Introduction to Machine Learning

Fabien Baradel
PhD Student - INSA Lyon
fabienbaradel.github.io

What is your definition of Machine Learning ?

Definition: Machine Learning

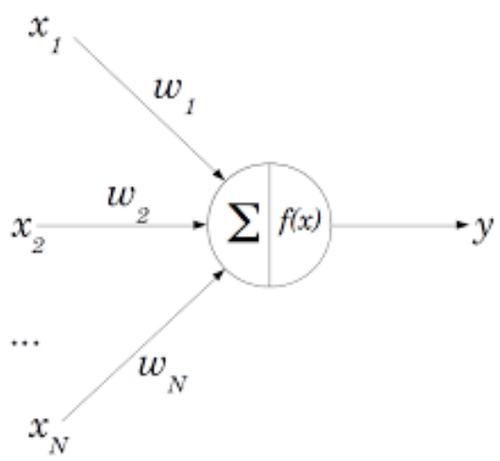
Statictics?
Maths?
Computer Science?
Big Data?
Artificial Intelligence?

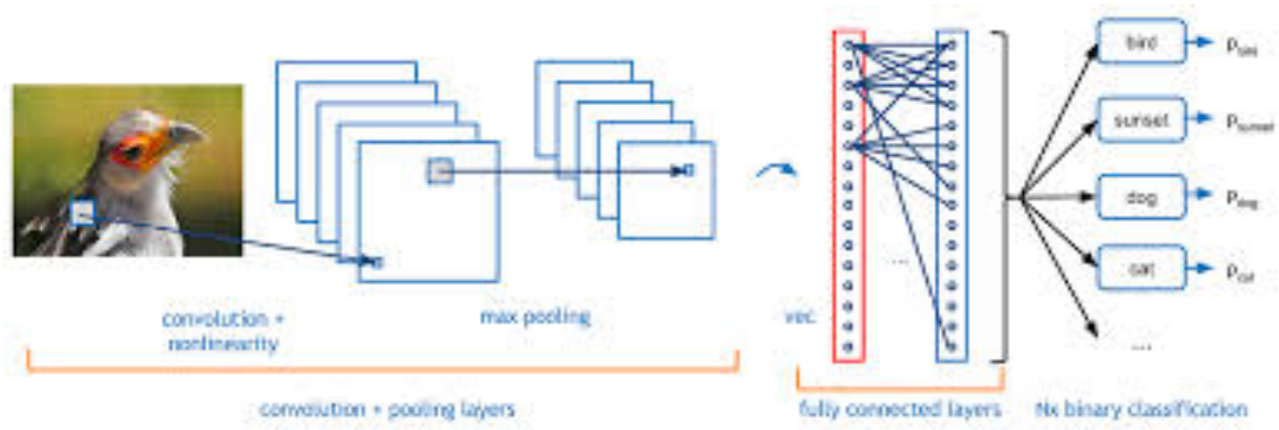
« Field of study that gives computers the ability to learn without being explicitly programmed »
Arthur Samuel, 1959

