# Deep Learning

## Part1: Introduction

Ali El Hadi Ismail Fawaz,
Germain Forestier, Hassan Ismail Fawaz

ENSISA 2A Info, Université Haute-Alsace

November 14, 2024

École nationale supérieure d'ingénieurs sud Alsace

Une école d'ingénieurs de l'Université de Haute-Alsace

RÉPUBLIQUE FRANÇAISE

Liberté Égalité Fraternité

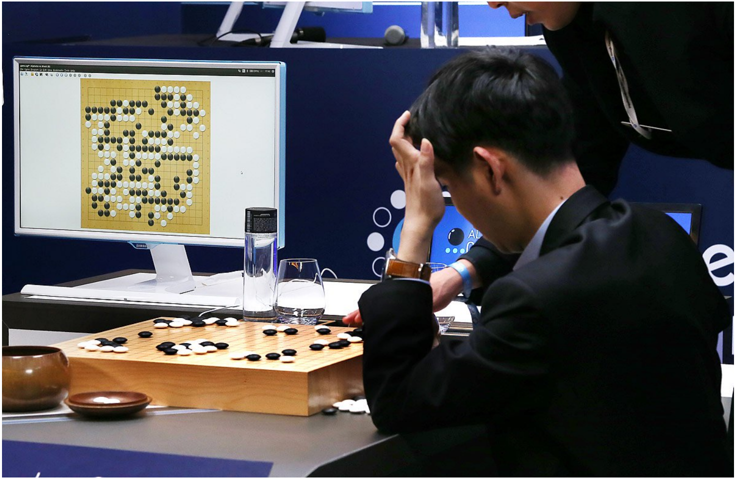
UNIVERSITÉ HAUTE-ALSACE

# Table of Contents

# Revolution of Deep Learning

# Revolution of Deep Learning

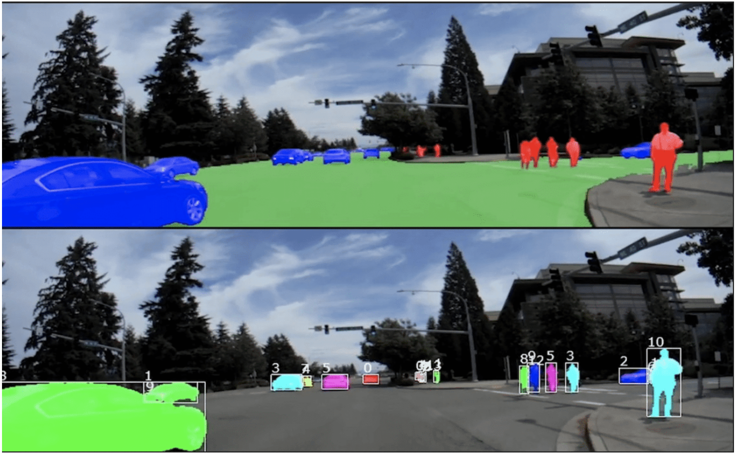
## In 2016, Google's AlphaGo Beat a Go World Champion



source: https://www.technologyreview.com/author/brenden-lake/

# Revolution of Deep Learning

## Object detection for self driving cars



source: https://neptune.ai/blog/self-driving-cars-with-convolutional-neural-networks-cnn

# Revolution of Deep Learning

## Large Language Models for AI Chat Bots


source: `https://www.linkedin.com/pulse/llm-revolution-how-ai-language-models-transforming-lives-ahmed-jawed/`

# Revolution of Deep Learning

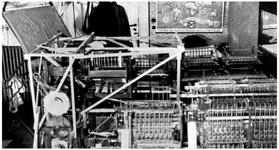
## Speech Recognition

# History of Deep Learning

# History of Deep Learning

**In the 1930s**

- Creation of the first binary programmable PC: Z1[1]

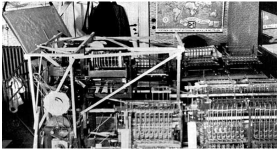


[1] https://dcmlr.inf.fu-berlin.de/rojas/index.html%3Fp=567.html

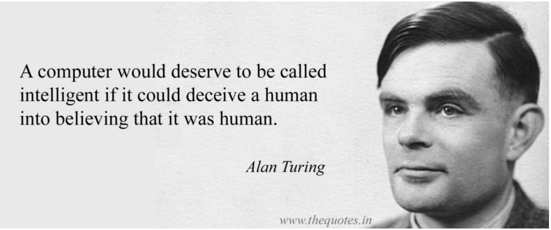
# History of Deep Learning

**In the 1930s**

- Creation of the first binary programmable PC: Z1[1]

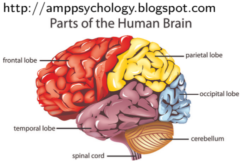


- Alan Turing asked himself : “ Can a machine really think ? “
- He did not have the machines of our time to answer this question




