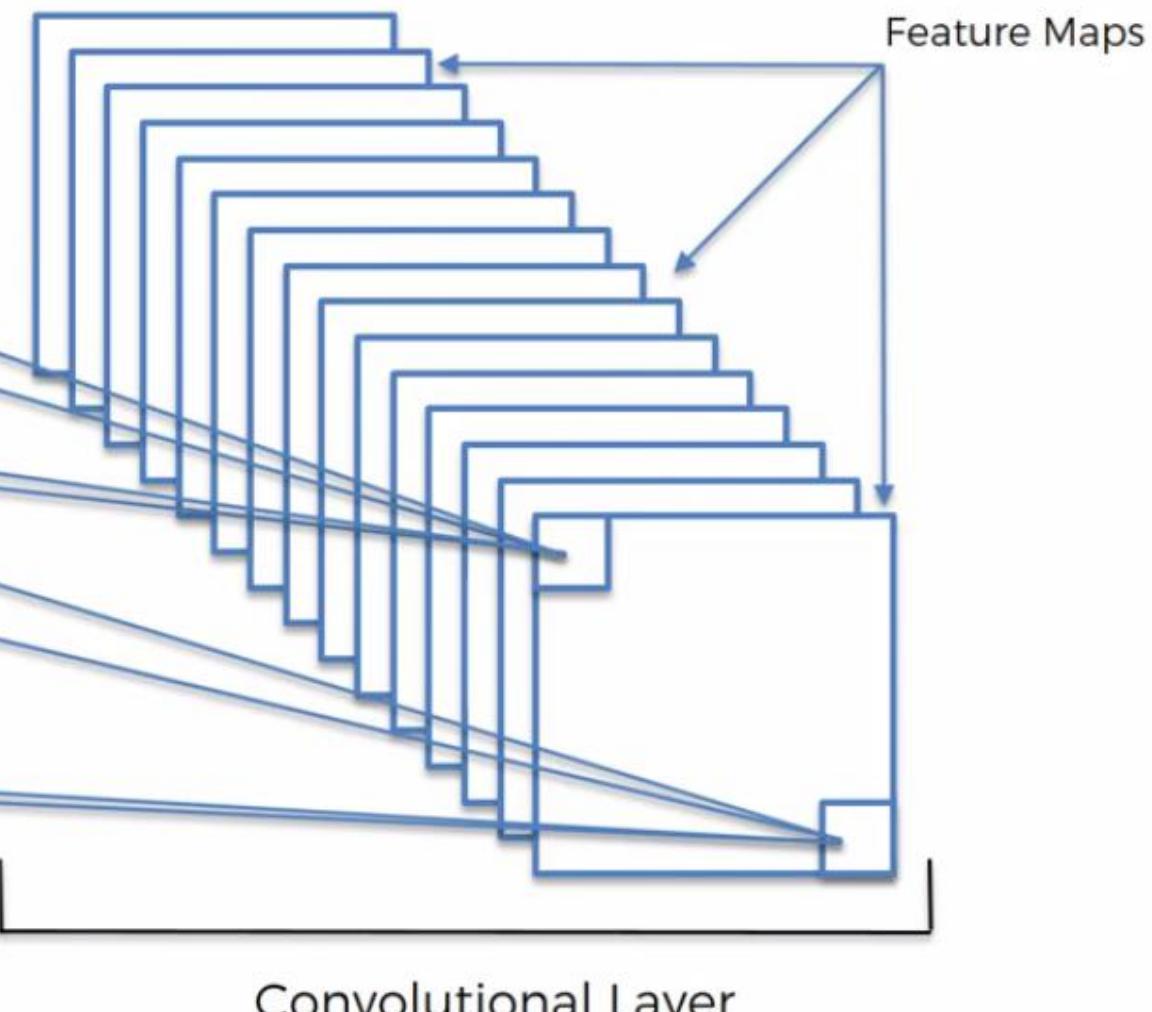
Many such feature maps
combine to make convolution
layer

Step 1 - Convolution

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Input Image

We create many feature maps to obtain our first convolution layer



Applying different filters to image.

(Note: Filters are like feature detectors.)

Step 1 - Convolution

Sharpen:

| | | | | |
|---|----|----|----|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | -1 | 0 | 0 |
| 0 | -1 | 5 | -1 | 0 |
| 0 | 0 | -1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

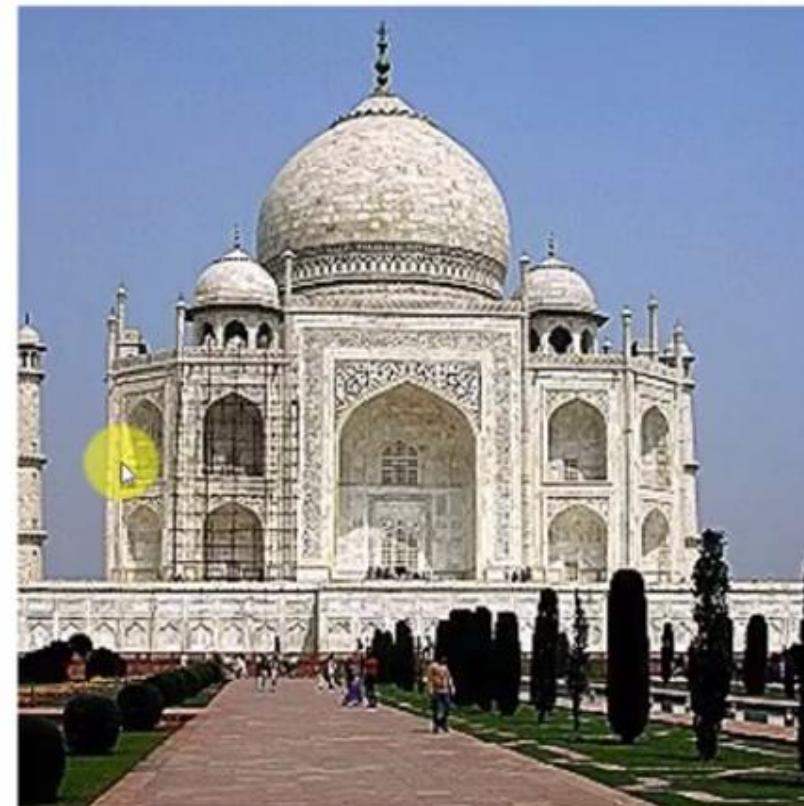


Image Source: docs.gimp.org/en/plug-in-convmatrix.html

Step 1 - Convolution

Blur:

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |



Image Source: docs.gimp.org/en/plug-in-convmatrix.html

Step 1 - Convolution

Edge Enhance:

$$\begin{matrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{matrix}$$



Image Source: docs.gimp.org/en/plug-in-convmatrix.html

Step 1 - Convolution

Edge Detect:

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

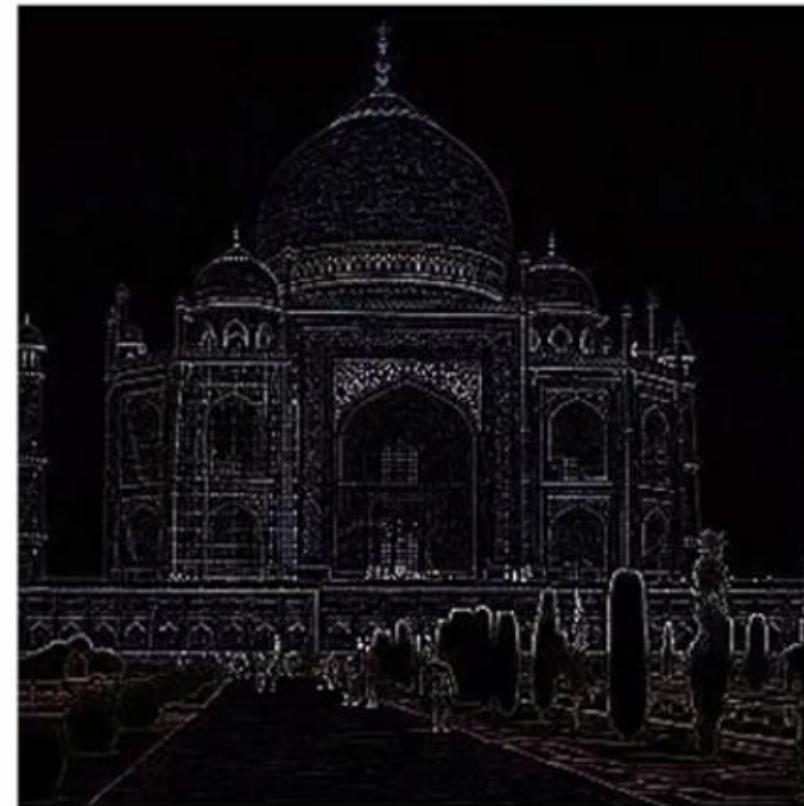


Image Source: docs.gimp.org/en/plug-in-convmatrix.html

Key Take Away from convolution:

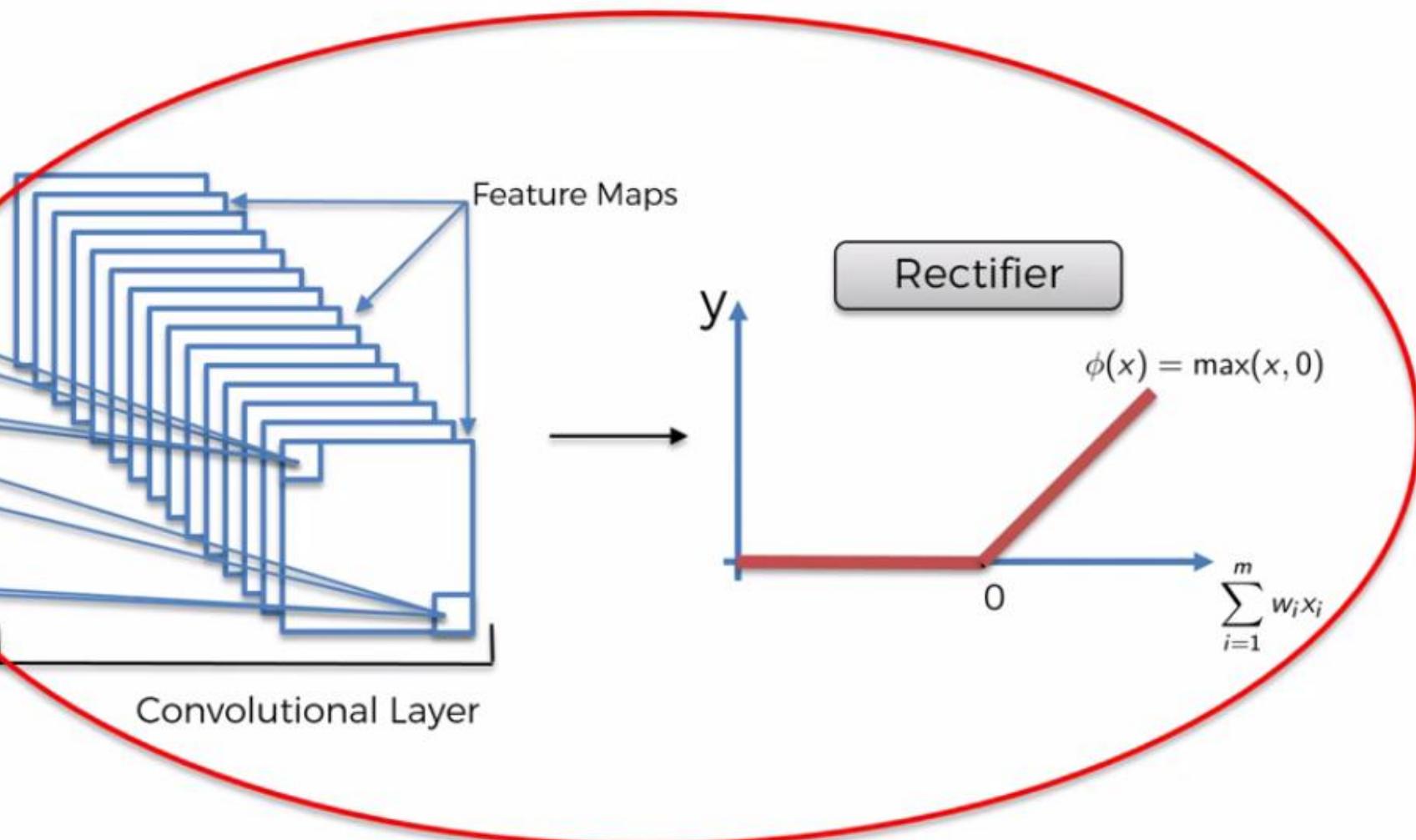
Find features in image using
feature detector and put it into
feature map.

Step 1(B) - ReLU Layer

Step 1(B) - ReLU Layer

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 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 | |

Input Image



Note: Rectification is applied to increase the non linearity in image.

