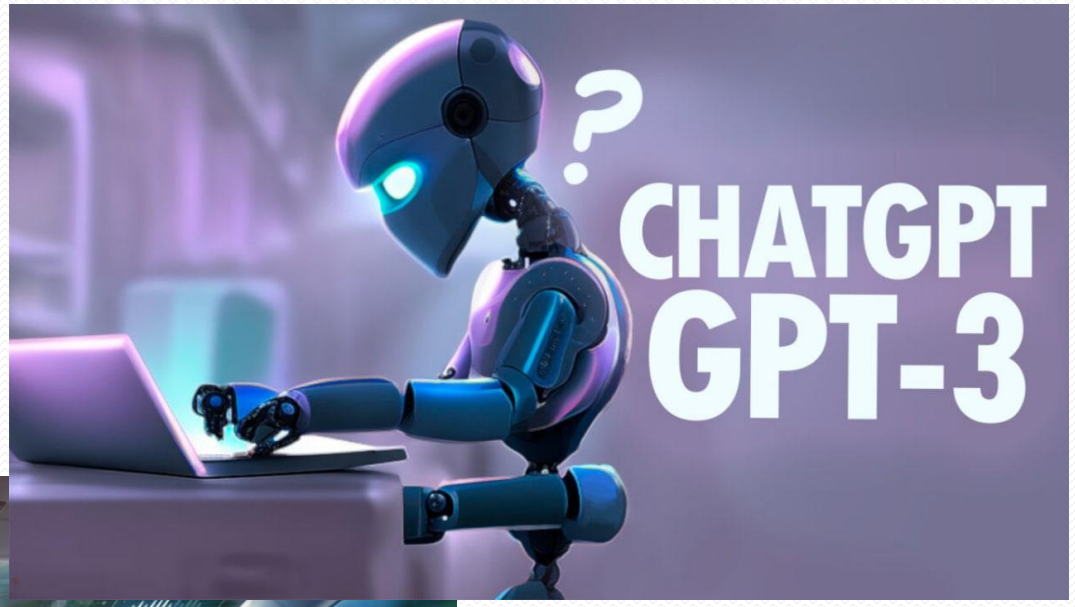
AI TRAINING SERIES

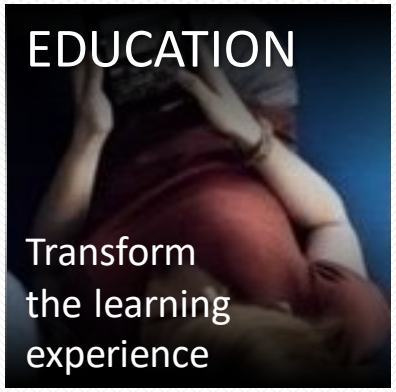
Perceptron & ANN

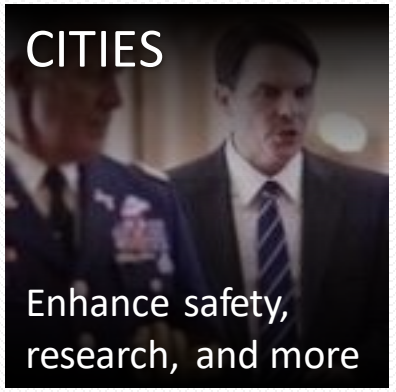
AI/DL introduction & history

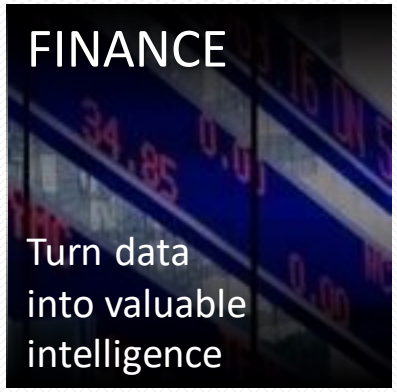
AI/DL Momentum

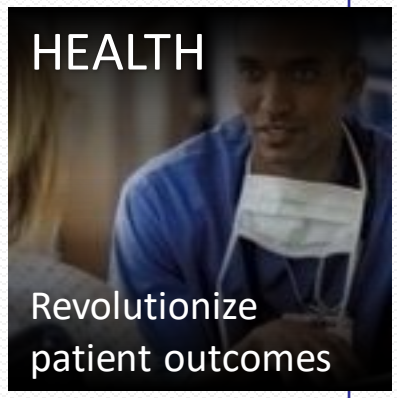


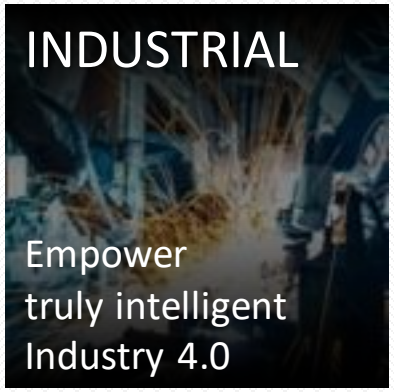
AI is changing every market

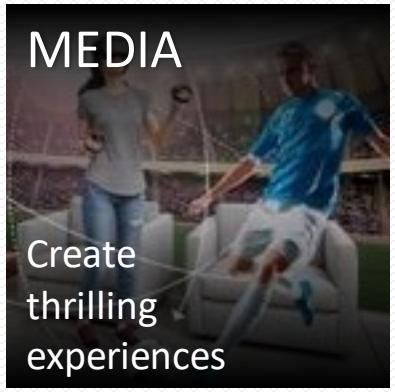


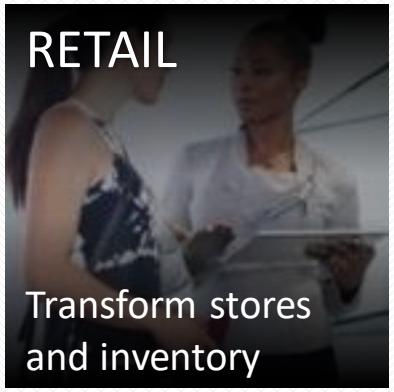


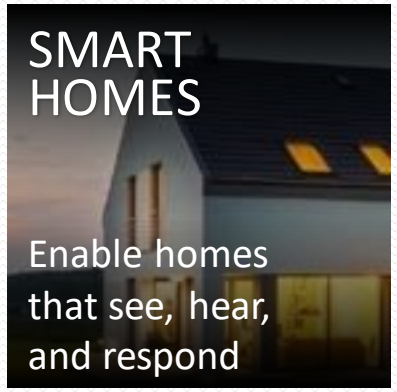


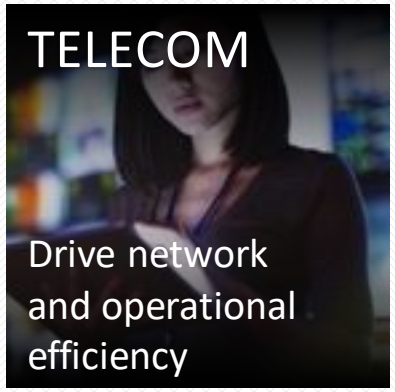




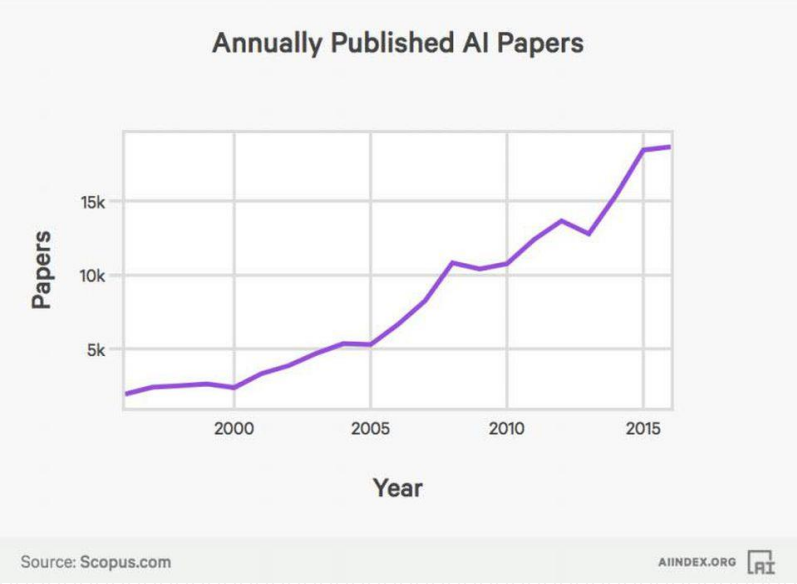


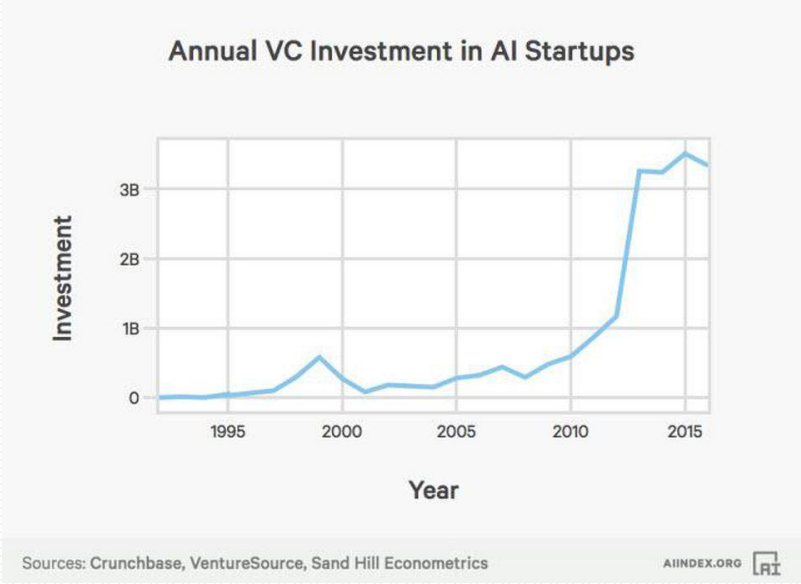


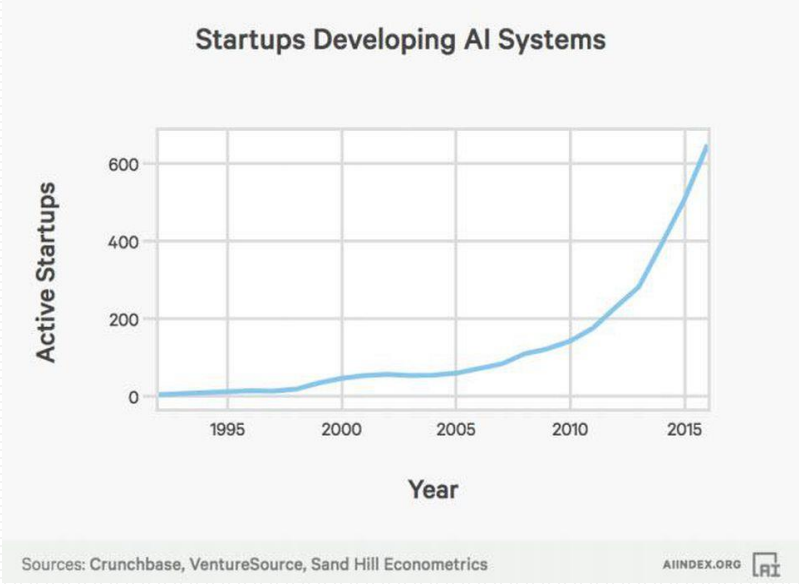


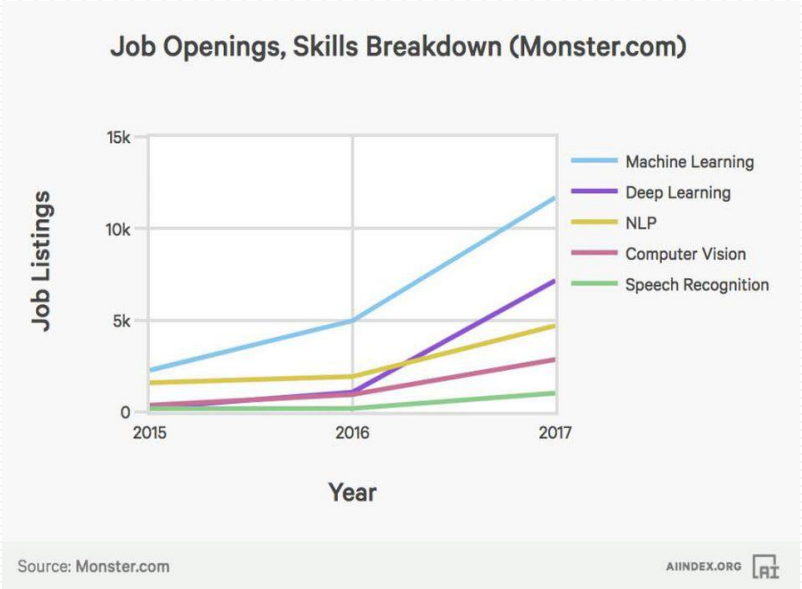


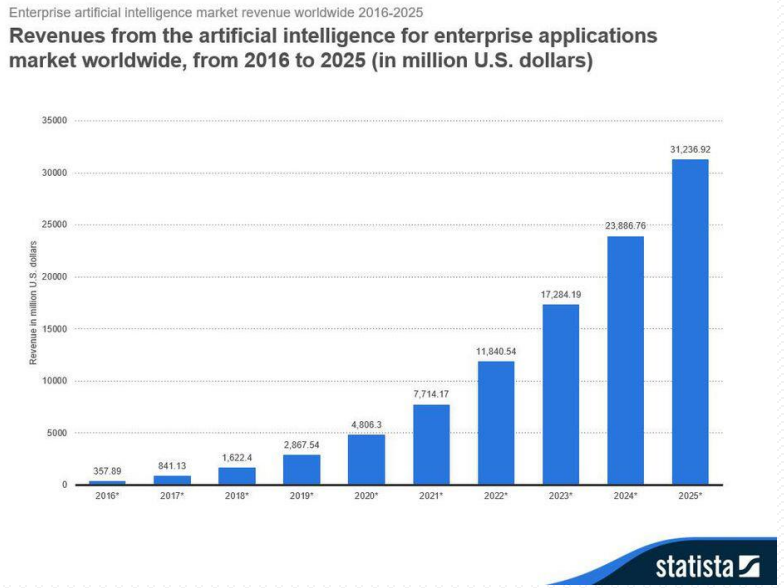
AI GROWTH







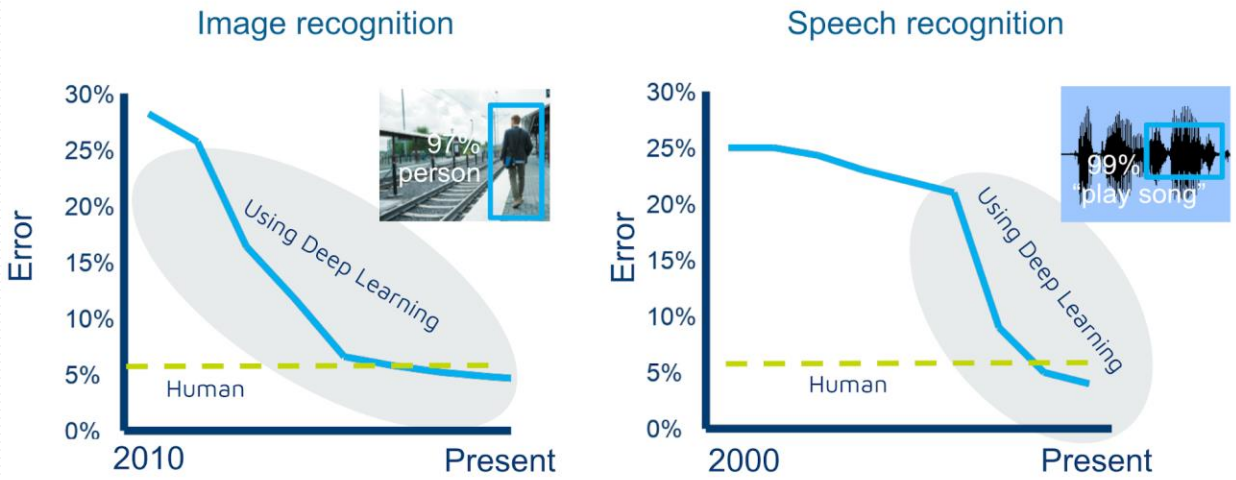




source

WHY NOW?

Bigger Datasets	Better Hardware	Smarter Algorithms
IMAGENET 10M labeled images YouTube 8M categorized videos	**Moore's Law** Cost / GB in 1995: $1000 Cost / GB in 2020: $0.02	**Recurrent Neural Networks** **Convolutional Neural Nets** **LSTM**



AI TRENDS IN MARKET

A.I. Here, There, Everywhere!

RETAIL

Personalized Shopping Experience

Inventory Management

In-store Surveillance

Supply Chain Optimization

INDUSTRIAL

Predictive Maintenance

Machine Vision

Robotics

Improved Quality of Service

Operational Efficiency

TRANSPORT

Enhanced In-cabin Experience

Aircraft Maintenance and Safety

Real-time Tracking of Packages/ Fleet

Autonomous Driving

SMART CITIES

Safety & Security

Resident Engagement

Traffic Management

Object Detection

HEALTHCARE

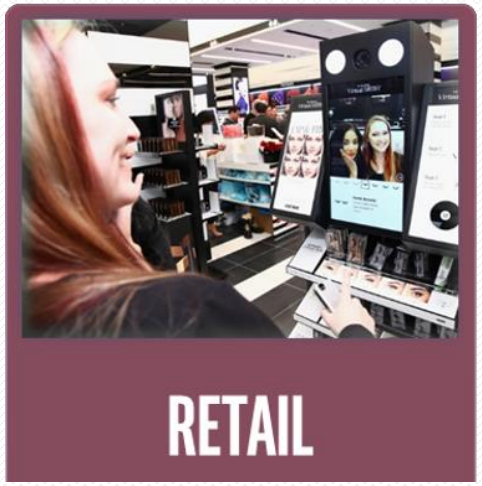
Enhanced Diagnostics

Drug Discovery

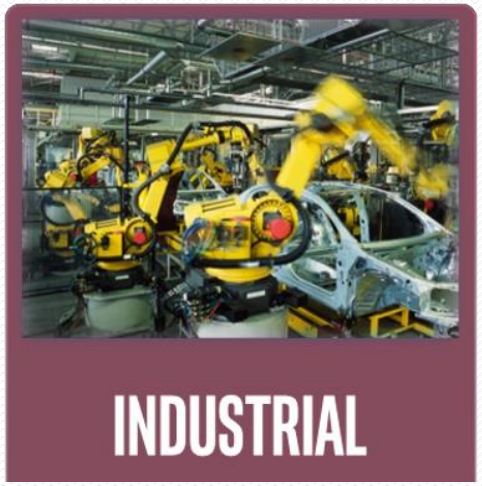
Customized Patient Care

Sensory Aids

AI Products



RETAIL



INDUSTRIAL

TRANSPORT

SMART CITIES

HEALTHCARE

BEAR ROBOTICS

LG Energy Solution

HYUNDAI

SAMSUNG

SAMSUNG

SAMSUNG MEDISON

emart

Hanwha

현대중공업

NAVER

VUNO

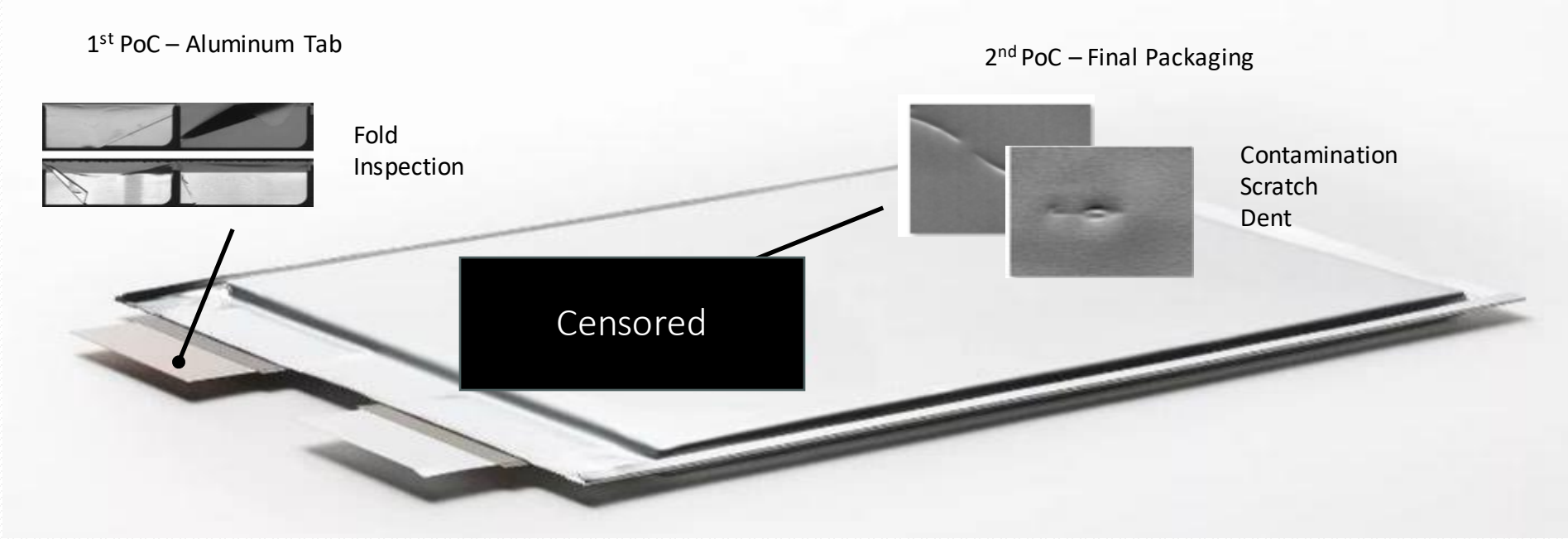
View the Invisible, Know the Unknown

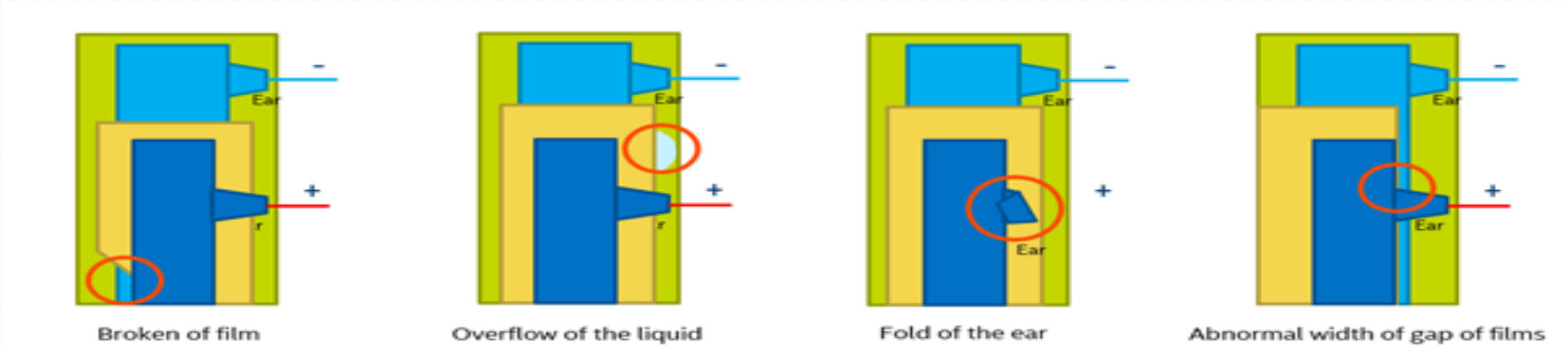
DELL

LOTTE

신한은행

불량검출


1st PoC – Aluminum Tab
2nd PoC – Final Packaging
Fold
Inspection
Contamination
Scratch
Dent
Censored


Ear
Broken of film
Overflow of the liquid
Fold of the ear
Abnormal width of gap of films

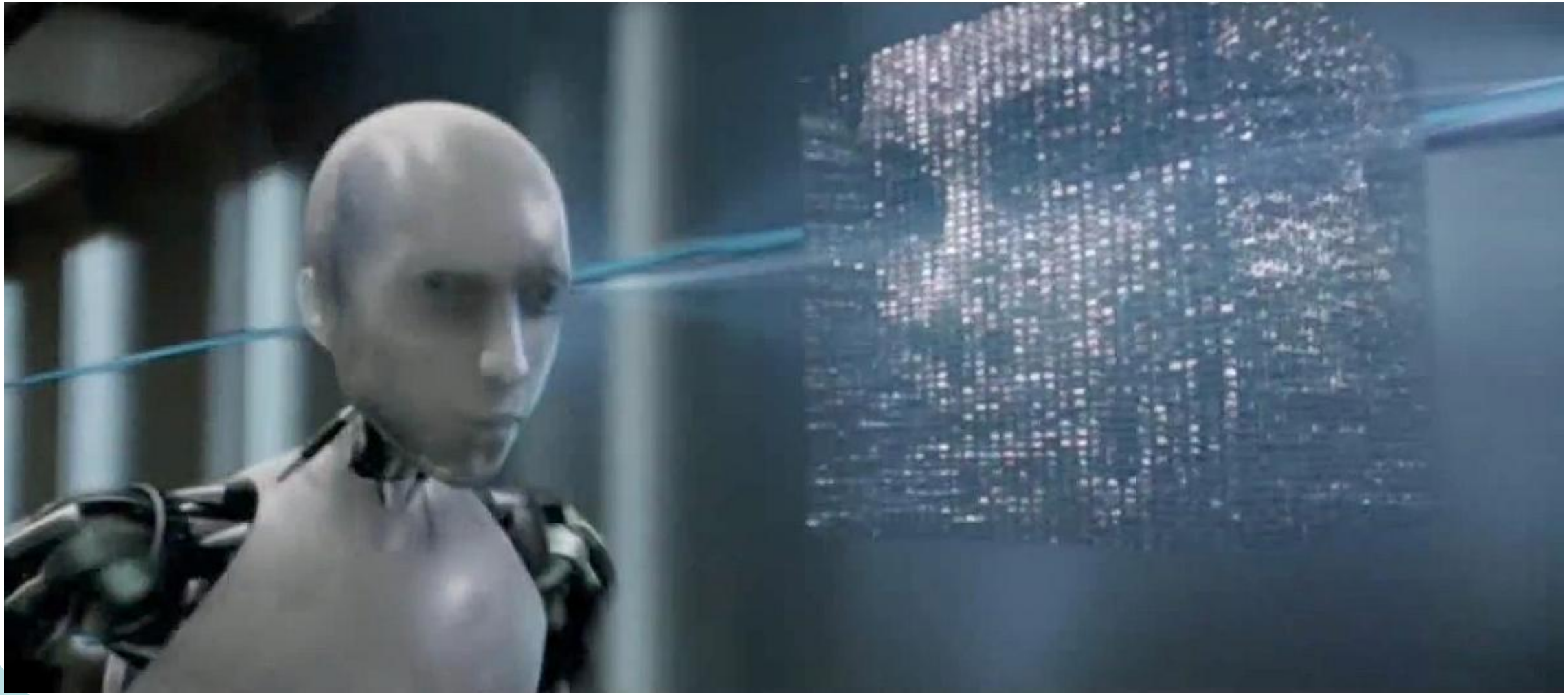
Session Break

AI TRAINING SERIES

AI

What is Artificial Intelligence? How Does AI Work?