[1] https://dcmlr.inf.fu-berlin.de/rojas/index.html%3Fp=567.html

# History of Deep Learning

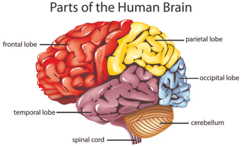



## In the 1940s

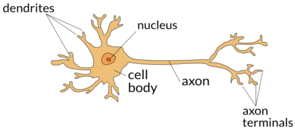
- Research in neuro-science had advanced

# History of Deep Learning

http://amppsychology.blogspot.com




https://appliedgo.net/perceptron/

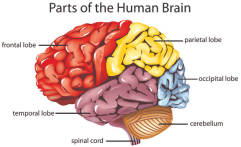



**In the 1940s**

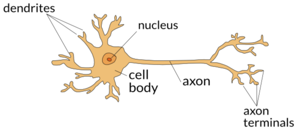
- Research in neuro-science had advanced
- We started to better understand the functionality of the brain

# History of Deep Learning

http://amppsychology.blogspot.com

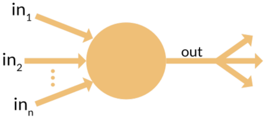



https://appliedgo.net/perceptron/



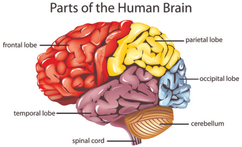

## In the 1940s

- Research in neuro-science had advanced
- We started to better understand the functionality of the brain
- Mathematical models developed to simulate the functionality of a neuron

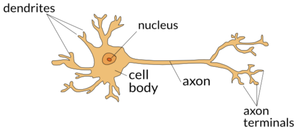


# History of Deep Learning

http://amppsychology.blogspot.com

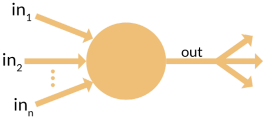


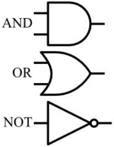

https://appliedgo.net/perceptron/




## In the 1940s

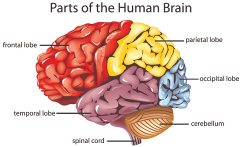
- Research in neuro-science had advanced
- We started to better understand the functionality of the brain
- Mathematical models developed to simulate the functionality of a neuron
- For example to reproduce the functions of logic gates AND OR etc.



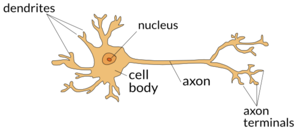



# History of Deep Learning

http://amppsychology.blogspot.com

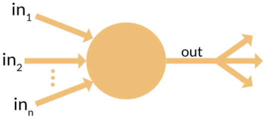


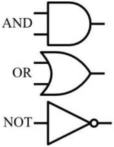

https://appliedgo.net/perceptron/




## In the 1940s

- Research in neuro-science had advanced
- We started to better understand the functionality of the brain
- Mathematical models developed to simulate the functionality of a neuron
- For example to reproduce the functions of logic gates AND OR etc.
- But still not learning: the developer should manually define the functionality of each neuron



# History of Deep Learning

**In the 1950s**

- Machines are still programmed without learning
- Instruction were well defined for them to follow

# History of Deep Learning

**In the 1950s**

- Machines are still programmed without learning
- Instruction were well defined for them to follow
- Exemple : ELIZA de MIT: one of the first chat bots
    - It was programmed in a very defined manner
    - It used techniques of word substitution
    - At first, it passed the Turing test

# History of Deep Learning

**In the 1950s**

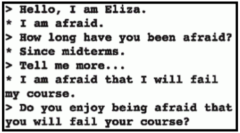
- Machines are still programmed without learning
- Instruction were well defined for them to follow
- Exemple : ELIZA de MIT: one of the first chat bots
    - It was programmed in a very defined manner
    - It used techniques of word substitution
    - At first, it passed the Turing test
    - A couple seconds later, we can notice its a machine
    - Reason to fail: the software is based on the idea that we can develop a machine for it to be intelligent

# History of Deep Learning

## In the 1950s

- Machines are still programmed without learning
- Instruction were well defined for them to follow
- Exemple : ELIZA de MIT: one of the first chat bots
  - It was programmed in a very defined manner
  - It used techniques of word substitution
  - At first, it passed the Turing test
  - A couple seconds later, we can notice its a machine
  - Reason to fail: the software is based on the idea that we can develop a machine for it to be intelligent
- Intelligence alone is not enough: we need learning !

knowledgeillusion.wordpress.com




# History of Deep Learning

**In the late 1950s and early 1960s**

- Frank Rosenblatt developed a physical perceptron
- Learn how to differentiate shapes like rectangles squares etc.
- The machine was able to learn alone how to solve this task

# History of Deep Learning

**In the late 1950s and early 1960s**

- Frank Rosenblatt developed a physical perceptron
- Learn how to differentiate shapes like rectangles squares etc.
- The machine was able to learn alone how to solve this task


- Instead of manual instructions, the machine learns from examples
- The first non-deep neural network able to learn and adapt
- People started to have an interest in neural network

