

Computer Vision

CVI620

Session 21
03/2025

What is Left?

9 sessions

1. Optimization and Loss Function
2. Code + Logistic Regression
3. ML and Images
4. Perceptron and Neural Networks
5. Deep Neural Networks
6. Convolution Neural Networks (CNN)
7. Advanced CNNs
8. Project
9. Segmentation
10. Introduction to object detection and image generation methods with AI
11. Project

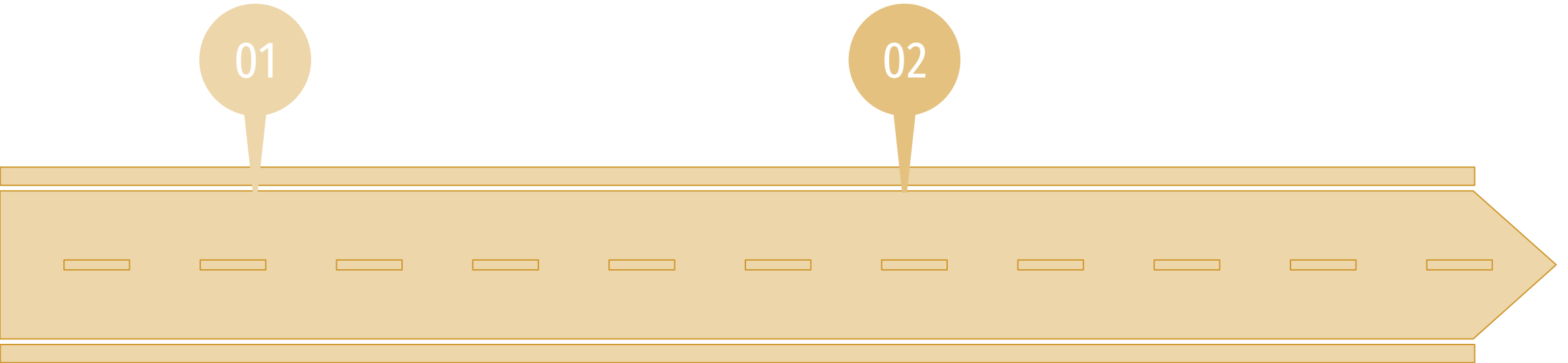
Agenda

Neural Networks Code

Convolutional
Neural Networks

01

02



Frameworks for Neural Networks

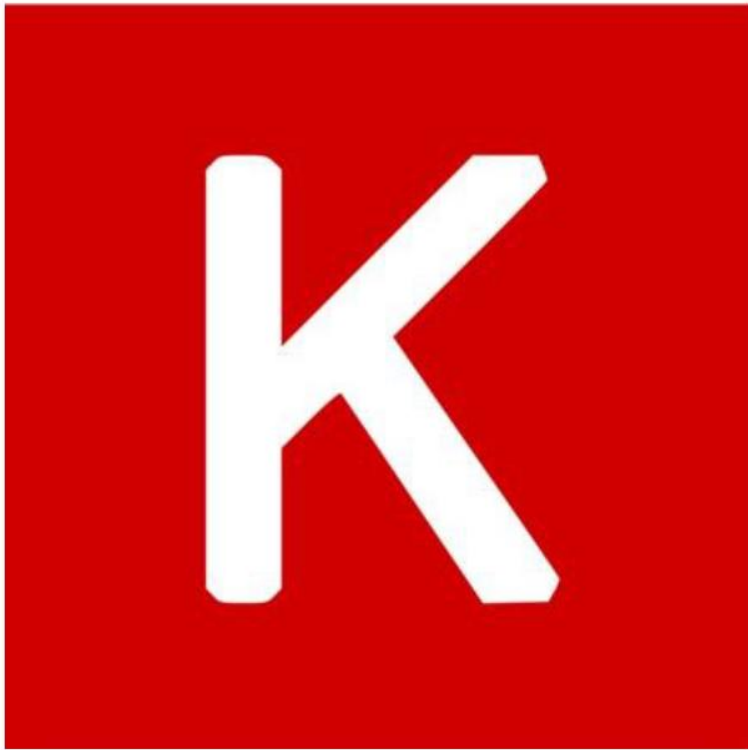


Google- 2015



Facebook - 2016

Keras

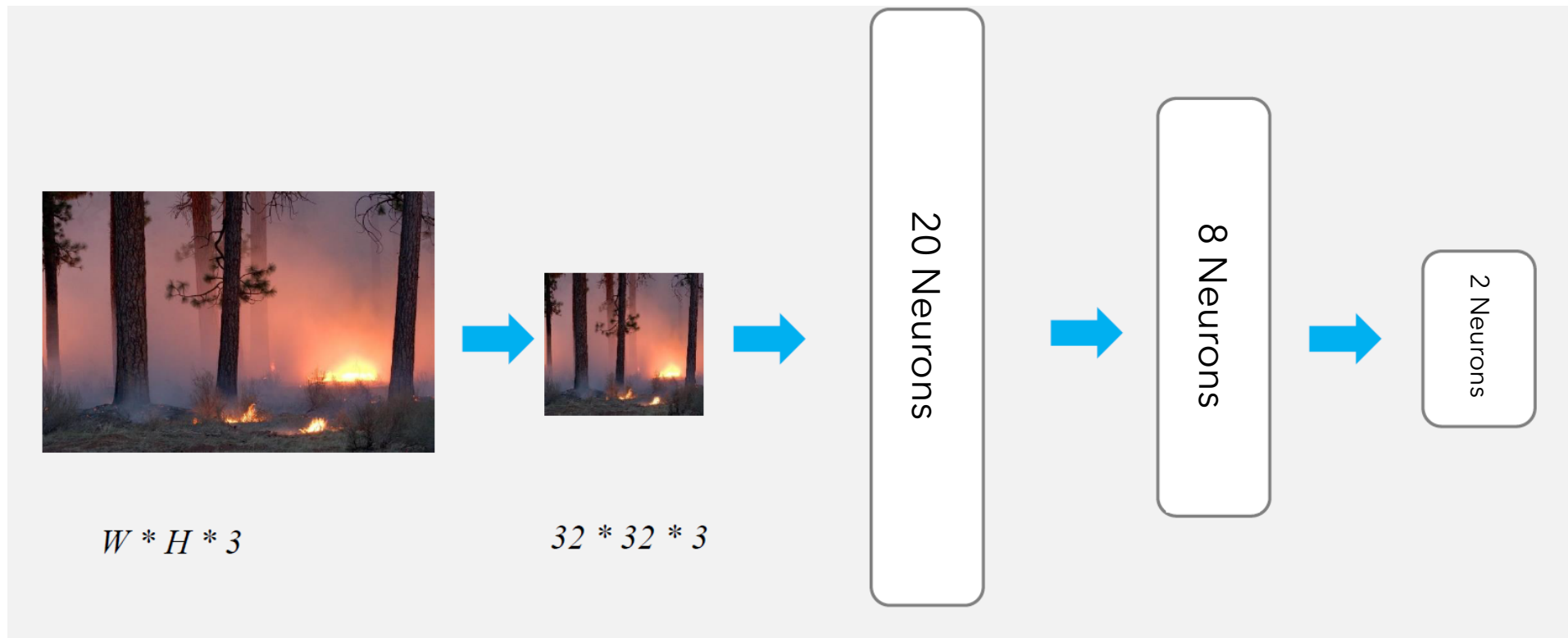


- By Francois Chollet
- Easier code
- From tf v2



Fire detection with Neural Networks

Architecture




Type of Encoding

Cat	→	0
Dog	→	1
Bird	→	2
Horse	→	3



Integer Encoding


```
from sklearn import preprocessing  
le = preprocessing.LabelEncoder()  
label = ["cat", "dog", "pandas", "fire"]  
out = le.fit_transform(label)  
print(out)
```



[1 2 0 1]

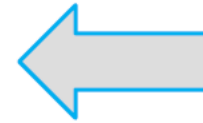
Type of Encoding

Cat → [1 0 0 0]

Dog → [0 1 0 0]

Bird → [0 0 1 0]

Horse → [0 0 0 0]



One Hot Encoding

```
from tensorflow.keras.utils import to_categorical  
  
labels = [1, 2, 0, 1]  
  
encode = to_categorical(labels)  
  
print(encode)
```

```
[[0.  1.  0.]  
 [0.  0.  1.]  
 [1.  0.  0.]  
 [0.  1.  0.]
```