AI 란 무엇일까?

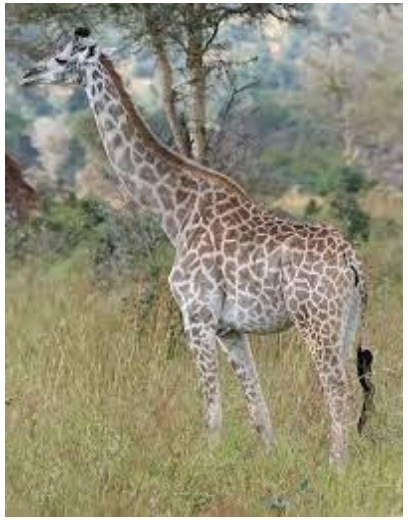


자연현상에서 출발

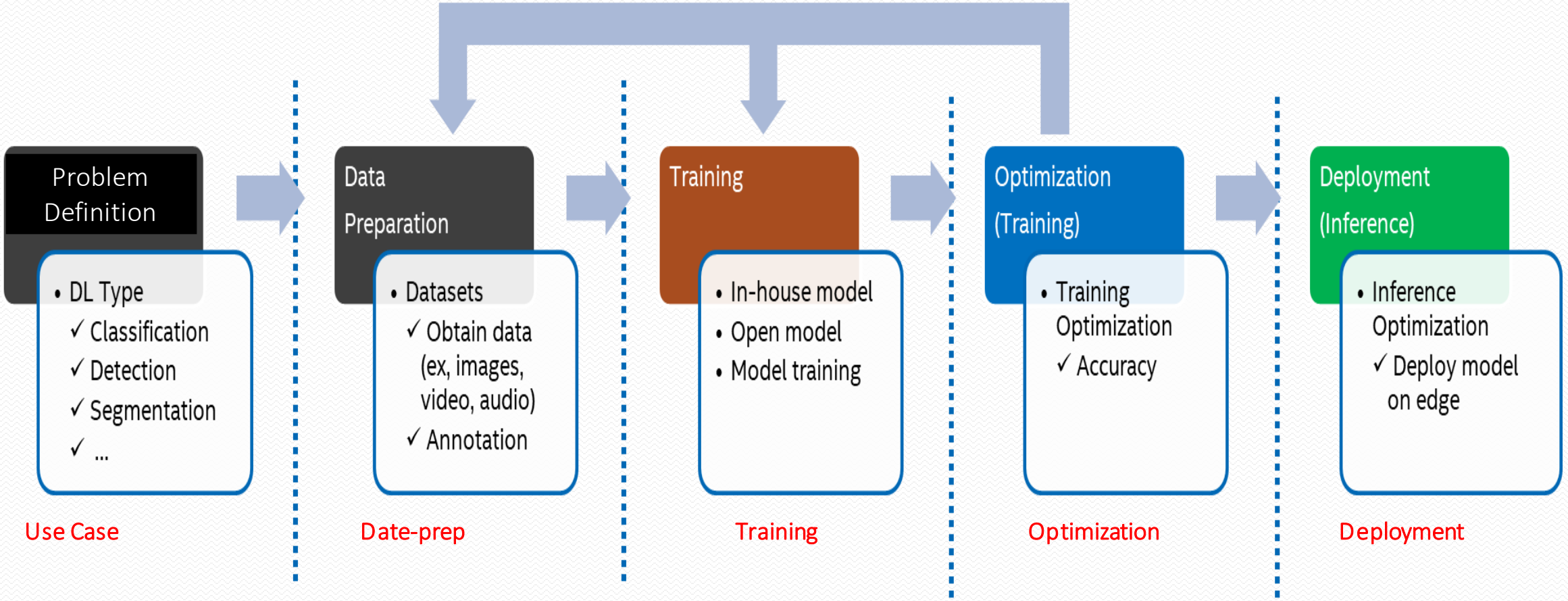
Sidewinder

Ultrasound

우리는 어떻게 생각했나

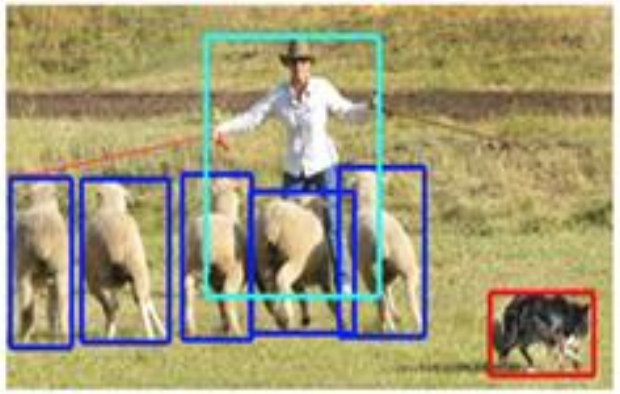


Deep Learning Workflow (생각하게 만들기)



1) Problem Definition

CLASSIFICATION



DETECTION

SEGMENTATION

NEURAL STYLE TRANSFER

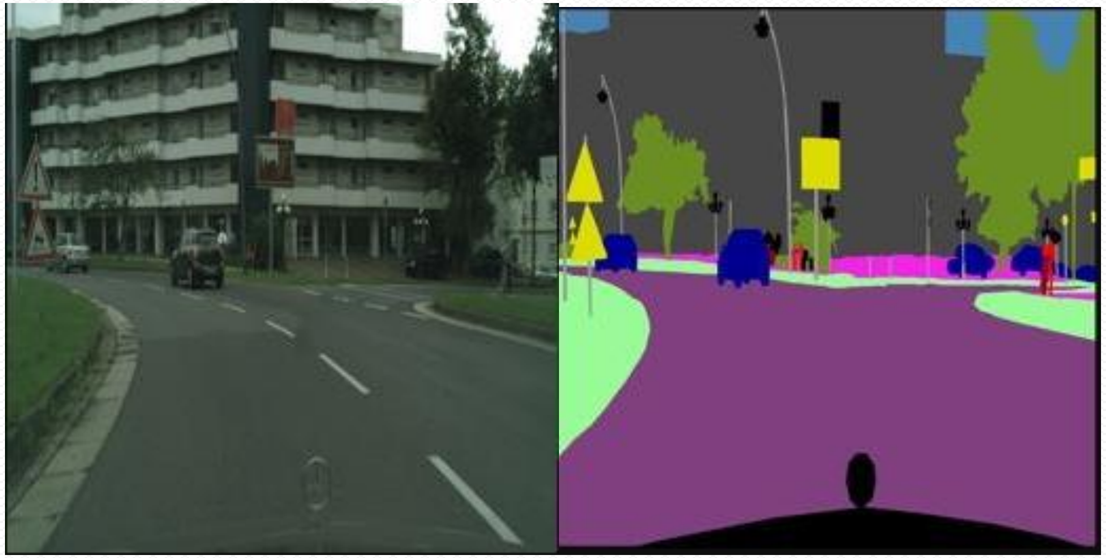
IMAGE CAPTION

COLORIZATION

2) Dataset preparation/annotation

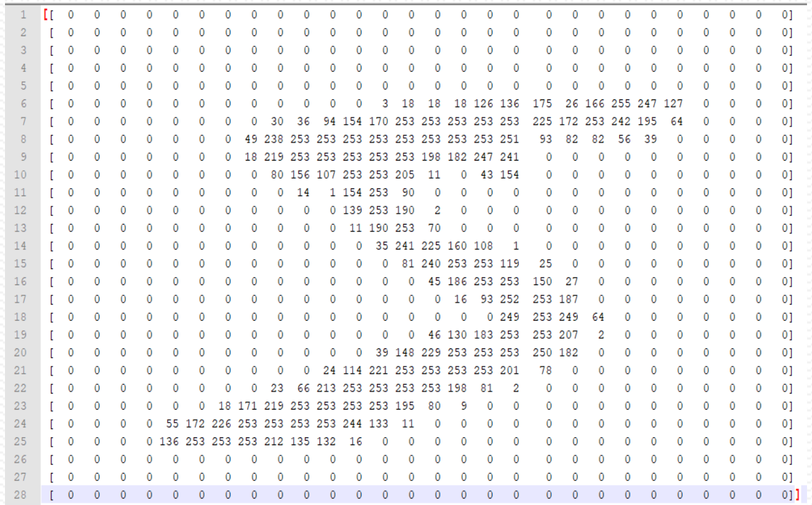
- Cityscape/Camvid

  Semantic Segmentation

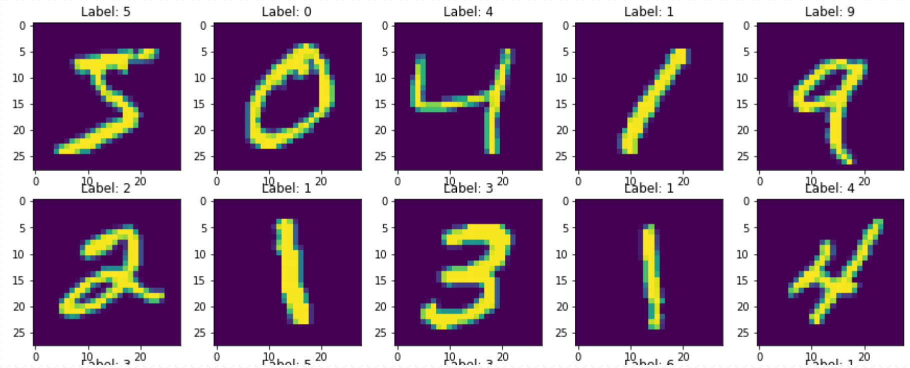


- MNIST

  Hand-written digit



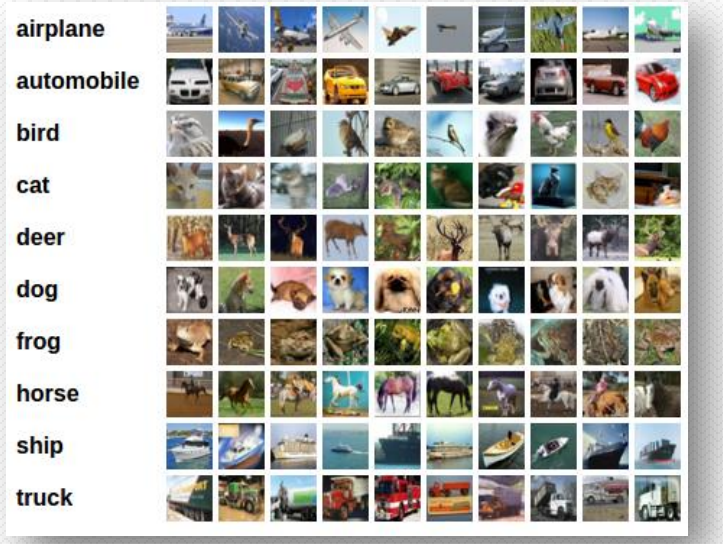
```
[[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 3 18 18 18 126 136 175 26 166 255 247 127 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 30 36 94 154 170 253 253 253 253 253 225 172 253 242 195 64 0 0 0 0]
 [ 0 0 0 0 0 0 0 49 238 253 253 253 253 253 253 253 253 251 93 82 82 56 39 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 18 219 253 253 253 253 253 198 182 247 241 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 80 156 107 253 253 205 11 0 43 154 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 14 1 154 253 90 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 139 253 190 2 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 11 190 253 70 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 35 241 225 160 108 1 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 81 240 253 253 119 25 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 45 186 253 253 150 27 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 16 93 252 253 187 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 249 253 249 64 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 46 130 183 253 253 207 2 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 39 148 229 253 253 253 250 182 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 24 114 221 253 253 253 253 201 78 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 23 66 213 253 253 253 253 198 81 2 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 18 171 219 253 253 253 253 195 80 9 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 55 172 226 253 253 253 253 244 133 11 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 136 253 253 253 212 135 132 16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
```



2) Dataset preparation/annotation – cont'd

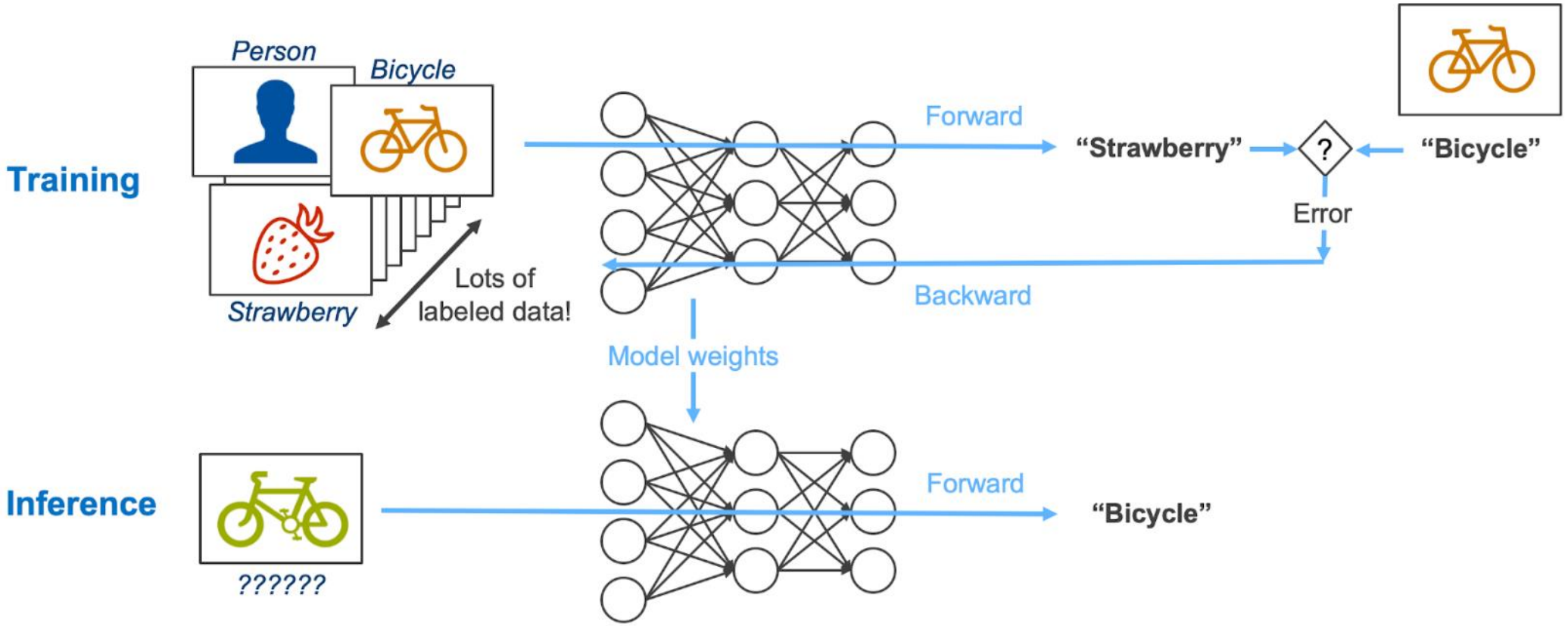
- MNIST
  - Label : digit 0-9
  - Consists of 60,000 training image & 10,000 test image 28x28 gray scale image

- CIFAR-10 dataset
  - Label : 10 classes
  - Consists of 60,000 (32x32) color images, with 6000 images per class.

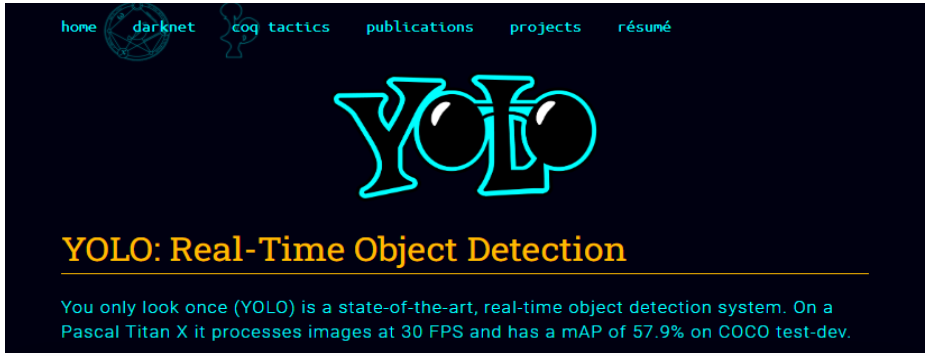


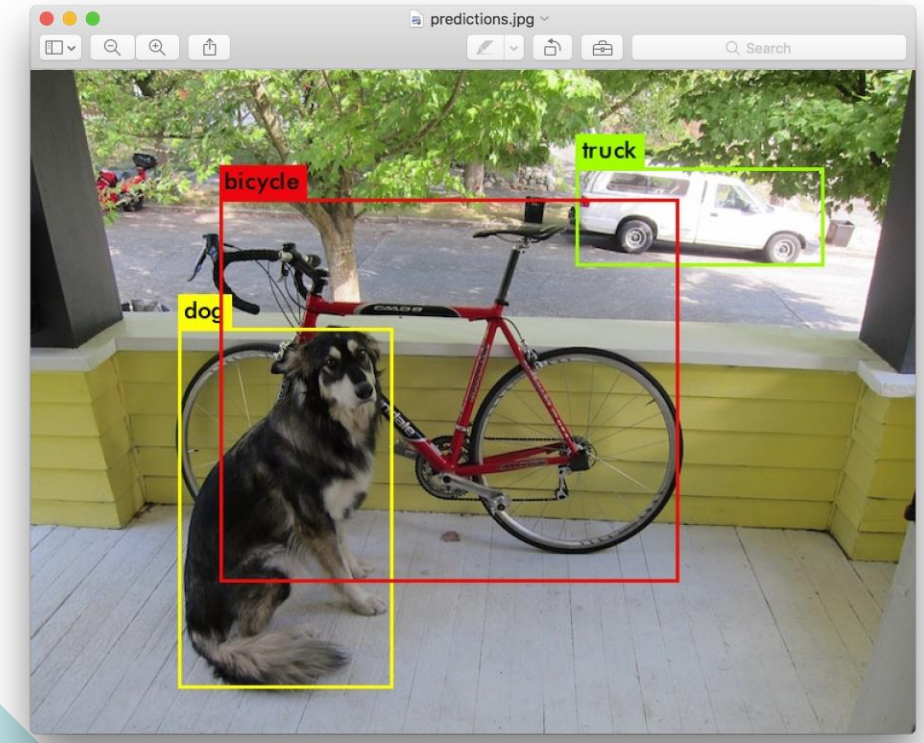
- IMAGENET
  - Label : 1000 classes
  - Consists of 800,000 (256x256 or 224x224) color Images, with 800 images per class.