# History of Deep Learning

**In the late 1950s and early 1960s**

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI)—The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

**Without Human Controls**

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

**Learns by Doing**

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

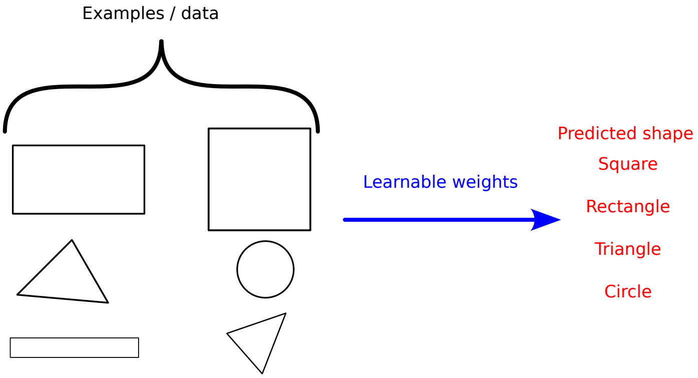
Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

source : New York Times Article, 1958

# History of Deep Learning

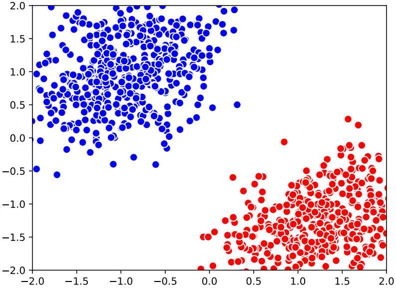
**In the late 1950s and early 1960s**



# History of Deep Learning

**In year 1969**

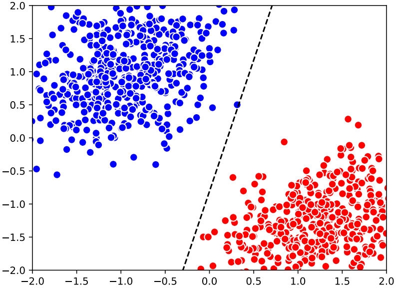
- Minsky and Papert showed that the Perceptron is not capable of learning non-linear functions with the optimization algorithms available at that time.



# History of Deep Learning

**In year 1969**

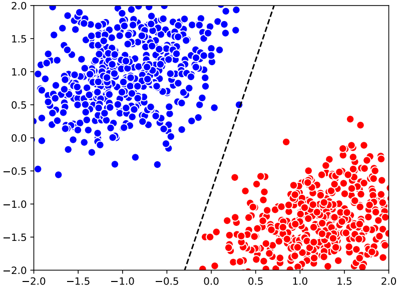
- Minsky and Papert showed that the Perceptron is not capable of learning non-linear functions with the optimization algorithms available at that time.

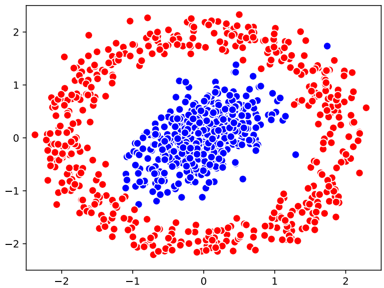


# History of Deep Learning

**In year 1969**

- Minsky and Papert showed that the Perceptron is not capable of learning non-linear functions with the optimization algorithms available at that time.

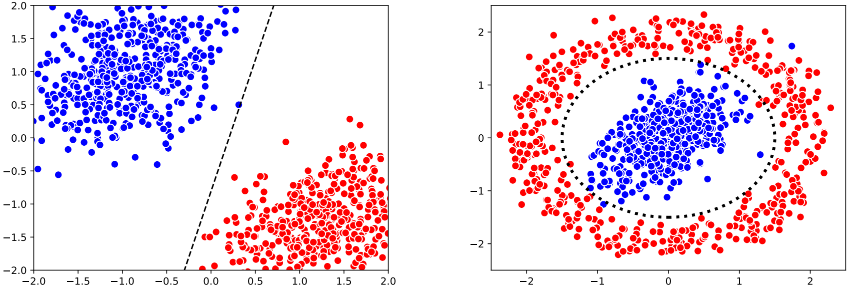


# History of Deep Learning

## In year 1969

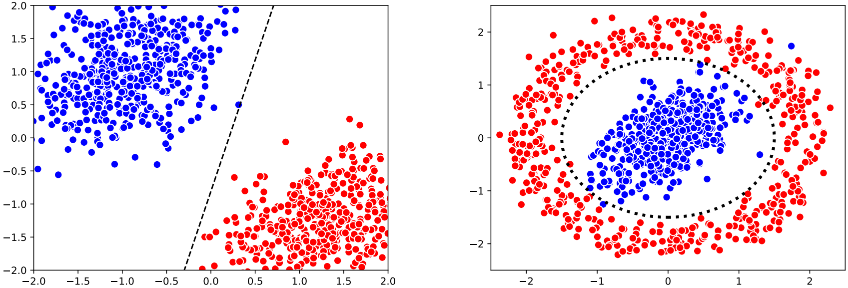
- Minsky and Papert showed that the Perceptron is not capable of learning non-linear functions with the optimization algorithms available at that time.