Automatic discover of patterns in data by a computer

Different from rule-based methods

Brief history of Machine Learning and AI





1950: Turing test

1952: 1st computer learning program by Arthur Samuel

1957: Perceptron by Frank Rosenblatt

1957: Nearest neighbor algorithm

1981: Gerald Dejong introduces concept of Explanation Based Learning

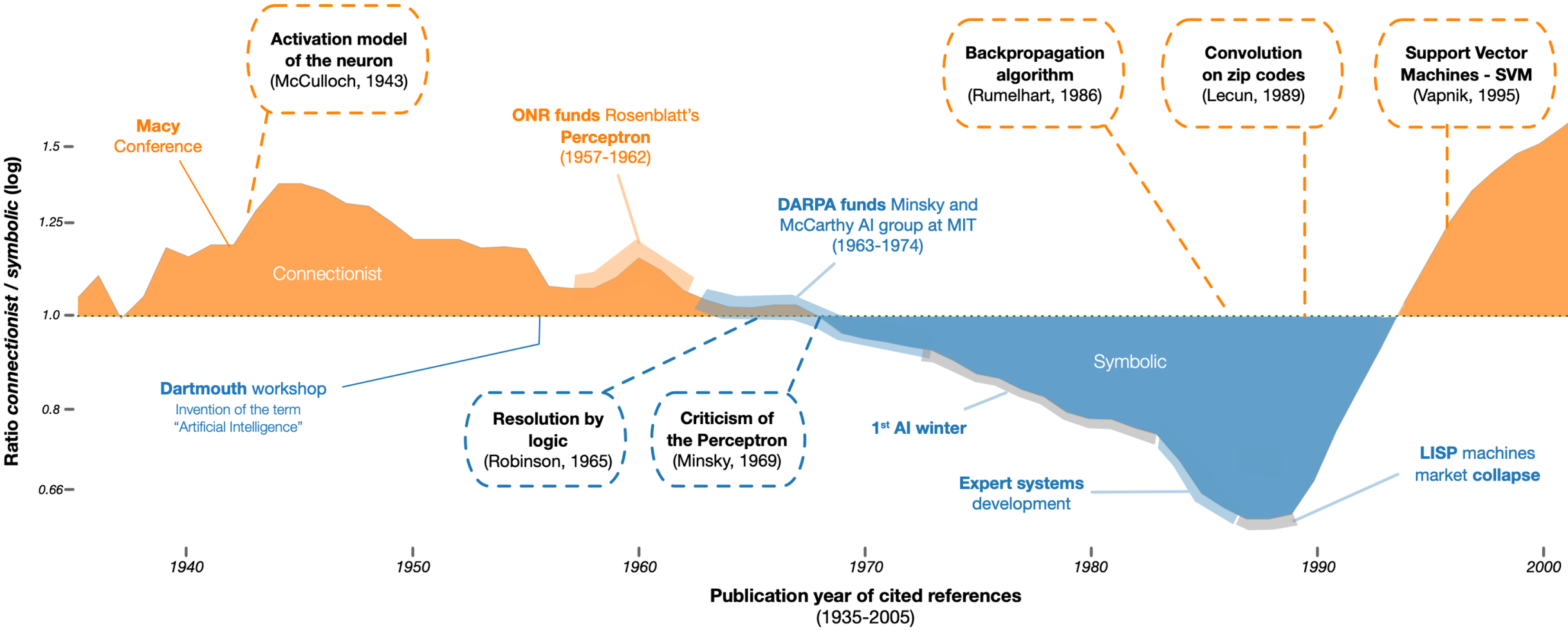
1990: ML shifts from knowledge-driven to data driven

1997: IBM Deep Blue

2006: Geoffrey Hinton coins the term « deep learning »

2012: AlexNet won Imagenet with CNN

Timeline AI



https://neurovenge.antonomase.fr

What do we want to learn?