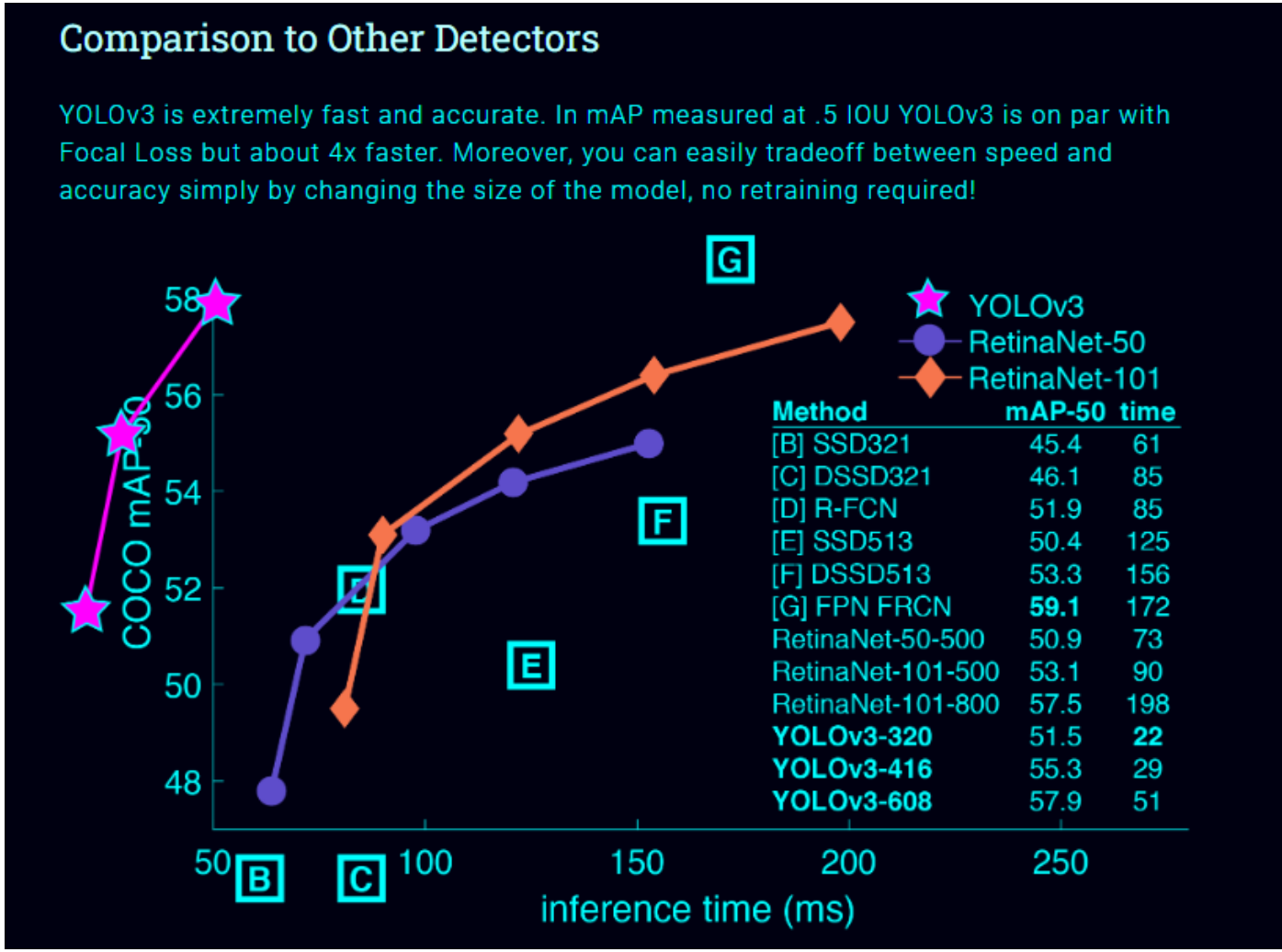
3) Model selection & Training



Deep Learning Model (YOLO)







4) Optimization

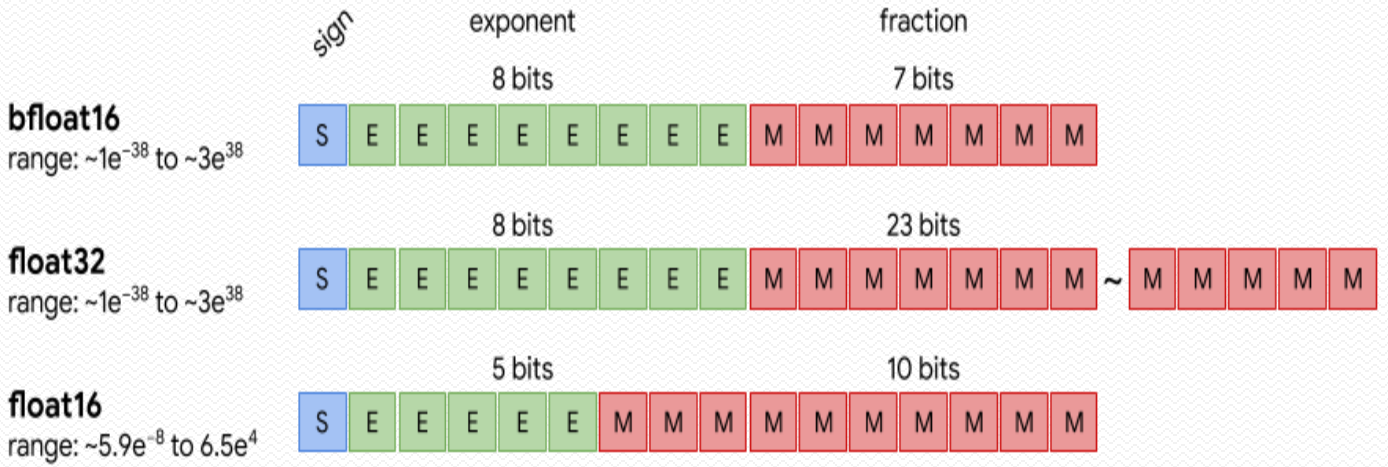
FP16 VS. FP32

FP16 improves speed (TFLOPS) and performance

FP16 reduces memory usage of a neural network

FP16 data transfers are faster than FP32

But converting from/to 32-bit floating-point can reduce accuracy



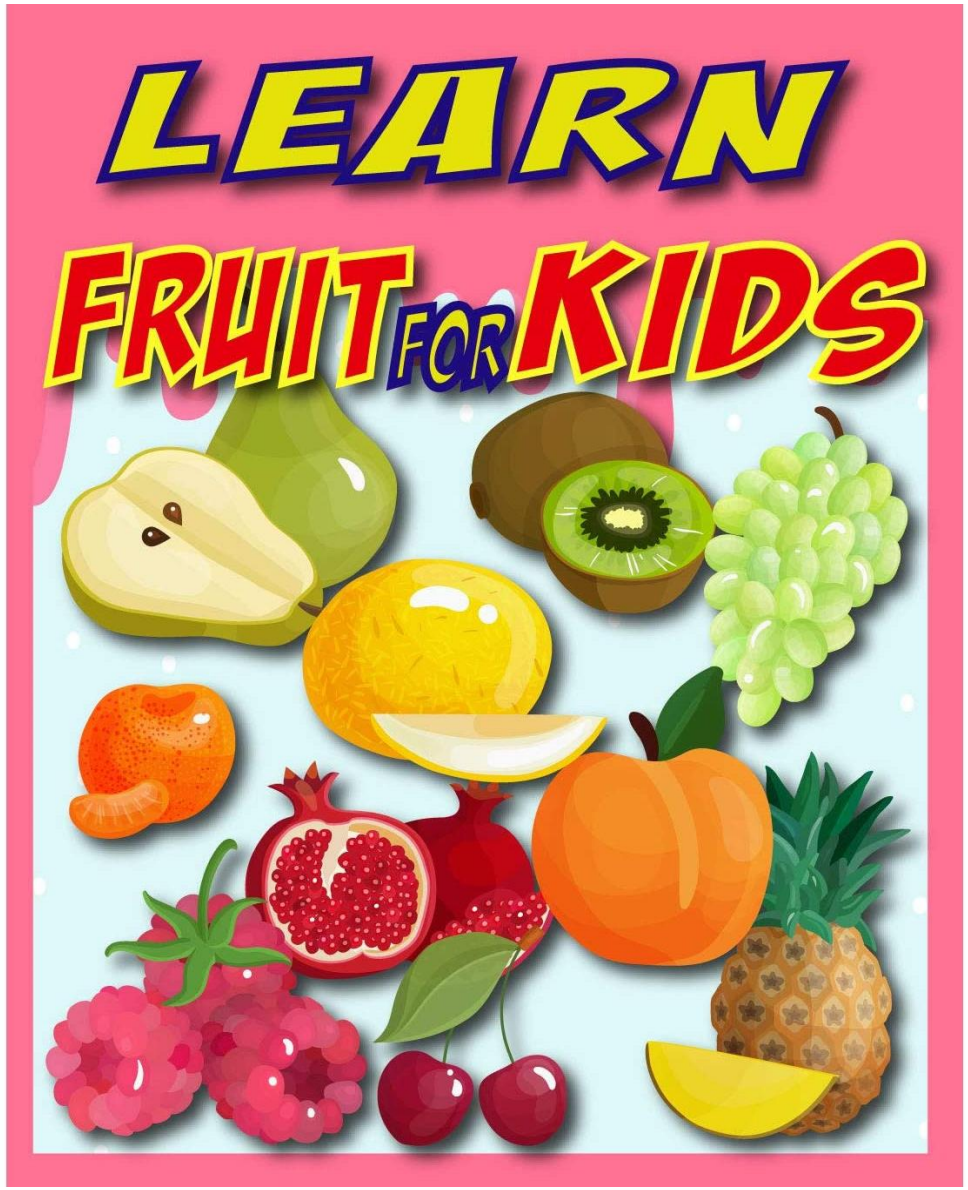
INT8 QUANTIZATION

Accelerate the performance of certain models

However, this comes at the cost of a small reduction in accuracy

사과!

??



Apple!

Pear!

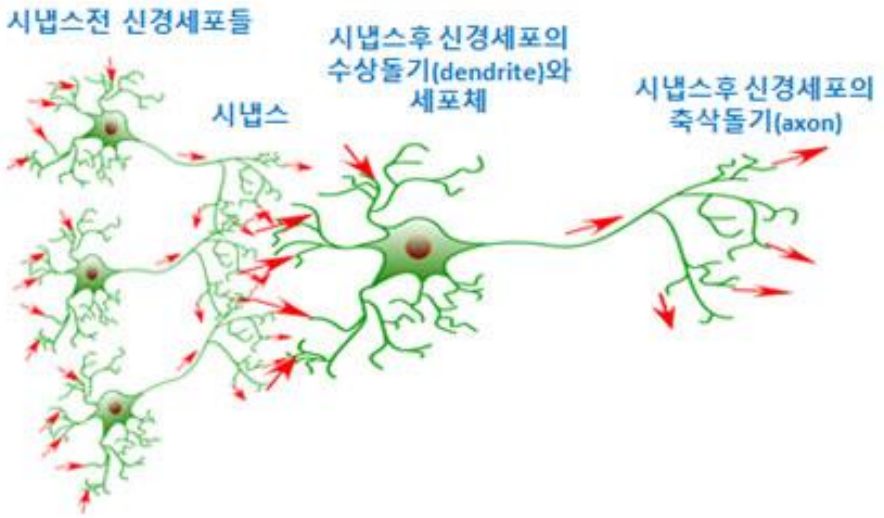
AI TRAINING SERIES

History

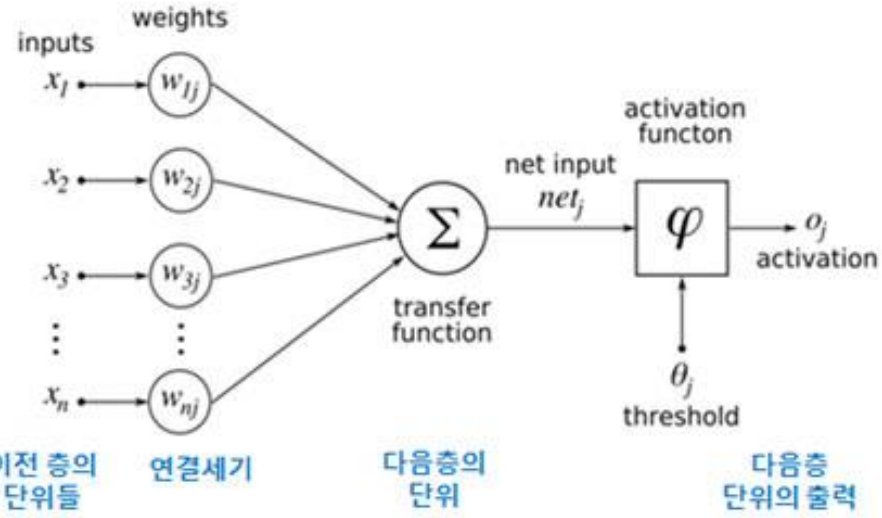
What is Artificial Intelligence? How Does AI Work?

신경망

뇌 속 신경망



인공 신경망



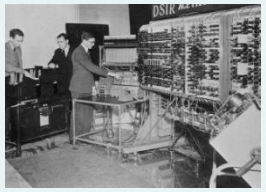
http://scienceon.hani.co.kr/397536

HISTORY OF A.I.

1950

Alan Turing

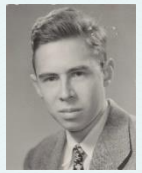
Turing Test

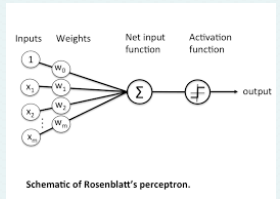


1958

Frank Rosenblatt

Perceptron



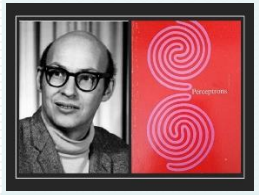


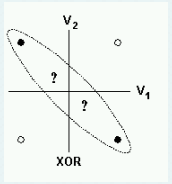
Schematic of Rosenblatt's perceptron.

1969

Marvin Minsky

A book of “Perceptrons”



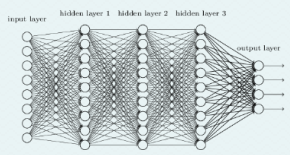


AI Winter

1986

David Rumelhart

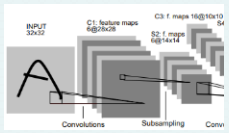
Multi Layer Perceptron (MLP)



1998

Yann Lecun

LeNet-5

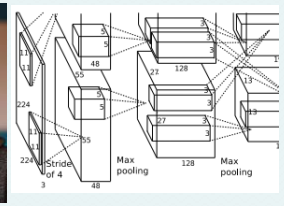


AI Winter

2012

Alex Krizhevsky

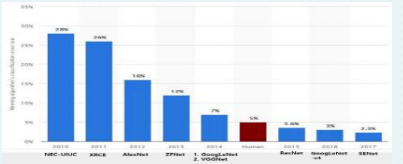
AlexNet



2015

ResNet

Accuracy is higher than human



2016

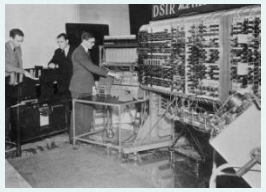
Google Brain

AlphaGo

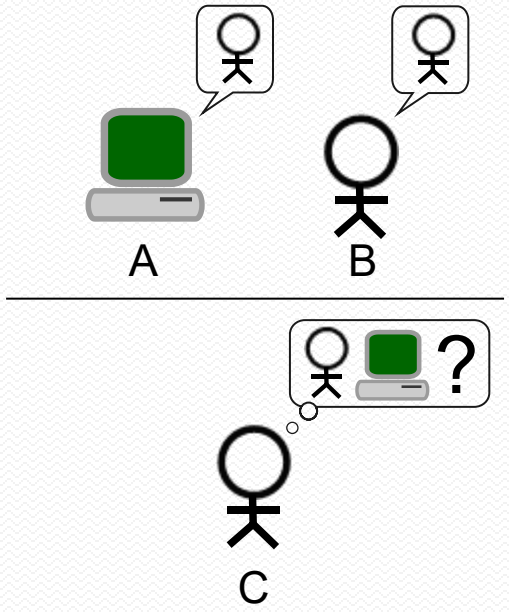
HISTORY OF A.I.

Alan Turing

Turing Test



1950



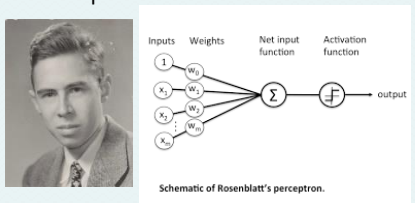
- 튜링 테스트(Turing test) 의 의미
  - 기계가 생각하는 것이 가능한가?
  - Computing Machinery and Intelligence 논문
- 각각 분리된 채, 하나는 인간, 다른 하나는 기계와 대화를 하는 Test
- 5분동안 기계와 대화한다는 것을 인지하지 못하면 성공
- 대화에서 질문에 올바른 대답을 하는것 보단 얼마나 인간같이 대화할 수 있느냐로 판단

HISTORY OF A.I.

1958

Frank Rosenblatt

Perceptron



Schematic of Rosenblatt's perceptron.

- 프랭크 로젠블렛(신경 생물 과학자, 심리학전공)
- **이미지 인식 성공** !! – 많은 투자와 관심
- Feed Forward Network
- 발표 논문
  - *THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN*

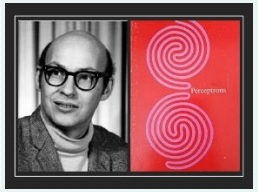
If we are eventually to understand the capability of higher organisms for perceptual recognition, generalization, recall, and thinking, we must first have answers to three fundamental question.

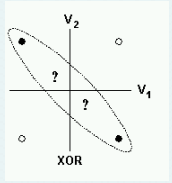
1. *How is information about the physical world sensed, or detected, by the biological system?*
2. *In what form is information stored, or remembered?*
3. *How does information contained in storage, or in memory, influence recognition and behavior?*

HISTORY OF A.I.

Marvin Minsky

A book of “Perceptrons”

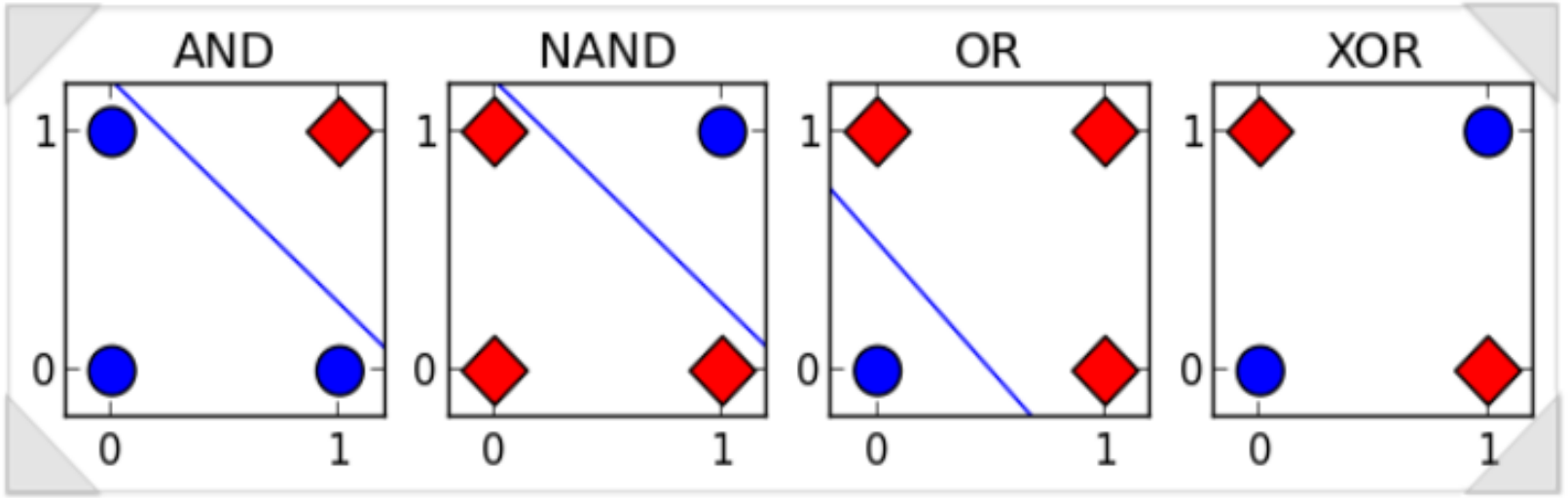




1969

마빈 민스키

- *Perceptrons* – Minsky & Papert 출간, 퍼셉트론 모델에 대해 철저한 분석 및 그 한계에 대해서도 논리정연하게 분석

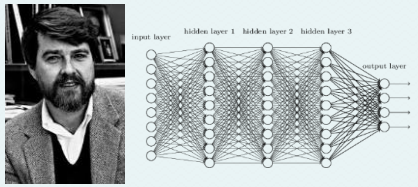


HISTORY OF A.I.

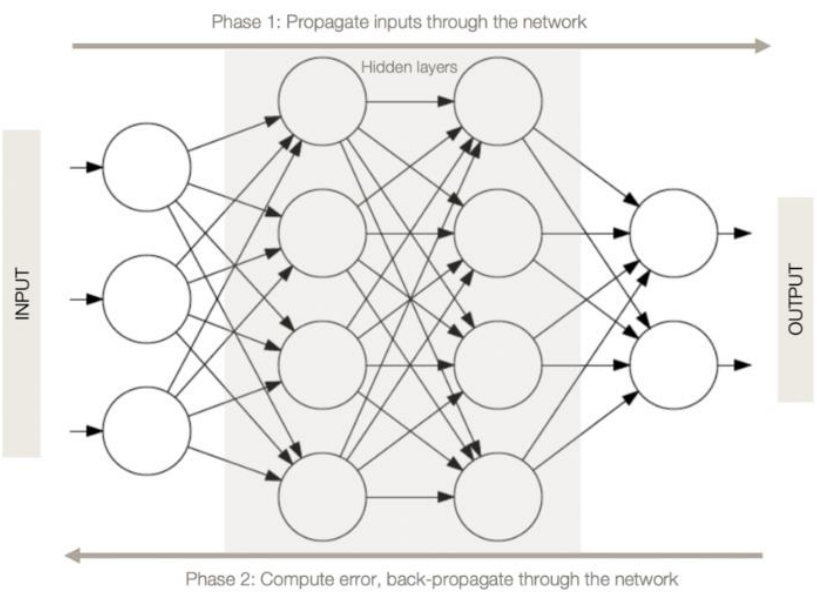
1986

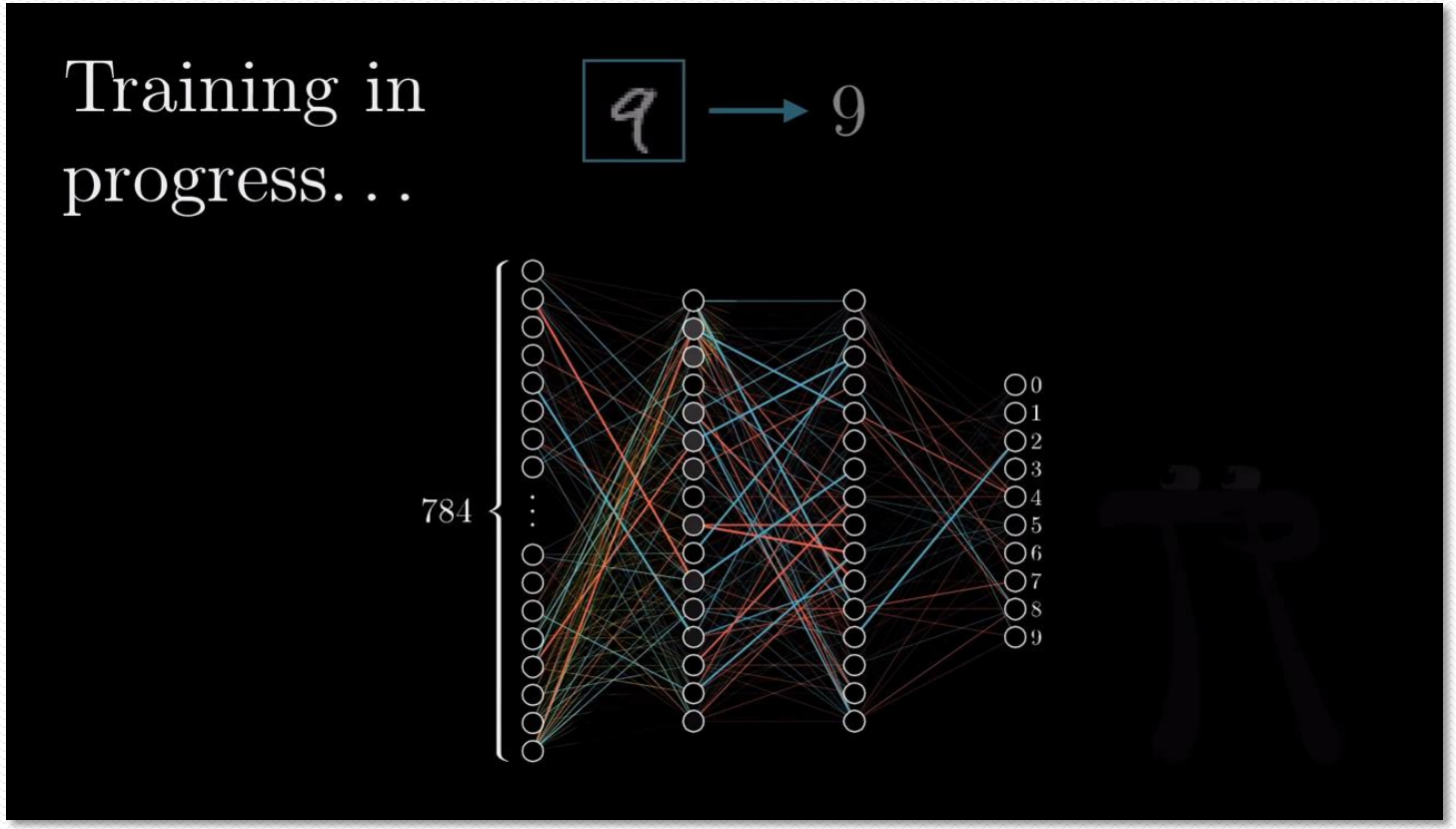
David Rumelhart

Multi Layer Perceptron (MLP)



- Geoffrey Hinton, Ronald Williams 와 함께 **Back Propagation** 에 대한 논문을 통해 Multi Layer Perceptron을 Training 할수 있음을 언급
- 다시 AI 에 대한 열기를 가져옴



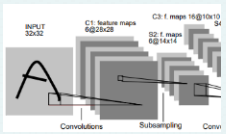


https://www.researchgate.net/figure/Deep-learning-backpropagation-in-a-nutshell-The-Economist-2016-Tim-Jones-2017_fig11_330382022

HISTORY OF A.I.