Defining Neural Network

Sequential

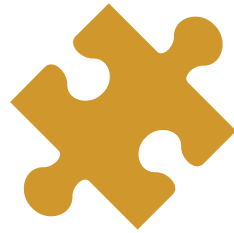
Functional
API

Model
Subclassing

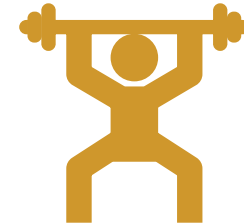
Steps



Design and define
NN



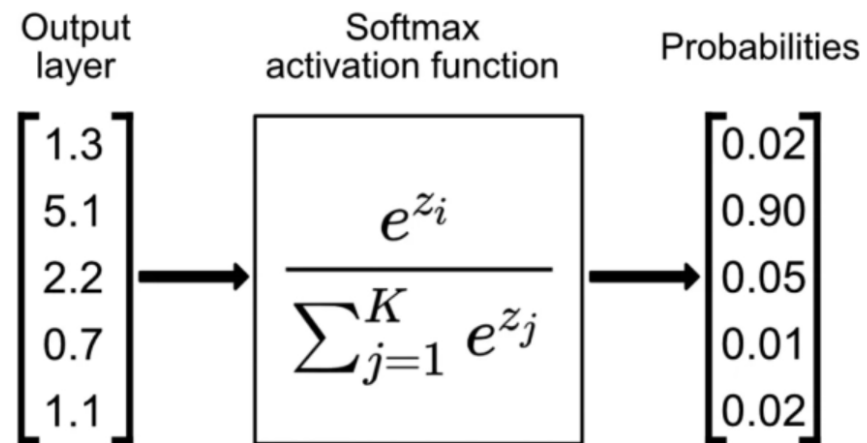