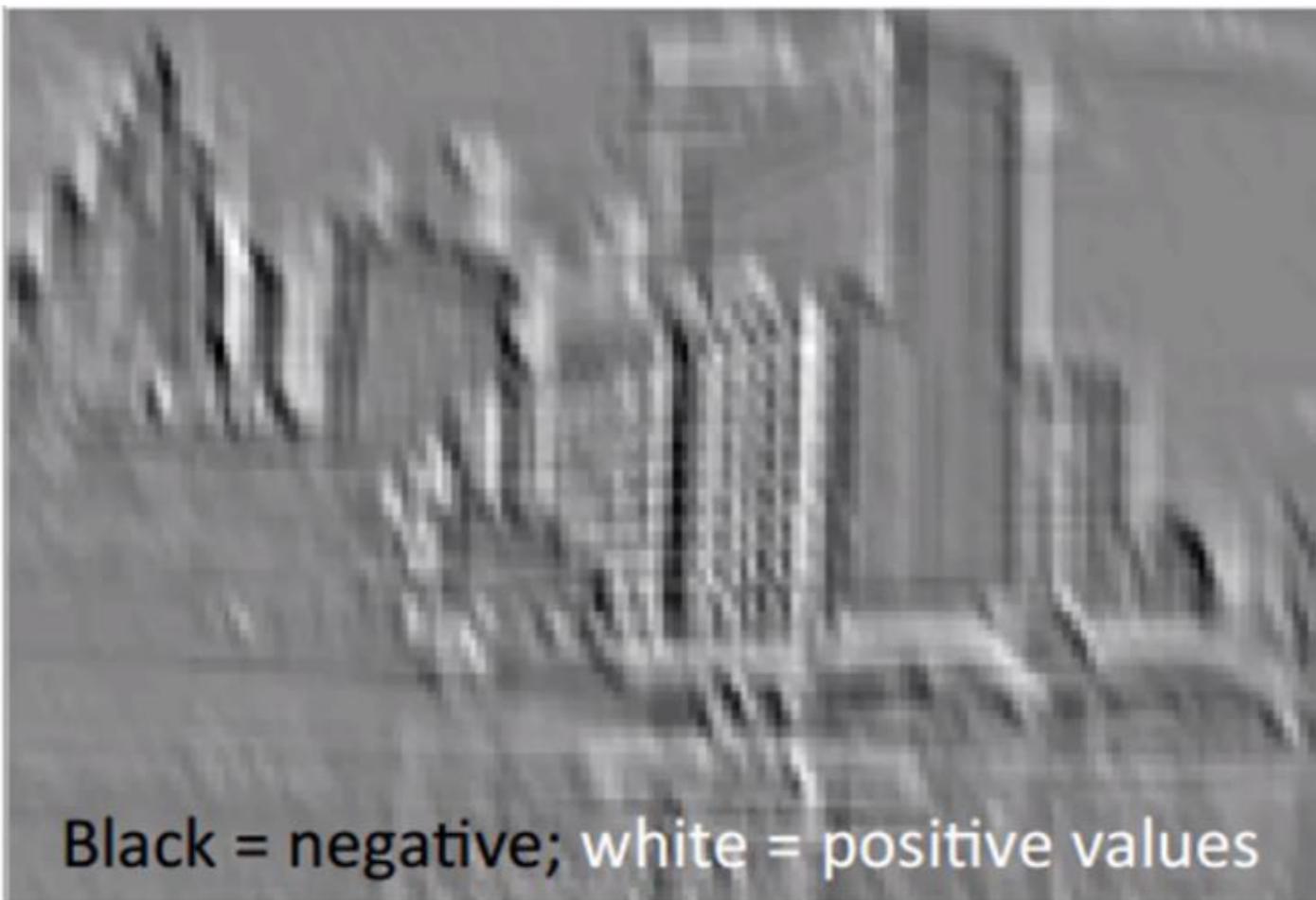
Step 1(B) - ReLU Layer



Image Source: http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf

After applying feature
detector on this image we
get-->

Step 1(B) - ReLU Layer



Black = negative; white = positive values

Image Source: http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf

After applying rectification
we get:

Step 1(B) - ReLU Layer



Only non-negative values

Image Source: http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf

*Take Away: In ReLU layer
we just apply rectification
function.*

Step 2 - Max Pooling

Step 2 - Max Pooling



Image Source: Wikipedia

Step 2 - Max Pooling



Image Source: Wikipedia

Note: There are different types of Pooling like mean pooling,min pooling, max pooling. But we will use here Max Pooling.

Step 2 - Max Pooling

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 2 | 1 |
| 1 | 4 | 2 | 1 | 0 |
| 0 | 0 | 1 | 2 | 1 |

Feature Map

Note: This is the feature map obtained from convolution step.

Max Pooling



| | | |
|--|--|--|
| | | |
| | | |
| | | |

Pooled Feature Map

Step 2 - Max Pooling

*Find maximum value out of this
2x2 box.*

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 2 | 1 |
| 1 | 4 | 2 | 1 | 0 |
| 0 | 0 | 1 | 2 | 1 |

Feature Map

Max Pooling



| | | |
|---|--|--|
| 1 | | |
| | | |
| | | |

Pooled Feature Map

Step 2 - Max Pooling

Note: Here we will use stride of two as it is commonly used.

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 2 | 1 |
| 1 | 4 | 2 | 1 | 0 |
| 0 | 0 | 1 | 2 | 1 |

Feature Map

Max Pooling



| | | |
|---|---|--|
| 1 | 1 | |
| | | |
| | | |

Pooled Feature Map

Step 2 - Max Pooling

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 2 | 1 |
| 1 | 4 | 2 | 1 | 0 |
| 0 | 0 | 1 | 2 | 1 |

Feature Map

Max Pooling



| | | |
|---|---|---|
| 1 | 1 | 0 |
| 4 | 2 | 1 |
| 0 | 2 | 1 |

Pooled Feature Map

Main purpose of pooling: To get rid of 75% of un-necessary information that is not the required feature or the required information we are looking for.

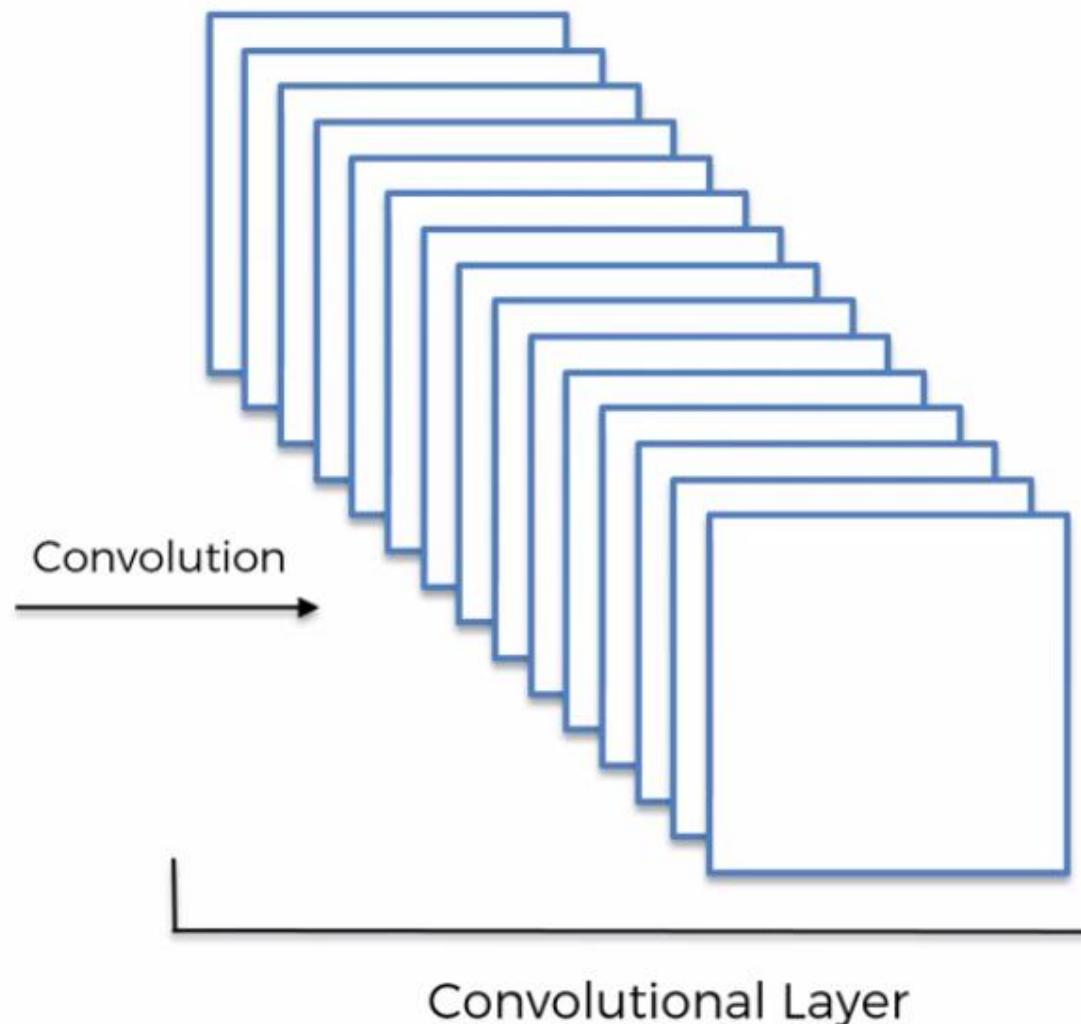
For E.g let us assume here the value 4 correspond to the tears of cheetah. So there are various images of cheetah come are rotated left some right etc. So with the help of pooling we are discarding that rotations and extracting our main information which is maximum value i.e 4.

