Modern Computer Vision & Generative AI

Powered by

Keras CV

Presenter

/arifmazumder

Mohammed Arif, PhD

Lead Data Scientist

Big Data | Machine Learning | AI

Microsoft CERTIFIED AZURE DATA SCIENTIST ASSOCIATE

Microsoft CERTIFIED AZURE DATA ENGINEER ASSOCIATE

CCA cloudera SPARK & HADOOP DEVELOPER

Mohammed Arif has more than nineteen (19) years of working experience in Information Communication and Technology (ICT) industry. The highlights of his career are more than seven (7) years of holding various senior management and/or C-Level and had five (5) years of international ICT consultancy exposure in various countries (APAC and Australia), specially on Big Data, Data Engineering, Machine Learning and AI arena.

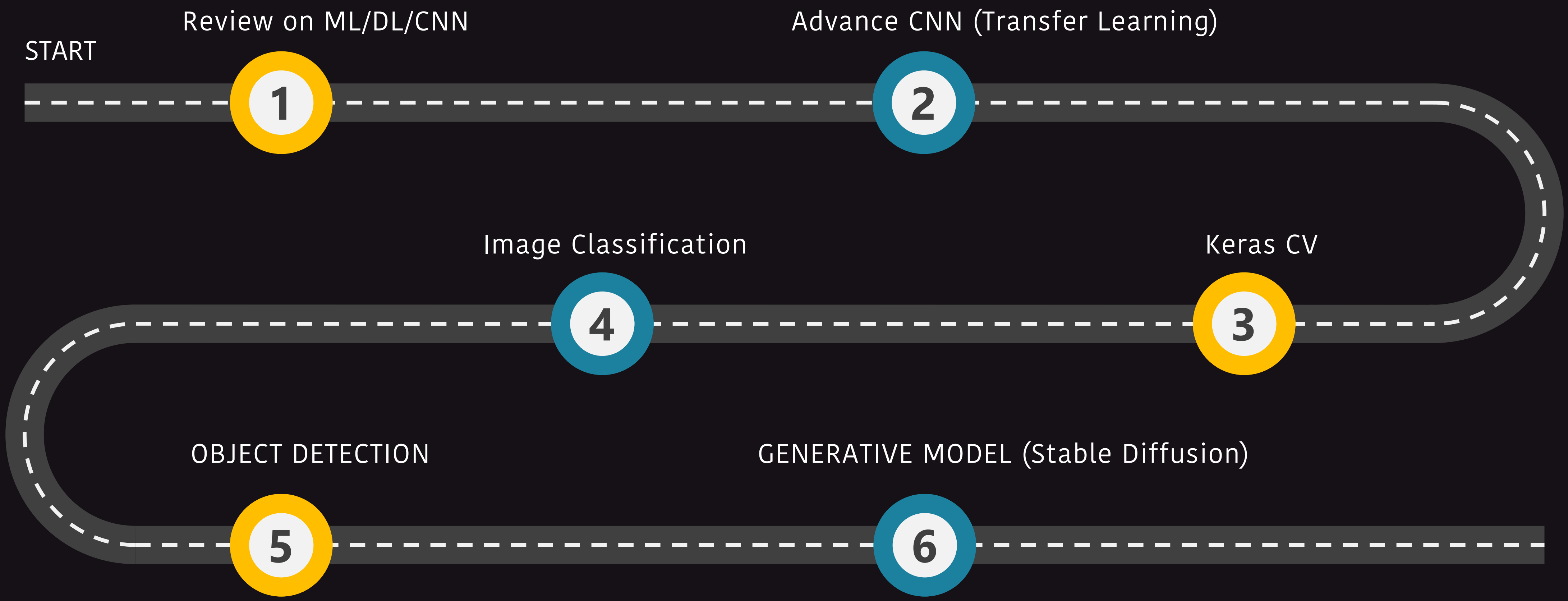
He is also Certified Trainer for Microsoft & Cloudera.

MICROSOFT CERTIFIED TRAINER Microsoft MCT Trainer 2023 - 2024

Related Resources

https://arif.works/unikl

LEARNING JOURNEY



Disclaimer

- Computer Vision is a vast domain
- Emphasize on Practical / Industry Implementation
- Take it as Experience Sharing Session

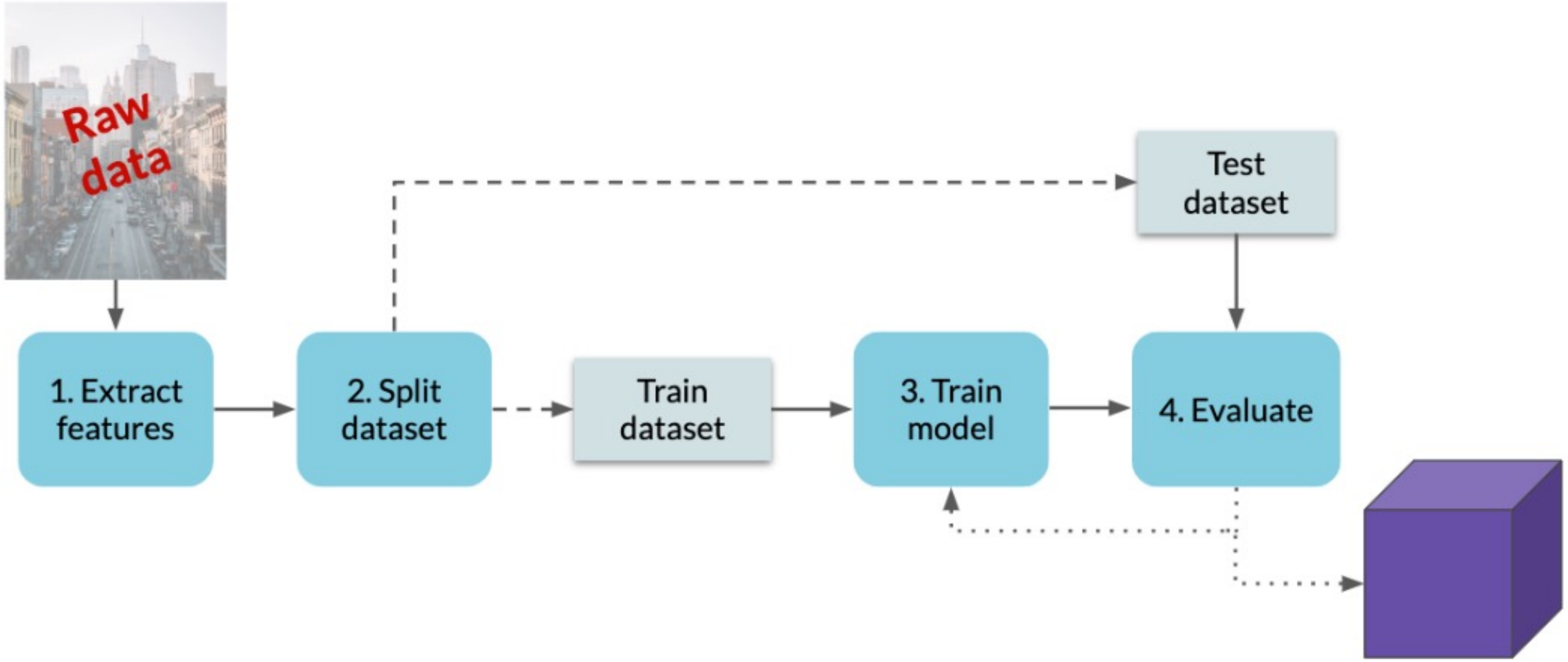
1.0 Review on ML & DL

- Machine Learning Process
- Deep Learning Process
- Convolutional Neural Network
- Hands-on