Compile



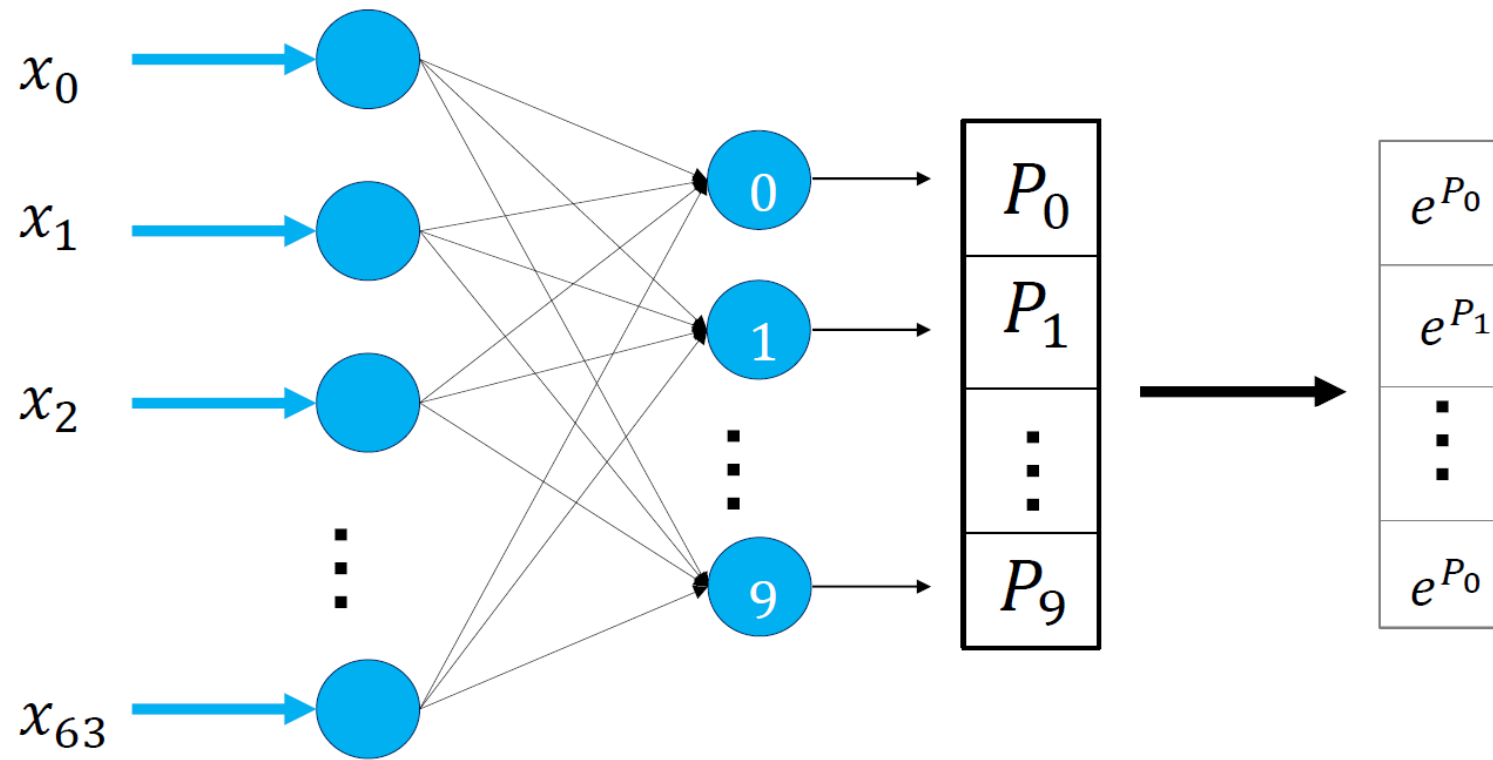
Fit

Softmax

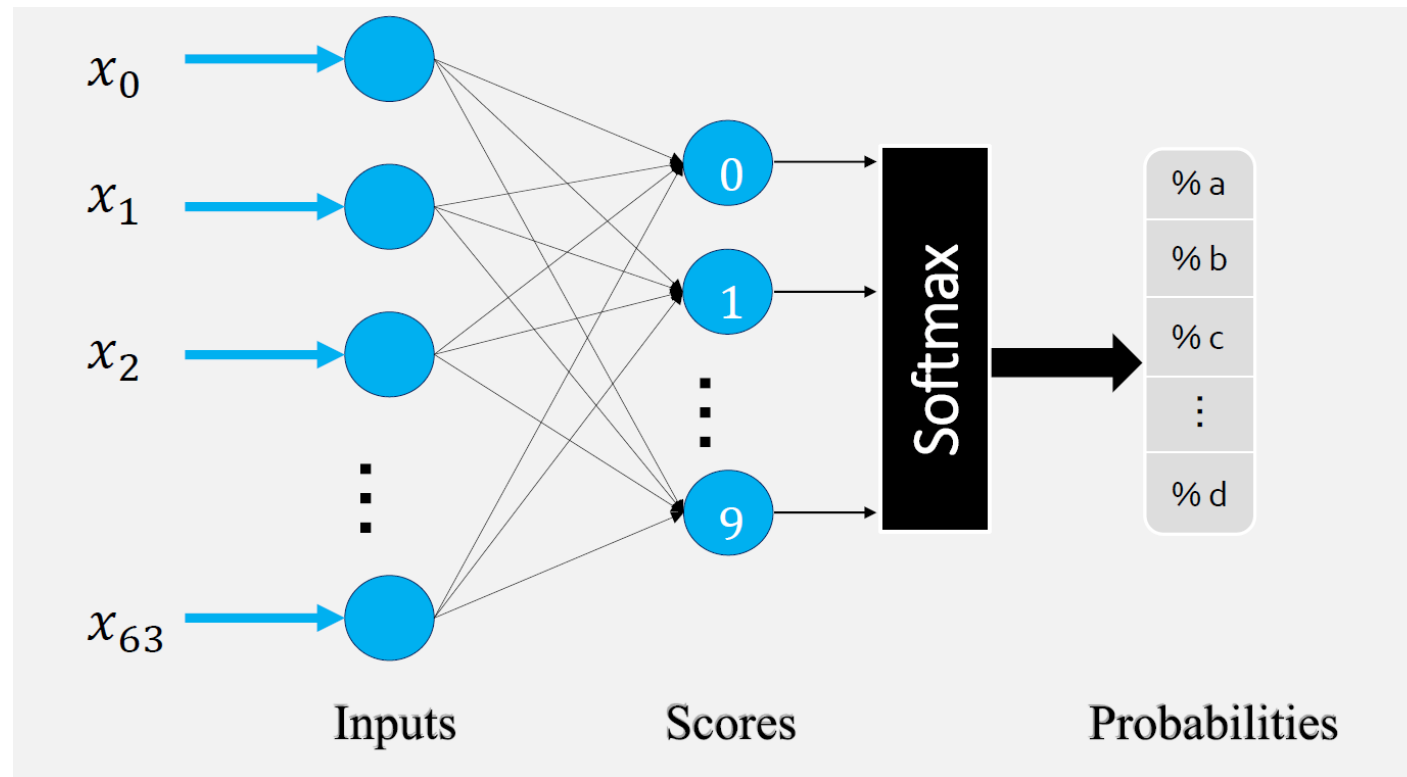
- An activation function
- Turns a vector of raw scores (logits) into probabilities that sum to 1.