# History of Deep Learning

**In year 1969**

- Minsky and Papert showed that the Perceptron is not capable of learning non-linear functions with the optimization algorithms available at that time.




- We need a Multi Layer Perceptron (MLP) to solve the non-linear separable points
- Issue: Back then, no algorithm could solve the MLP

# History of Deep Learning

**In the 1970s**

- The learning problem intensifies with Minsky and Papert
- We were unable to make the machine learn

# History of Deep Learning

**In the 1970s**

- The learning problem intensifies with Minsky and Papert
- We were unable to make the machine learn
- We did not have the algorithms or the resources at that time
- The research in AI is starting to decrease and retreat

# History of Deep Learning

**In the 1970s**

- The learning problem intensifies with Minsky and Papert
- We were unable to make the machine learn
- We did not have the algorithms or the resources at that time
- The research in AI is starting to decrease and retreat
- We were receiving less and less funding for AI
- A period known as AI Winter

# History of Deep Learning

**In the 1980s**

- In 1974, Werbos, during his thesis, proposed the algorithm of **backpropagation**
- This solves the problem discovered by Minsky and Papert

[2] https://www.youtube.com/watch?v=Ilg3gGewQ5U

# History of Deep Learning

**In the 1980s**

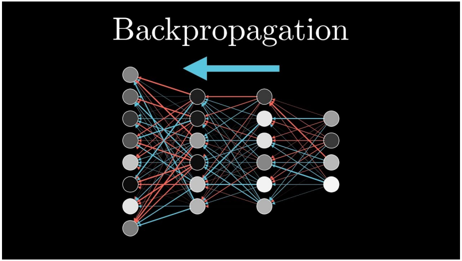
- In 1974, Werbos, during his thesis, proposed the algorithm of **backpropagation**
- This solves the problem discovered by Minsky and Papert
- This algorithm is still used in Deep Learning to this day
- Due to the AI winter, the algorithm was only published in 1982

[2] https://www.youtube.com/watch?v=Ilg3gGewQ5U

# History of Deep Learning

**In the 1980s**

- In 1974, Werbos, during his thesis, proposed the algorithm of **backpropagation**
- This solves the problem discovered by Minsky and Papert
- This algorithm is still used in Deep Learning to this day
- Due to the AI winter, the algorithm was only published in 1982
- In 1986, Rumelhart, Hinton, and Williams rediscovered the algorithm
- With this second publication, the algorithm gained popularity [2]




[2] https://www.youtube.com/watch?v=Ilg3gGewQ5U

# History of Deep Learning

**In the 1990s**

- Yann LeCun proposed Convolutional Neural Networks
- Convolutions gave networks the ability to see
- Early industrial applications of Deep Learning
- 10% of checks and postal codes are read by CNNs

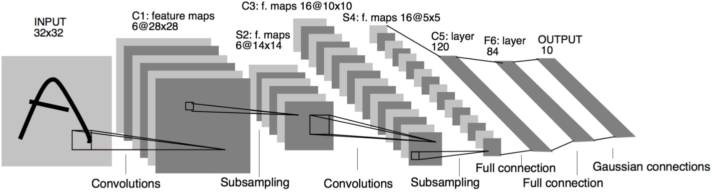
[3] Gradient-Based Learning Applied to Document Recognition

# History of Deep Learning

## In the 1990s

- Yann LeCun proposed Convolutional Neural Networks
- Convolutions gave networks the ability to see
- Early industrial applications of Deep Learning
- 10% of checks and postal codes are read by CNNs

LeNet-5 Architecture [3]




[3] Gradient-Based Learning Applied to Document Recognition

# History of Deep Learning

**In the 2000s**

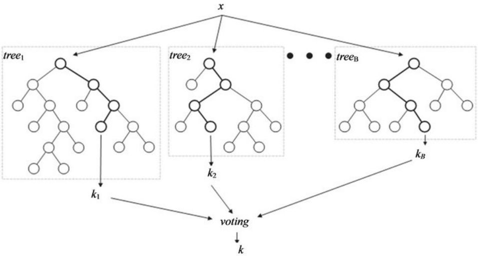
- A dip in neural networks
- We couldn't train on complex data

[4] Galvanize: What Counting Jelly Beans Can Teach Us About Machine Learning

# History of Deep Learning

**In the 2000s**

- A dip in neural networks
- We couldn't train on complex data
- Other learning algorithms gained more popularity[4]
  - Do not require enormous computational power
  - Do not need too much data to converge
  - Examples include SVMs and decision trees