Yann Lecun

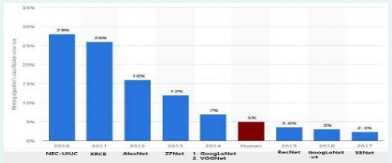
LeNet-5



1998

ResNet

Accuracy is higher than human

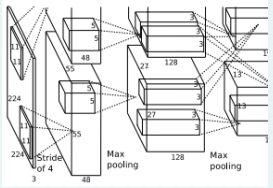


2015

2012

Alex Krizhevsky

AlexNet



2016

Google Brain

AlphaGo

2번째 위기

- Hidden layer가 깊어지면 back propagation이 제대로 안됨

LeNet-5

- Paper : Gradient-based learning applied to document recognition
- Multilayer Neural Network의 한계를 극복하기 위해 CNN을 활용
- MNIST Dataset

AlexNet

- Paper: ImageNet Classification with Deep Convolutional Neural Networks
- 2012년 ILSVRC 우승

ResNet

- 2015년 ILSVRC 우승
- Top-5 test error : 3.57%
- Residual neural network

ILSVRC : (ImageNet Large Scale Visual Recognition Challenge)

Session Break

AI TRAINING SERIES

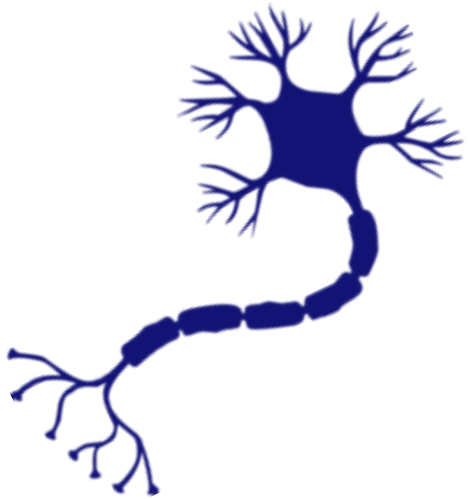
ANN

ARTIFICIAL NEURAL NETWORKS

CONTENTS

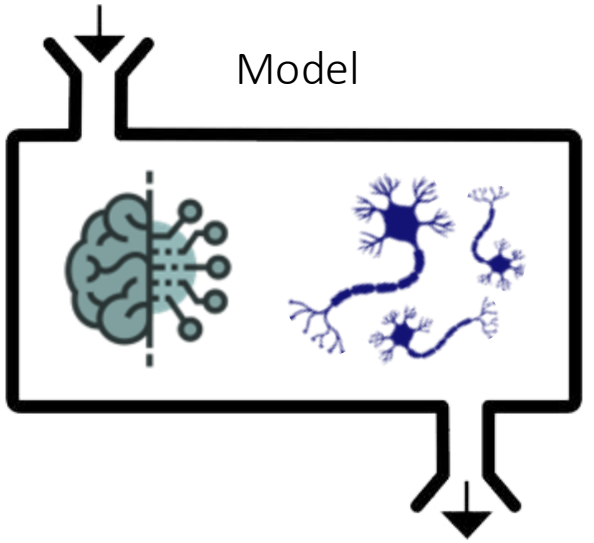
NEWRAL NETWORKS OVERVIEW

Neural networks

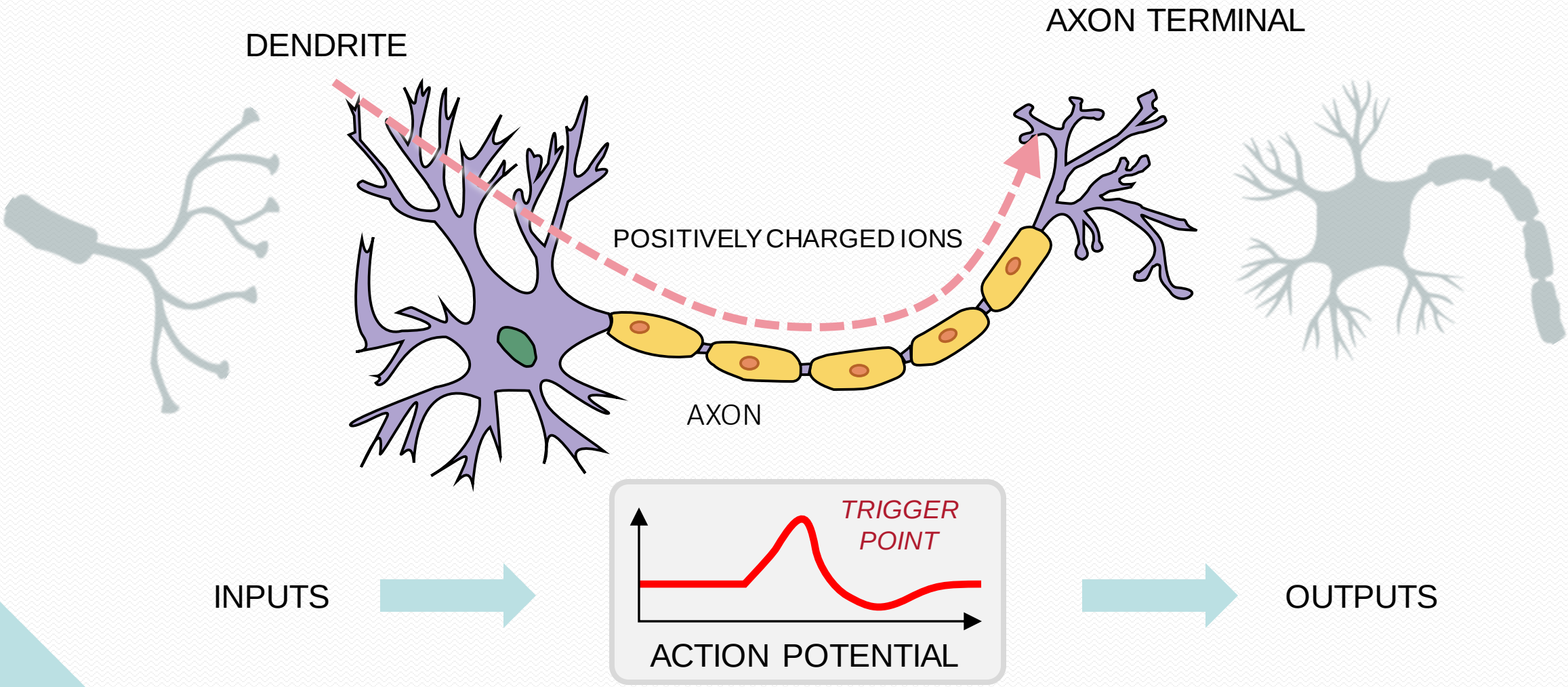


Neuron in human brain

Model

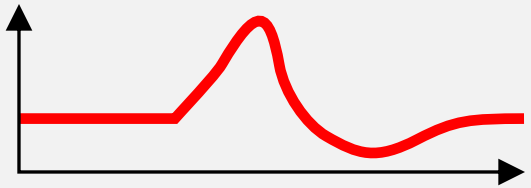


NEURON IN OUR BRAINS



ACTIVATION

INPUTS



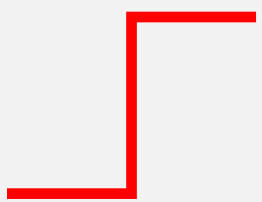
ACTION POTENTIAL

OUTPUTS

PERCEPTRON

INPUTS

CELL BODY



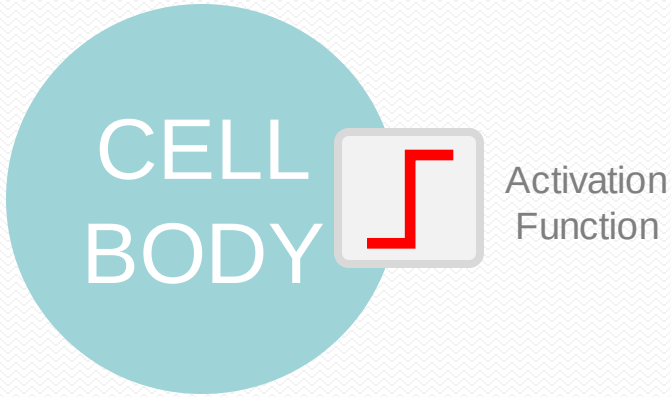
ACTIVATION FUNCTION

OUTPUT(S)

WEIGHTS AND BIASES

WEIGHTS

- CONTROL THE SIGNAL BETWEEN TWO NEURONS
- DETERMINES HOW THE INPUT AFFECTS THE OUTPUT



$$Y = \Sigma(weight \cdot input) + bias$$

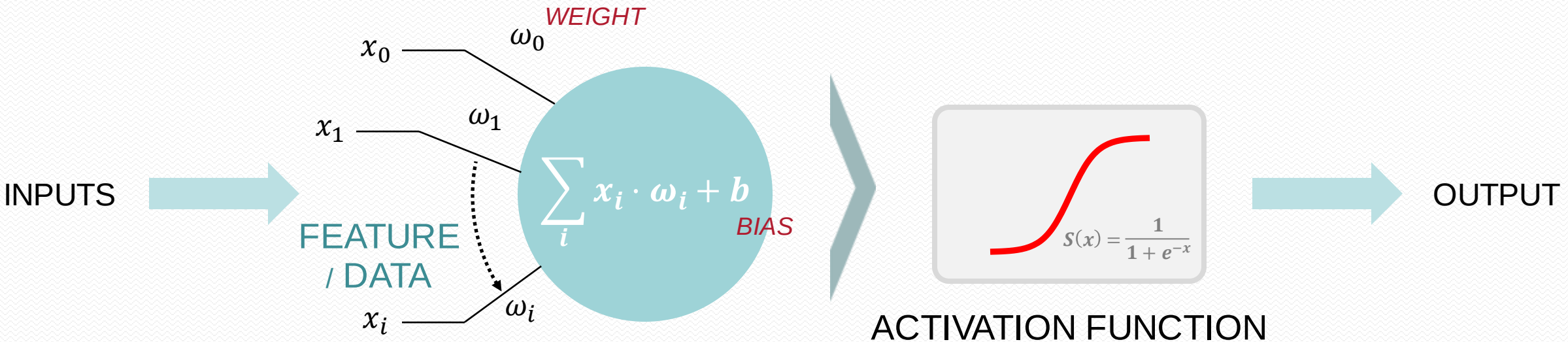
BIASES

- ADDITIONAL INPUT CONSTANT VALUES FOR NEXT NEURONS
- NOT INFLUENCED BY THE PREVIOUS NEURONS

  These are the connections that go out of their own weights

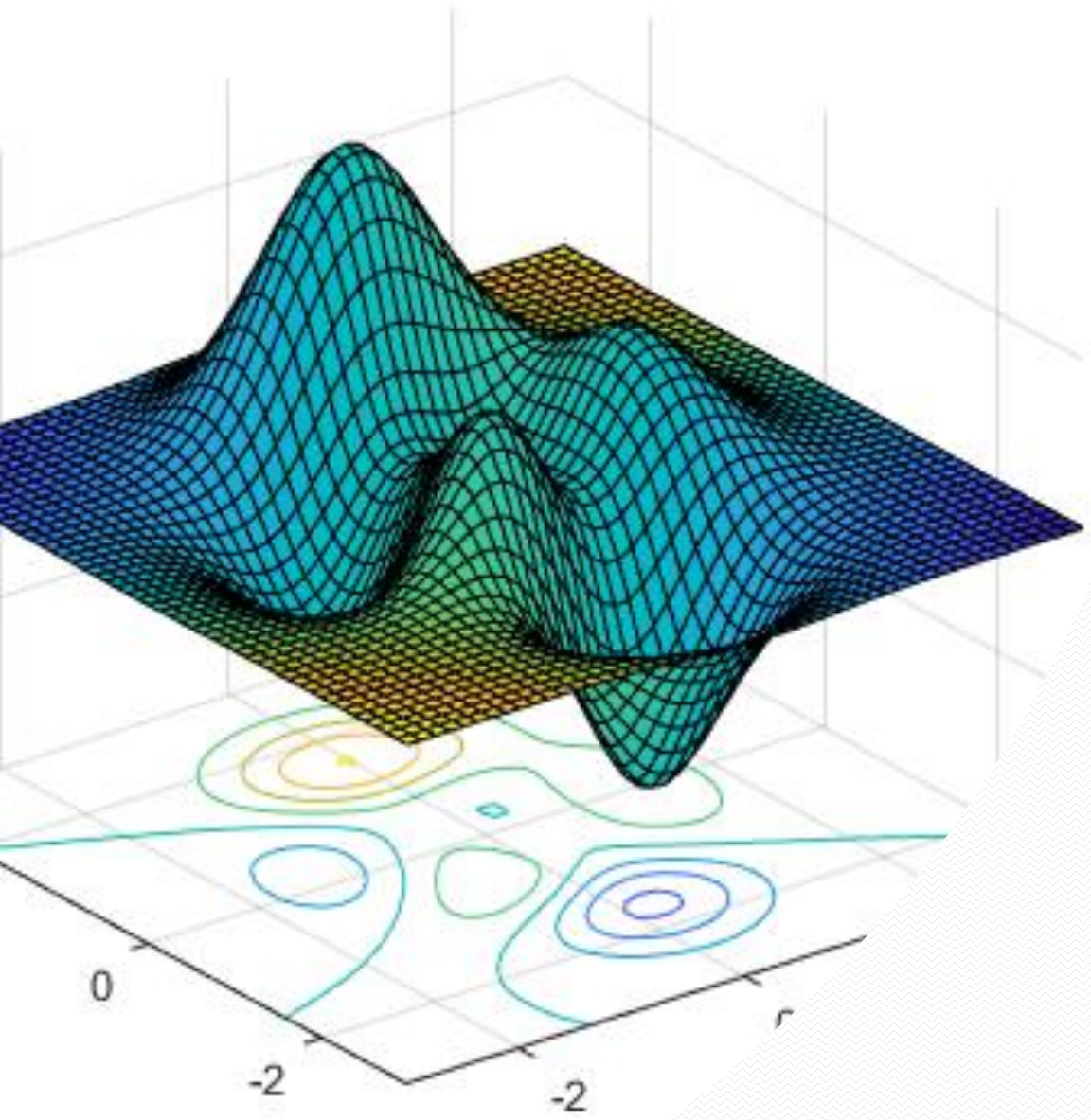
PERCEPTRON

ALGORITHM FOR SUPERVISED LEARNING OF BINARY CLASSIFIERS



$$H(x) = W_x + b$$

HYPOTHESIS



Perceptron

Logic gate

PERCEPTRON EXAMPLE

GO OUT

Alice

H | w | M

Bob

h | w | M

Charlie

H | w | M

	ω_h	ω_w	ω_m
Alice	**.4**	.3	**.3**
Bob	.2	.1	**.7**
Charlie	**.2**	.5	**.3**

Alice *(.7)*

Go out

Bob *(.7)*

Go out

Charlie *(.5)*

Stay

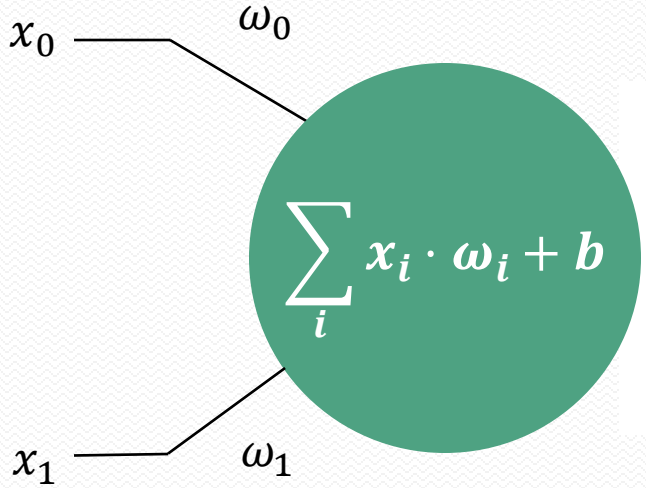
Perceptron Example – **Logical OR**

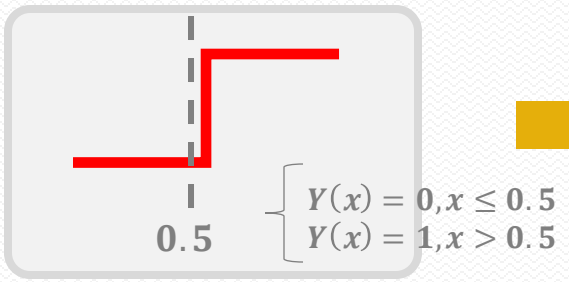
- Q. Find w_0, w_1 and b that makes logical **OR** output.

x_0	x_1	w_0	w_1	b	H(x)	Output
0	0	0.2	0.2	0.4	0.4	0
0	1				0.6	1
1	0				0.6	1
1	1				0.8	1

W0	W1	b
0.1	0.1	0.5
0.2	0.2	0.5
0.2	0.3	0.4
0.3	0.2	0.4
	.	
	.	
	.	

x_0	x_1	Output
0	0	0
0	1	1
1	0	1
1	1	1





output

Activation Function

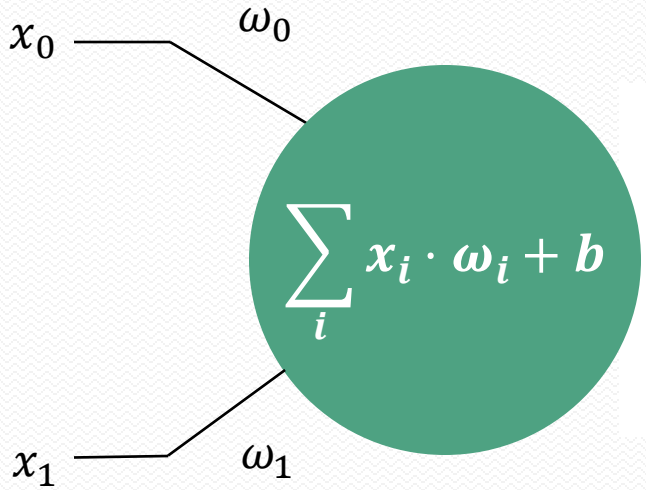
Perceptron Example – **Logical NAND**

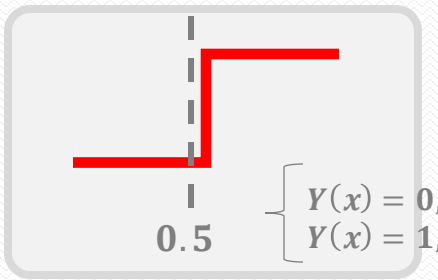
- Q. Find w_0, w_1 and b that makes logical **NAND** output.

x_0	x_1	w_0	w_1	b	H(x)	Output
0	0	-0.5	-0.5	1.1	1.1	1
0	1				0.6	1
1	0				0.6	1
1	1				0.1	0

W0	W1	b
-0.2	-0.2	0.8
-0.3	-0.3	0.9
-0.4	-0.4	1.0
-0.5	-0.5	1.1
	.	
	.	
	.	

x_0	x_1	Output
0	0	1
0	1	1
1	0	1
1	1	0





Activation Function

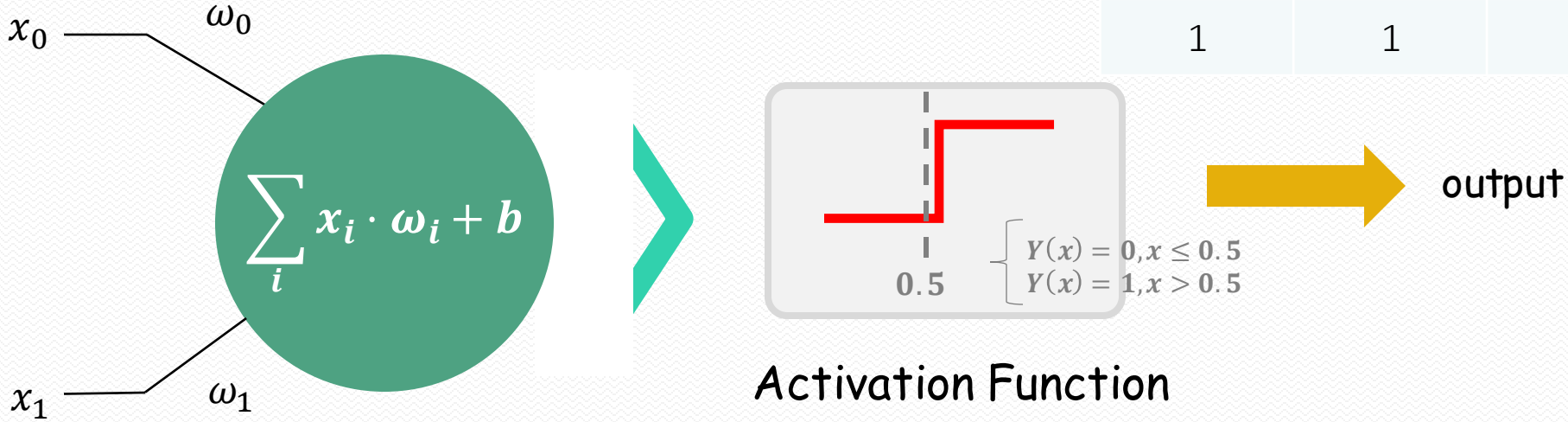
output

Perceptron Example – **Logical XOR**

- Q. Find w_0, w_1 and b that makes logical **XOR** output.

x_0	x_1	w_0	w_1	b	H(x)	Output
0	0				0	0
0	1				0	0
1	0				0	0
1	1				0	0

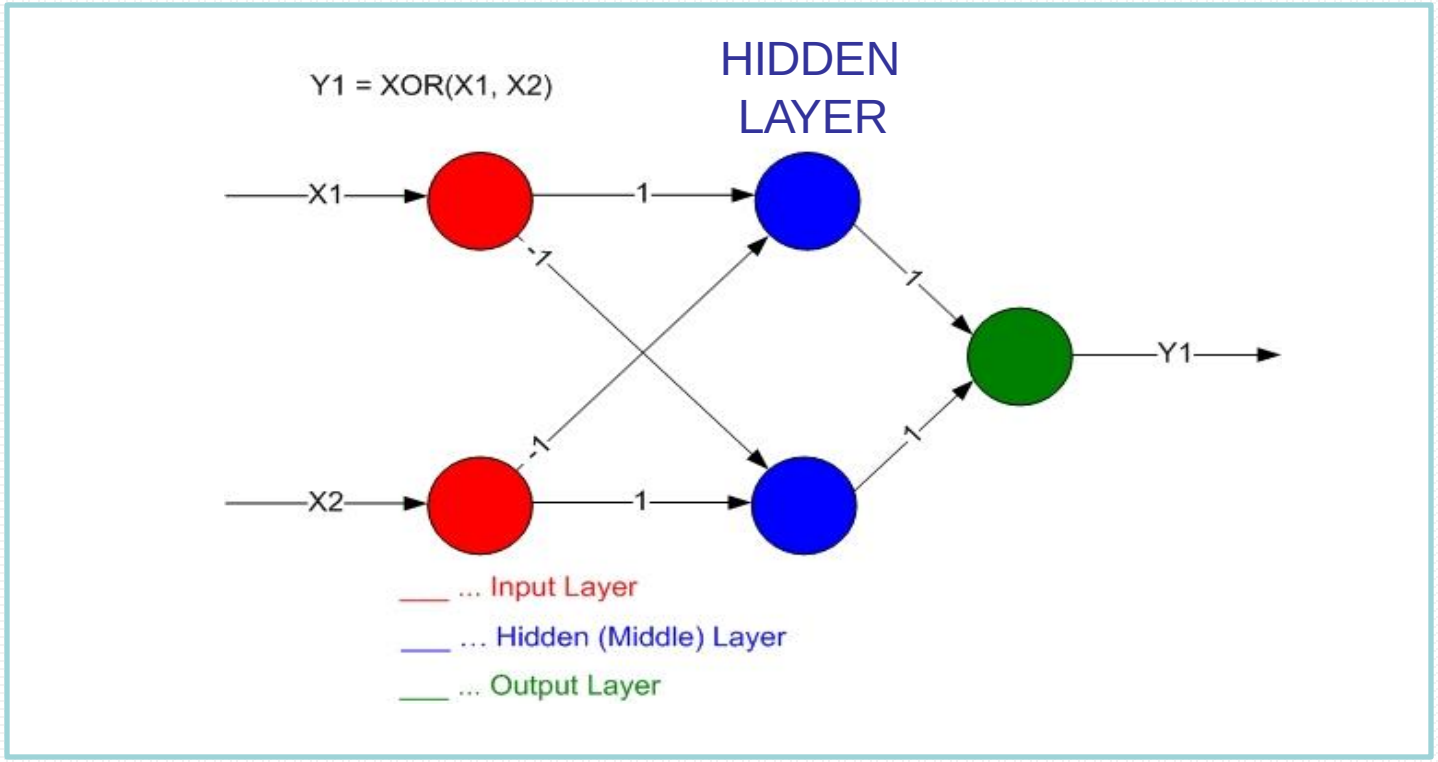
x_0	x_1	Output
0	0	0
0	1	1
1	0	1
1	1	0

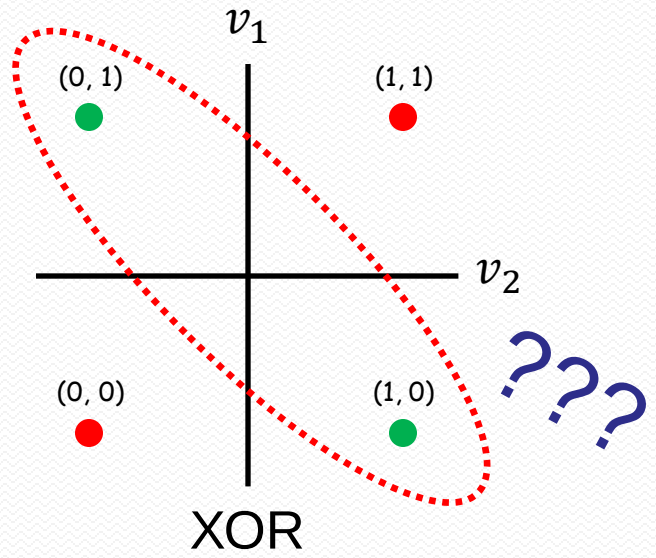


PROBLEM OF PERCEPTRON

PERCEPTRON IS NOT AN ALL-ROUND PLAYER

PROBLEMS THAT ARE NOT CLASSIFIED AS LINEAR CANNOT BE CALCULATED
e.g., XOR logic / impossible to solve with a single layer of perceptron