Review on Machine Learning

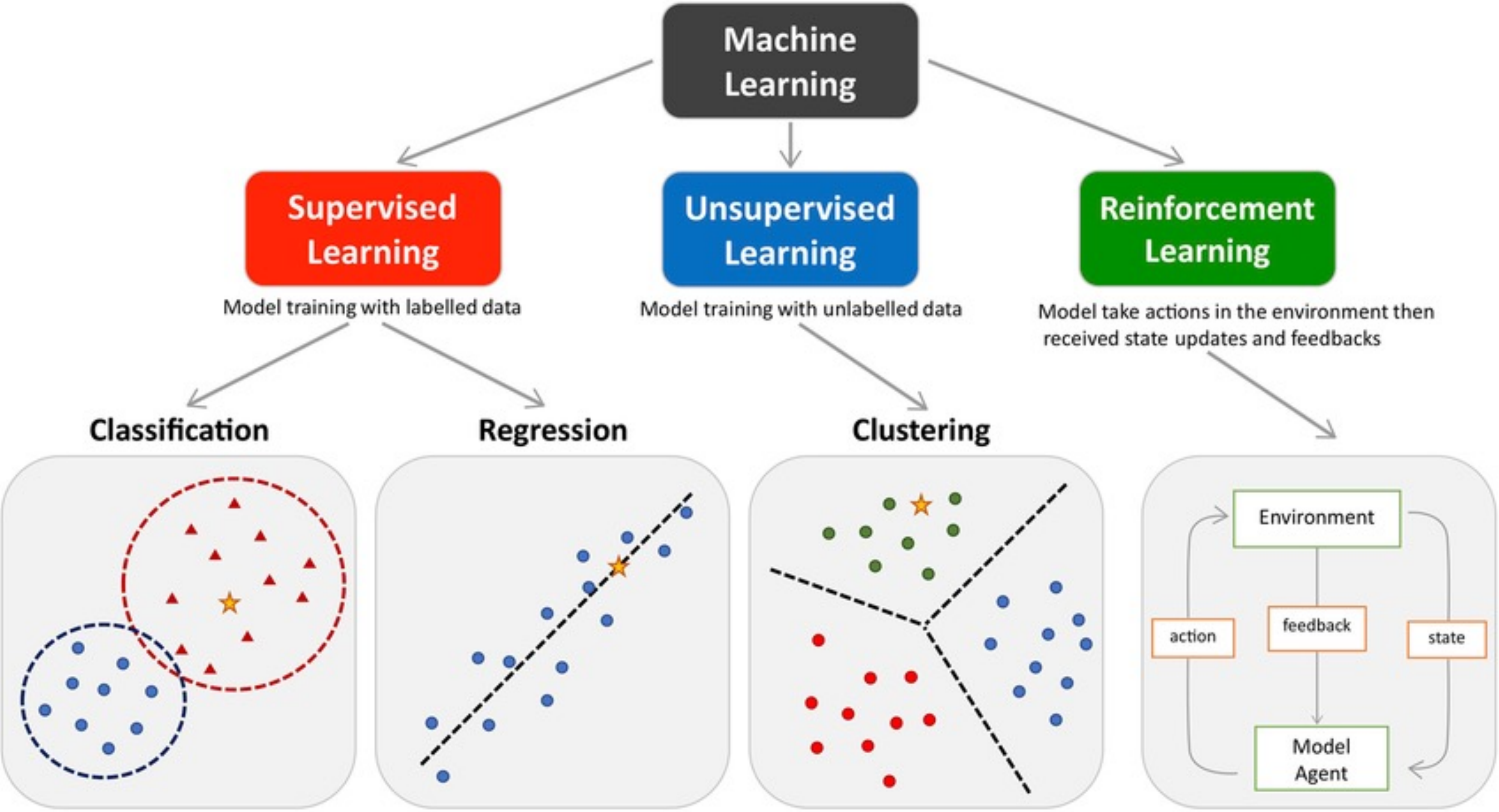
Machine Learning

Machine Learning allows computers to find **hidden insights** without being explicitly programmed where to look.



Review on Machine Learning

Types of Machine Learning



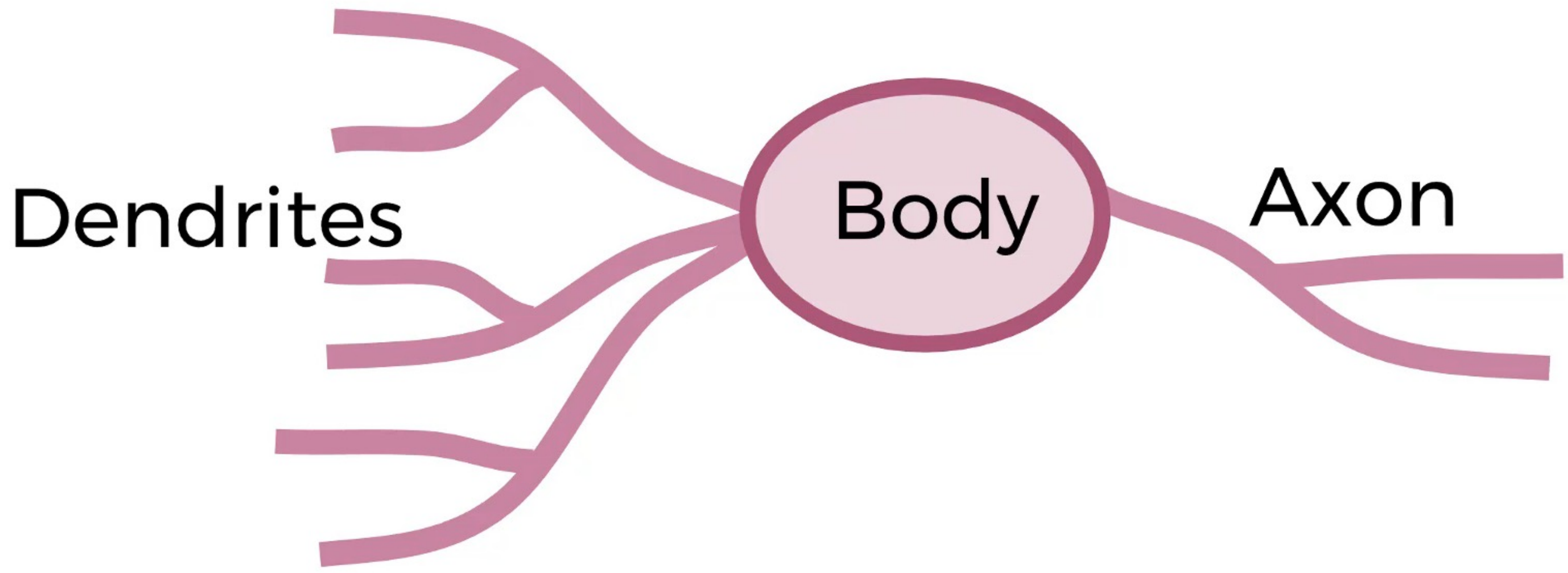
Review on Machine Learning

Model Evaluation

		predicted condition		
	total population	prediction positive	prediction negative	Prevalence $= \frac{\Sigma \text{ condition positive}}{\Sigma \text{ total population}}$
true condition	condition positive	**True Positive (TP)**	**False Negative (FN)** (type II error)	True Positive Rate (TPR), Sensitivity, Recall, Probability of Detection $= \frac{\Sigma \text{ TP}}{\Sigma \text{ condition positive}}$
	condition negative	**False Positive (FP)** (Type I error)	**True Negative (TN)**	False Positive Rate (FPR), Fall-out, Probability of False Alarm $= \frac{\Sigma \text{ FP}}{\Sigma \text{ condition negative}}$
	Accuracy $= \frac{\Sigma \text{ TP} + \Sigma \text{ TN}}{\Sigma \text{ total population}}$	Positive Predictive Value (PPV), Precision $= \frac{\Sigma \text{ TP}}{\Sigma \text{ prediction positive}}$	False Omission Rate (FOR) $= \frac{\Sigma \text{ FN}}{\Sigma \text{ prediction negative}}$	Positive Likelihood Ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$
		False Discovery Rate (FDR) $= \frac{\Sigma \text{ FP}}{\Sigma \text{ prediction positive}}$	Negative Predictive Value (NPV) $= \frac{\Sigma \text{ TN}}{\Sigma \text{ prediction negative}}$	Negative Likelihood Ratio (LR–) $= \frac{\text{FNR}}{\text{TNR}}$