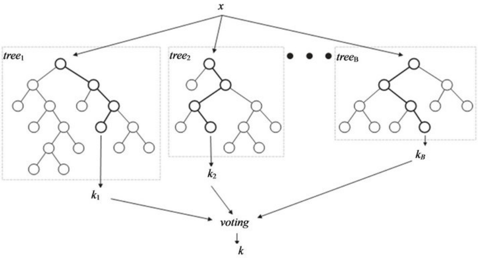



4 Galvanize: What Counting Jelly Beans Can Teach Us About Machine Learning

# History of Deep Learning

**In the 2000s**

- A dip in neural networks
- We couldn't train on complex data
- Other learning algorithms gained more popularity[4]
    - Do not require enormous computational power
    - Do not need too much data to converge
    - Examples include SVMs and decision trees
- In 2006, development of autoencoders, but still without significant real-world applications




4 Galvanize: What Counting Jelly Beans Can Teach Us About Machine Learning

# History of Deep Learning

**In the 2010s**

- In 2012, a deep neural network revolutionized computer vision
- AlexNet demonstrated exceptional performance in ImageNet

# History of Deep Learning

**In the 2010s**

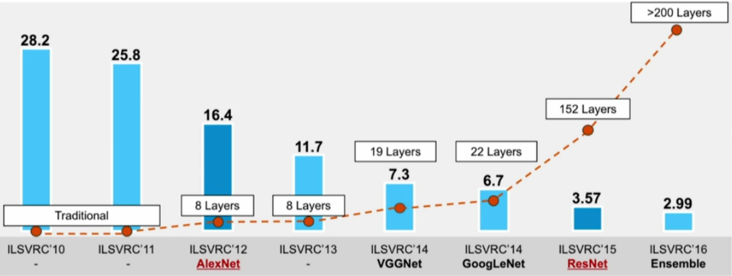
- In 2012, a deep neural network revolutionized computer vision
- AlexNet demonstrated exceptional performance in ImageNet
- It's worth noting that AlexNet doesn't differ too much from CNNs of 1998
- Since that year, many ideas and applications have emerged

# History of Deep Learning

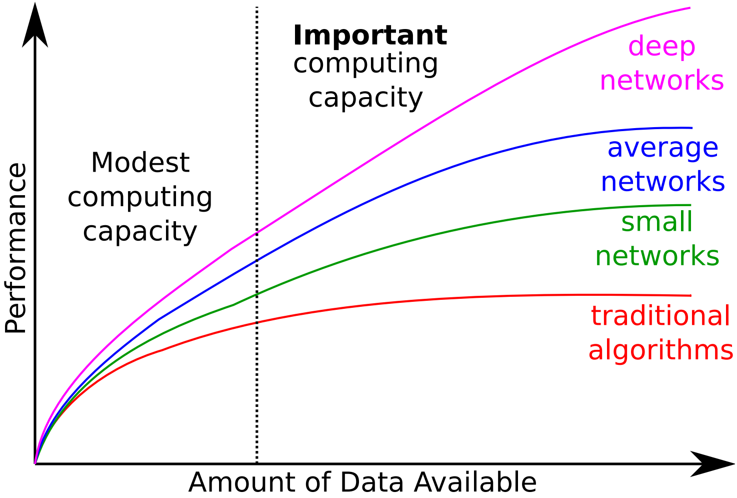
**In the 2010s**

- In 2012, a deep neural network revolutionized computer vision
- AlexNet demonstrated exceptional performance in ImageNet
- It's worth noting that AlexNet doesn't differ too much from CNNs of 1998
- Since that year, many ideas and applications have emerged

Error rate on ImageNet



# Why now ?


Performance
Amount of Data Available
Modest computing capacity
**Important** computing capacity
deep networks
average networks
small networks
traditional algorithms

# Supervised Learning

# Supervised Learning

**There exists two types of machine learning**

- Unsupervised learning (no existence of classes for each example)
- Supervised learning (association of a class label for each example)

**Two known tasks on supervised learning**

- Regression (predict a continuous value like temperature, pressure etc.)
- Classification (predict a discrete value, class label, like categorizing cats and dogs images etc.)

# Supervised Learning

**There exists two types of machine learning**

- Unsupervised learning (no existence of classes for each example)
- Supervised learning (association of a class label for each example)

**Two known tasks on supervised learning**

- Regression (predict a continuous value like temperature, pressure etc.)
- Classification (predict a discrete value, class label, like categorizing cats and dogs images etc.)