Supervised Learning

Object classification

$f(.)$ → cat

Human Pose estimation

Unsupervised Learning

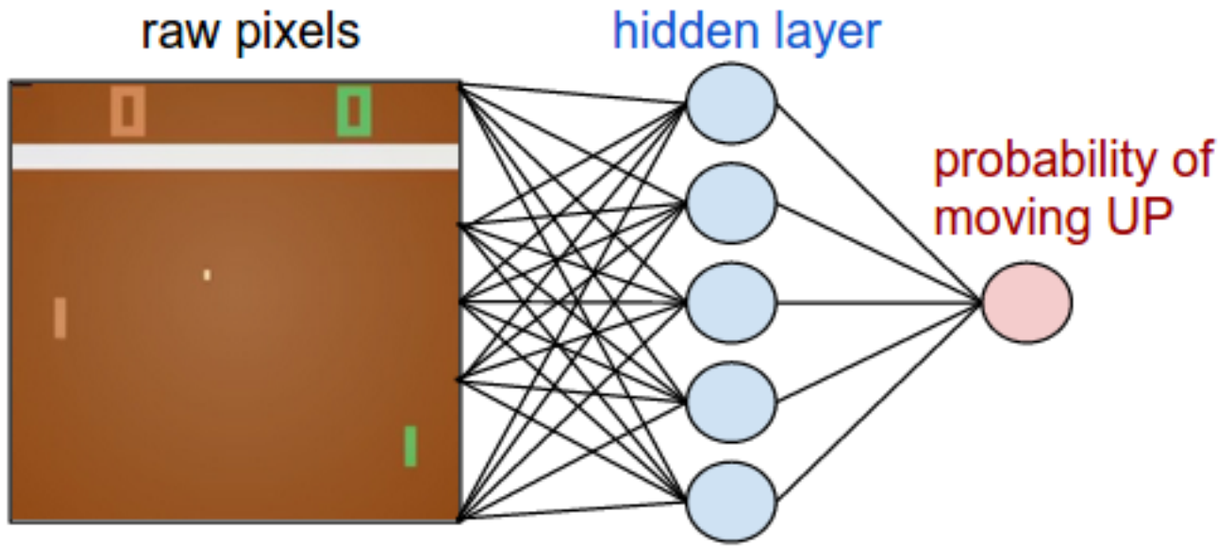
Image Generation

Future forecasting

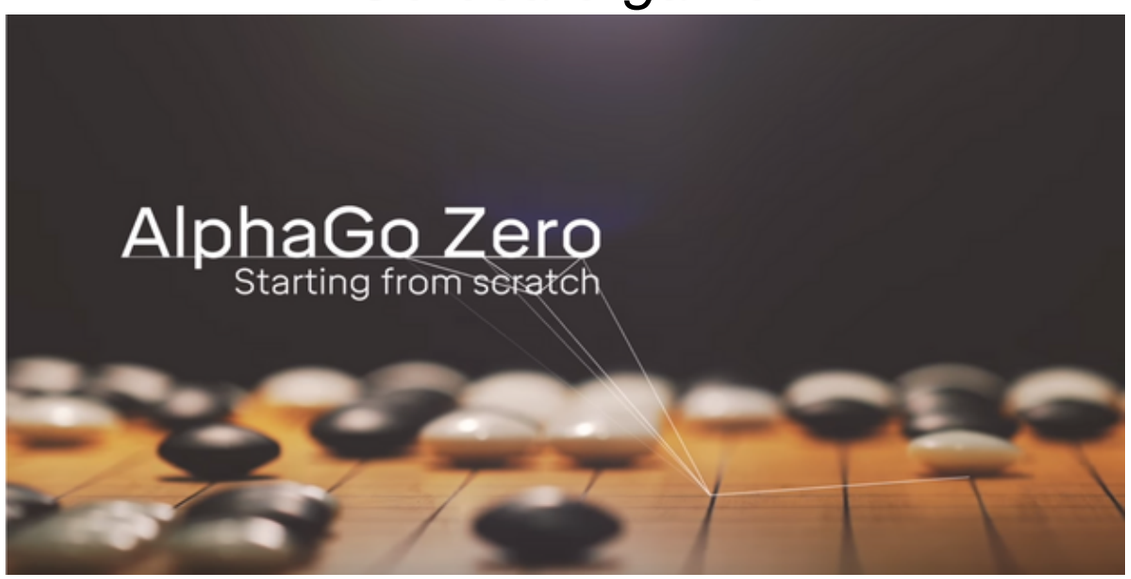
What do we want to learn?