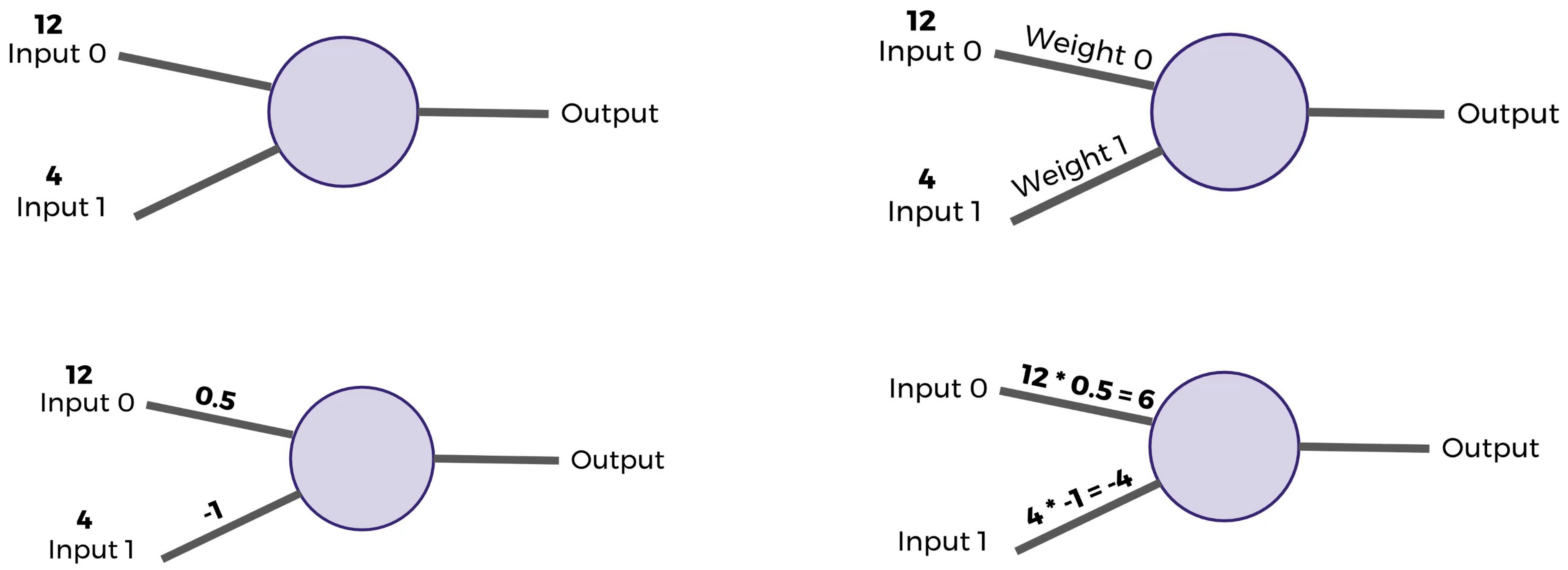
Review on Deep Learning

Biological Neuron



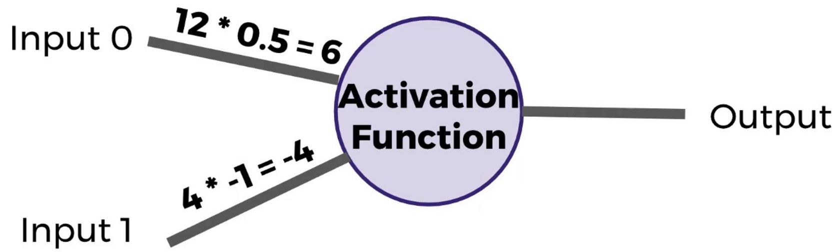
Review on Deep Learning

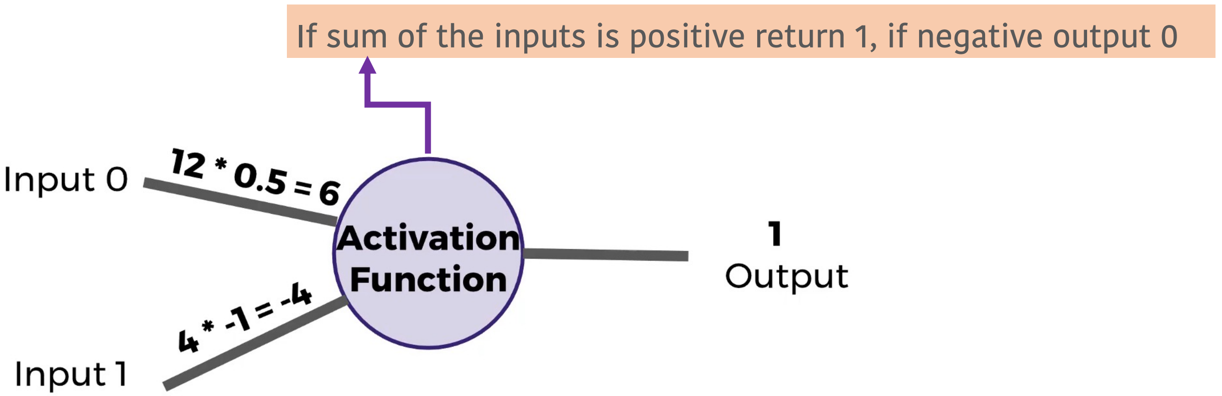
Perceptron

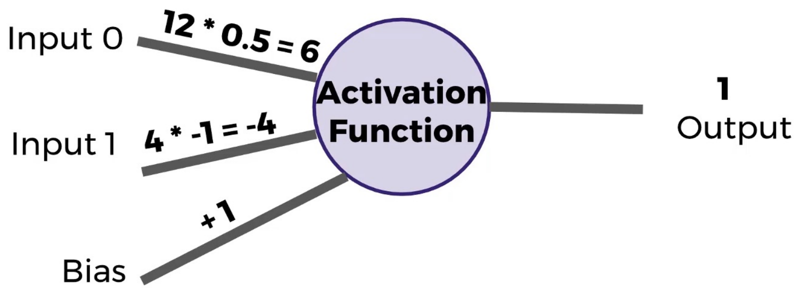


Review on Deep Learning

Perceptron



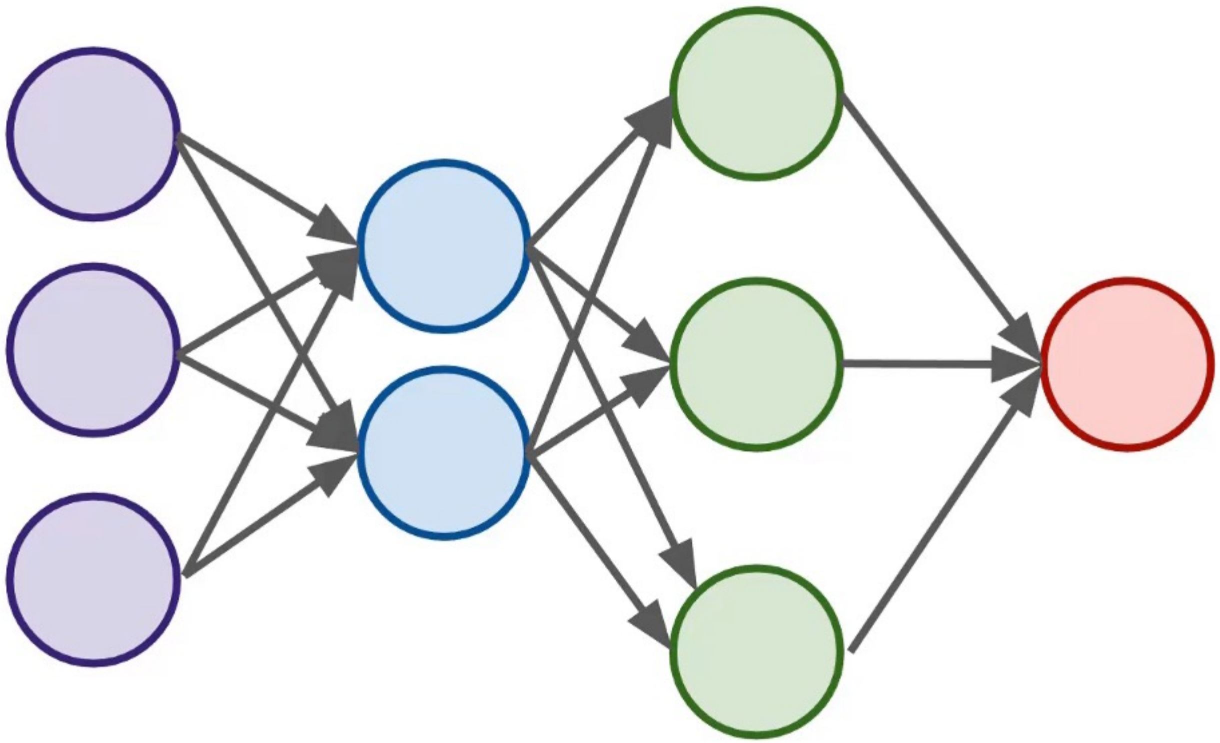




$$\sum_{i=0}^{n} w_i x_i + b$$

Review on Deep Learning

Multiple Perceptron Network

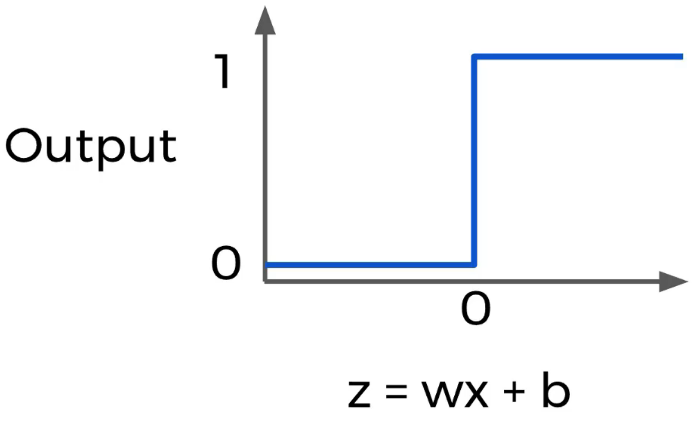


Input Layers, 2 Hidden Layers, Output Layer

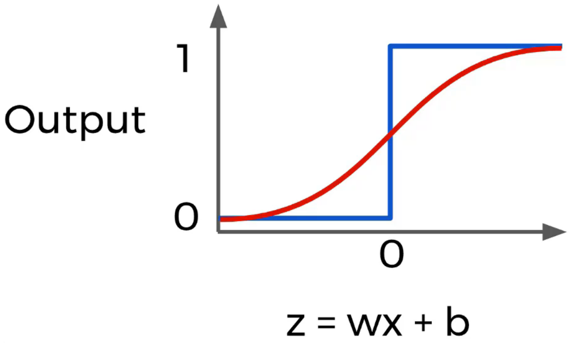
Review on Deep Learning

Activation Functions:

The primary role of activation functions is to **transform the weighted sum of inputs into a non-linear output**, enabling the network to learn complex patterns beyond simple linear relationships.

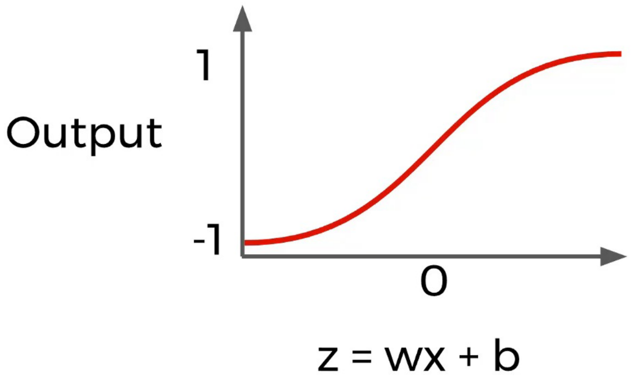


Sigmoid Functions



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

Hyperbolic Tangent: tanh(z)

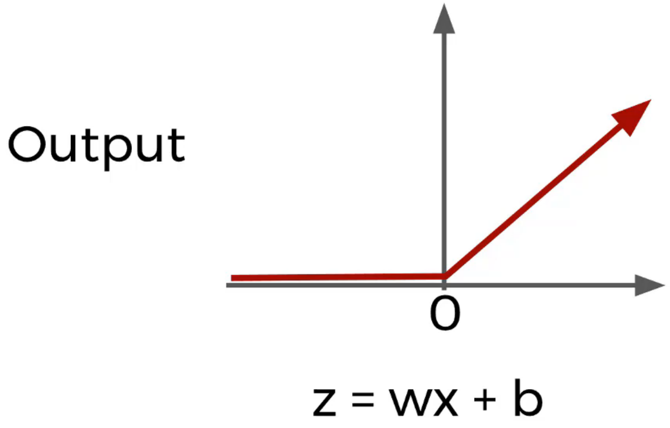


$$\cosh x = \frac{\mathbf{e}^{x} + \mathbf{e}^{-x}}{2}$$

$$\sinh x = \frac{\mathbf{e}^{x} - \mathbf{e}^{-x}}{2}$$

$$\tanh x = \frac{\sinh x}{\cosh x}$$

Rectified Linear Unit (ReLU): max (0,z)



Review on Deep Learning

Cost Functions:

Cost functions allow us to measure how well these neurons are performing or evaluating the performance of a neuron. It will measure how far off we are from expected value.