# Supervised Learning

**Use case example:**

# Supervised Learning

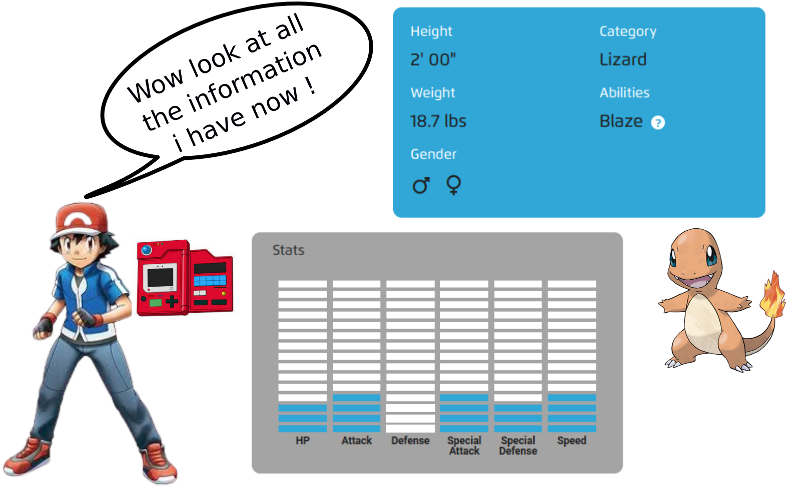
**Use case example:**


# Supervised Learning

**Use case example:**

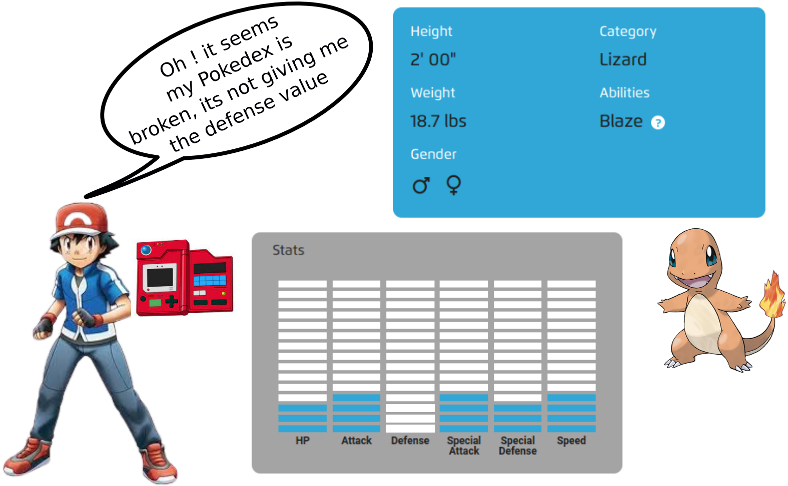

# Supervised Learning

**Use case example:** https://www.pokemon.com



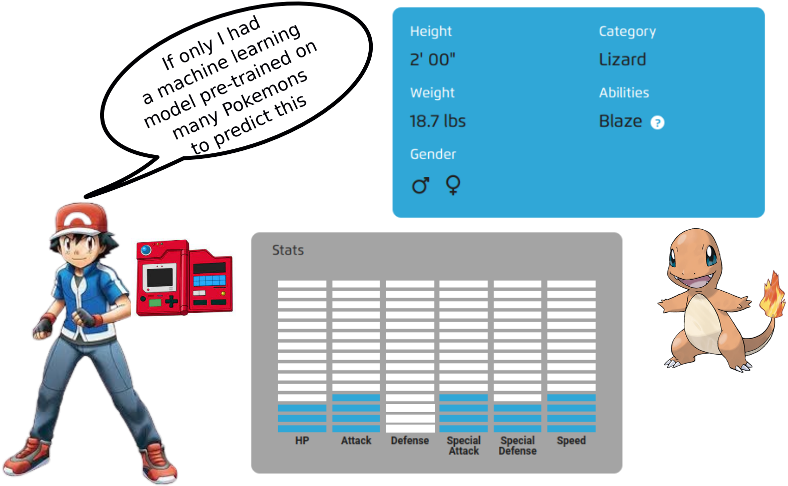
# Supervised Learning

**Use case example:**



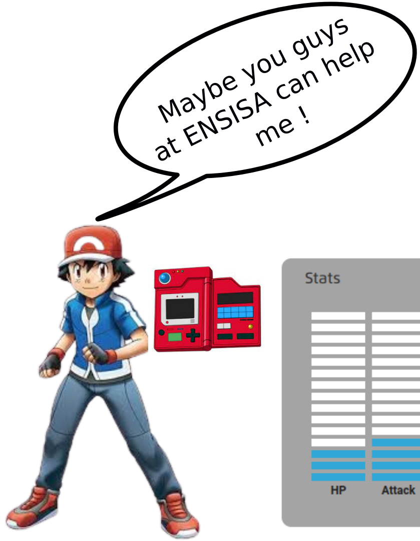
# Supervised Learning

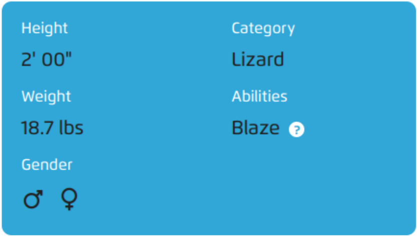
**Use case example:**

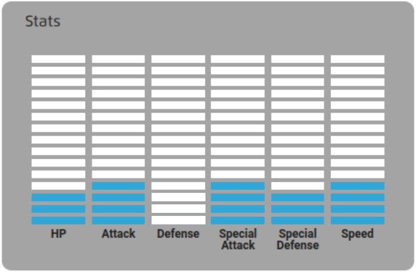


# Supervised Learning

**Use case example:**




# Supervised Learning

**Example on Pokemon stats data**[5]

| name | weight_kg | speed | sp_attack | sp_defense | type |
|---|---|---|---|---|---|
| Bulbasaur | 6.9 | 45 | 65 | 65 | grass |
| Charmander | 8.5 | 65 | 60 | 50 | fire |
| Butterfree | 32 | 70 | 90 | 80 | bug |
| Squirtle | 9 | 43 | 50 | 64 | water |