Softmax

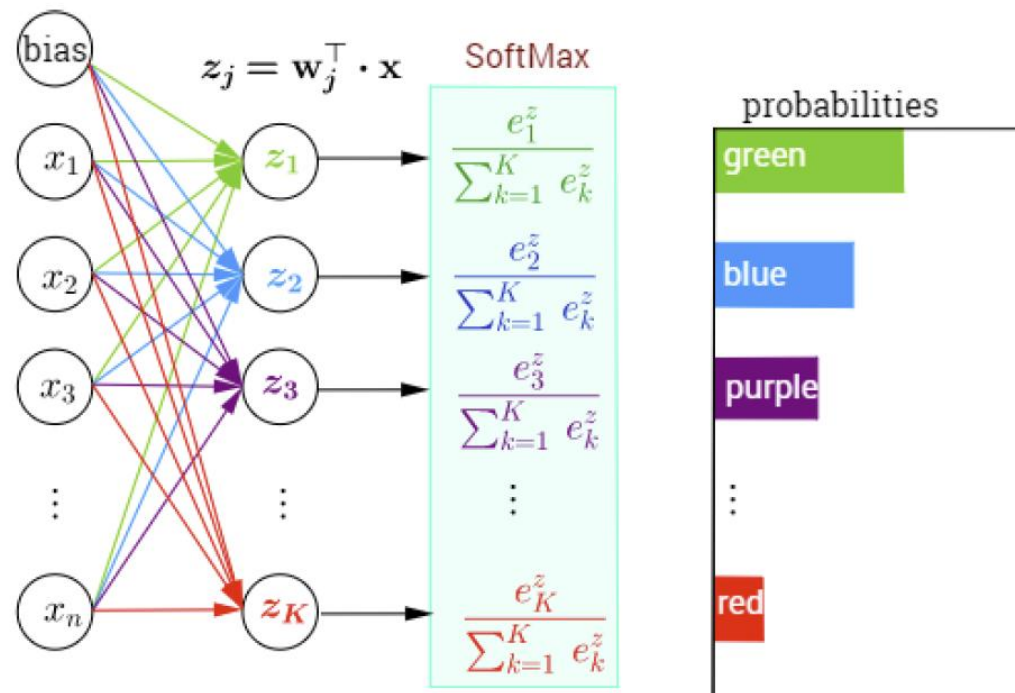


Softmax in Neural Networks



Summary

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_K \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^\top \\ \mathbf{w}_2^\top \\ \mathbf{w}_3^\top \\ \vdots \\ \mathbf{w}_K^\top \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

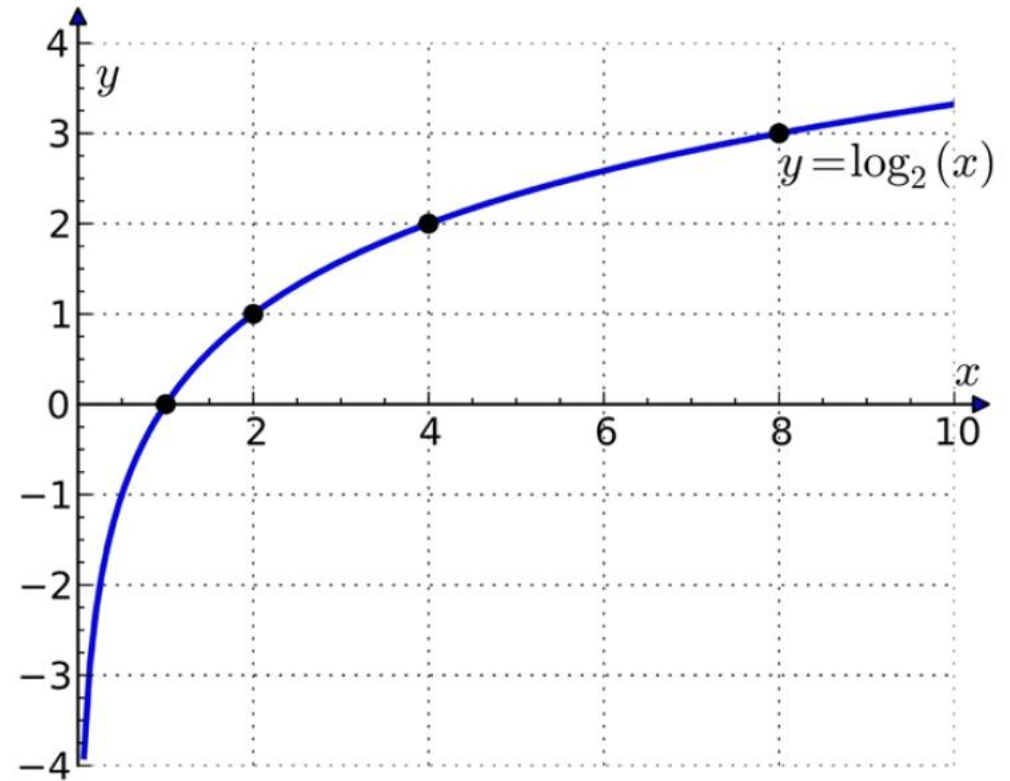




Loss Function For Classification

Cross Entropy Loss

$$loss = - \sum_{i=1}^n y_i \log(y'_i)$$




Cross Entropy Loss

Prediction

0.15	0.11	0.19	0.37	0.20	0.9	0.12	0.32	0.13	0.07
-------------	-------------	-------------	-------------	-------------	------------	-------------	-------------	-------------	-------------