Quadratic Cost Function

$$C = \Sigma(y-a)^2 / n$$

Here,
y represent true value
a represent neuron's prediction

This calculation can cause a slowdown in our learning speed.

Cross Entropy

$$C = (-1/n)\ \Sigma\ (y \cdot \ln(a) + (1-y) \cdot \ln(1-a))$$

This cost function allows for faster learning.
The larger the difference, the faster the neuron can learn.

Review on Deep Learning

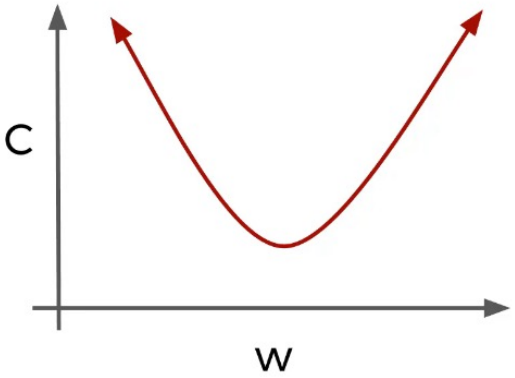
We need to figure out how we can use of our neurons and the measurement of error (our cost function) and then attempt to correct our prediction, in other words, “Learn”

Review on Deep Learning

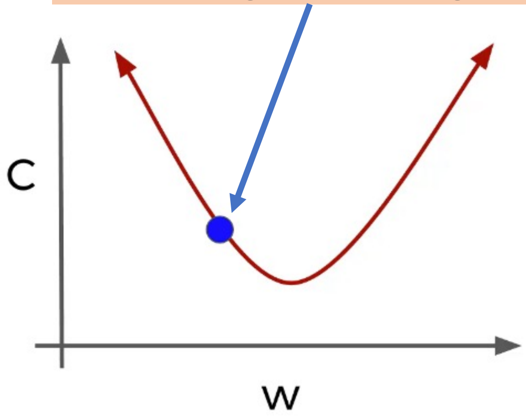
Gradient Descent:

Gradient descent is an optimization algorithm for finding the minimum of a function.

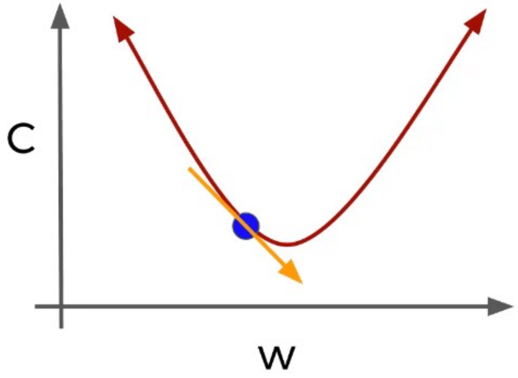
Why? We want to minimize the cost function.



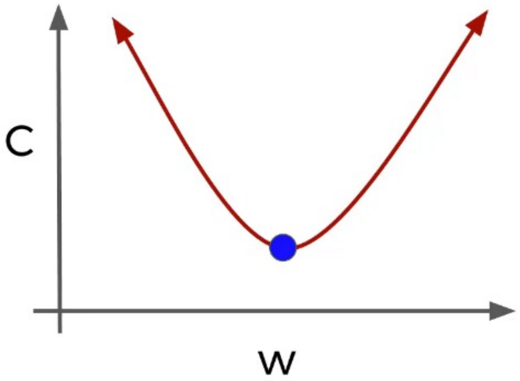
Random weights at the Beginning



We take steps proportional to the negative of the gradient



We end up taking step-by-step a descent along that gradient until we finally see that we get that minimum in our cost



Review on Deep Learning

Gradient Descent

Using gradient descents, we can figure out the best parameters for minimizing our costs. So for example, finding the best values for the weights of the neuron inputs.

Review on Deep Learning

Backpropagation

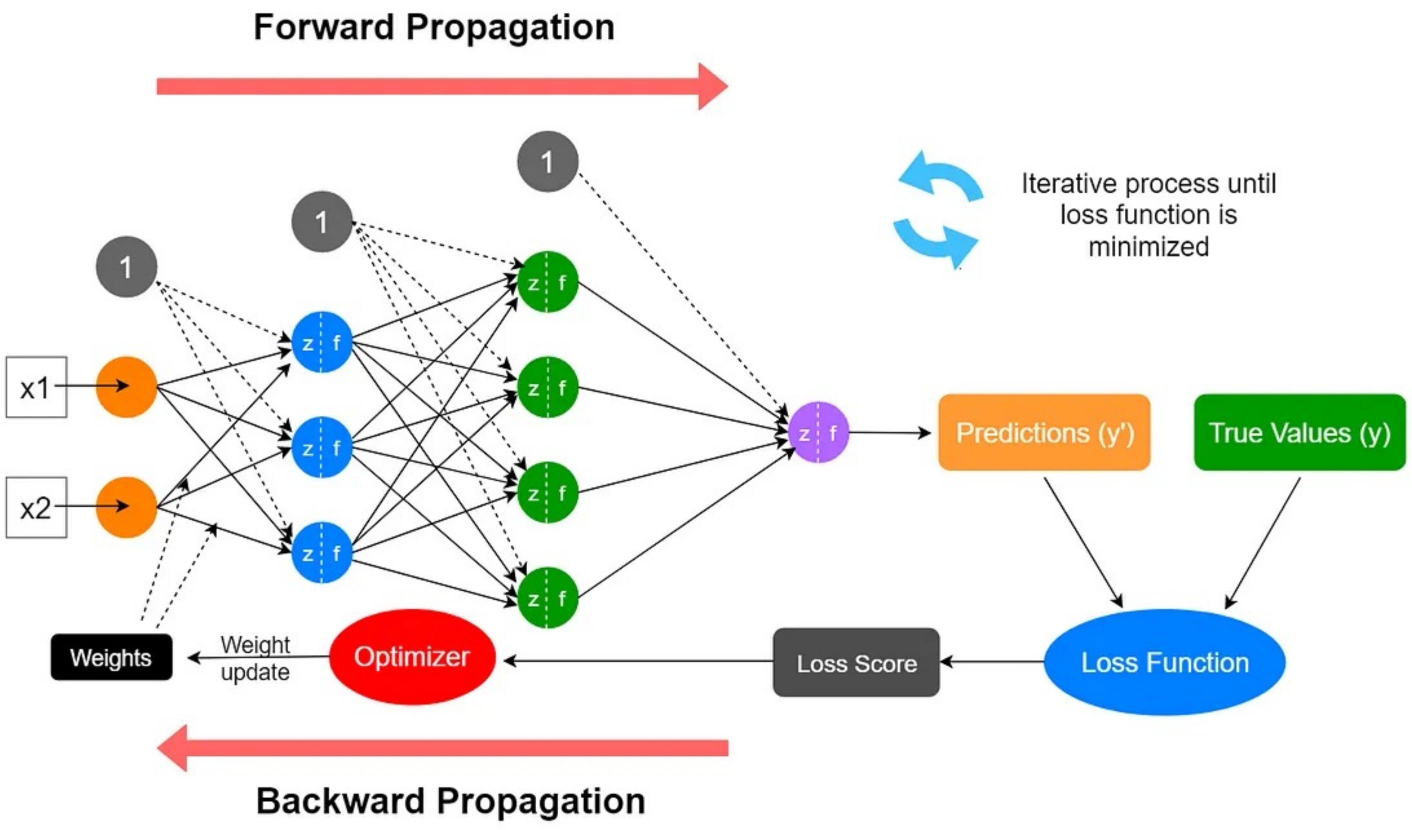
We now just have one issue to solve, how can we quickly adjust the optimal parameters or weights across our entire network?

There is where backpropagation comes in !