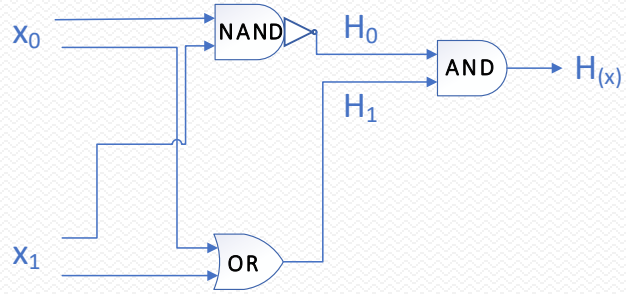


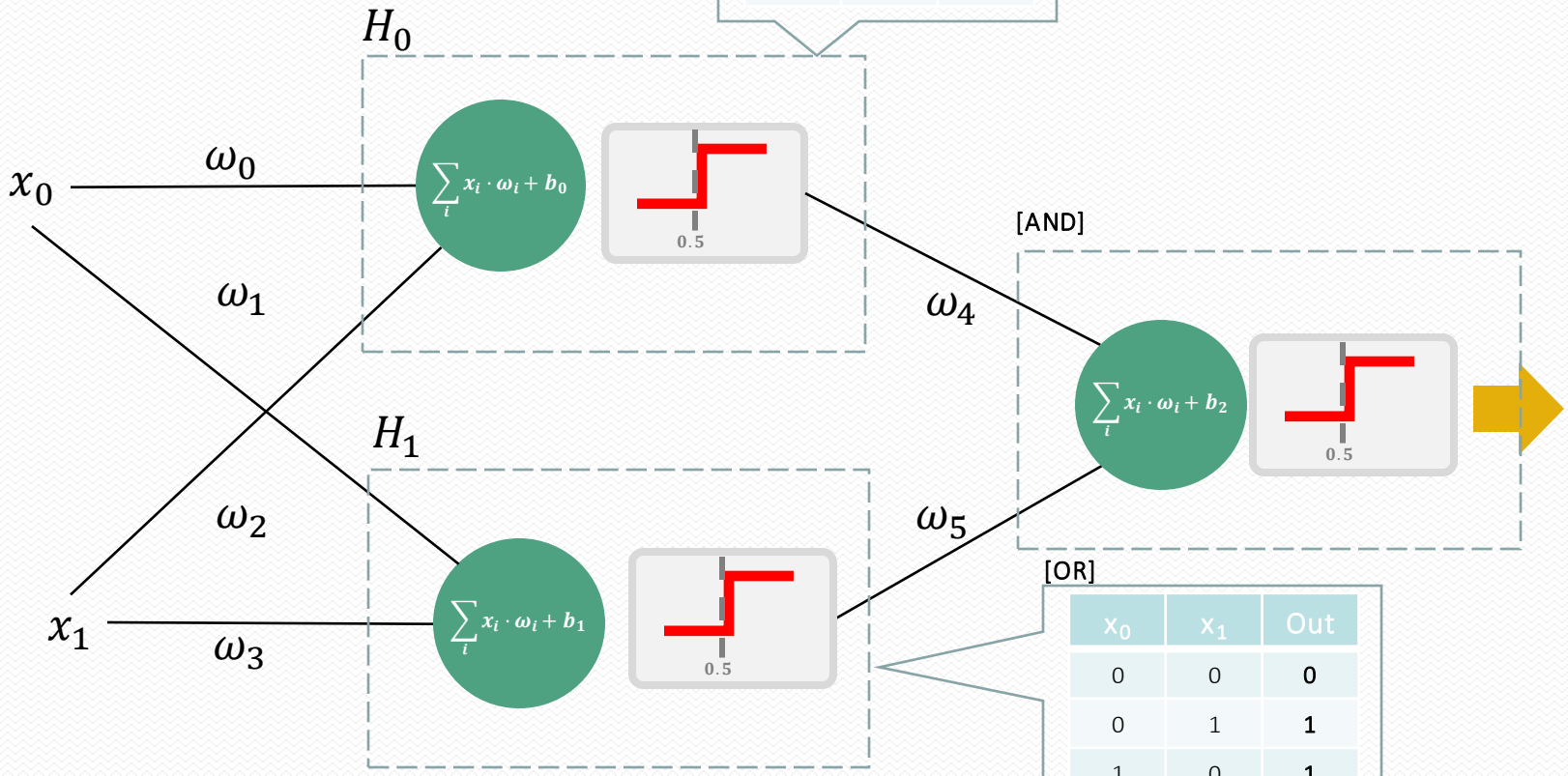


Multi Layer Perceptron - XOR

[NAND]

x_0	x_1	Out
0	0	1
0	1	1
1	0	1
1	1	0



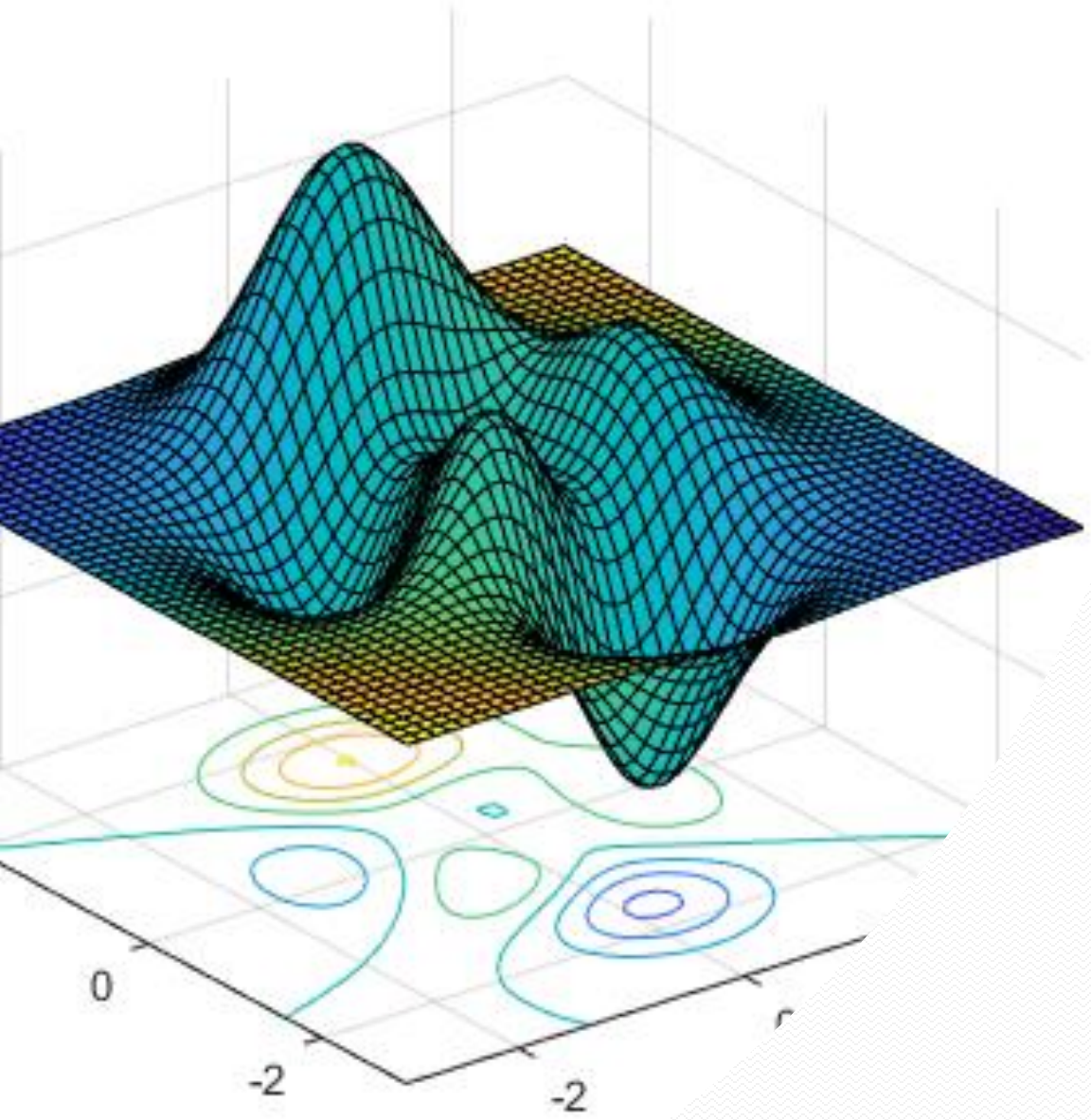


[XOR]

w_0	x_1	Out
0	0	0
0	1	1
1	0	1
1	1	0

[OR]

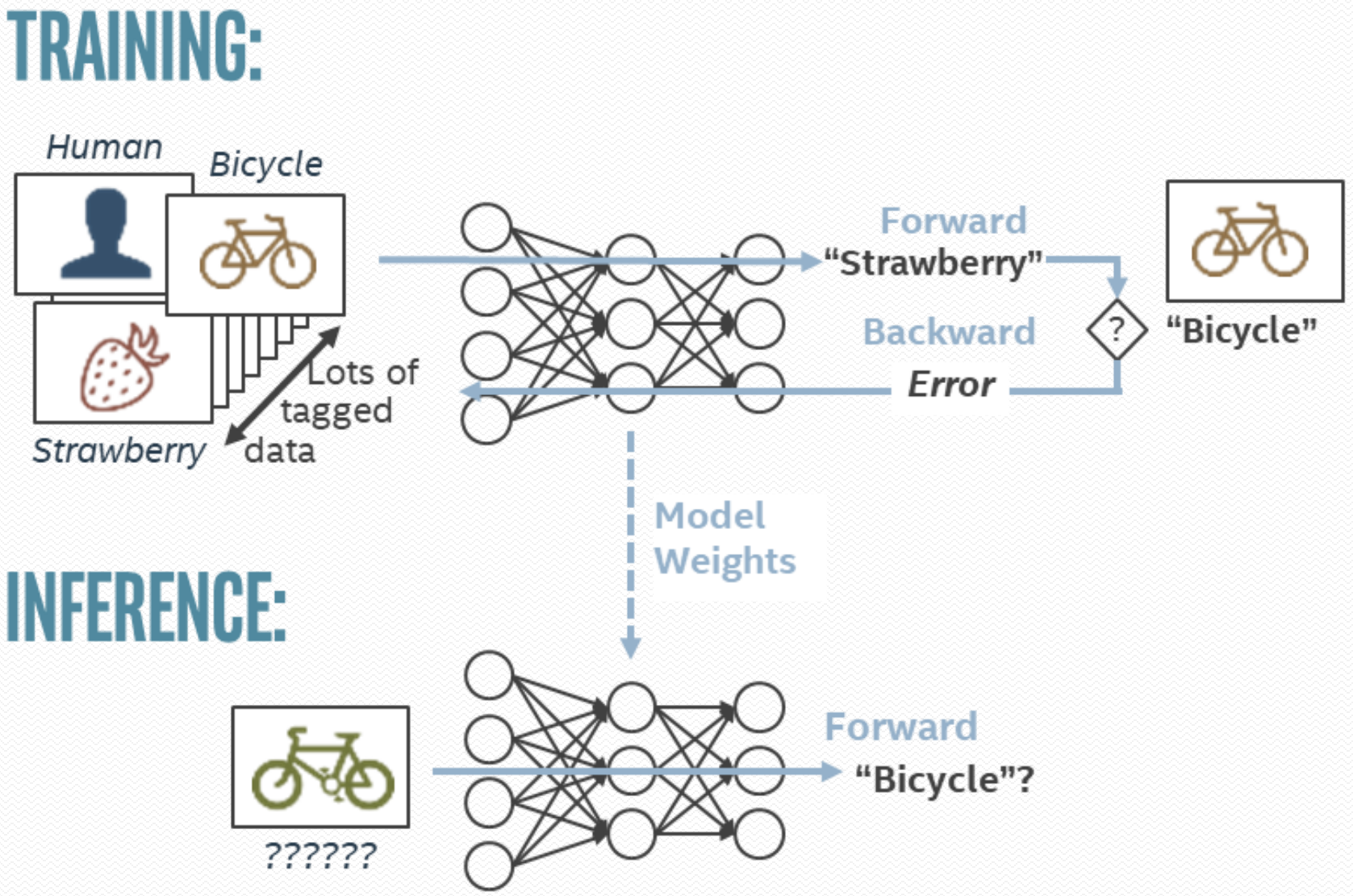
x_0	x_1	Out
0	0	0
0	1	1
1	0	1
1	1	1



Deep-dive

Technical

TRAINING & INFERENCING

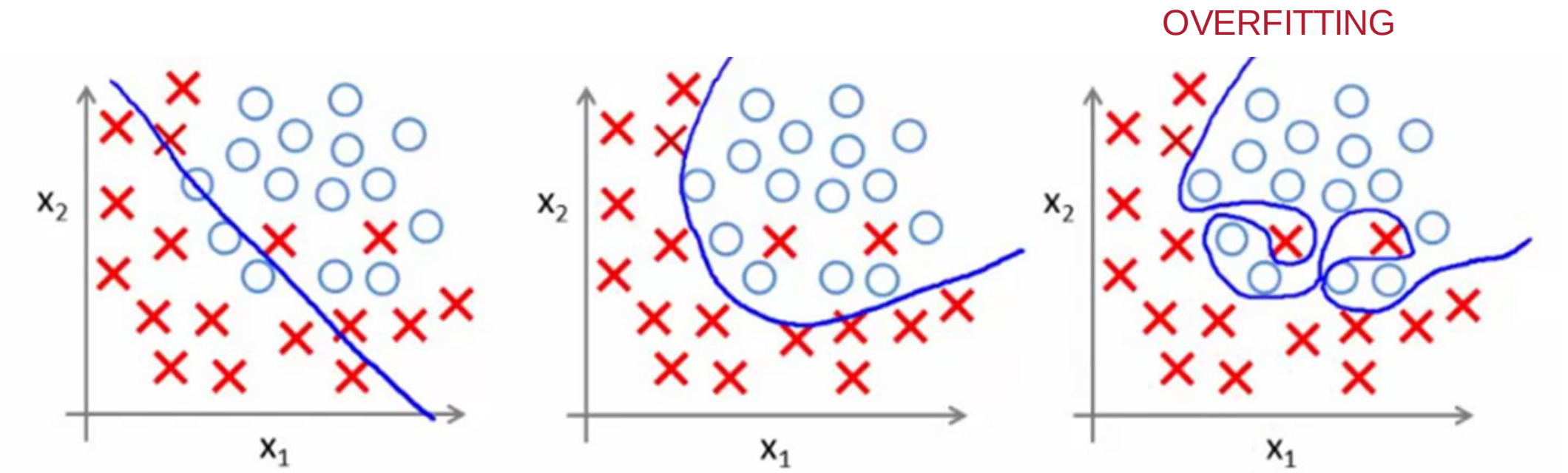


OVERFITTING / UNDERFITTING

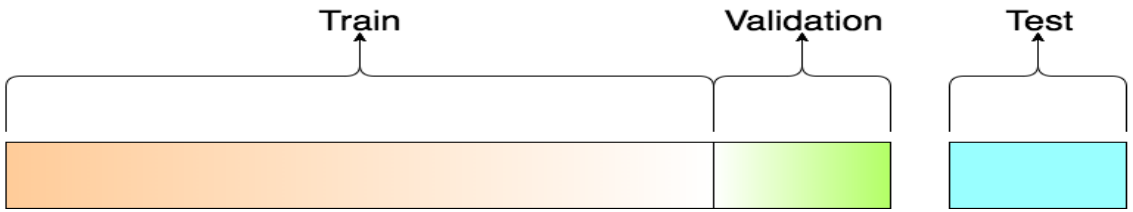
HIGH COMPLEXITY MODEL = HIGH POSSIBILITY OF OVERFITTING

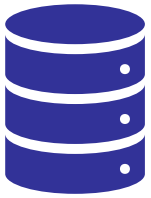
If the model is too specific, generalization is reduced

When new test data sets come in → they may not be well categorized



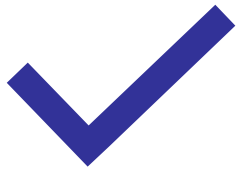
DATASET





TRAIN

The actual dataset that we use to train the model (**weights** and **biases** in the case of a Neural Network).



VALIDATION

The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model **hyperparameters**.

TEST

The sample of data used to provide an **unbiased** evaluation of a final model fit on the training dataset.

HYPERPARAMETERS

 # OF HIDDEN LAYER

determines how many hidden layers is in neural network

 # OF EPOCH

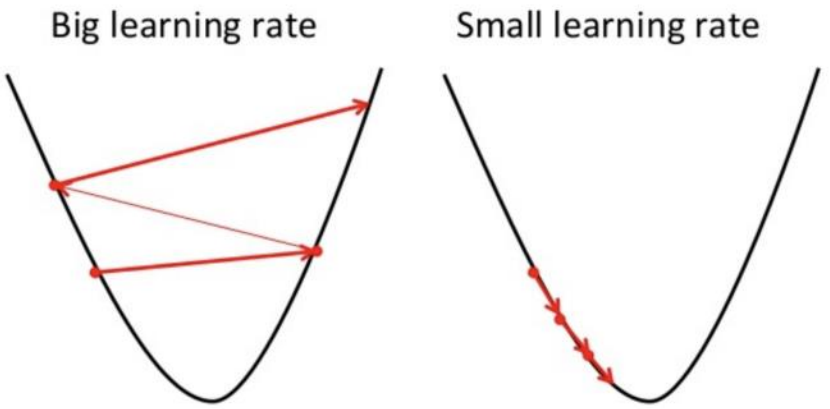
finds the boundary value of the epoch where the error is minimized

BATCH SIZE

means the size of the data to be divided into when learning

LEARNING RATE

determines how fast or slow to move per iteration



INITIAL VALUE OF EACH PARAMETER

determines the initial value when running iteration

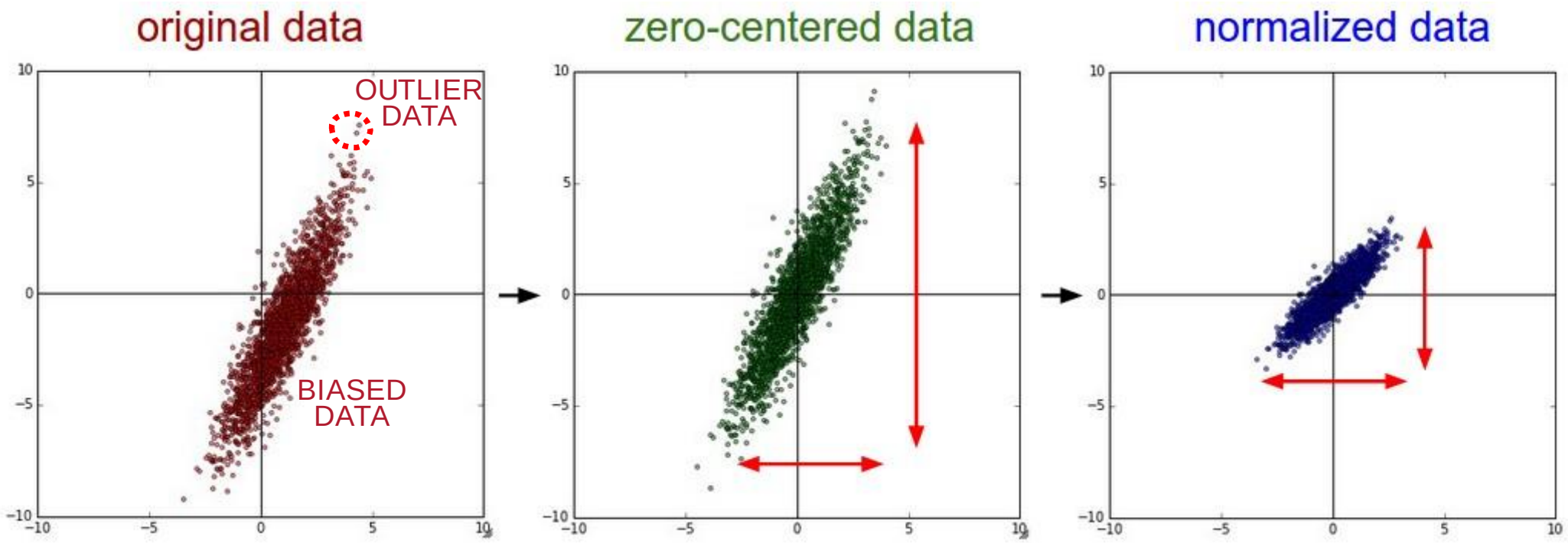
LOSS / COST FUNCTION

Decides how to calculate between actual result and calculated output

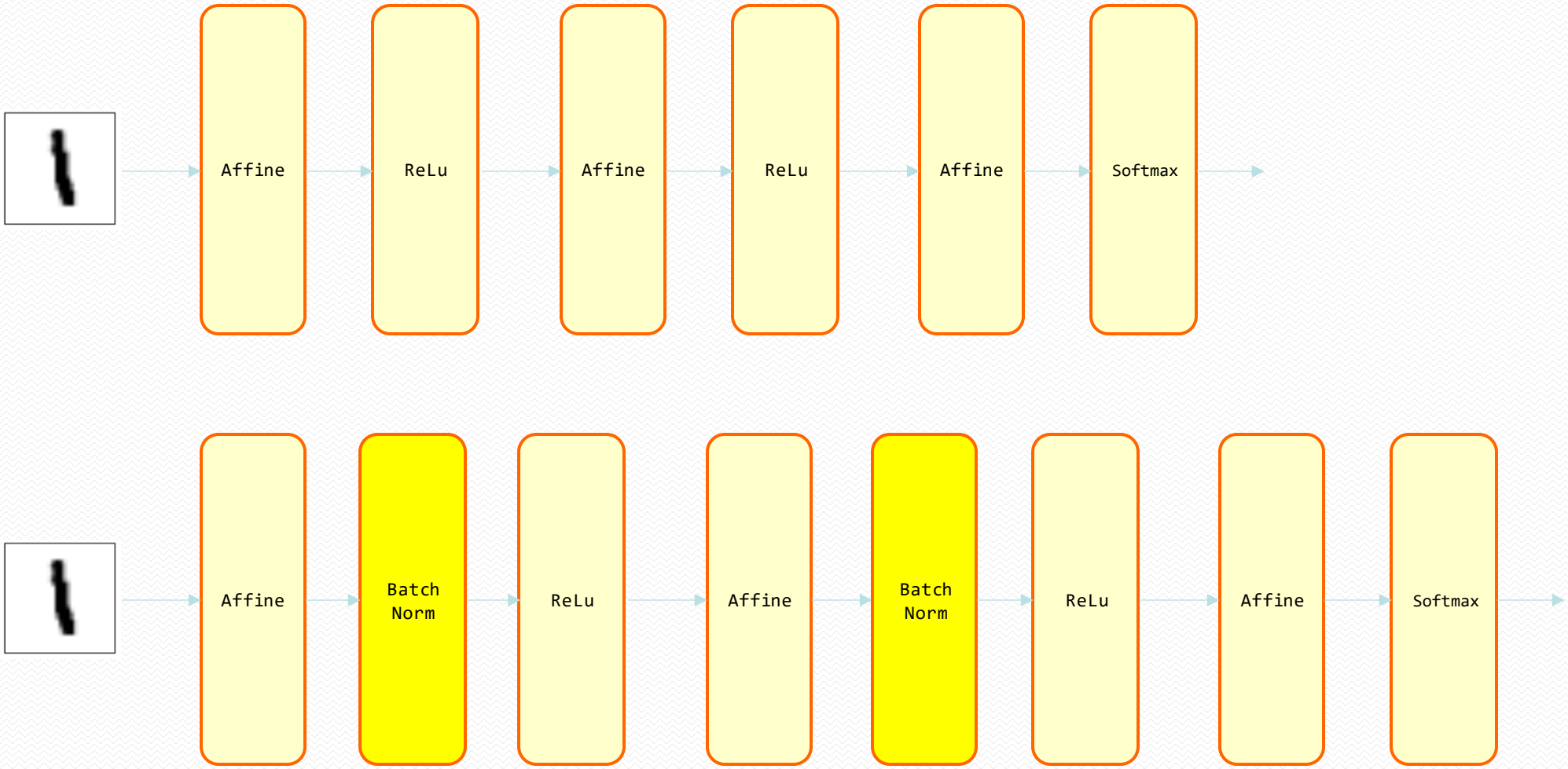
BATCH NORMALIZATION

NORMALIZING THE DATA DIMENSIONS

Method to help the network learn better



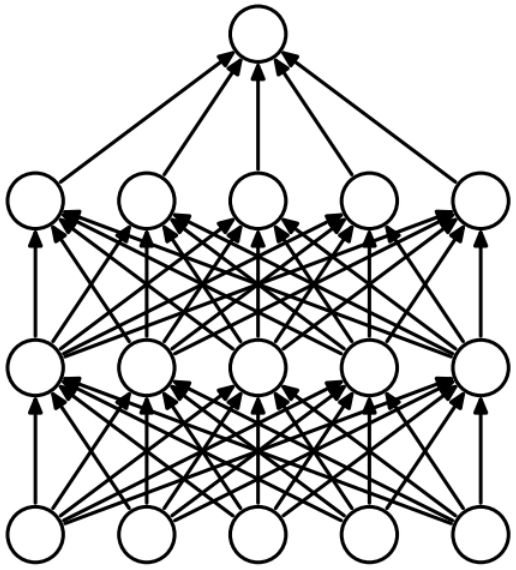
배치정규화 알고리즘


Affine
ReLu
Affine
ReLu
Affine
Softmax
Affine
Batch Norm
ReLu
Affine
Batch Norm
ReLu
Affine
Softmax