Reinforcement Learning

Superhuman performance in video game



Go board game

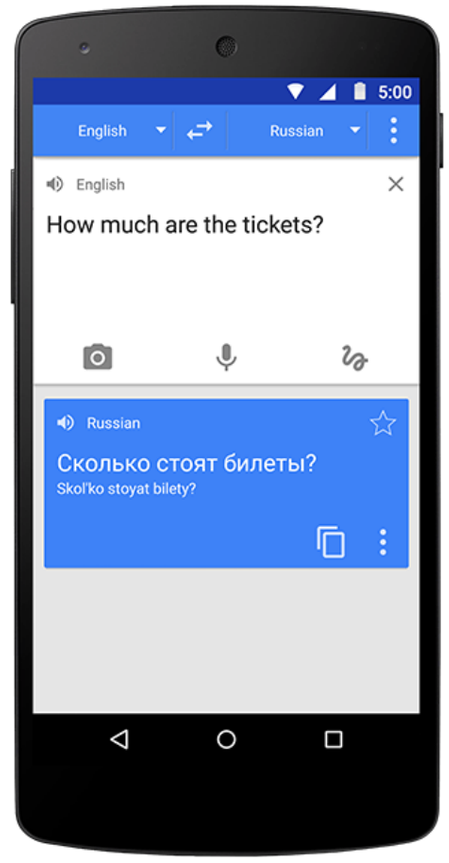


Strong AI

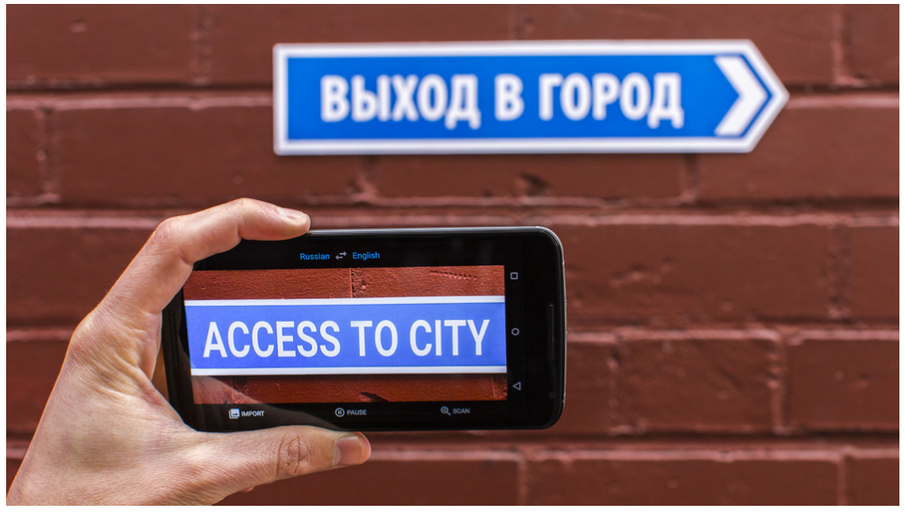
Specific task
Memorization
Shift between train and test
Adaptation to new task

Real World Applications

Tesla



Google



Criteo

Industry

Huge investment
R&D Centers
USA - Canada - China - Europe (France!)