Review on Deep Learning

Backpropagation

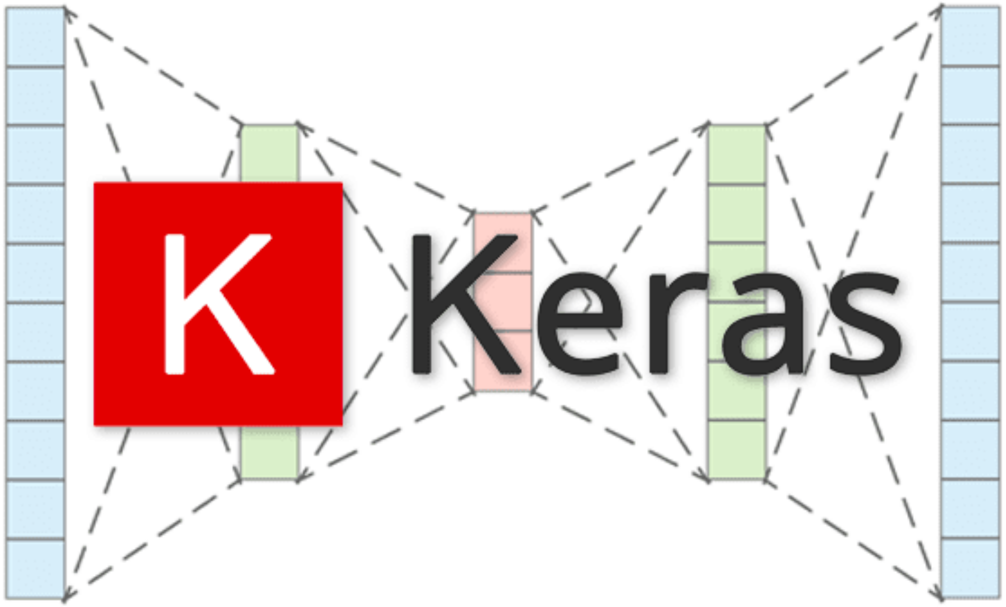
- ✓ Basically, backpropagation is used to calculate the **error contribution of each neuron** after a batch of data is processed.
- ✓ And it relies heavily on the **chain rule from calculus** to go back through the network and calculate these errors.
- ✓ The way backpropagation works is just it **calculates the error at the output and then distributes back through the network layers,** and it requires a known desired output for each input value.

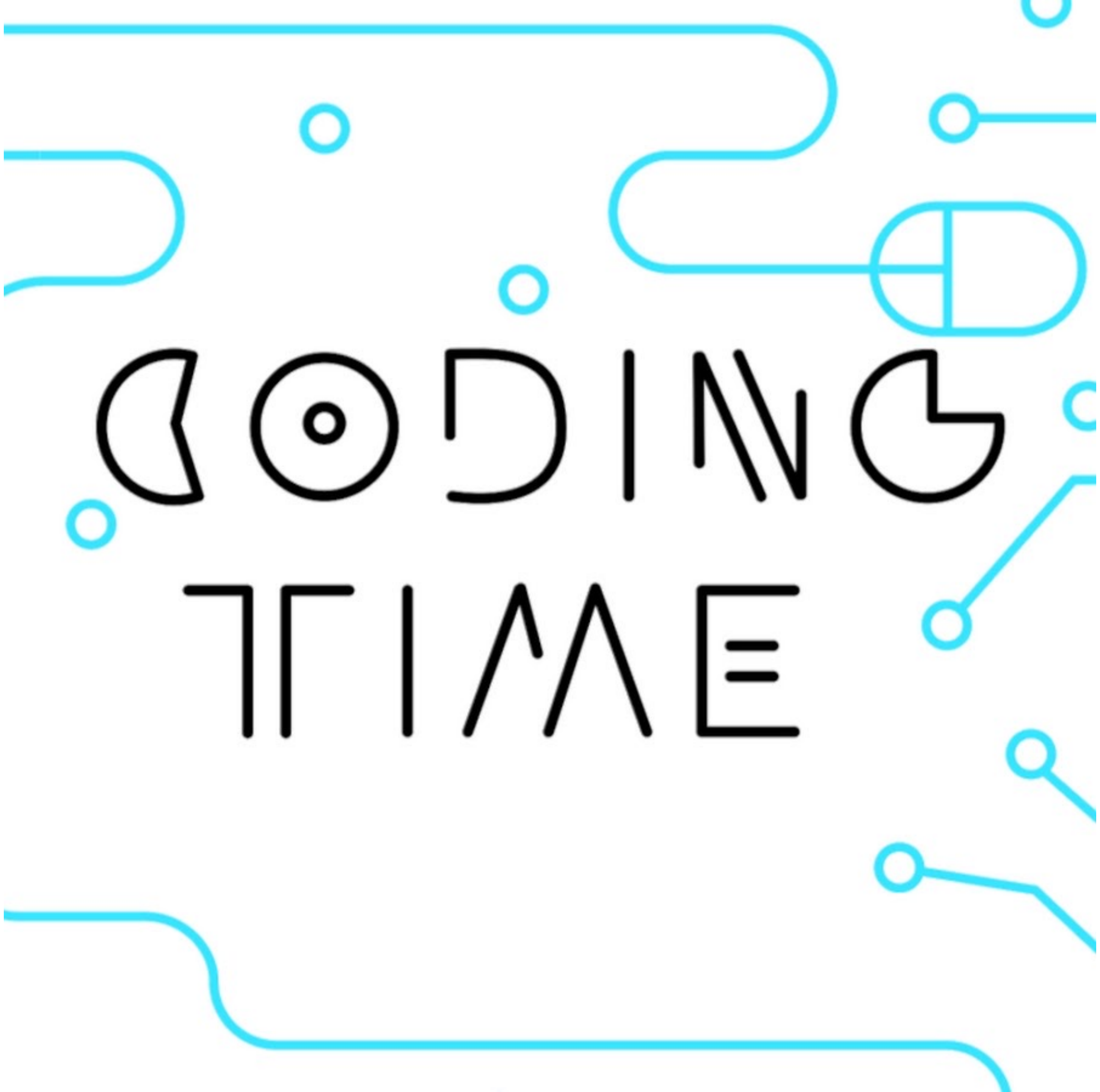


Review on Deep Learning

Keras

- ✓ We're going to be using a data set of features of counterfeit versus real bills.
- ✓ Keep in mind we're still working on just numerical features. We're not actually reading in image data directly.

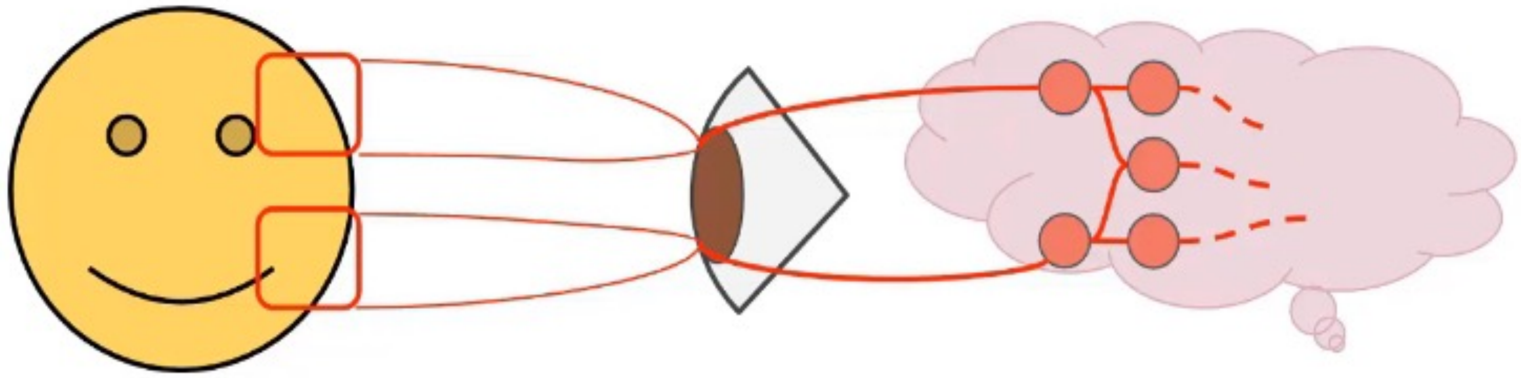



CODING
TIME

Review on Deep Learning

Convolutional Neural Network (CNN)

CNN also have their origins in biological research. And the biological research we base this off of is from a pair of scientists, David Hubel and Torson Weisel, who were studying the structure of the visual cortex in mammals.

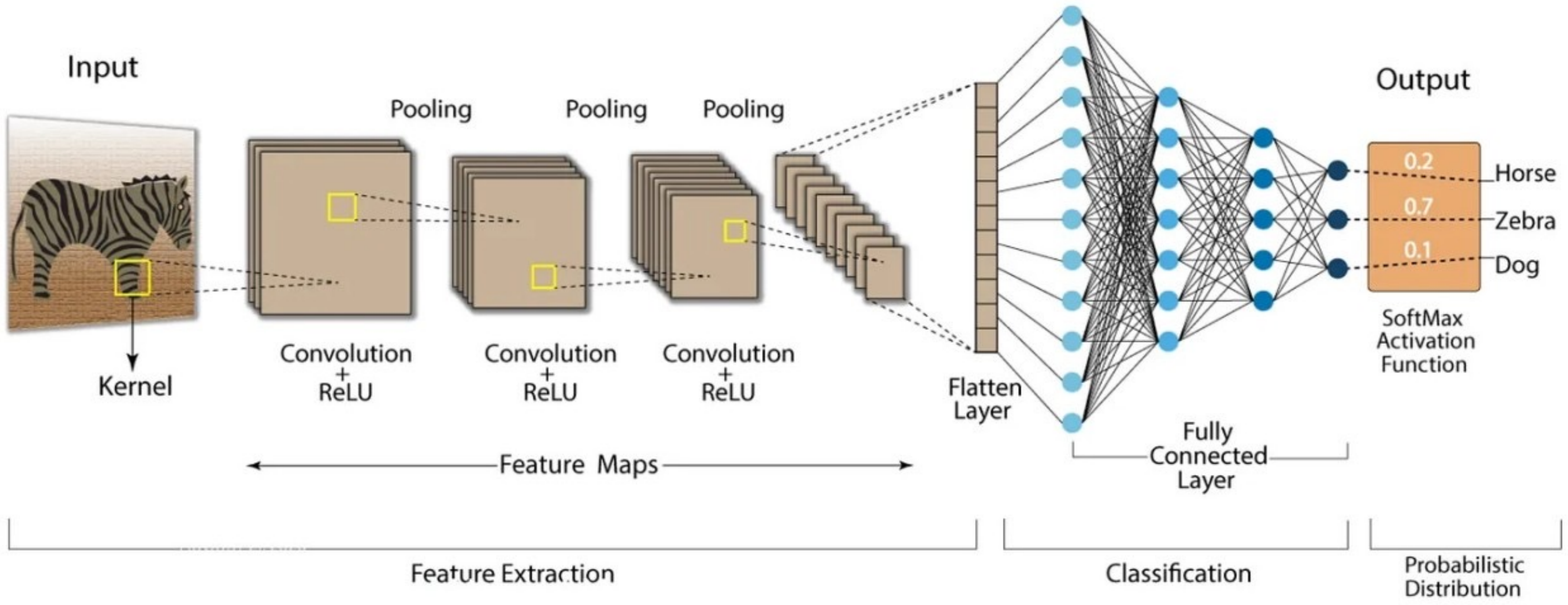


- ✓ One of the key components that their research revealed was that neurons in the visual cortex had what they call a small local receptive field.
- ✓ These neurons are actually only looking at a local, a smaller subsection of the entire image that the person is viewing.
- ✓ These local subsections can then later on overlap and create a larger image and visual field. What's also important to note is that these certain neurons in the visual cortex are only activated when they detect certain things