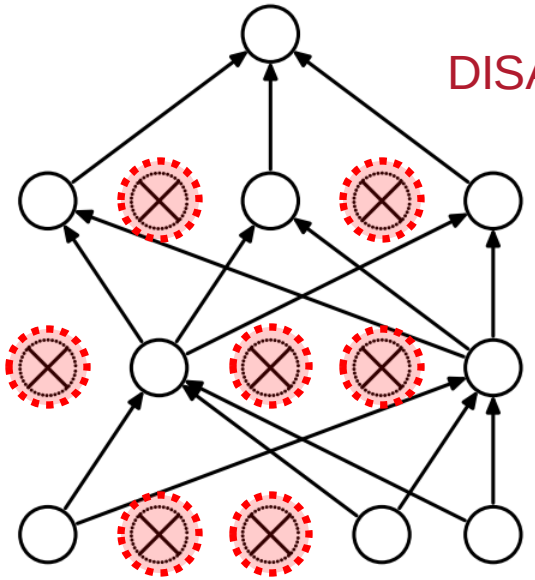
DROPOUT

RANDOMLY SET SOME NEURONS TO ZERO IN THE FORWARD PASS

Voting effect → makes the result not biased
Anti co-adaptation → creates a robust network



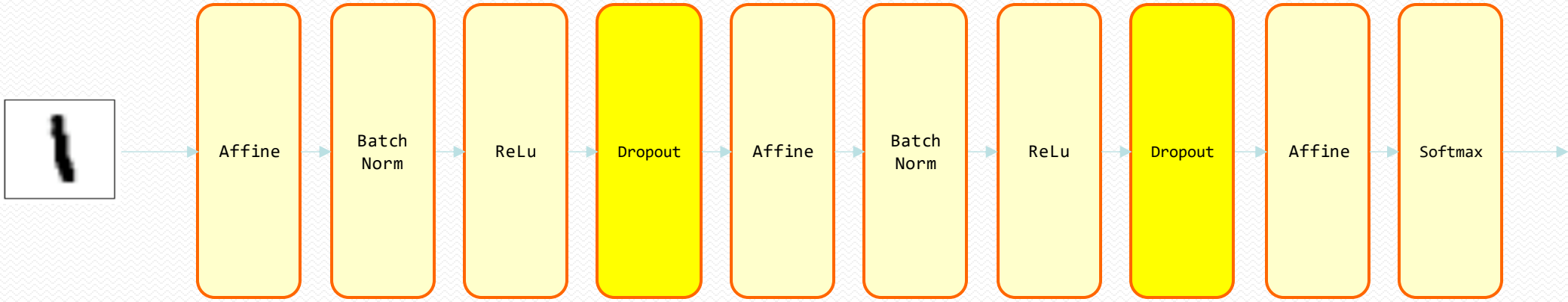
(a) Standard Neural Net

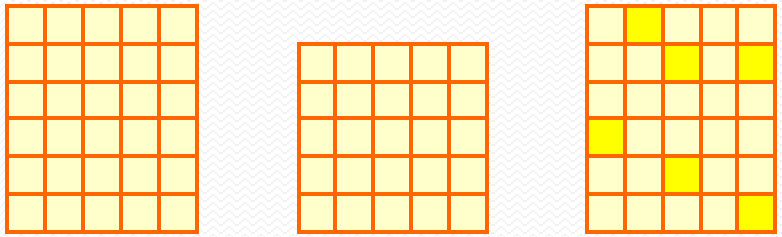


(b) After applying dropout.

DISABLE RANDOMLY

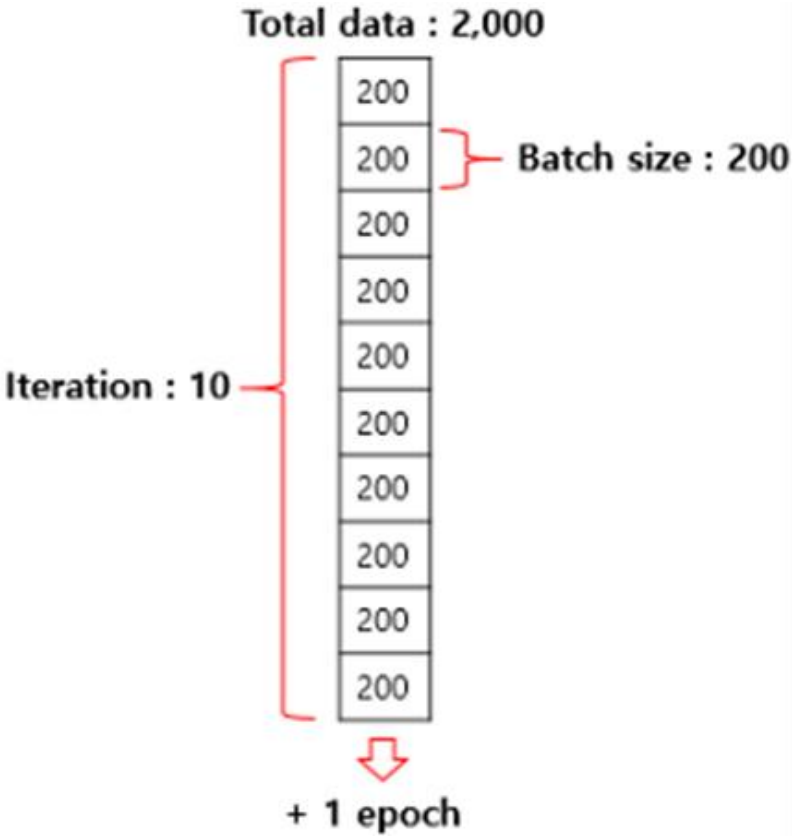
드롭아웃


Affine
Batch Norm
ReLu
Dropout
Affine
Batch Norm
ReLu
Dropout
Affine
Softmax



(6,5) (5,5) (6,5)

EPOCH & ITERATION & BATCH



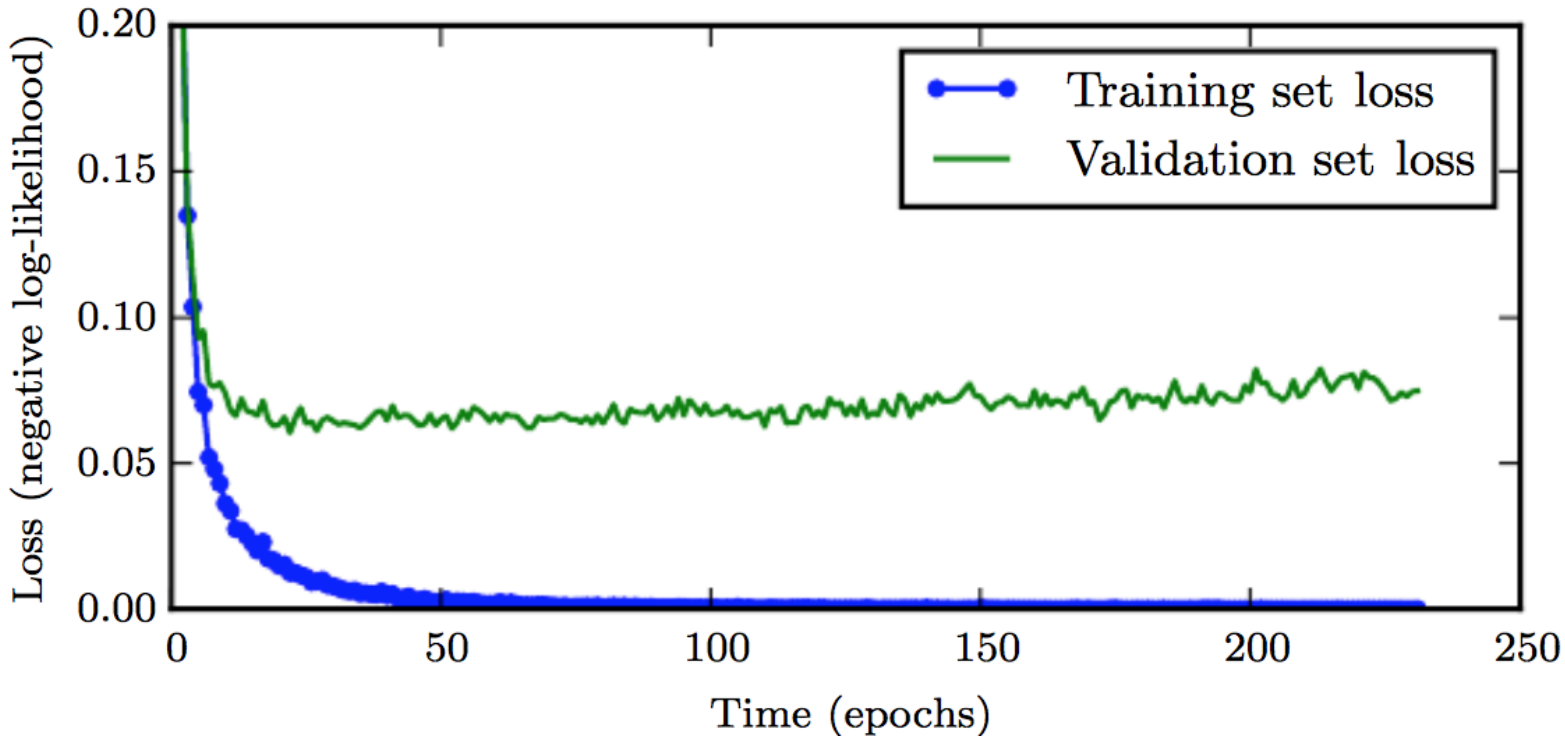
GENERALIZATION / RIZATION

EARLY STOPPING

Stop training when the gap between validation error and training error becomes large

Simple to do without additional hyperparameters, but mixing two problems

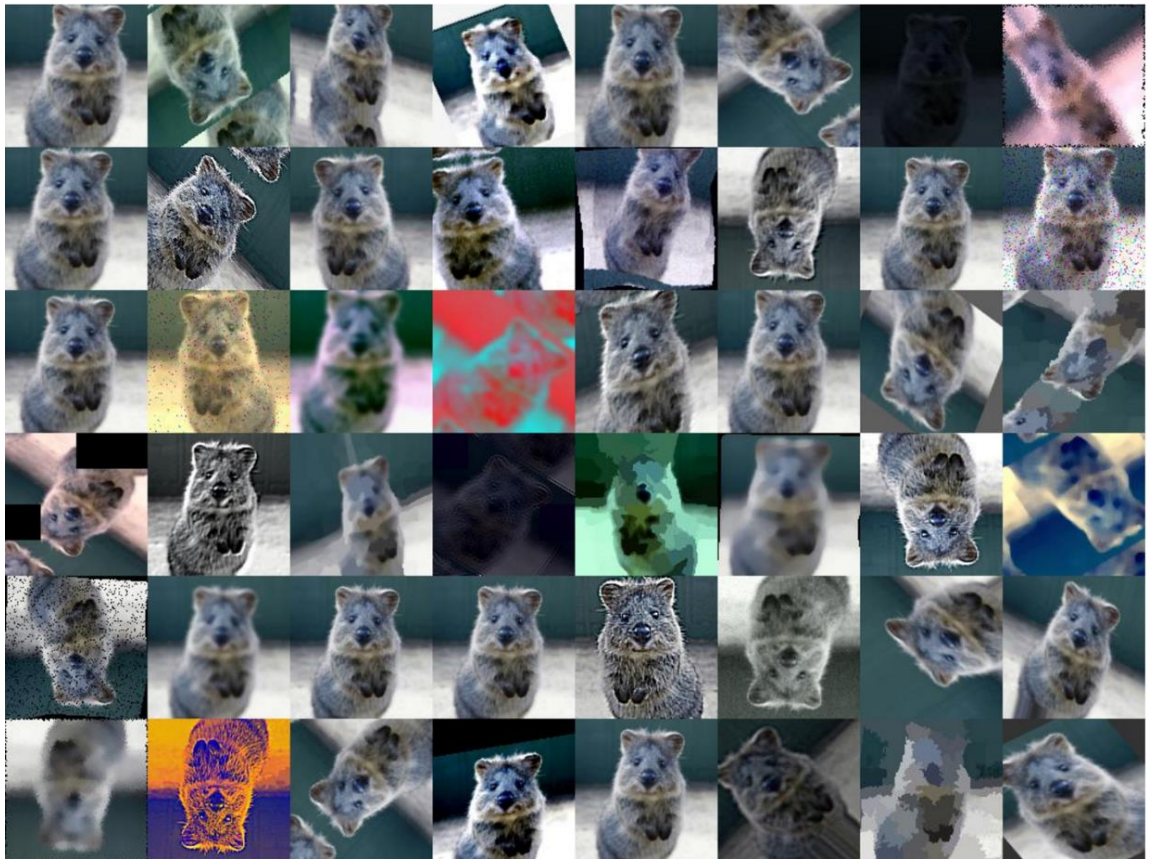
→ *UNDERFITTING AND OVERFITTING*

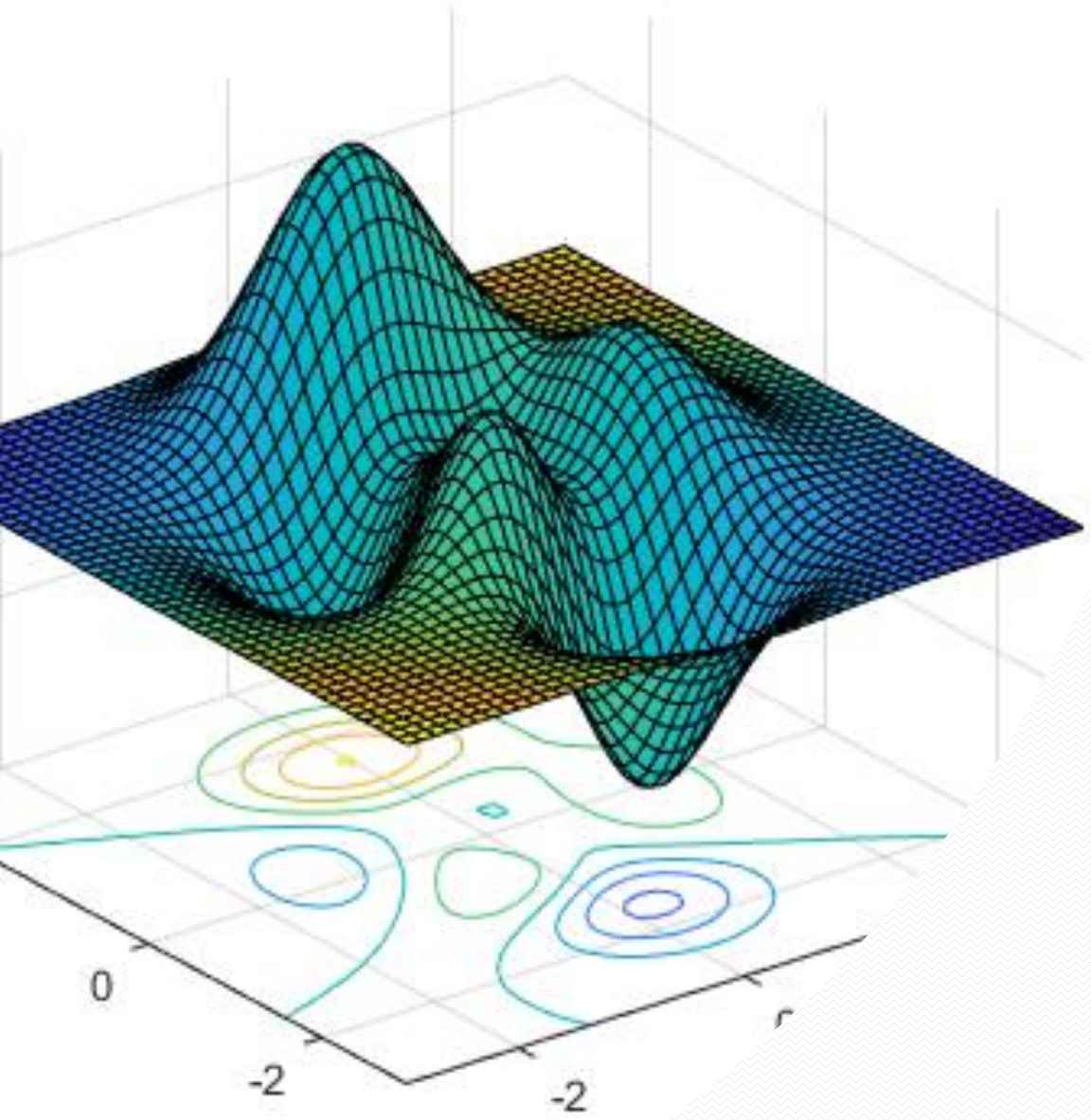


GENERALIZATION / REGULARIZATION

DATA AUGMENTATION

Strategy for improve CNN performance that enables user to significantly increase the diversity of data available for training models, without collecting new data





Deep-dive

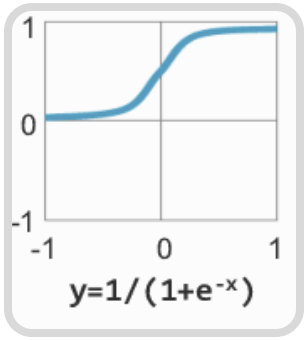
Activation Function

ACTIVATION FUNCTIONS

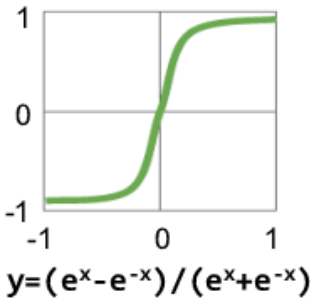
DEFINES THE OUTPUT OF A NODE USING A GIVEN INPUT OR SET OF INPUTS

Traditional Non-Linear Activation Functions

Sigmoid



Hyperbolic Tangent

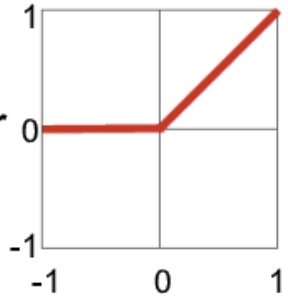


For binary classification

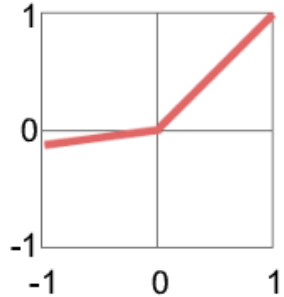
Widely used in hidden layers

Modern Non-Linear Activation Functions

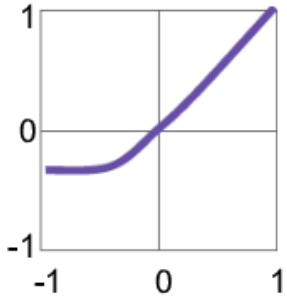
Rectified Linear Unit (ReLU)



Leaky ReLU



Exponential LU



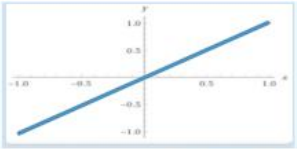
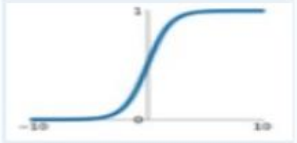
α = small const. (e.g. 0.1)

ACTIVATION FUNCTIONS

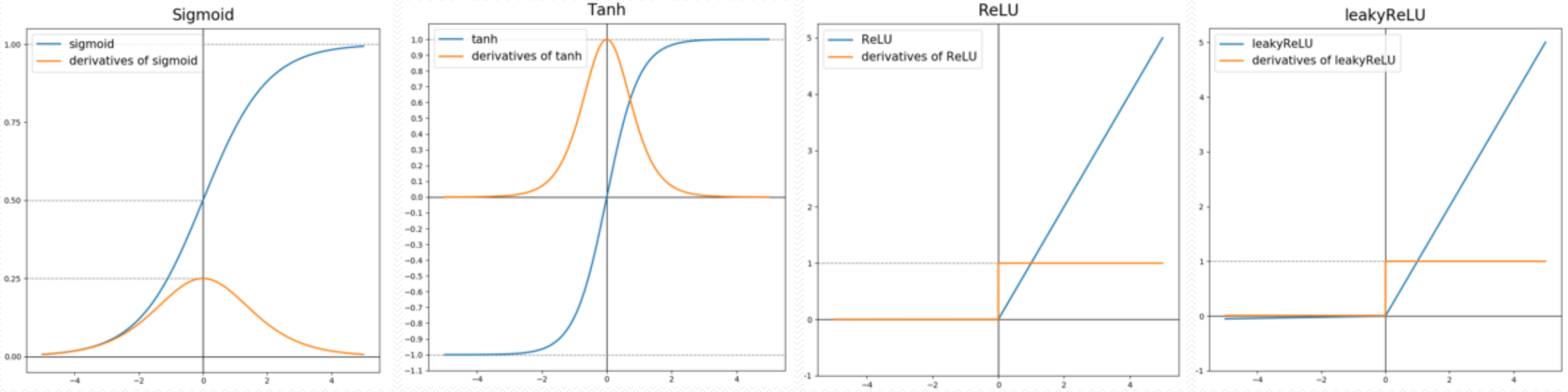
ACTIVATION FUNCTIONS

Why derivative/differentiation is used?

To know in which direction and how much to change or update the curve depending upon the slope.