[5] https://www.kaggle.com/datasets/hasanarcas/pokemon-stats-dataset/data

# Supervised Learning

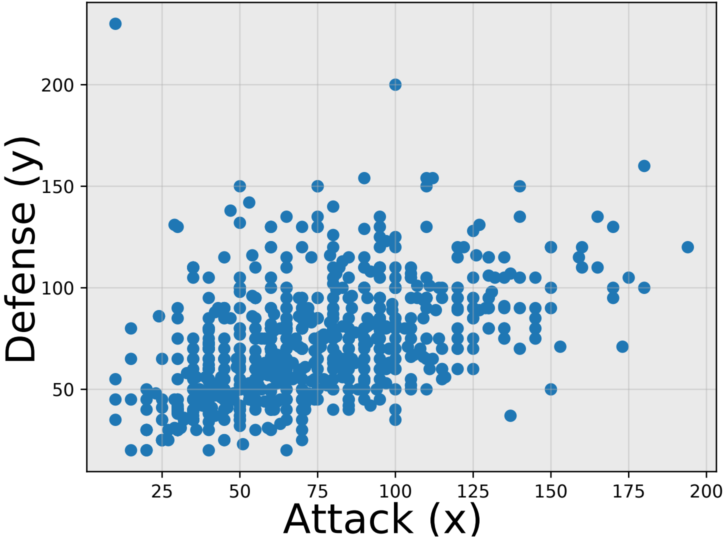
**Example on Pokemon stats data**[5]

| name | weight_kg | speed | sp_attack | sp_defense | type |
|---|---|---|---|---|---|
| Bulbasaur | 6.9 | 45 | 65 | 65 | grass |
| Charmander | 8.5 | 65 | 60 | 50 | fire |
| Butterfree | 32 | 70 | 90 | 80 | bug |
| Squirtle | 9 | 43 | 50 | 64 | water |

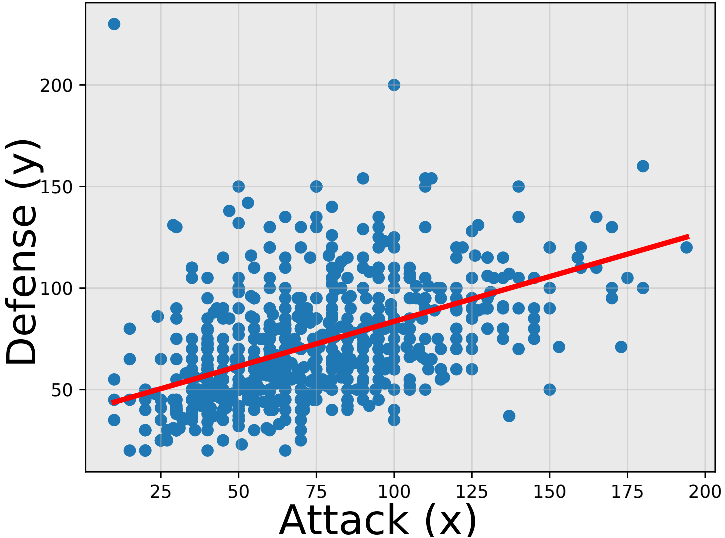
Predicting the Defense value of a Pokemon in function of its Attack value.

[5] https://www.kaggle.com/datasets/hasanarcas/pokemon-stats-dataset/data

# Supervised Learning



# Supervised Learning



# Supervised Learning

**Linear Model**

- We predict $y$ (defense) in function of $x$ (attack)

# Supervised Learning

**Linear Model**

- We predict $y$ (defense) in function of $x$ (attack)
- Linear $\implies$ straight line $\implies$ of the form: $y = w.x + b$

# Supervised Learning

**Linear Model**

- We predict $y$ (defense) in function of $x$ (attack)
- Linear $\implies$ straight line $\implies$ of the form: $y = w.x + b$
- Unbiased version: consider $b = 0 \implies y = w.x$

# Supervised Learning

## Linear Model

- We predict $y$ (defense) in function of $x$ (attack)
- Linear $\implies$ straight line $\implies$ of the form: $y = w.x + b$
- Unbiased version: consider $b = 0 \implies y = w.x$
- How to determine $w$ ?