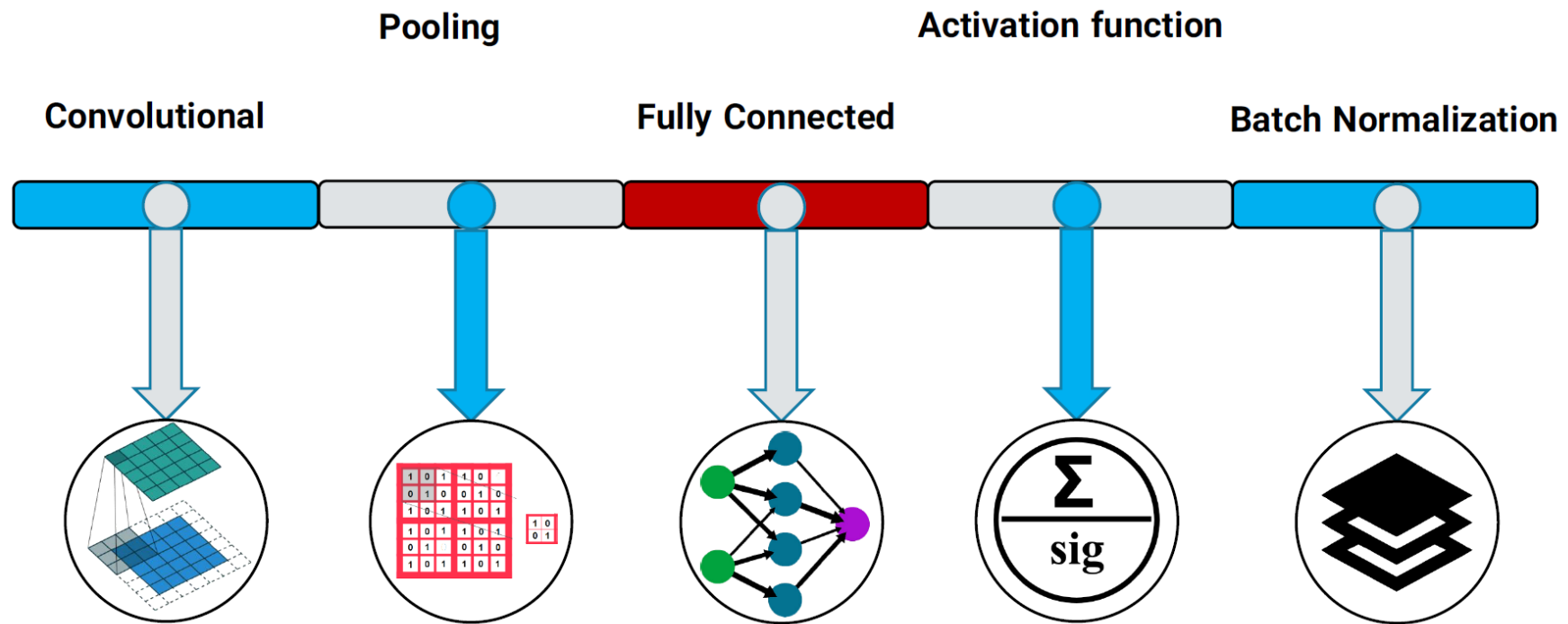
True Labels

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

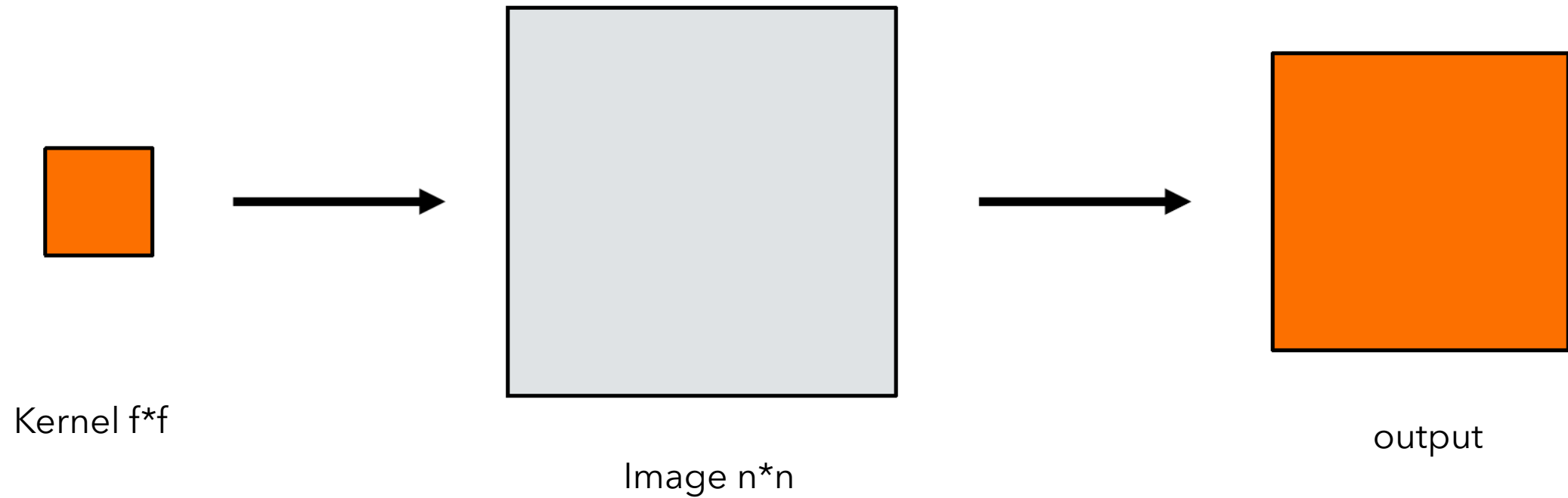


Convolutional Neural Networks (CNNs)