Quick Recap

Step 2 - Max Pooling

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |

Input Image

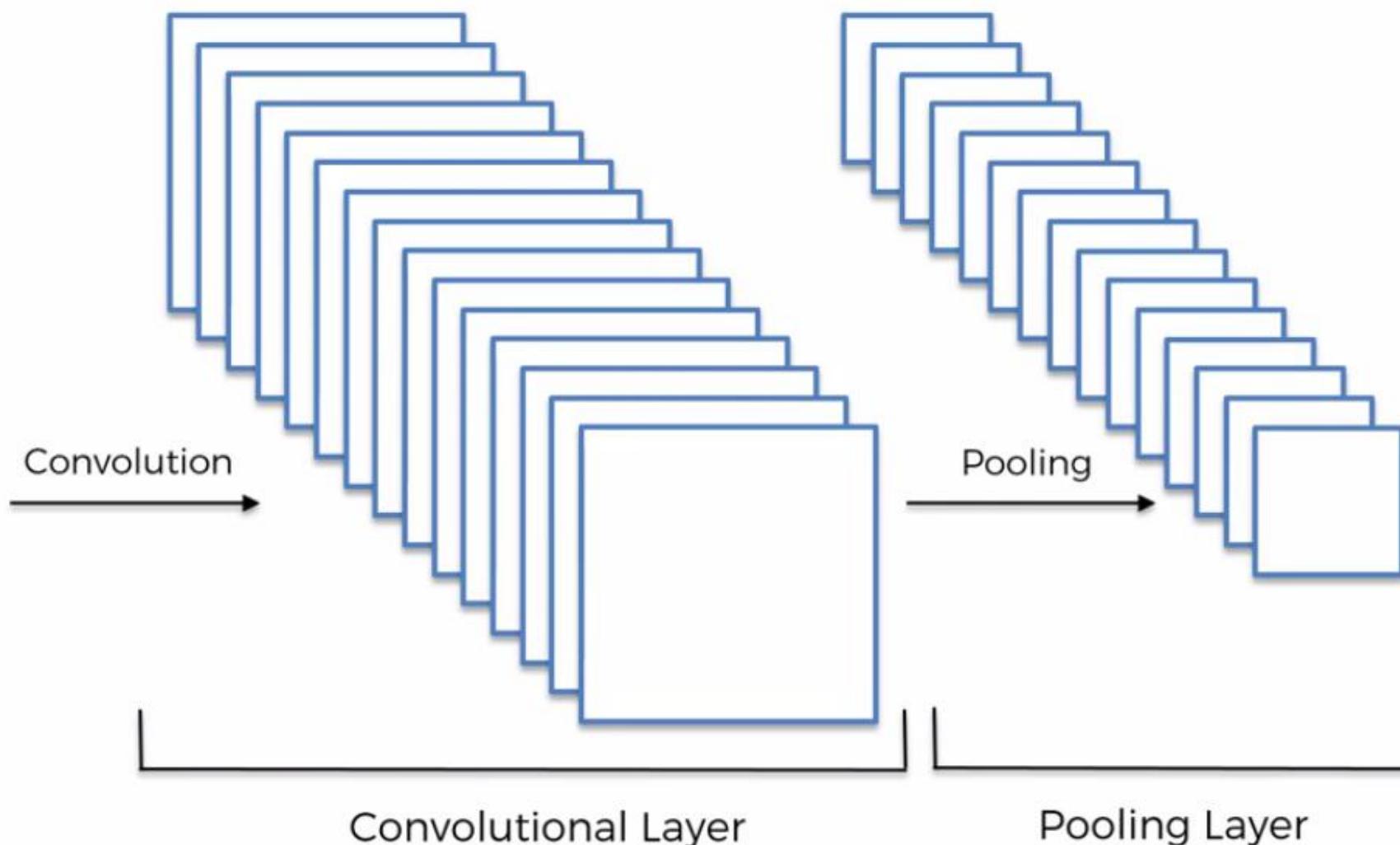


Convolutional Layer

Step 2 - Max Pooling

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |

Input Image

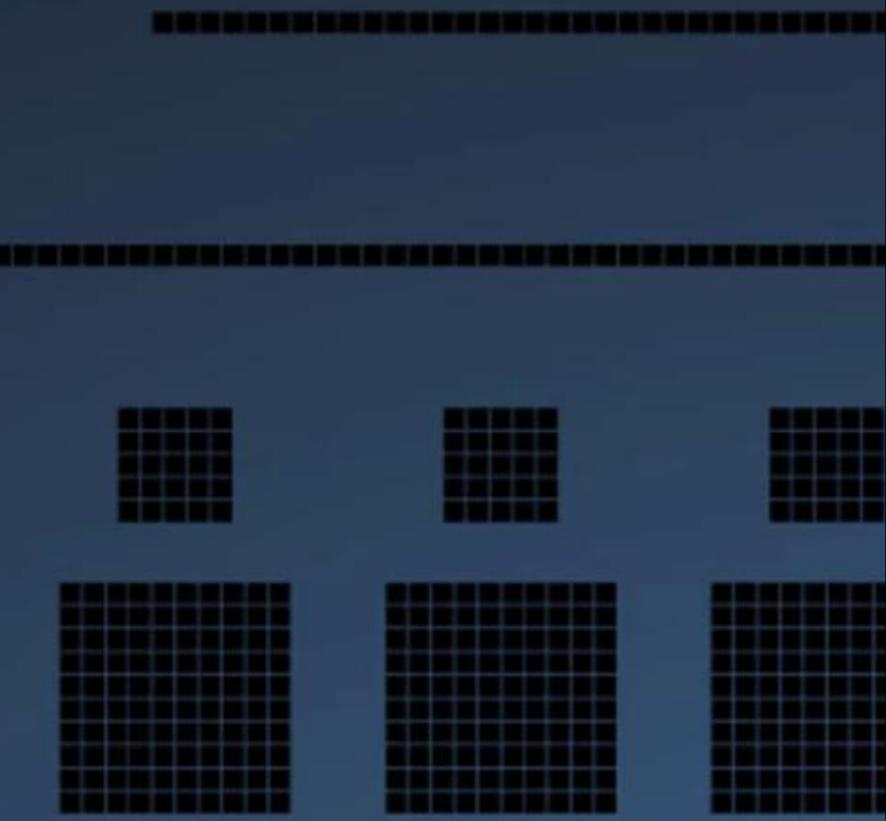
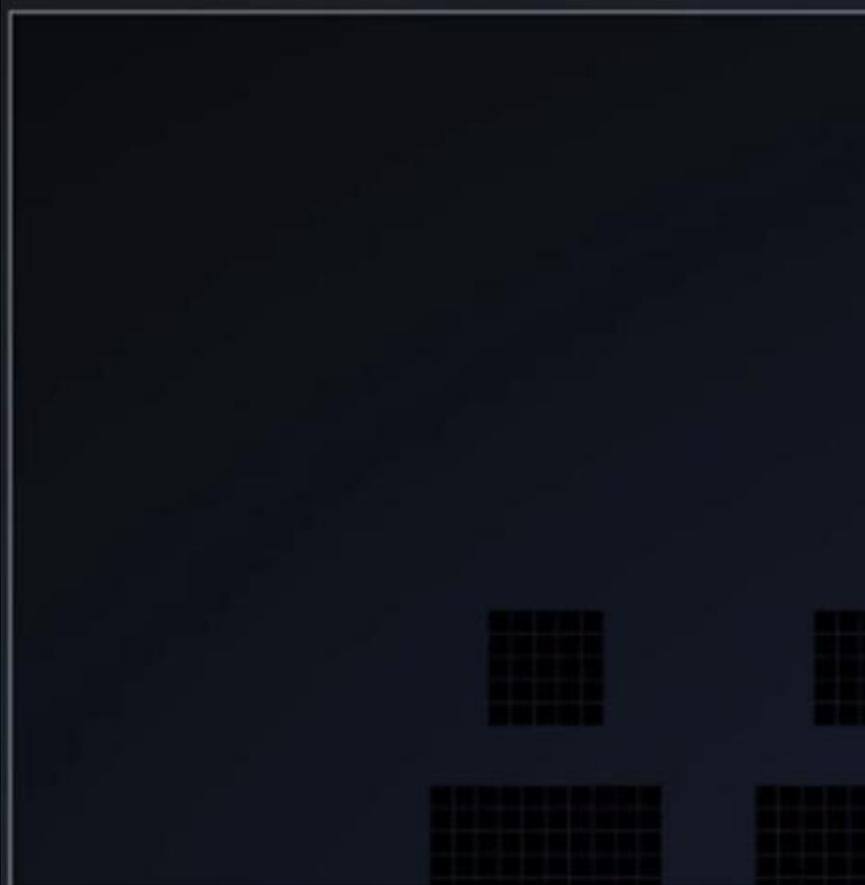