Review on Deep Learning

Convolutional Neural Network (CNN)

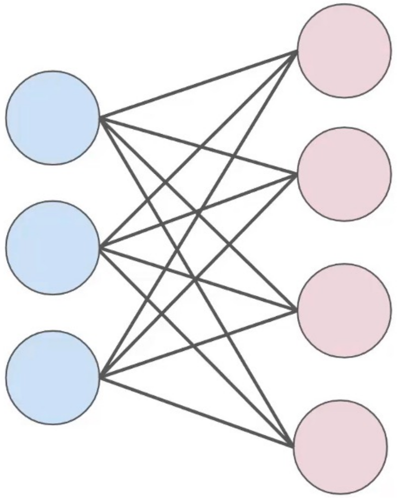
- ✓ This idea inspired an ANN architecture that would become CNN
- ✓ Famously implemented in the 1998 paper by Yann LeCun et al.
- ✓ The LeNet-5 architecture was first used to classify the MNIST dataset.



Review on Deep Learning

Convolutional Neural Network (CNN)

Every neuron in one layer is directly connected to every other neuron in the next layer

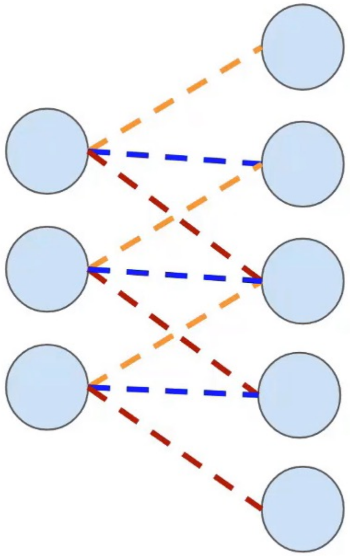


DNN

VS

CNN

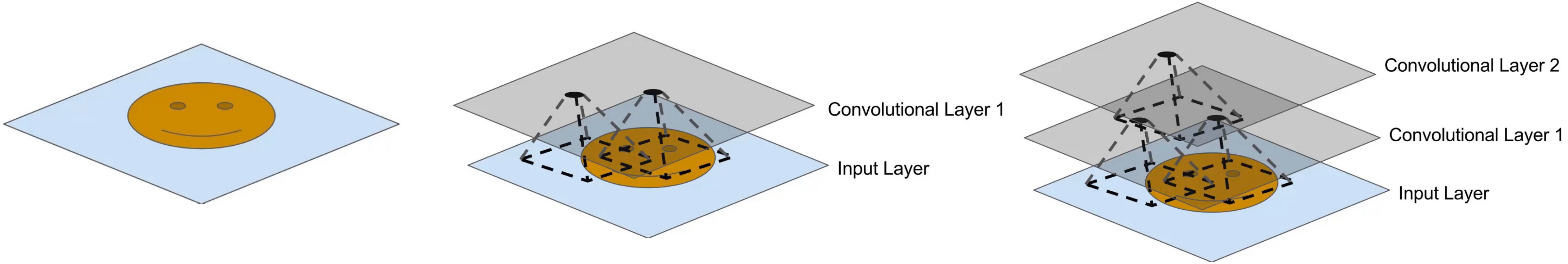
Each unit is connected to a smaller number of nearby unit in the next layer



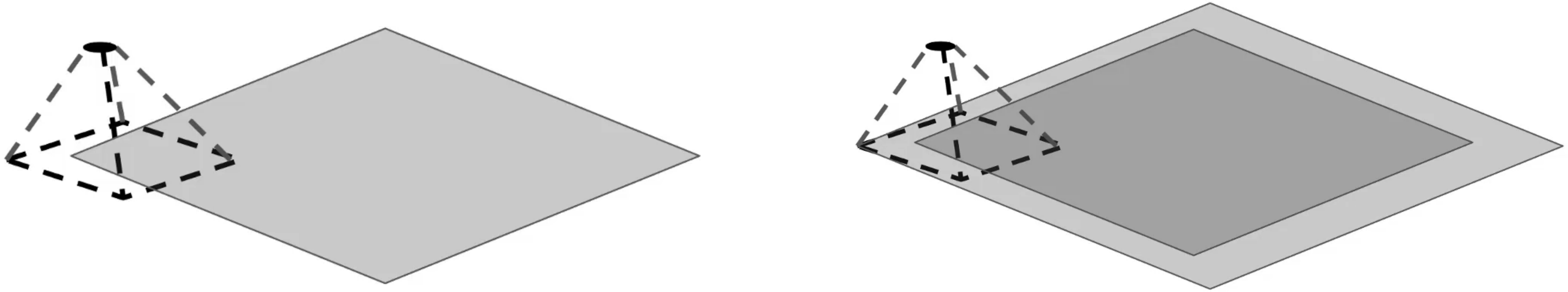
Review on Deep Learning

Convolutional Neural Network (CNN)

Convolution layers are only connected to pixels in their respective fields



Edge of image, no input neuron ! Fix this by adding a “Padding” of zeros around the image



Review on Deep Learning

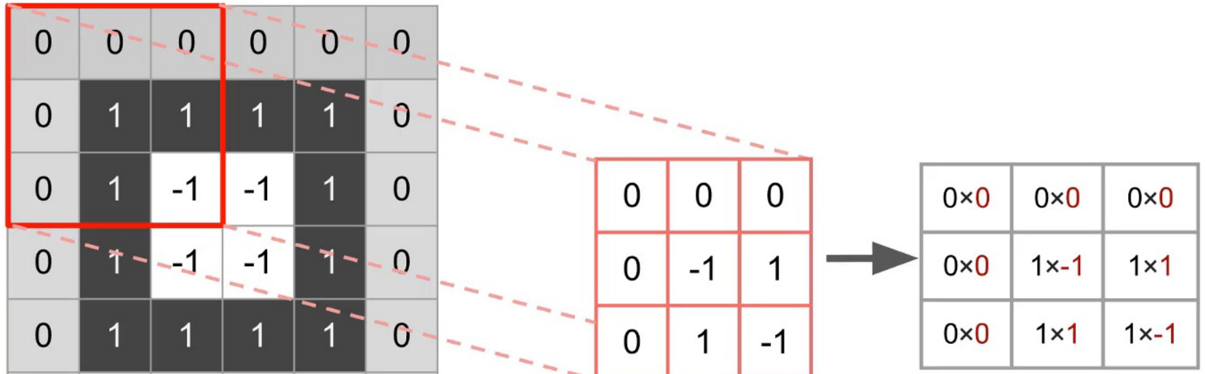
2D CNN (Filter for Convoluted Image)