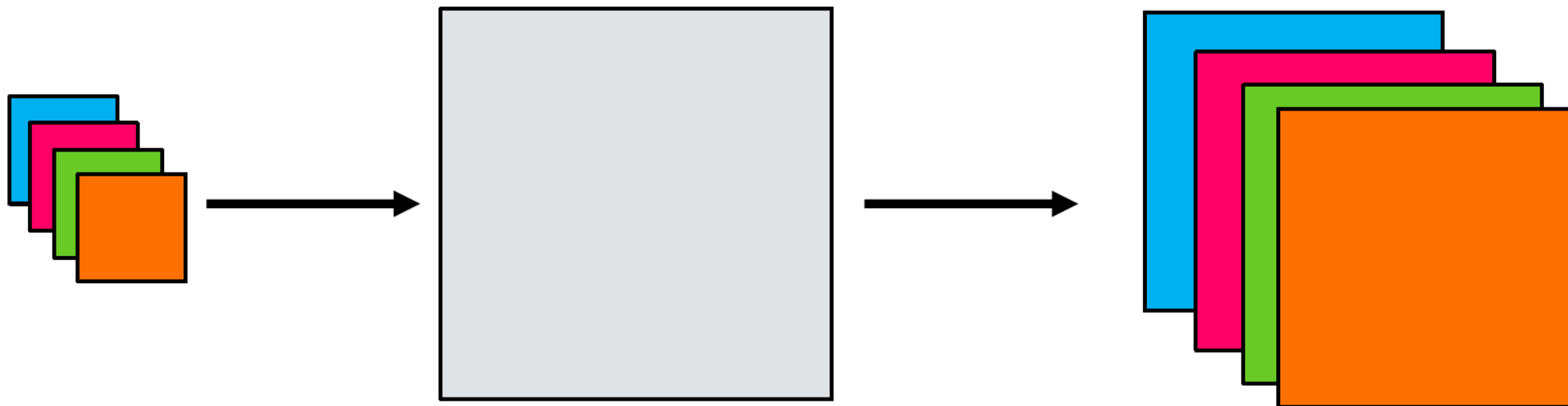
Layers in CNNs



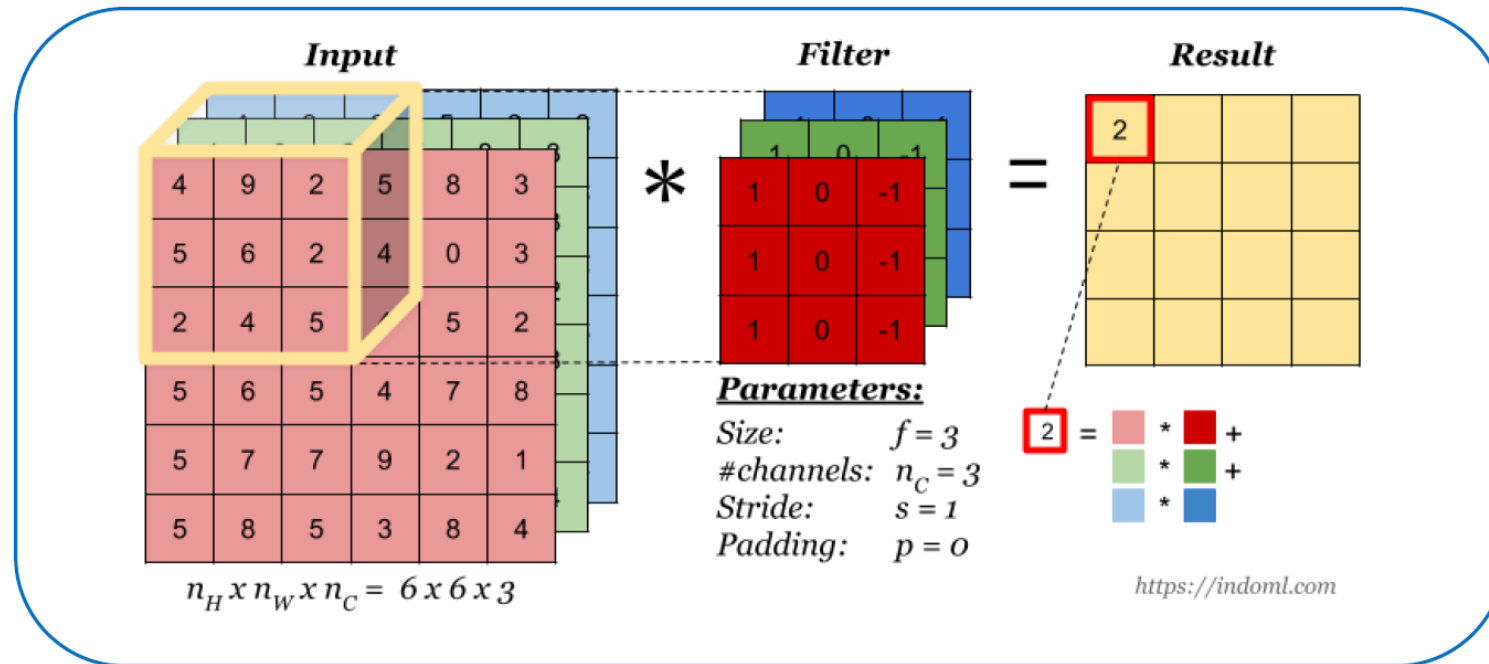
Convolution Layer



Multiple Convolutions



Convolution for Colored Images



Padding

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

$6 \times 6 \rightarrow 8 \times 8$

*

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