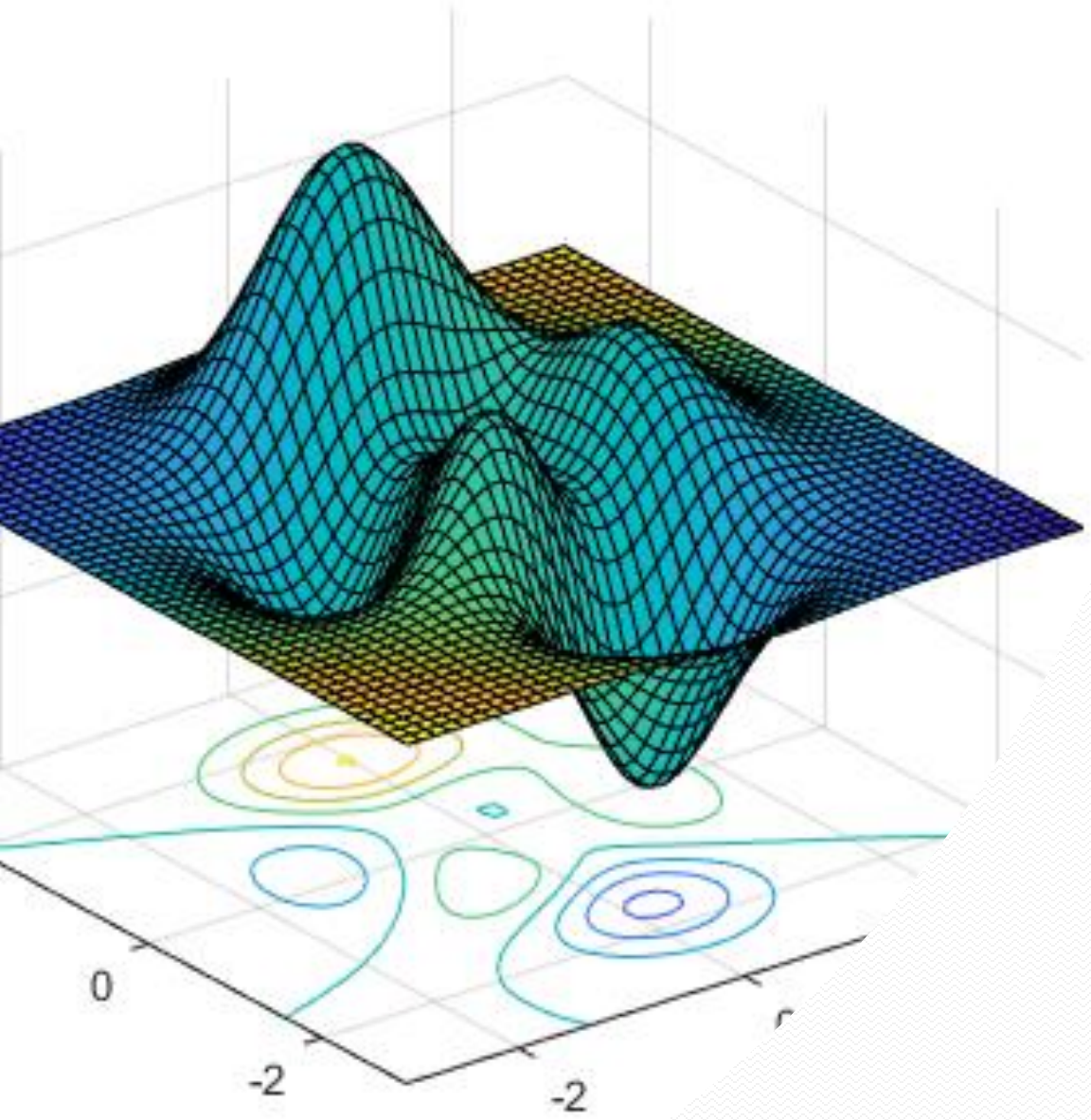
Name	Plot	Equation	Derivative
Linear		$f(x) = x$	$f'(x) = 1$
Sigmoid		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 \ for \ x < 0 \\ x \ for \ x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 \ for \ x < 0 \\ 1 \ for \ x \geq 0 \end{cases}$
Leaky ReLU		$f(x) = \begin{cases} ax \ for \ x < 0 \\ x \ for \ x \geq 0 \end{cases}$	$f'(x) = \begin{cases} a \ for \ x < 0 \\ 1 \ for \ x \geq 0 \end{cases}$

ACTIVATION FUNCTIONS



	Sigmoid	Tanh	ReLU	Leaky ReLU
Disadvantage	Vanishing Gradient Computational penalty	Vanishing Gradient	Dying ReLU problem	Result not consistent for negative
Advantage	1 or 0, binary value	Zero centered	Computational efficient Non-linear	Prevent negative Values become zero

https://ganghee-lee.tistory.com/32



Deep-dive

Optimizer

Gradient descent optimization Algo.

BGD (Batch Gradient Descent)

- FULL TRAIN DATASET, REQUIRES A LOT OF COMPUTATION

SGD (Stochastic Gradient Descent)

- USE Some of training sample, calculate gradient

Momentum

- Method that helps accelerate SGD in the relevant direction and dampens oscillations by adding update vector of the past time step to the current update vector

Image 2: SGD without momentum

Image 3: SGD with momentum

Gradient descent optimization Algo.

Adagrad (Adaptive Gradient)

- Adapts learning rate to the parameters, performing smaller updates(i.e. low learning rates) for parameters associated with frequently occurring features, and larger updates (i.e. high learning rates) for parameters associated with infrequent features.

Adadelta

- Extension of Adagrad that seeks to reduce its aggressive, monotonically decreasing learning rate. Instead of accumulating all past squared gradients, Adadelta restricts the window of accumulated past gradients to some fixed size.

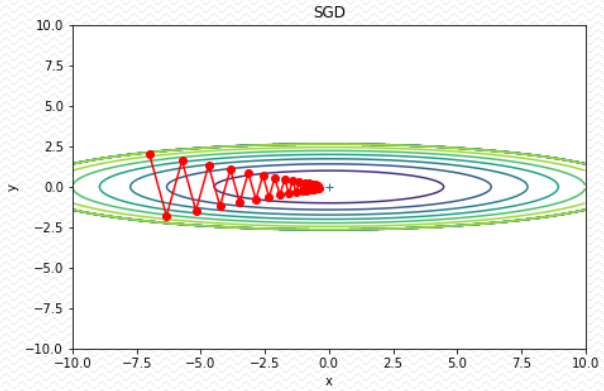
RMSProp

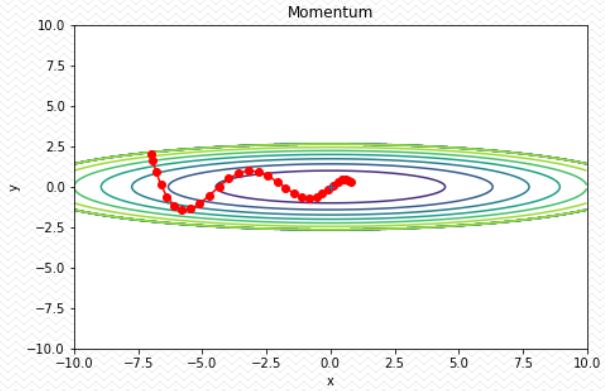
- Need to resolve Adagrad's radically diminishing learning rates. RMSprop divides the learning rate by an exponentially decaying average of squared gradients

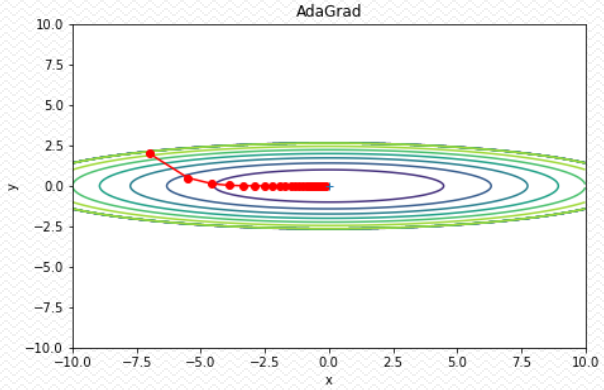
Adam (Adaptive Moment Estimation)

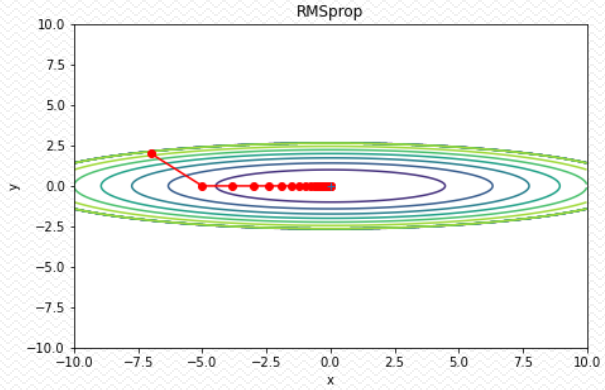
- Another approach to compute adaptive learning rates for each parameter. In addition to storing an exponentially decaying average of past squared gradients like Adadelta and RMSprop, Adam also keeps an exponentially decaying average of past gradients similar to momentum