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

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

0	0	0
0	-1	1
0	1	-1

3 by 3 Filter

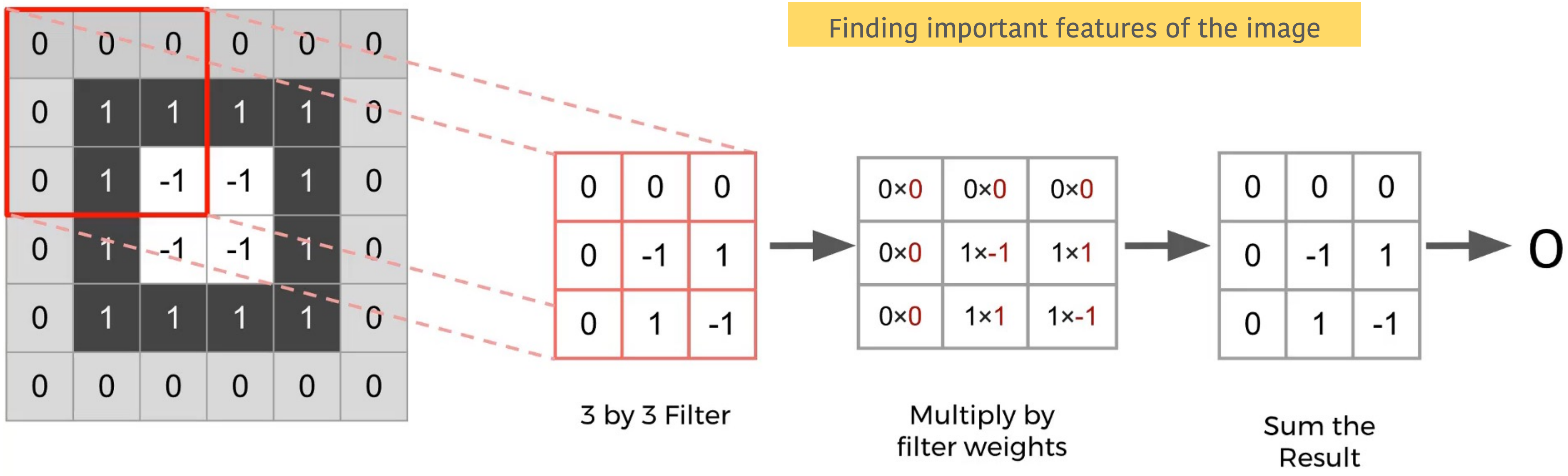


3 by 3 Filter

Multiply by filter weights

Review on Deep Learning

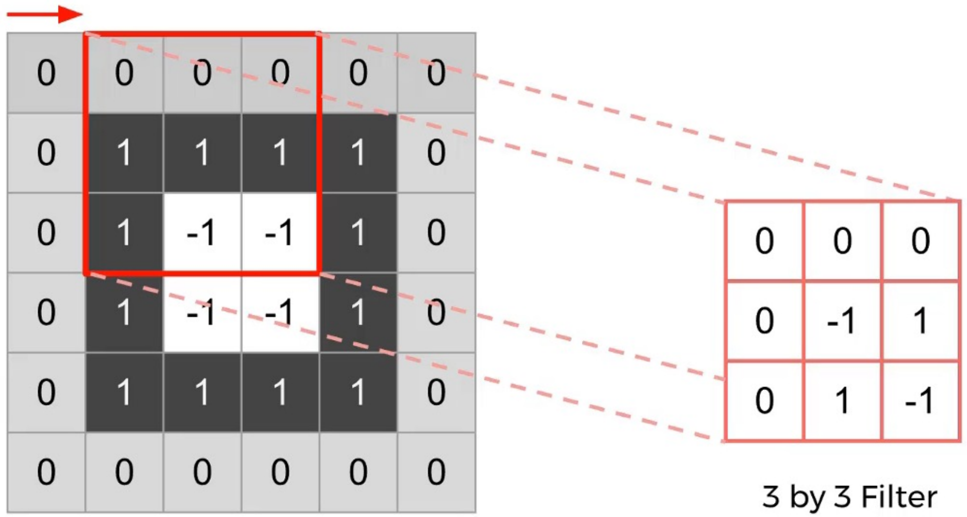
2D CNN (Filter for Convoluted Image)



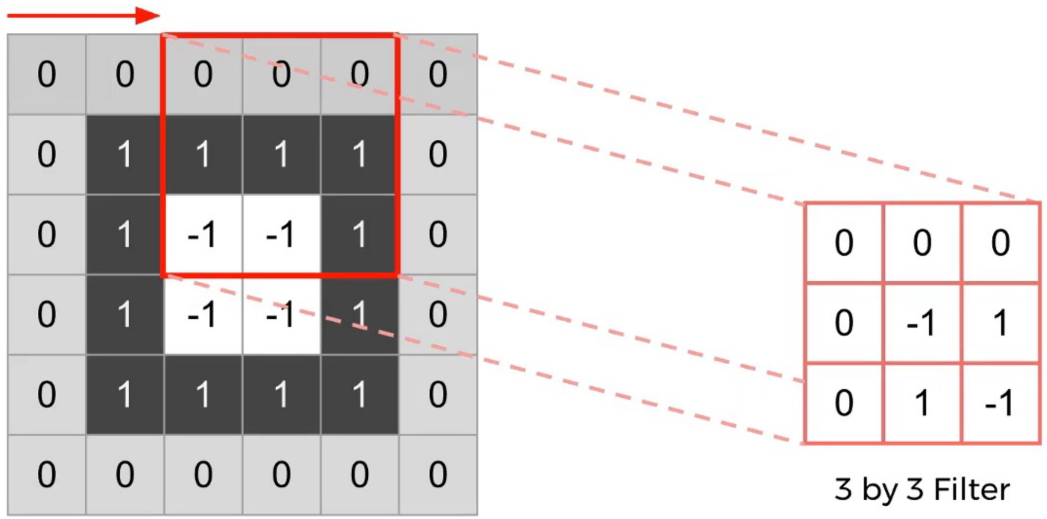
Review on Deep Learning

2D CNN (Stride: How fast you move across your image)

Stride Distance by 1 Pixel



Stride Distance by 2 Pixel



Review on Deep Learning

2D CNN (Filter)

https://setosa.io/ev/image-kernels/

To see how they work, let's start by inspecting a black and white image. The matrix on the left contains numbers, between 0 and 255, which each correspond to the brightness of one pixel in a picture of a face. The large, granulated picture has been blown up to make it easier to see; the last image is the "real" size.

Let's walk through applying the following 3x3 **sharpen** kernel to the image of a face from above.

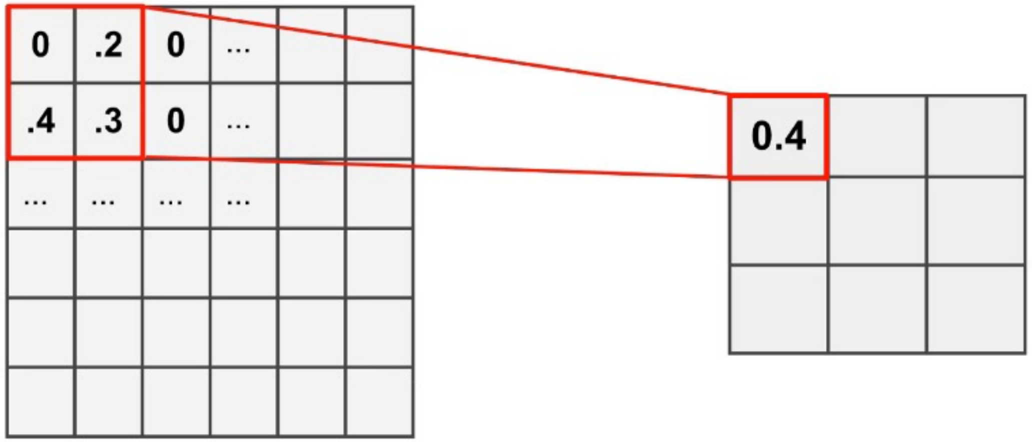
sharpen

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

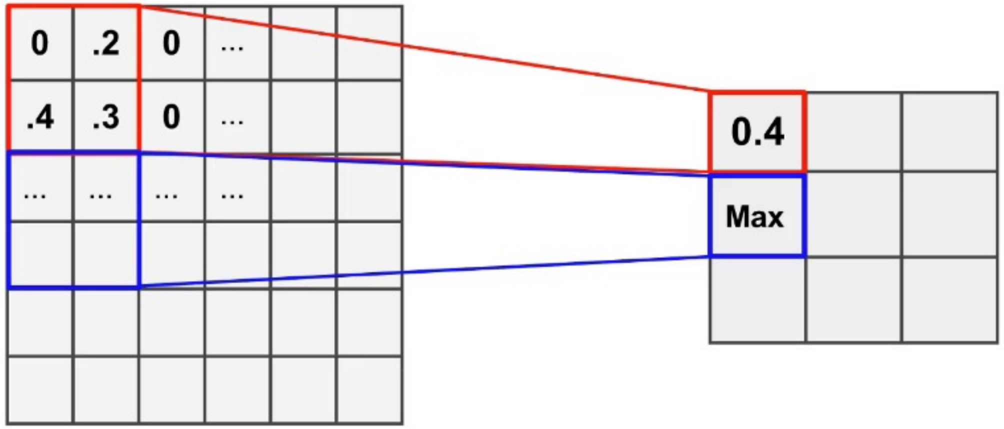
Review on Deep Learning

2D CNN (Subsampling / Pooling)