3×3

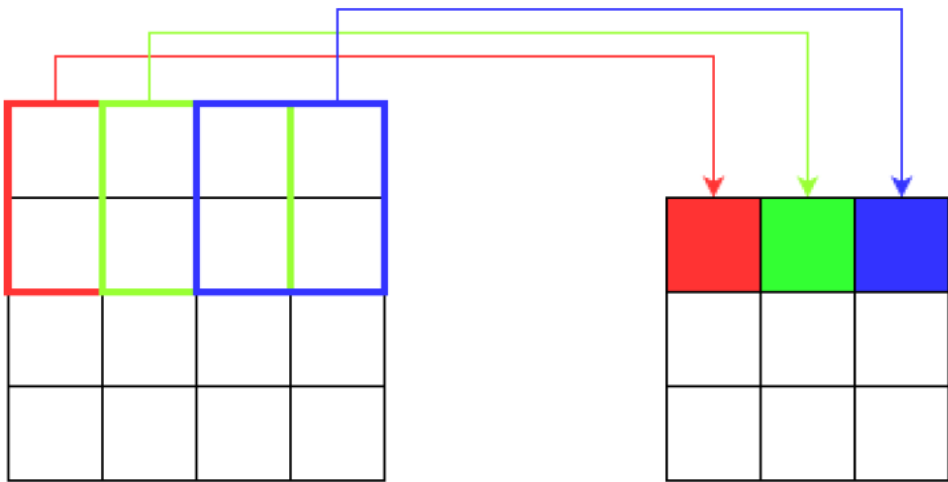
=

-10	-13	1			
-9	3	0			

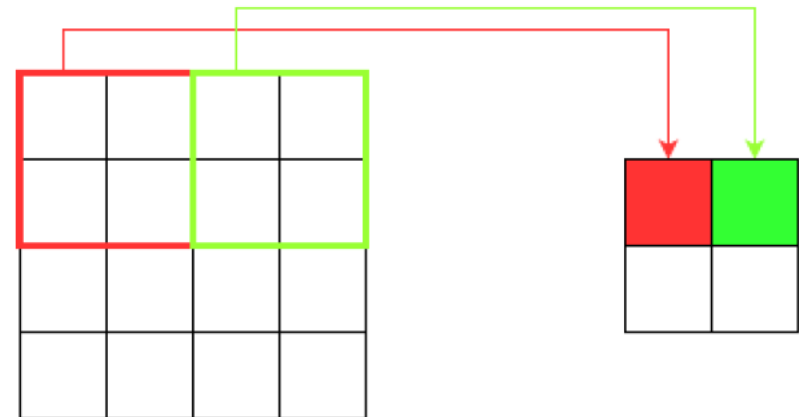
6×6

Stride

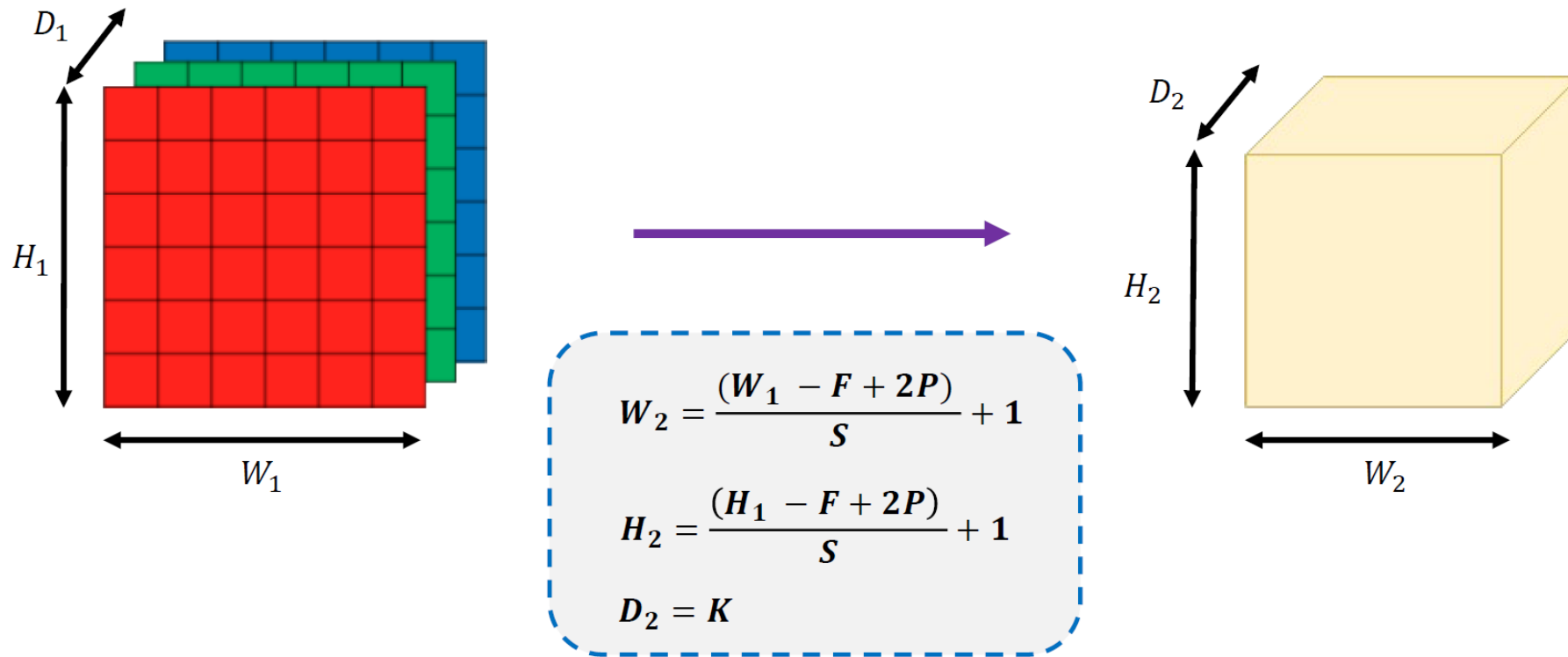
S = 1



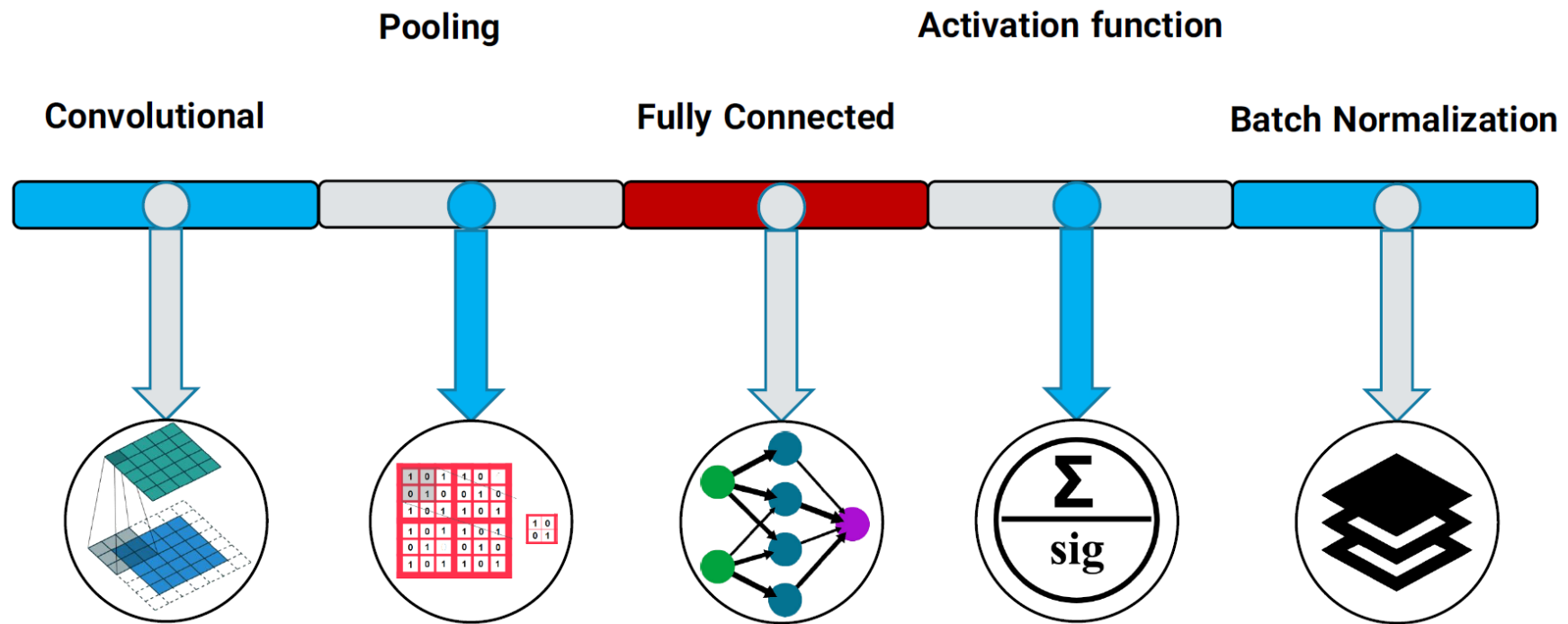
S = 3



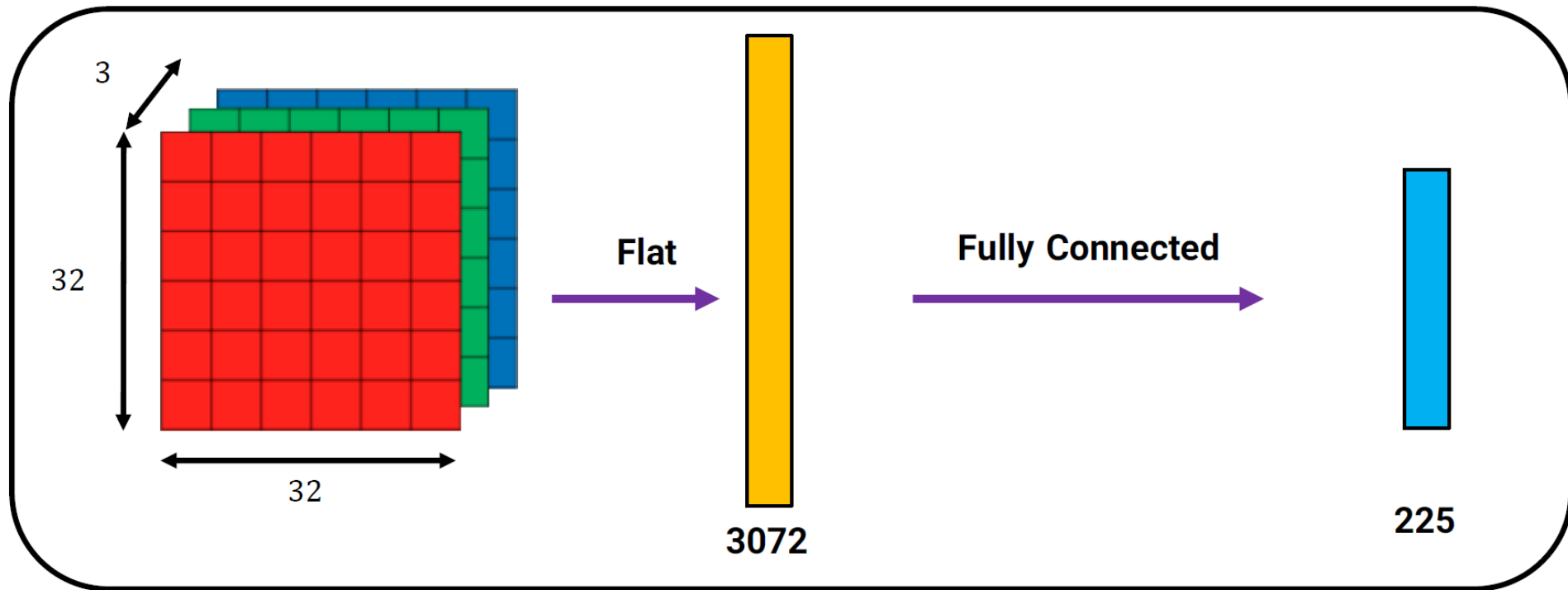
Post Convolution Dimensions



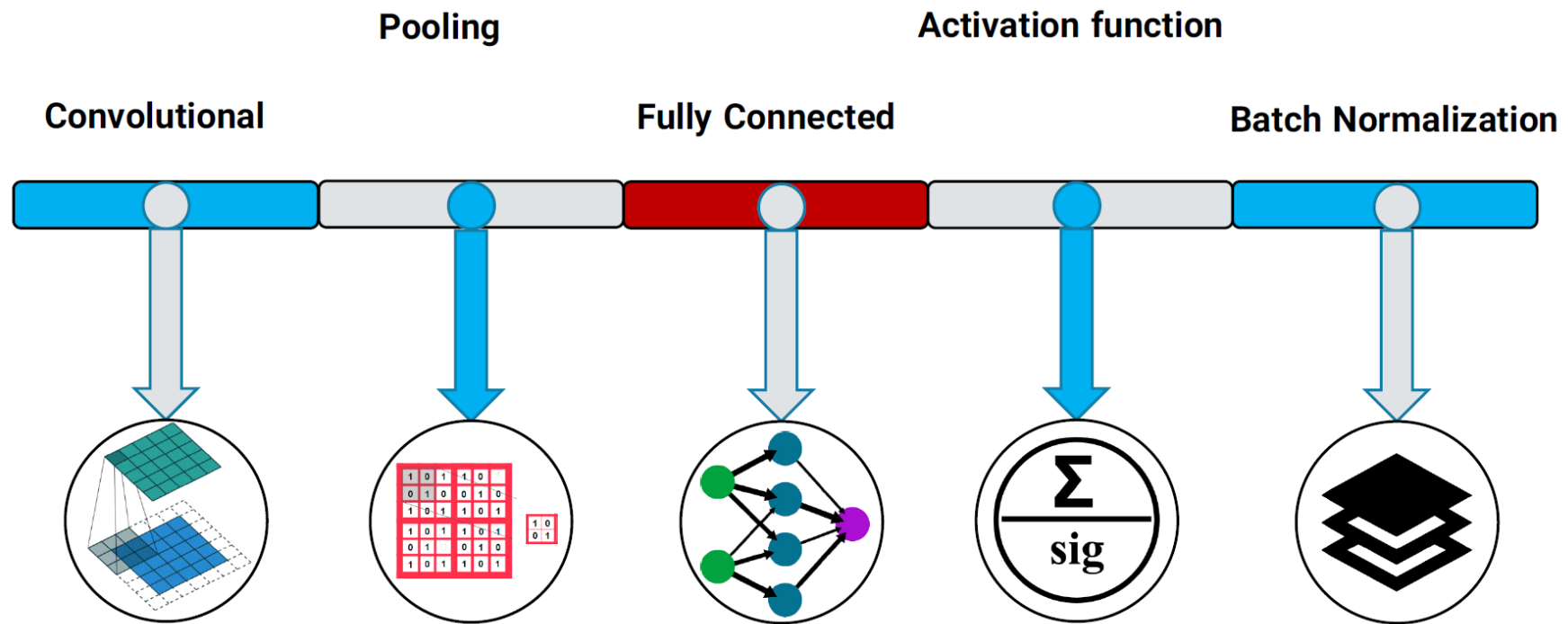
Layers in CNNs



MLP



Layers in CNNs



MaxPooling

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

Max Pool
→
Filter - (2 x 2)
Stride - (2, 2)

9	7
8	6

Average Pooling

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

Average Pool
→
Filter - (2 x 2)
Stride - (2, 2)

4.25	4.25
4.25	3.5

CNNs