Explanation of CNN through Web.

Open :

*http://scs.ryerson.ca/~aharley/
vis/conv/*

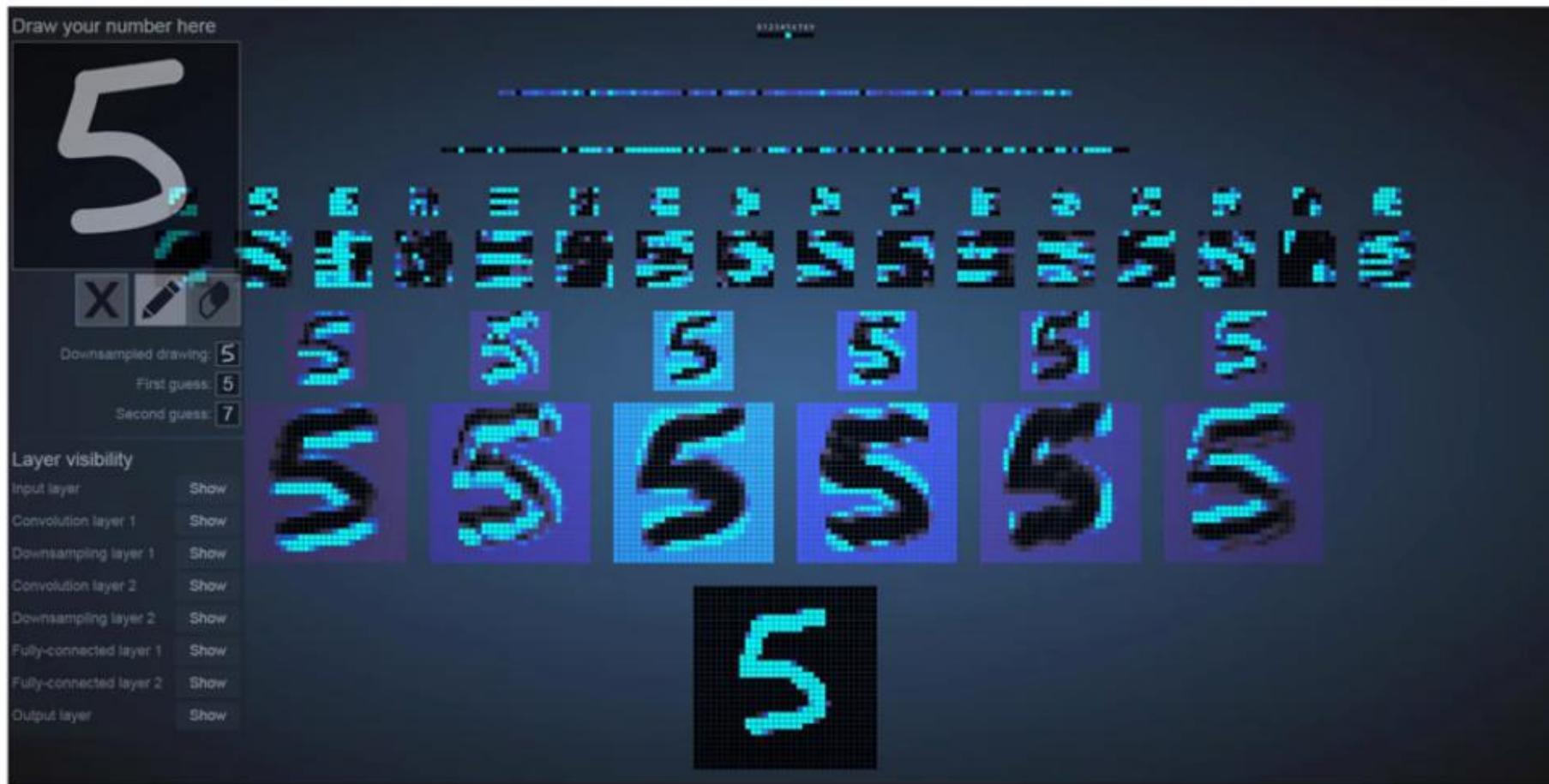
Draw your number here



Note: This tool is trained on digits
from 0-9

Example

Draw shape
here.



Fully
connected
layer.

Max Pooled
layer.

Convoluted/
Downsampl
ed layer.