# Supervised Learning

**Linear Model**

- We predict $y$ (defense) in function of $x$ (attack)
- Linear $\implies$ straight line $\implies$ of the form: $y = w.x + b$
- Unbiased version: consider $b = 0 \implies y = w.x$
- How to determine $w$ ?
- Many approaches exist

# Supervised Learning

**Linear Model**

- We predict $y$ (defense) in function of $x$ (attack)
- Linear $\implies$ straight line $\implies$ of the form: $y = w.x + b$
- Unbiased version: consider $b = 0 \implies y = w.x$
- How to determine $w$ ?
- Many approaches exist
  - Optimal solution

# Supervised Learning

**Linear Model**

- We predict $y$ (defense) in function of $x$ (attack)
- Linear $\implies$ straight line $\implies$ of the form: $y = w.x + b$
- Unbiased version: consider $b = 0 \implies y = w.x$
- How to determine $w$ ?
- Many approaches exist
  - Optimal solution
  - Gradient Descent method

# Supervised Learning

**Linear Model**

- We predict $y$ (defense) in function of $x$ (attack)
- Linear $\implies$ straight line $\implies$ of the form: $y = w.x + b$
- Unbiased version: consider $b = 0 \implies y = w.x$
- How to determine $w$ ?
- Many approaches exist
  - Optimal solution
  - Gradient Descent method
- For today's session, we will try the brute force method:

# Supervised Learning

**Linear Model**

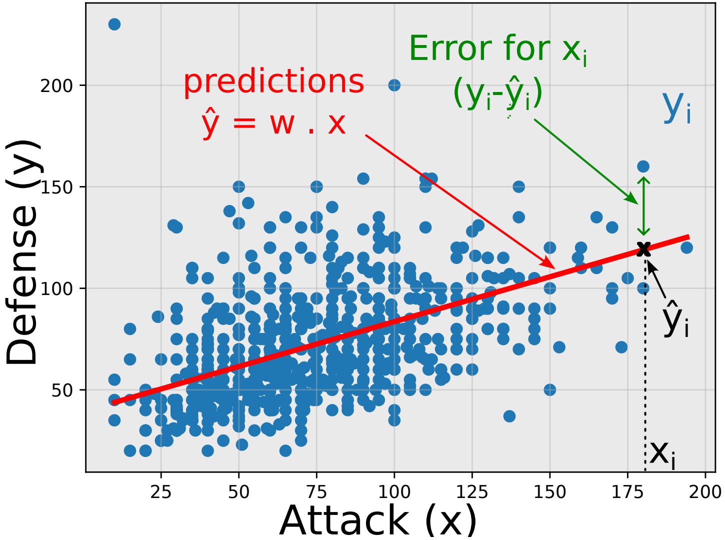
- We predict $y$ (defense) in function of $x$ (attack)
- Linear $\implies$ straight line $\implies$ of the form: $y = w.x + b$
- Unbiased version: consider $b = 0 \implies y = w.x$
- How to determine $w$ ?
- Many approaches exist
  - Optimal solution
  - Gradient Descent method
- For today's session, we will try the brute force method:
  - We try many possible values of $w$ until we find the best one

# Supervised Learning

**Linear Model**

- We predict $y$ (defense) in function of $x$ (attack)
- Linear $\implies$ straight line $\implies$ of the form: $y = w.x + b$
- Unbiased version: consider $b = 0 \implies y = w.x$
- How to determine $w$ ?
- Many approaches exist
  - Optimal solution
  - Gradient Descent method
- For today's session, we will try the brute force method:
  - We try many possible values of $w$ until we find the best one
  - Not an efficient method given the high running time

# Supervised Learning



# Supervised Learning

**How to evaluate the choice of $w$ ?**

- Notations:
    - $x$ is the attack value of a Pokemon
    - $y$ is the ground truth defense value of a Pokemon
    - $\hat{y}$ is the predicted defense value of a Pokemon using the model $\hat{y} = w.x$
    - $L$ is the cost function of the model
    - $n$ is the number of examples the dataset provides

# Supervised Learning

**How to evaluate the choice of $w$ ?**

- Notations:
  - $x$ is the attack value of a Pokemon
  - $y$ is the ground truth defense value of a Pokemon
  - $\hat{y}$ is the predicted defense value of a Pokemon using the model $\hat{y} = w.x$
  - $L$ is the cost function of the model
  - $n$ is the number of examples the dataset provides
- Cost function that measures the error in the model's predictions:

$$L(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i(w))^2 \tag{1}$$

# Supervised Learning

**How to evaluate the choice of $w$ ?**

- Notations:
  - $x$ is the attack value of a Pokemon
  - $y$ is the ground truth defense value of a Pokemon
  - $\hat{y}$ is the predicted defense value of a Pokemon using the model $\hat{y} = w.x$
  - $L$ is the cost function of the model
  - $n$ is the number of examples the dataset provides
- Cost function that measures the error in the model's predictions:

$$L(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i(w))^2 \tag{1}$$

- So the goal is to find the value of $w$ that minimizes the cost function $L$
- We find $w_{best}$ that minimizes the error $L \implies$ maximize precision

# Supervised Learning