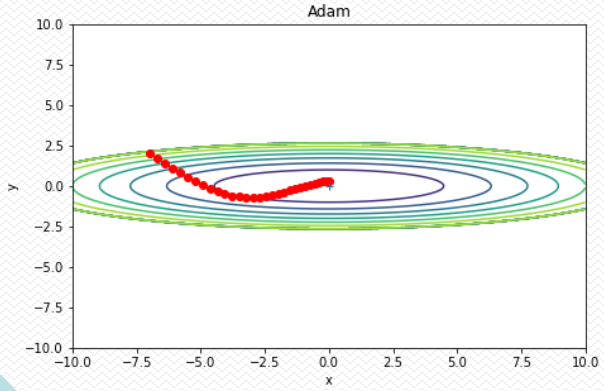
Behavior & Performance

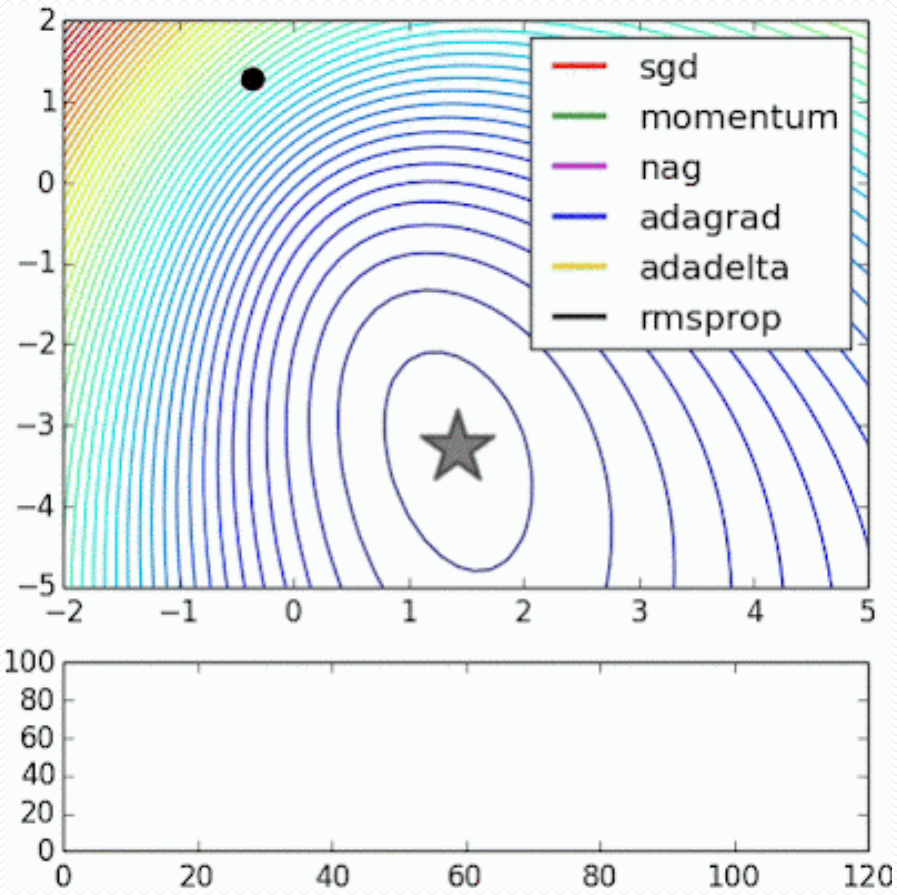




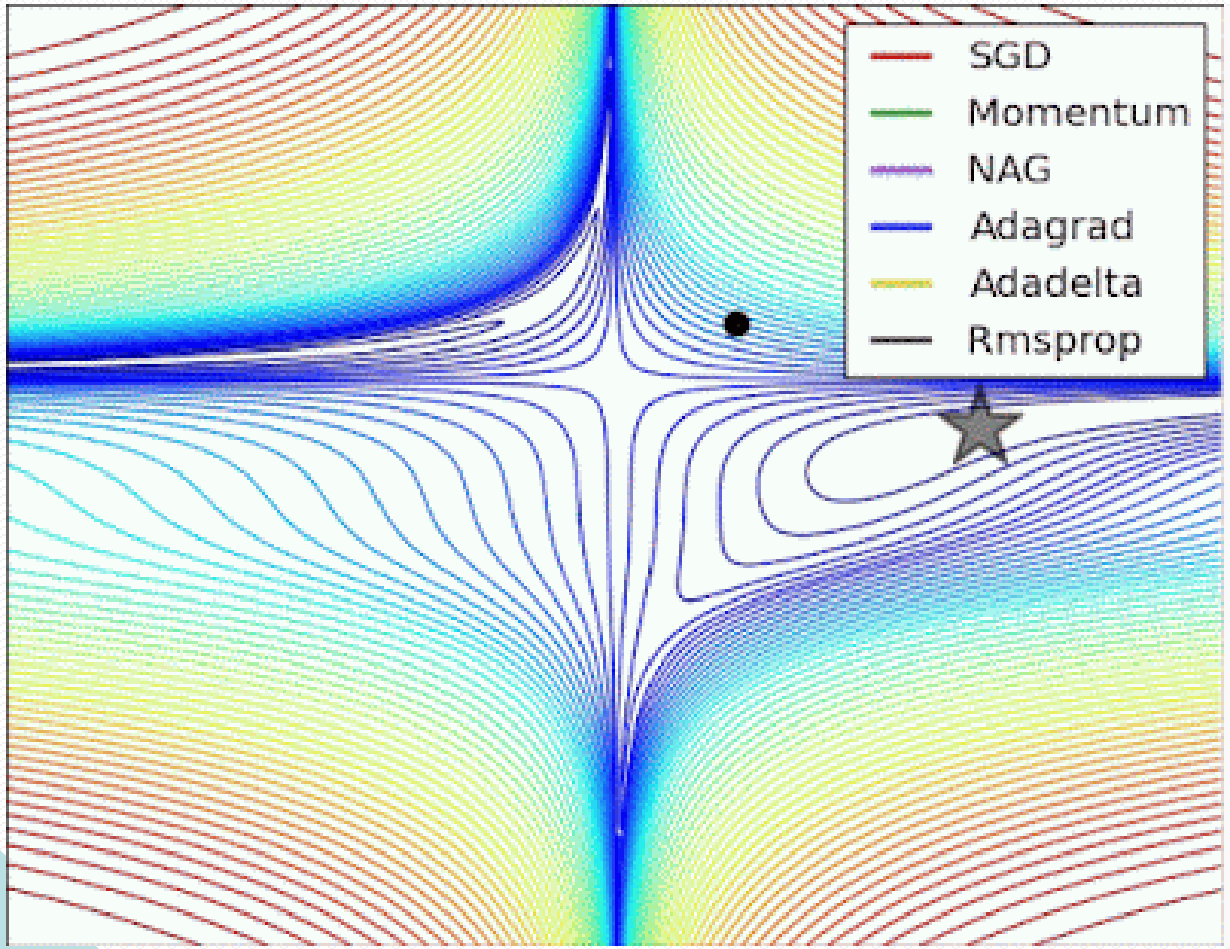




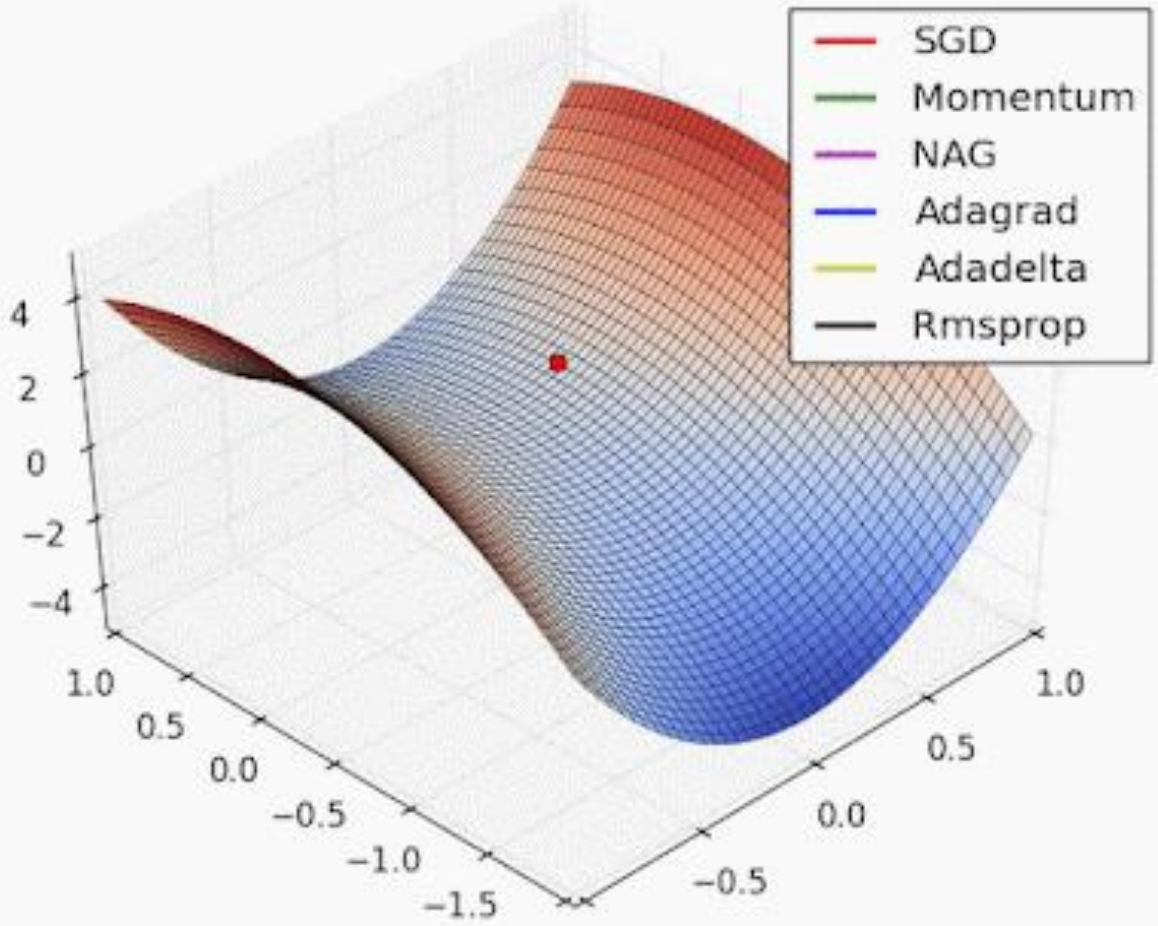




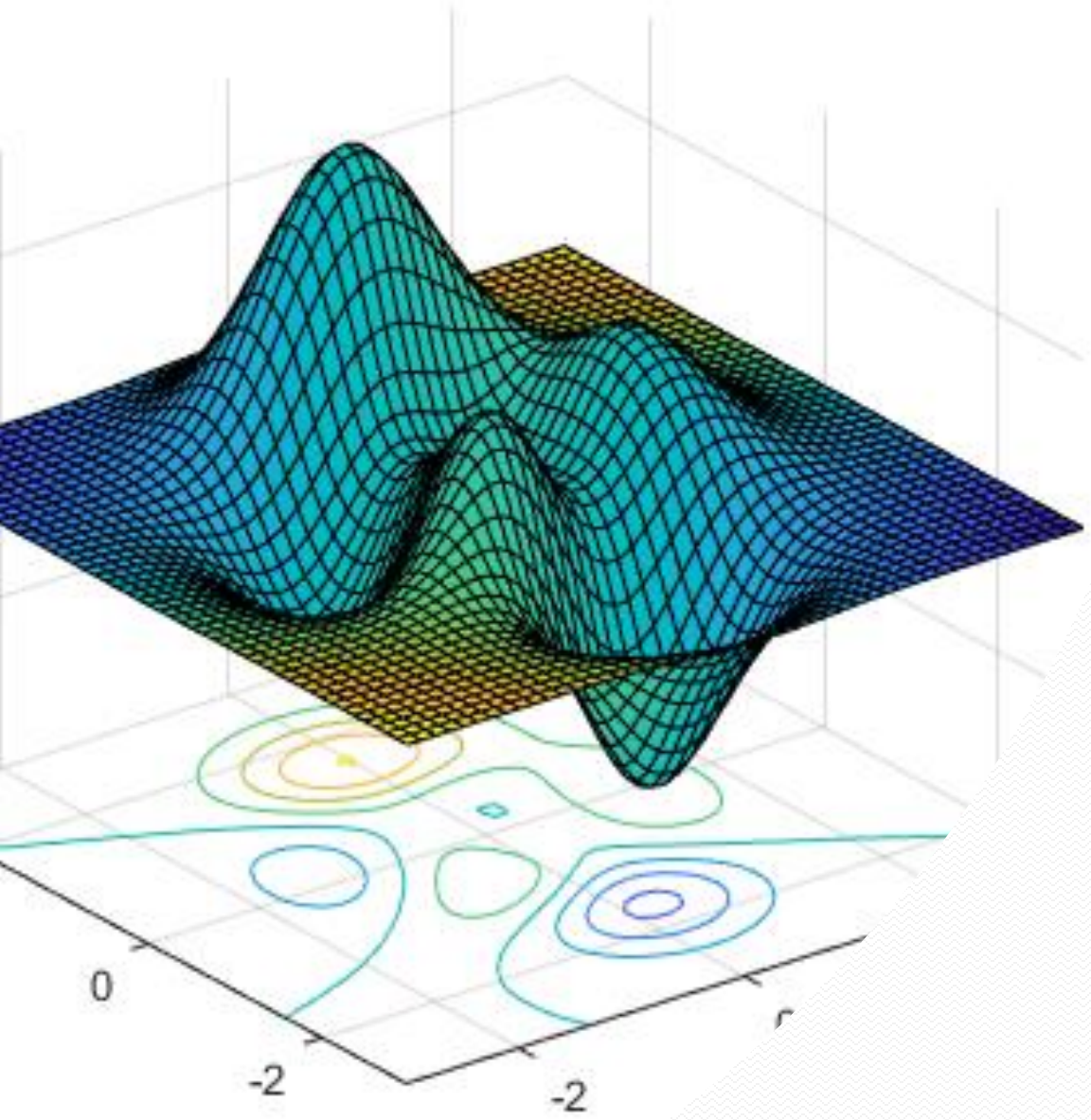
Behavior & Performance



Saddle Point

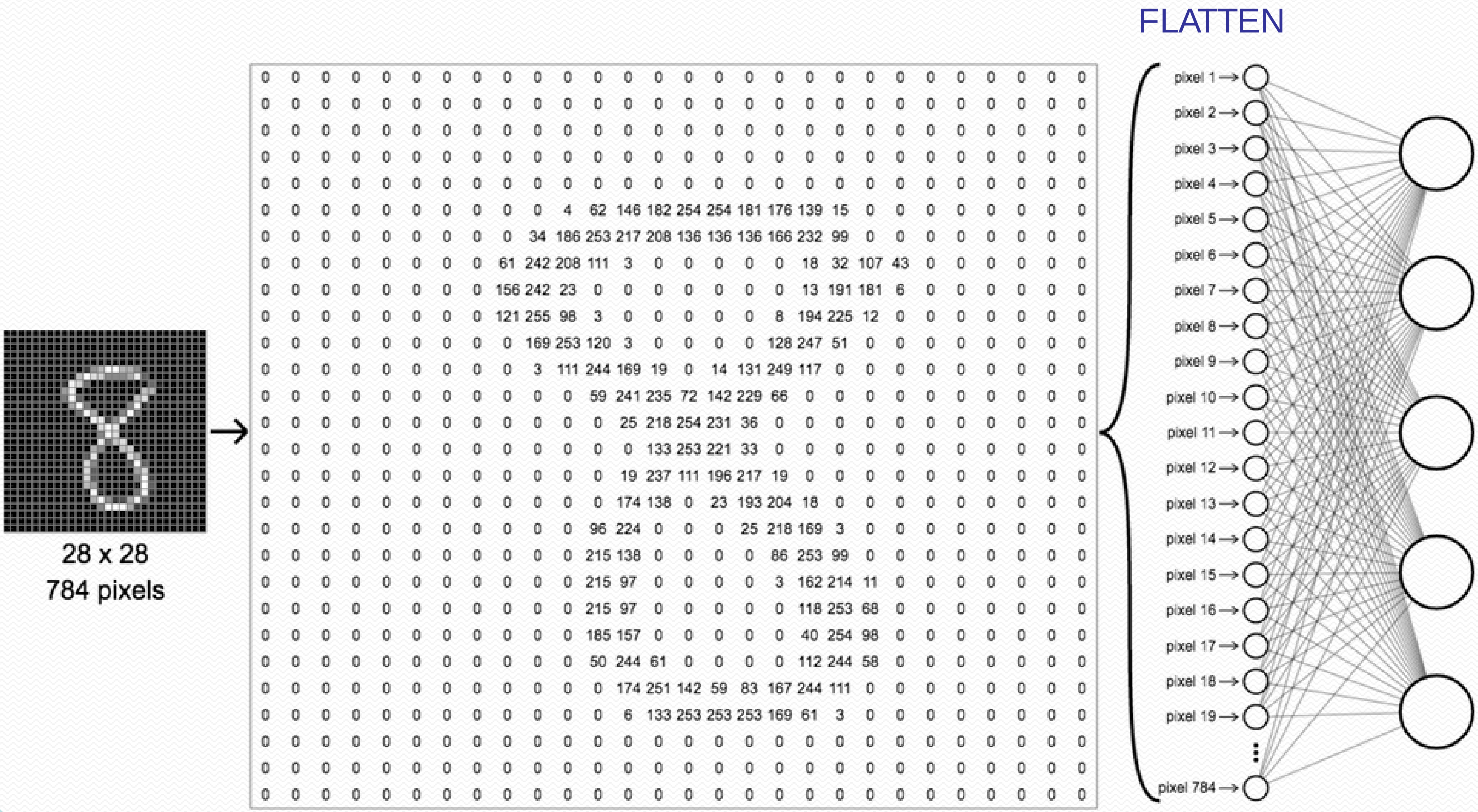


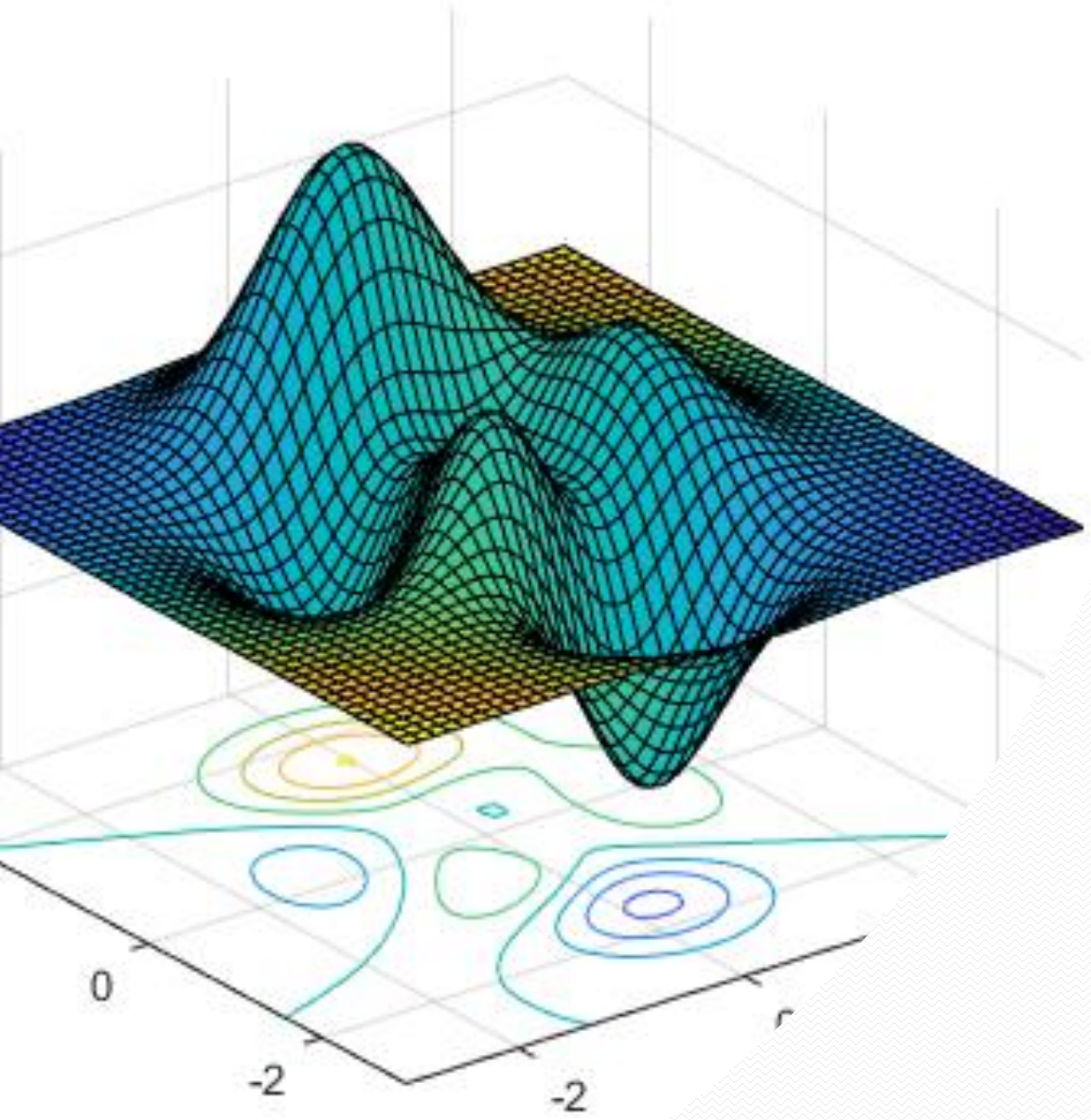
Alec Radford has created some great animations comparing optimization algorithms



Hands-on

FEED THE IMAGE DATA





Back Propagation

MLP TRAINING & INFERENCING

FEEDFORWARD

Predict

BACKPROPAGATE

Weight Update

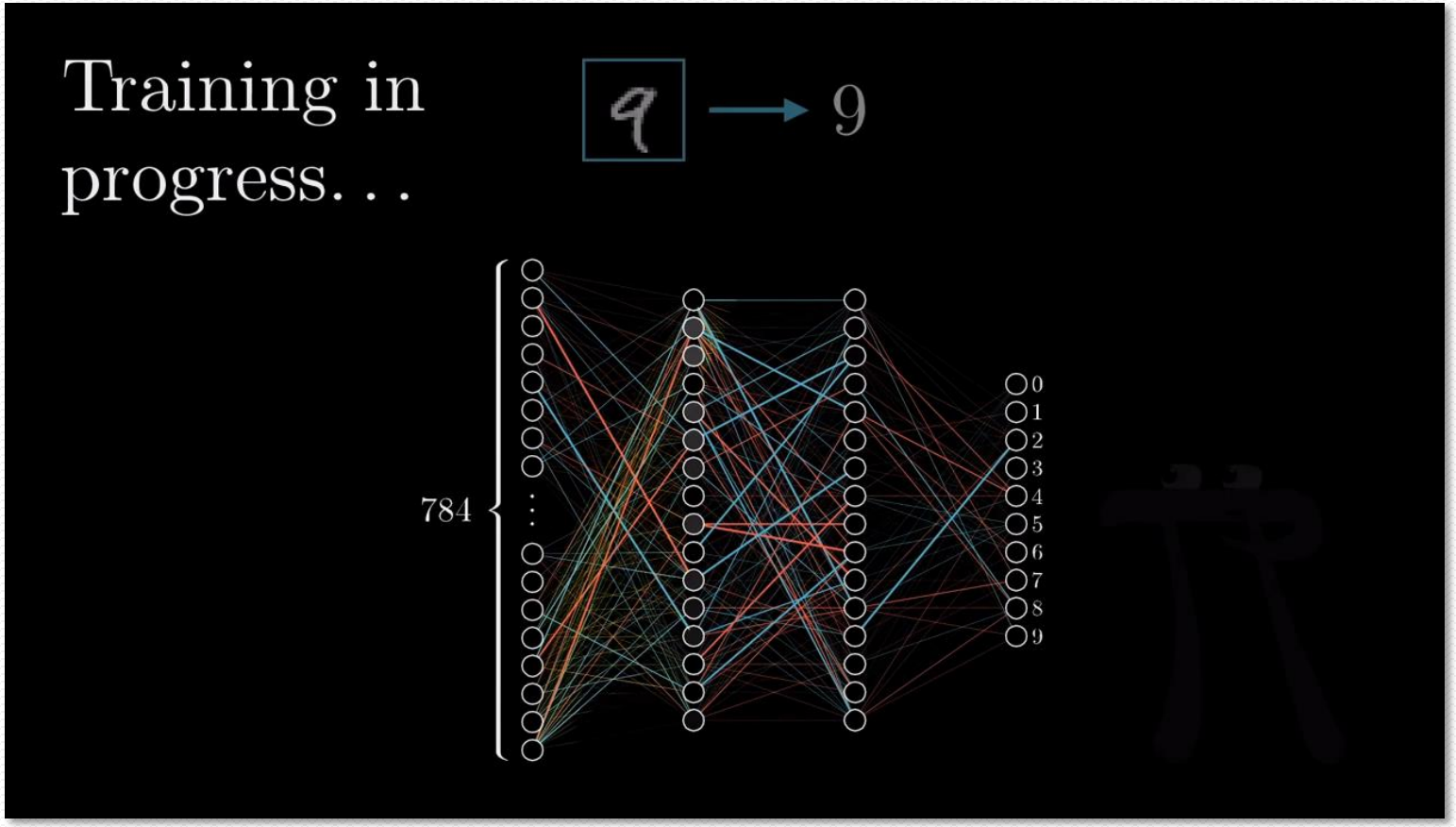
FEEDFORWARD

Predict

BACKPROPAGATE

Weight Update

...



CONTENTS

Session Break

THANK YOU

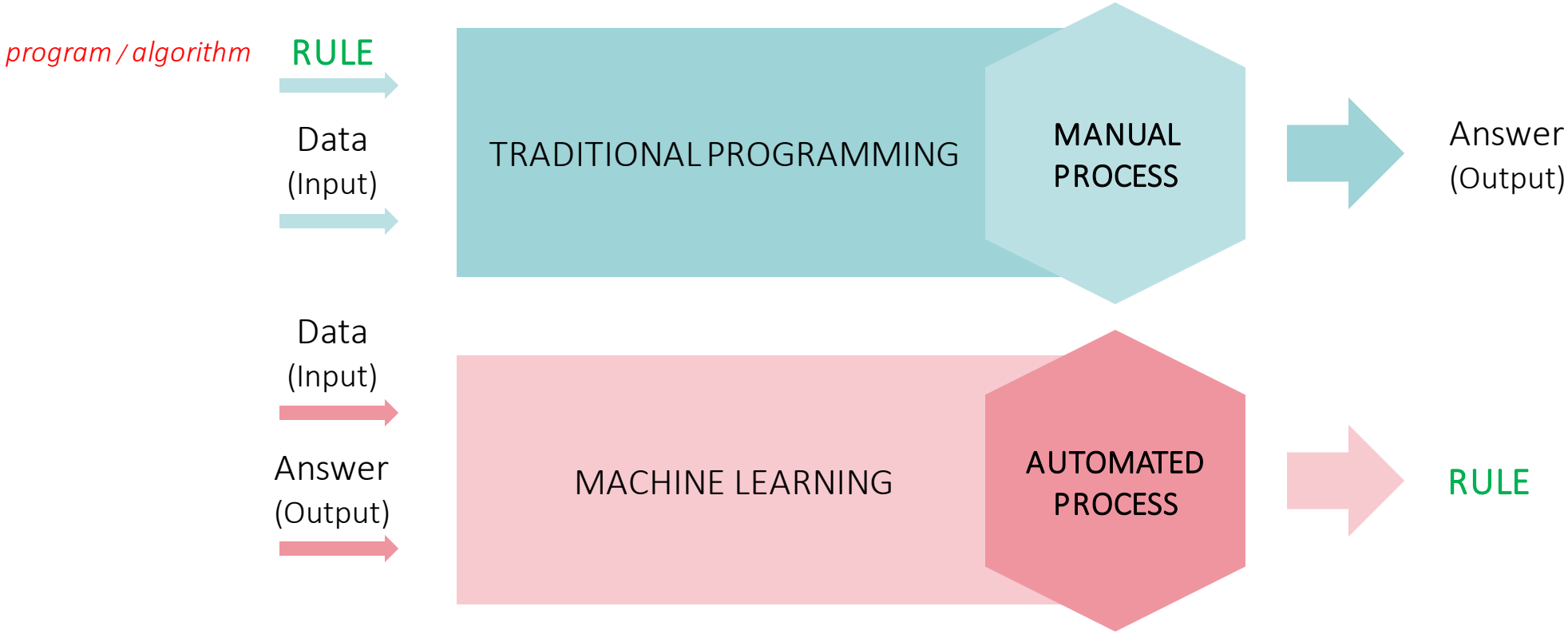
Hands-on Colab

- [Hands-on] MNIST Dataset
  - https://colab.research.google.com/drive/122uCWPt1eR7yrasyG96ak0vXKKQGqa_Y?usp=sharing
- [Hands-on] Coco Dataset
  - https://colab.research.google.com/drive/1CrOUX7Ta-phwMI_Ngj1Ay9_DtyJUezLK?usp=sharing
- [Hands-on] ANN MNIST
  - https://colab.research.google.com/drive/1_ZhB7hwtYCEtfHTwkUthEsLHx3rlehjz?usp=sharing
- [Hands-on]Benchmark app
  - https://colab.research.google.com/drive/1mF99L-U5NJ0KYjf_VY2ZkA6vhcpX7CSh?usp=sharing

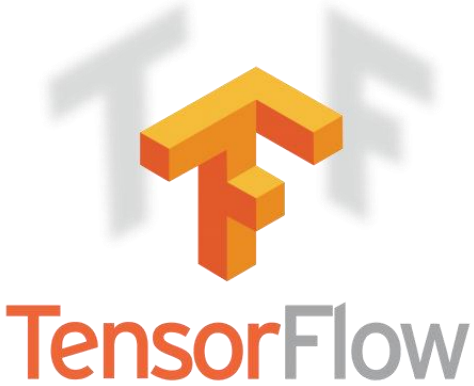
TRADITIONAL PROGRAMMING / ML

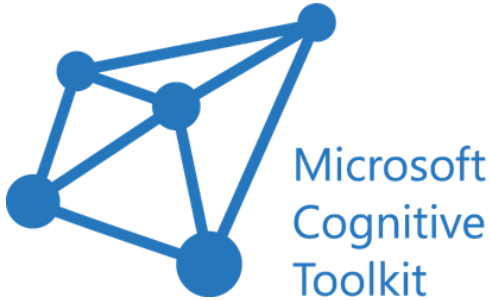
Traditional Programming refers to any manually created program

uses input data and runs on a computer to produce the output



DEEP LEARNING FRAMEWORKS





SUMMARY

AI SIMULATING LIVING BEHAVIOR

ML TRAINING MODEL FROM DATASETS

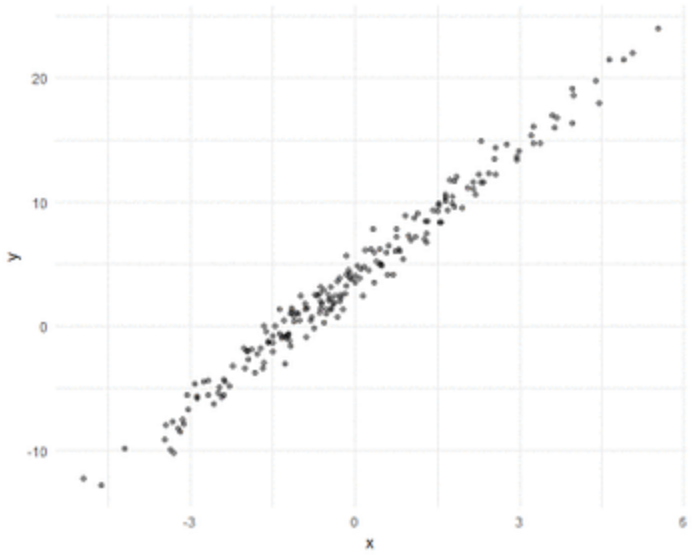
DL INSPIRED BY THE NEURON NETWORKS IN BRAIN

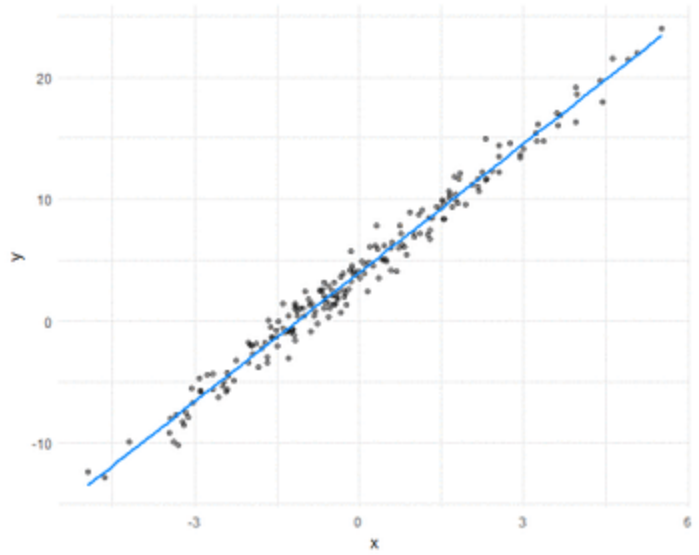
FEATURE EXTRACTION / DL

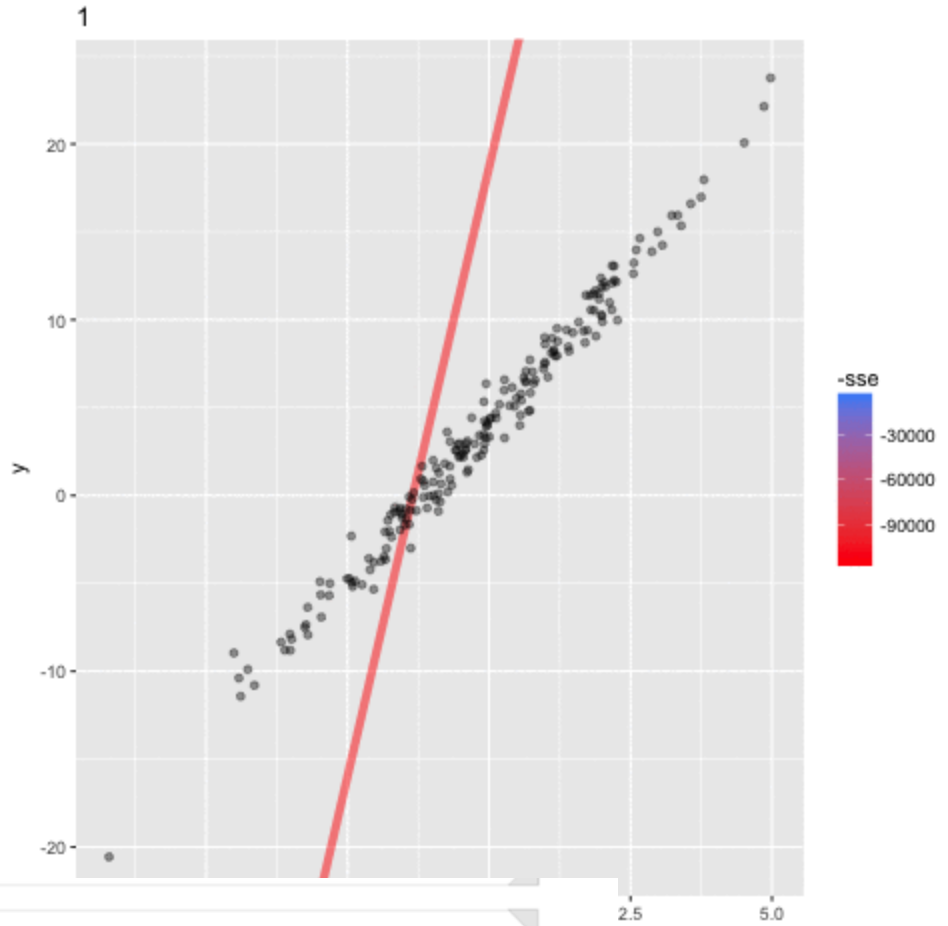
No need to manually extract features from the image

Layered feature extraction

Lowest cost







Hypothesis :

$$\hat{y} = wx + b$$

Parameters:

$$w, b$$

Cost(Loss) Function

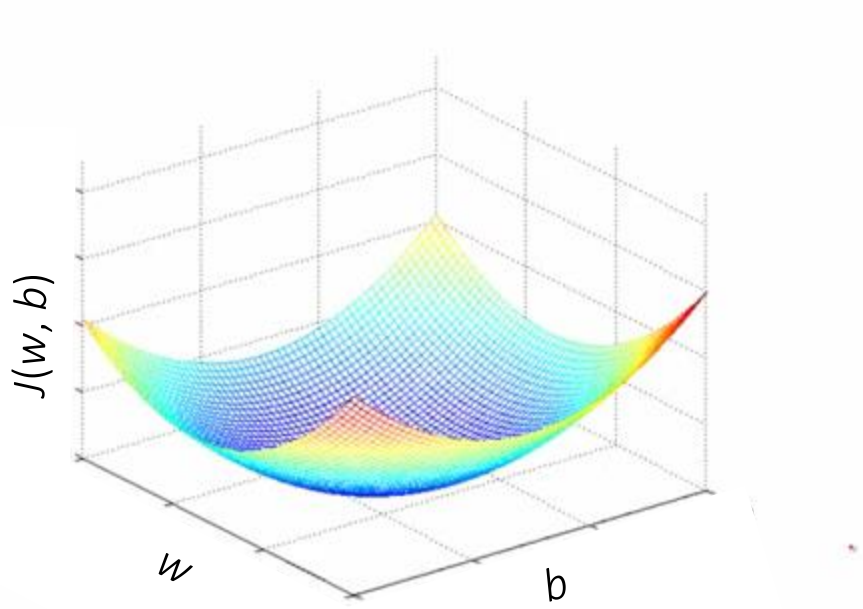
$$\text{Ccost}(w, b) = \frac{1}{2}(\hat{y} - y)^2$$

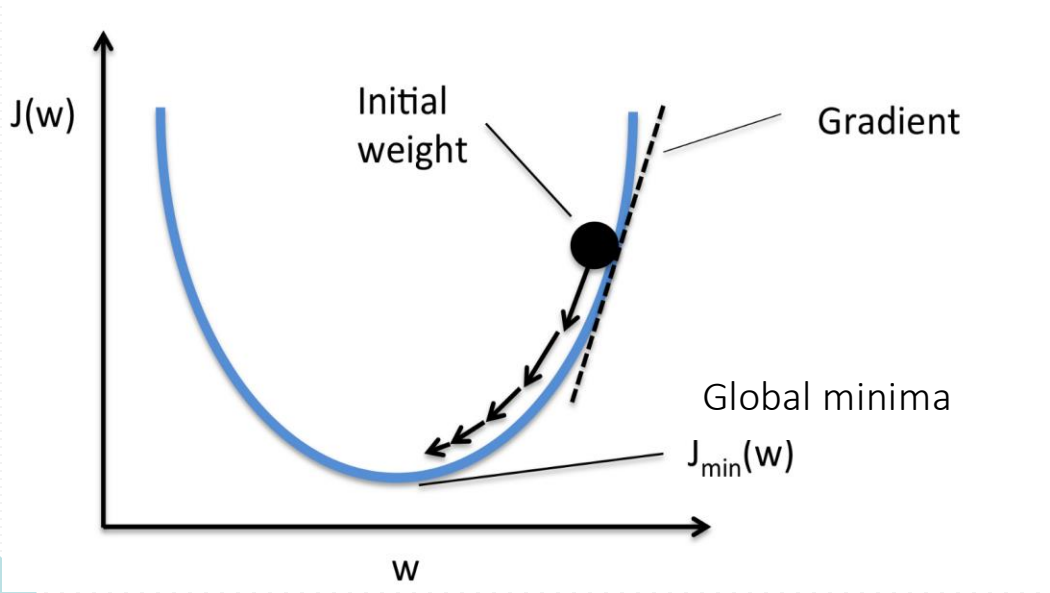
$$Cost = \frac{1}{n}\sum_{1}^{n}(H(x_n) - y_n)^2$$

- https://towardsdatascience.com/machine-learning-fundamentals-via-linear-regression-41a5d11f5220

Gradient Descent

- Example: mean squared error as loss function
- Cost function is a convex function
- **Goal**: Find values of *w* and *b* such that *J* is minimu
  i.e. find the global minima of the cost funct
- Move in the direction of negative gradient





$$w' = w - \alpha \frac{\partial J}{\partial w}$$

learning rate

$$b' = b - \alpha \frac{\partial J}{\partial b}$$