Find the max value of Kernel

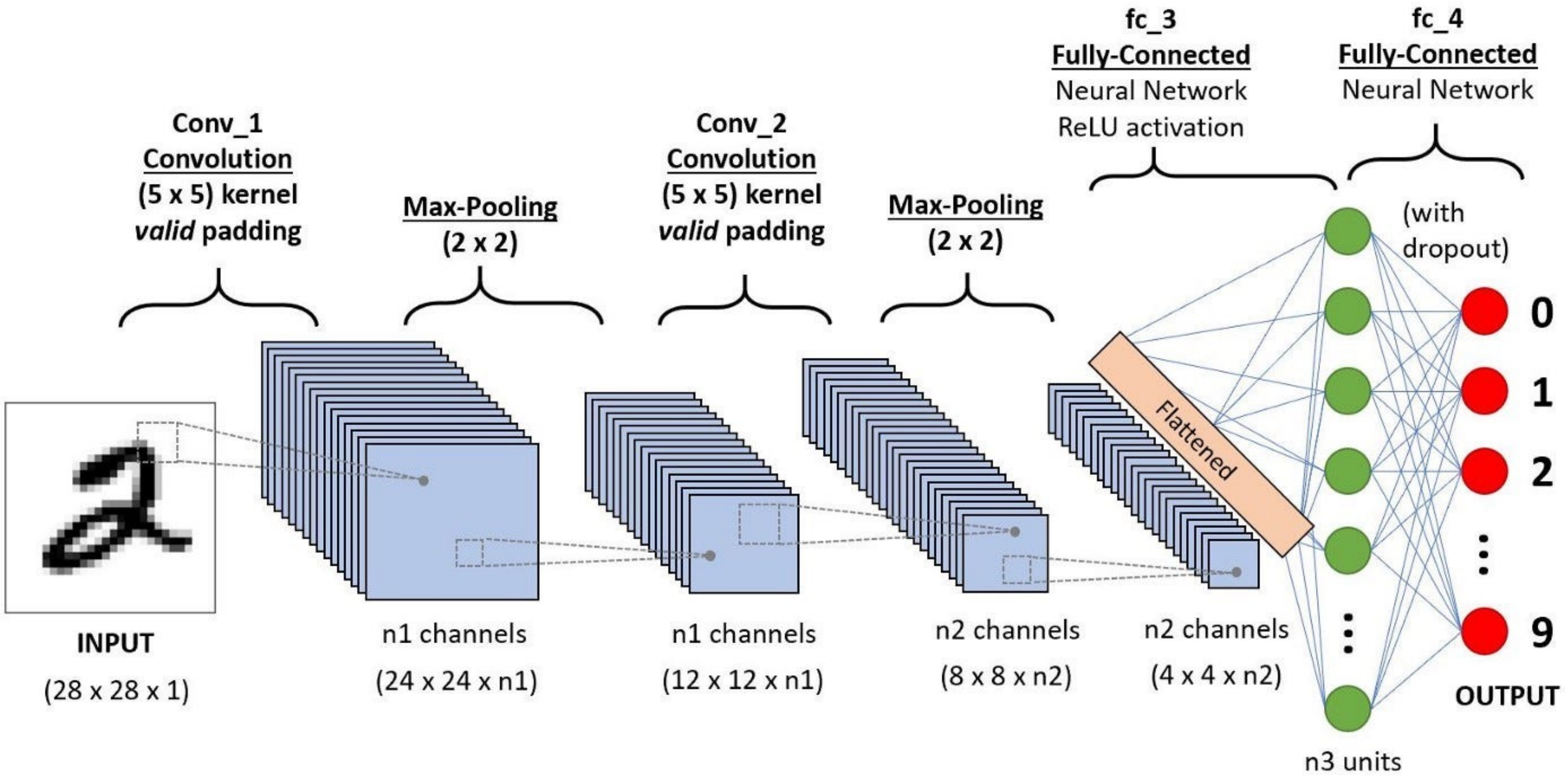


Stride Distance by 2 Pixel



Review on Deep Learning

2D CNN



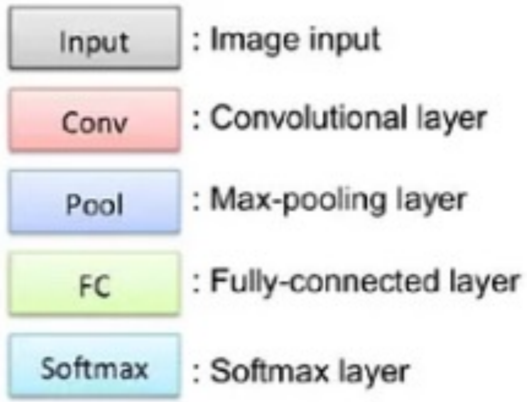
Review on Deep Learning

Convolutional Neural Network (CNN) Architecture

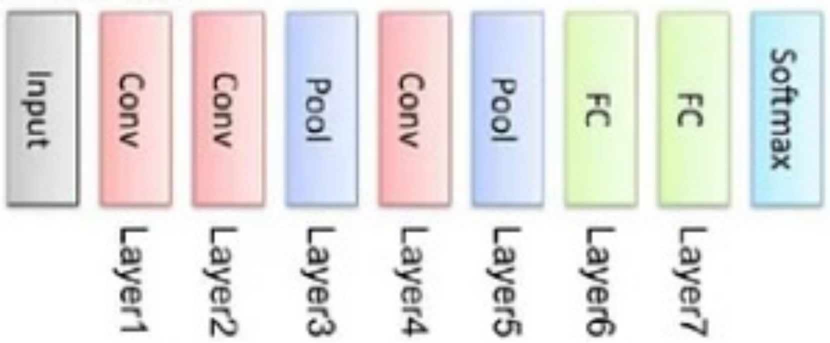
- ✓ LeNet-5 : Yann LeCun
- ✓ AlexNet : Alex Krizhenvsky et al.
- ✓ GoogLeNet : Szegedy at Google Research
- ✓ ResNet : Kaiming He et al.
- ✓ VGGNet : Visual Geometry Group at Oxford University (VGG-16 | VGG-19)

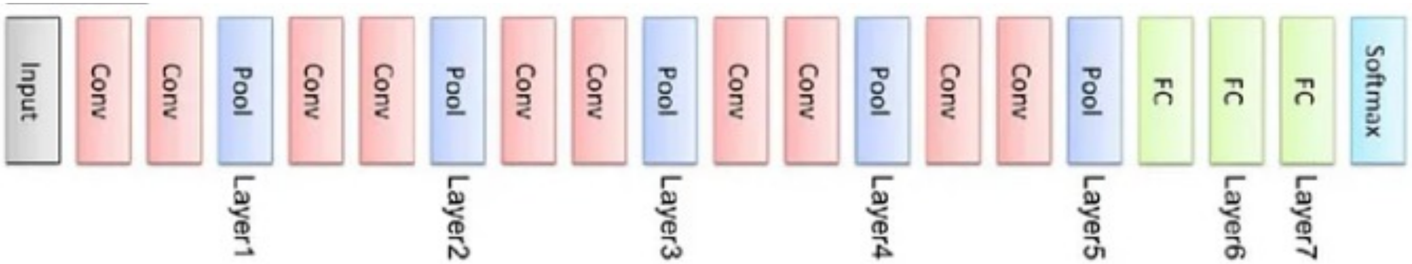
Review on Deep Learning

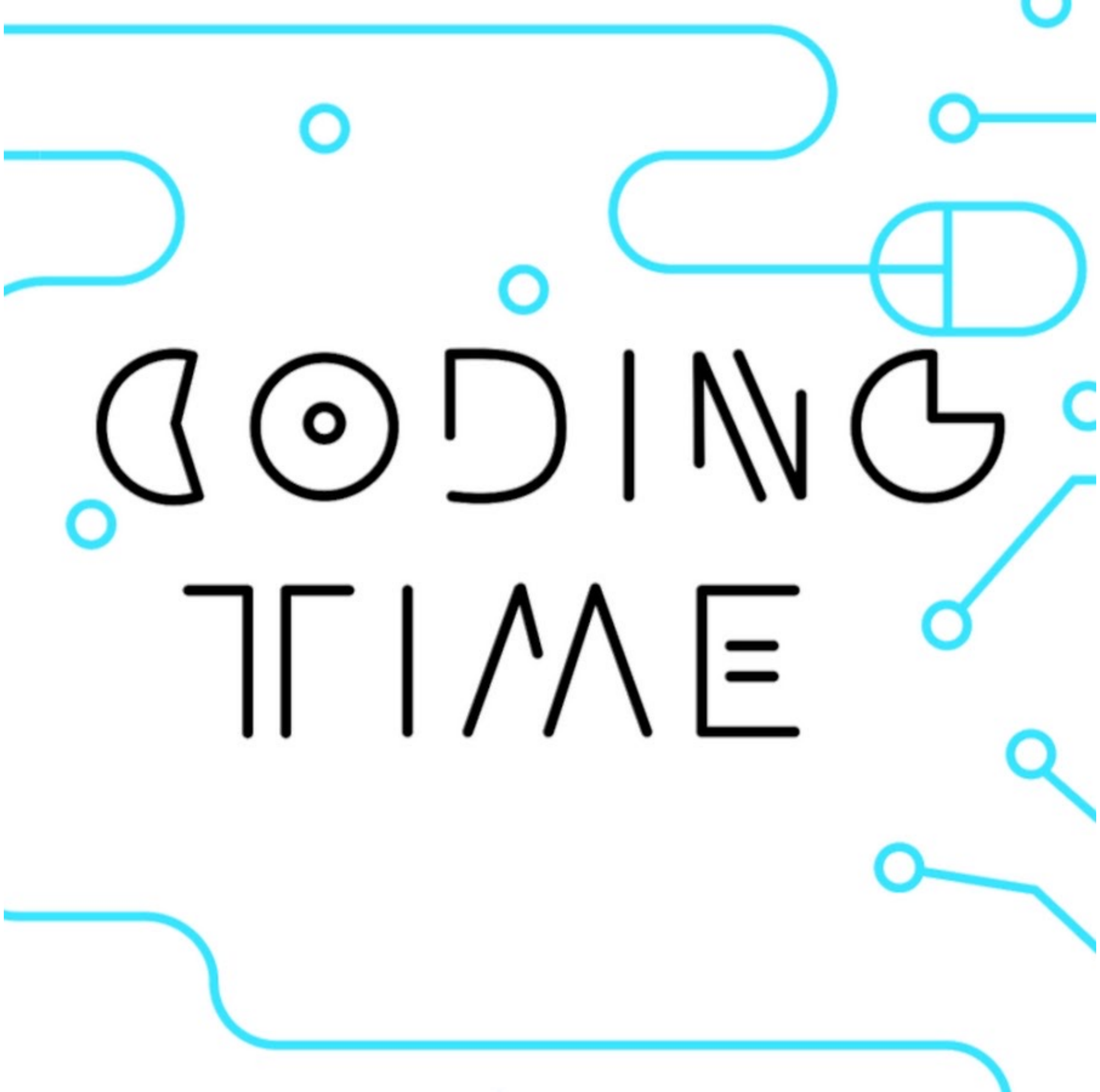
Convolutional Neural Network (CNN) Architecture



AlexNet VS VGGNet






CODING
TIME