Image Source: scs.ryerson.ca/~aharley/vis/conv/flat.html

Step 3 - Flattening

Step 3 - Flattening

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 4 | 2 | 1 |
| 0 | 2 | 1 |

Pooled Feature Map

Step 3 - Flattening

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 4 | 2 | 1 |
| 0 | 2 | 1 |

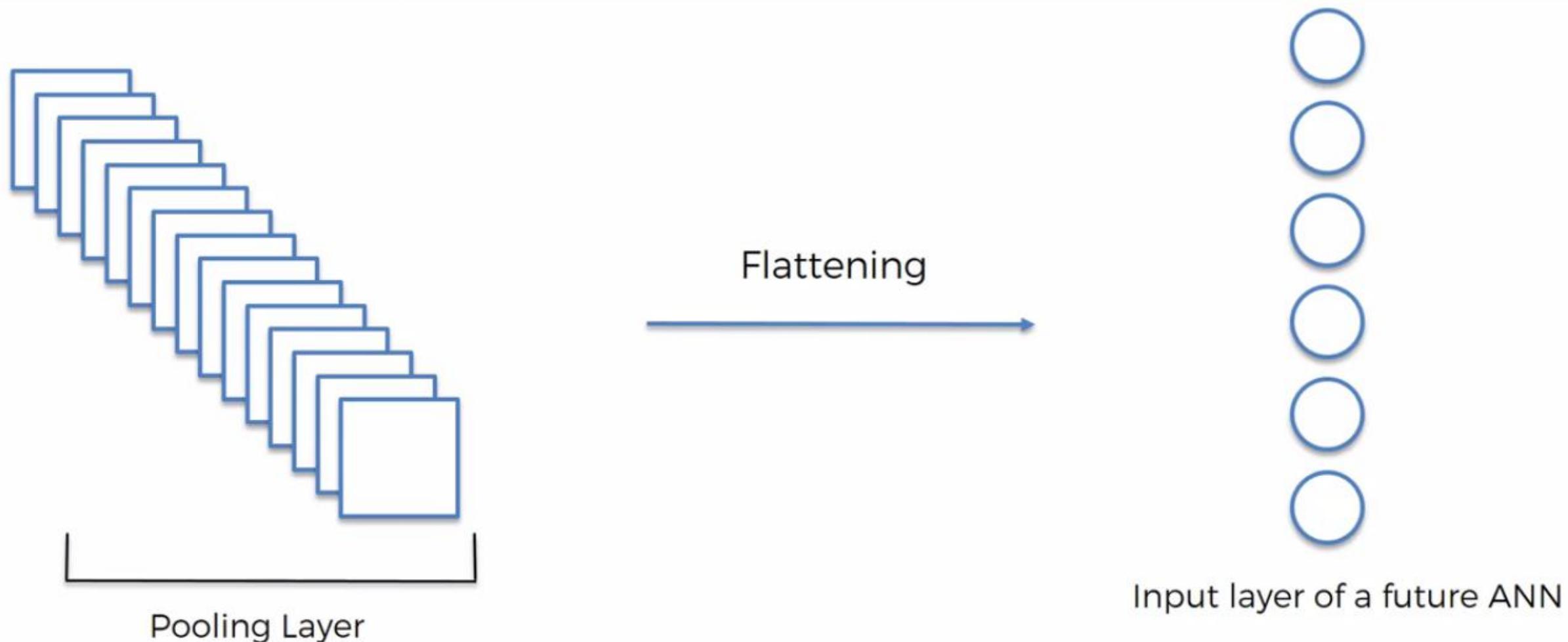
Pooled Feature Map

Flattening



| |
|---|
| 1 |
| 1 |
| 0 |
| 4 |
| 2 |
| 1 |
| 0 |
| 2 |
| 1 |

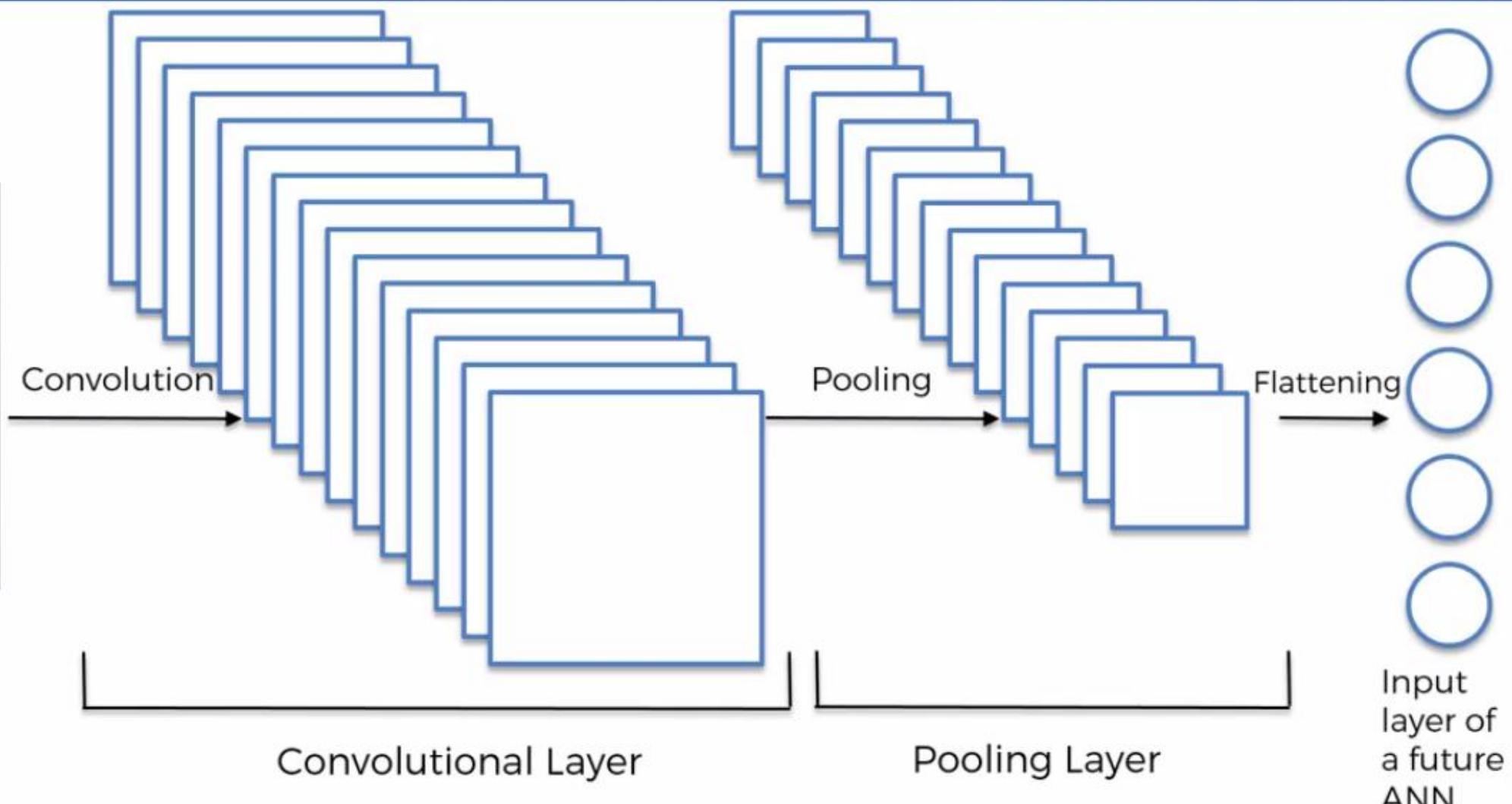
Step 3 - Flattening



Step 3 - Flattening

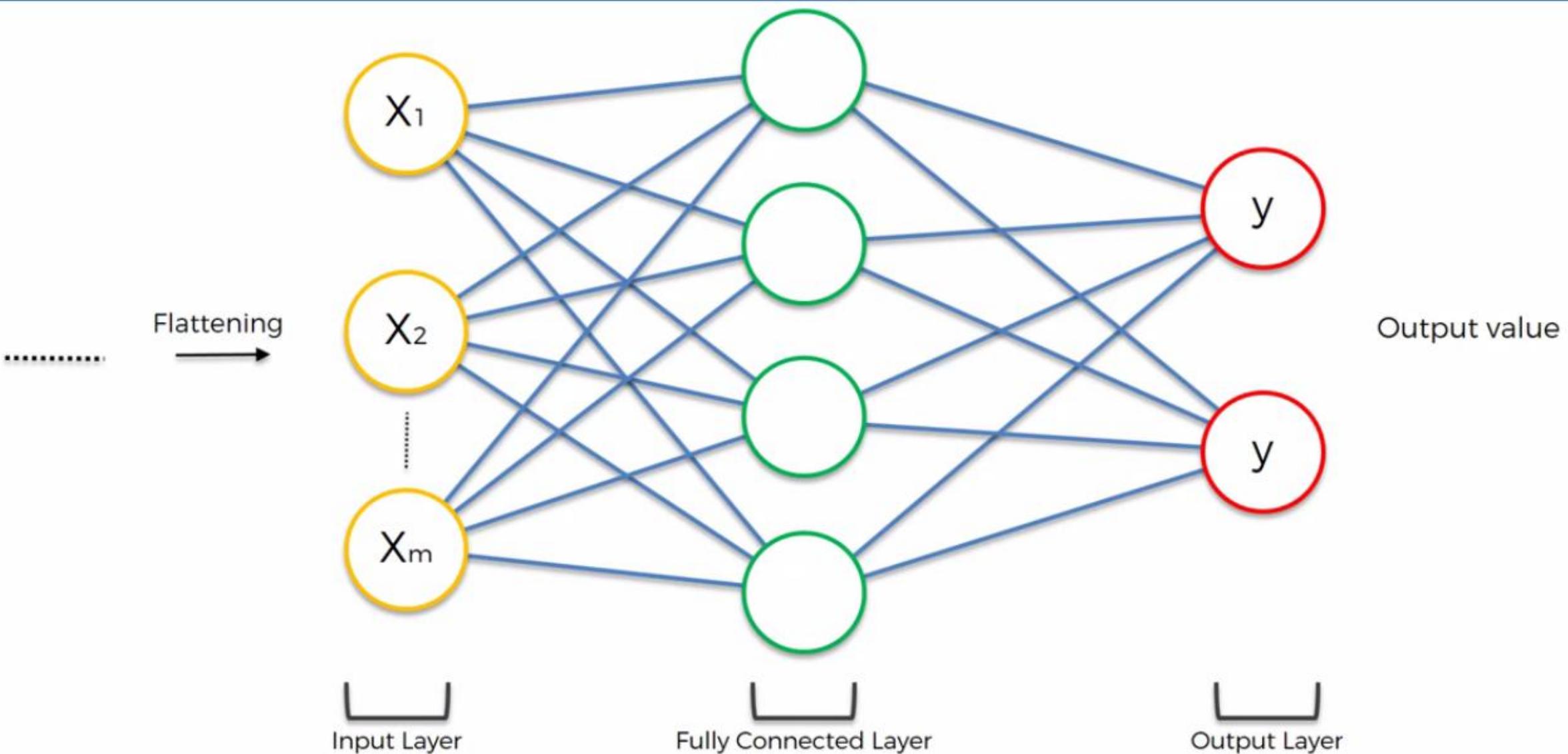
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Input Image

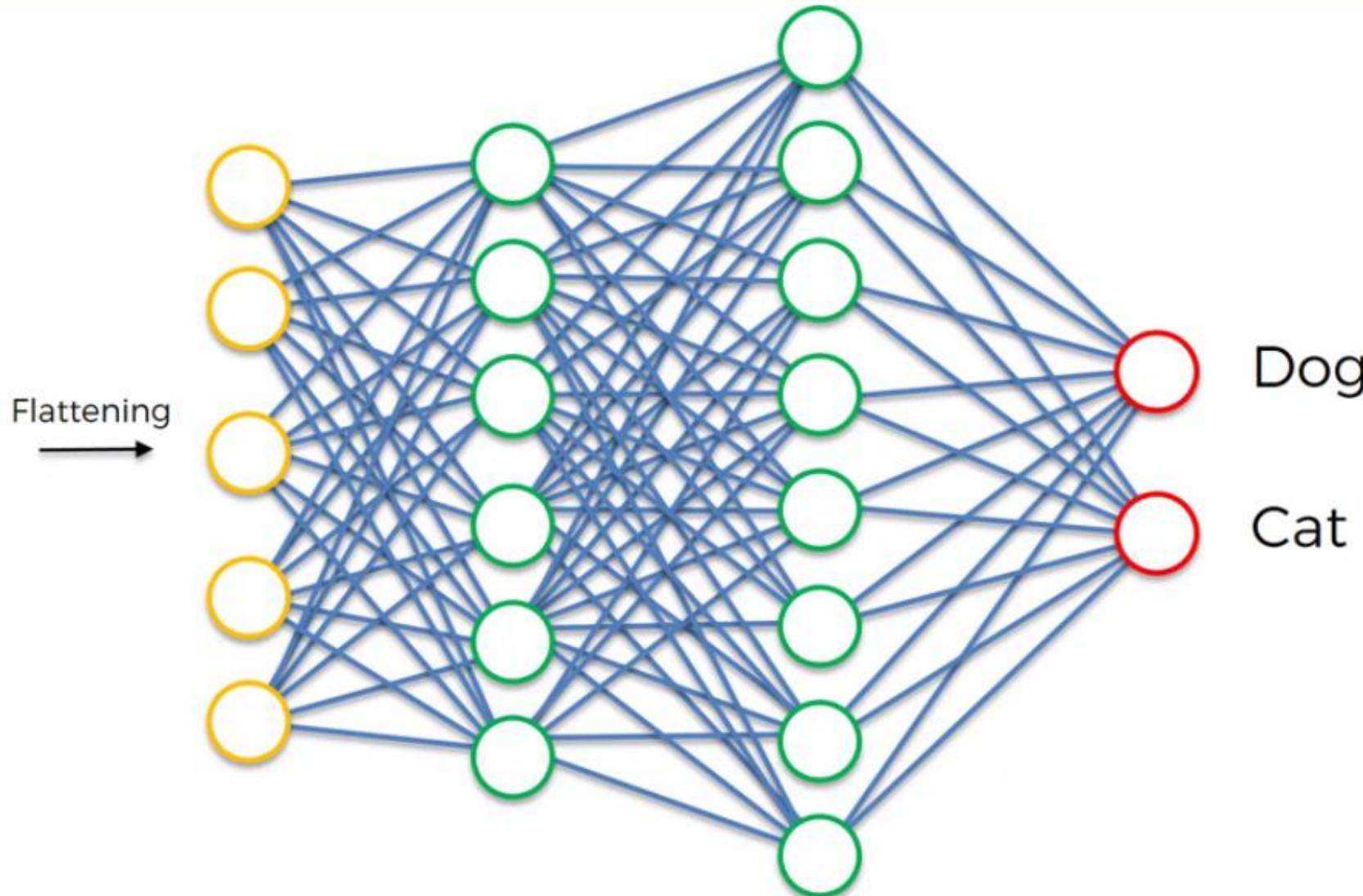


Step 4 - Full Connection

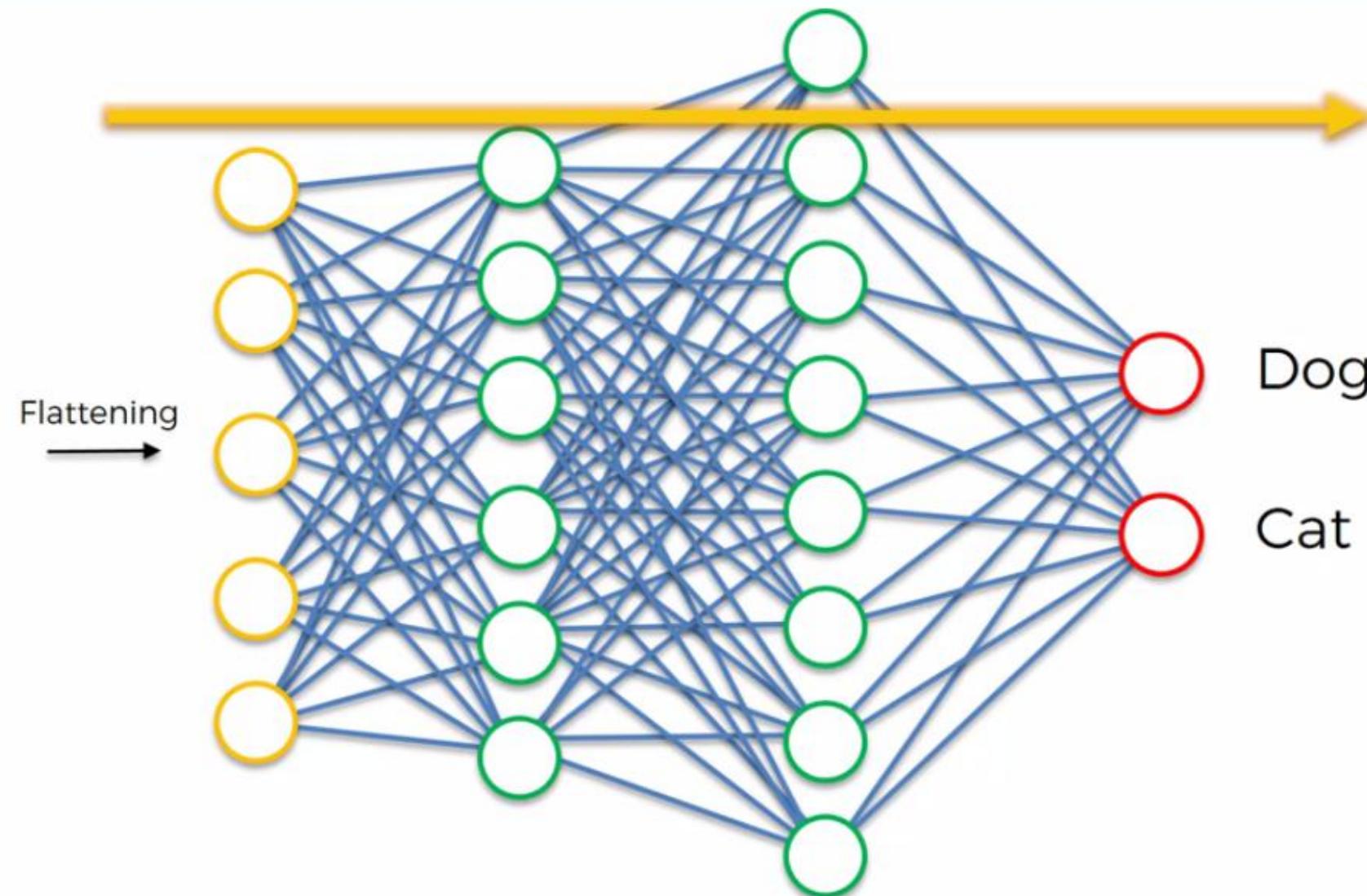
Step 4 - Full Connection



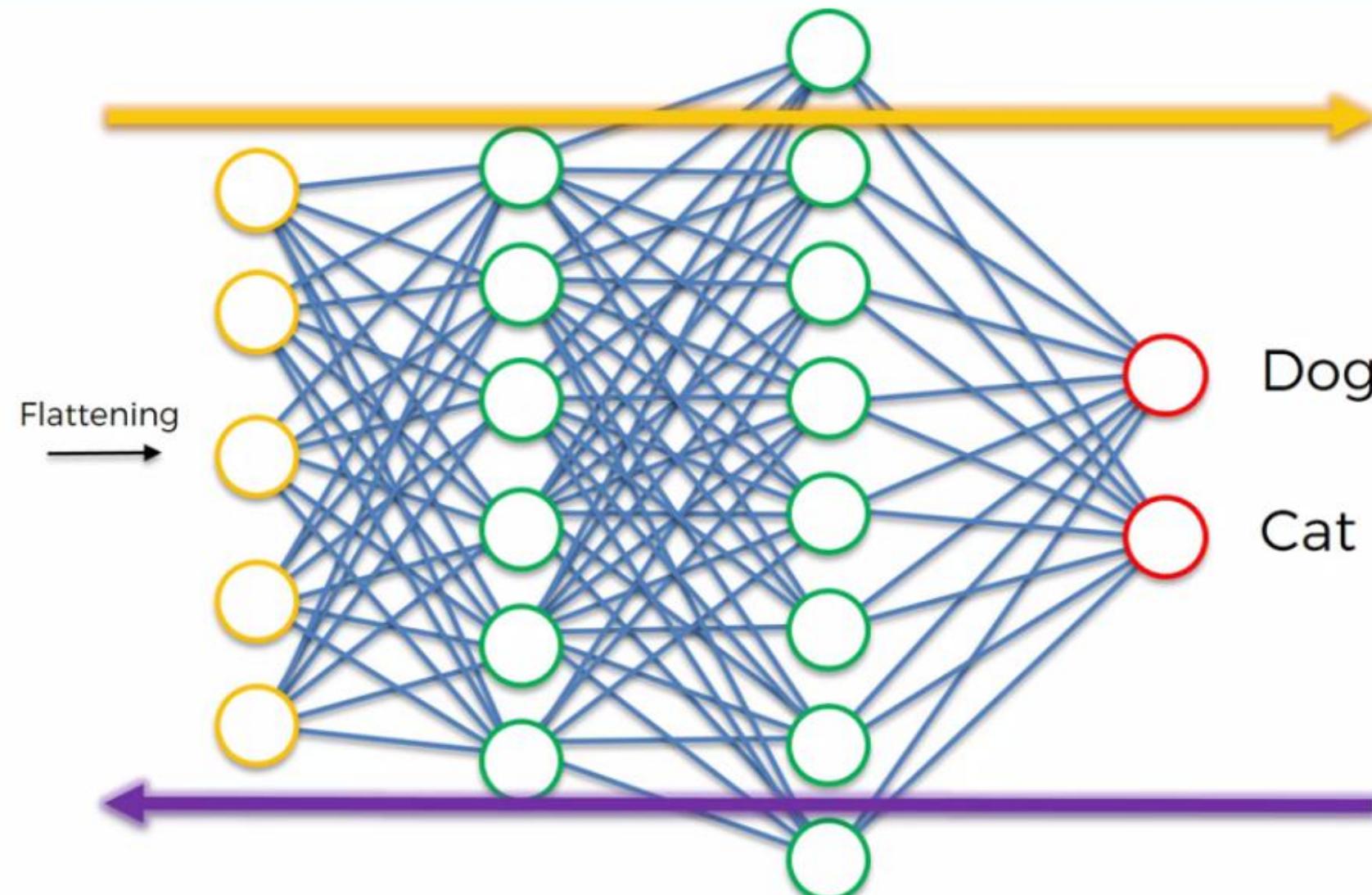
Step 4 - Full Connection



Step 4 - Full Connection



Step 4 - Full Connection



After calculating mean square error
backpropagation is done and weights are adjusted
thereafter.

Summary