Software

ANACONDA®

Python Data Science Platform

Conda - environnement

NVIDIA®

NumPy

SciPy

matplotlib

IP[y]: IPython

Interactive Computing

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

scikit learn

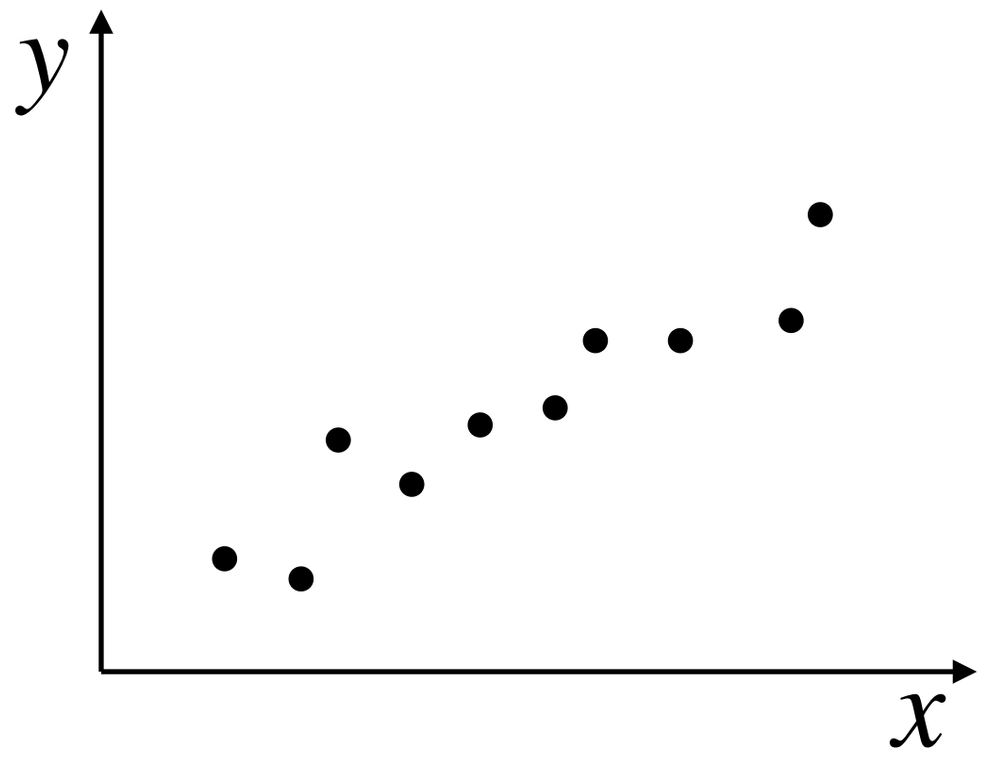
Supervised Learning

Regression

Problem Statement

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

$$y = f(x)$$



Solution

Model

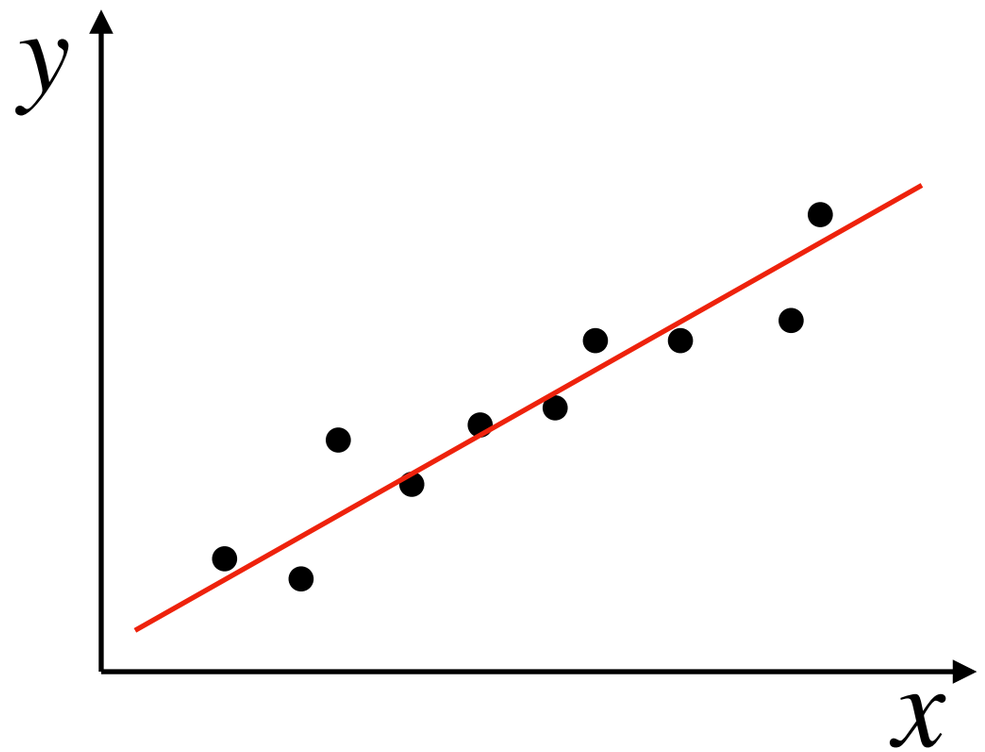
$$f_{w,b}(x) = wx + b$$

Prediction

$$y = f_{w,b}(x)$$

Optim

$$\min_{w,b} \frac{1}{N} \sum_{i=1\ldots N} (f_{w,b}(x_i) - y_i)^2$$



Closed-form solution

We set $\beta = [b \quad w]$ $X = \begin{bmatrix} 1 & x_1 \\ \cdots & \cdots \\ 1 & x_N \end{bmatrix}$ $Y = \begin{bmatrix} y_1 \\ \cdots \\ y_N \end{bmatrix}$

Optimization $\min_{\beta} ||\beta X - y||^2$

Optimal solution $\beta^* = \hat{\beta} = (X^T X)^{-1} X^T y$

Pros and Cons

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

We can add statistical hypothesis

Robust modelling

Model checking

Invertibility

Difficult to compute in some case

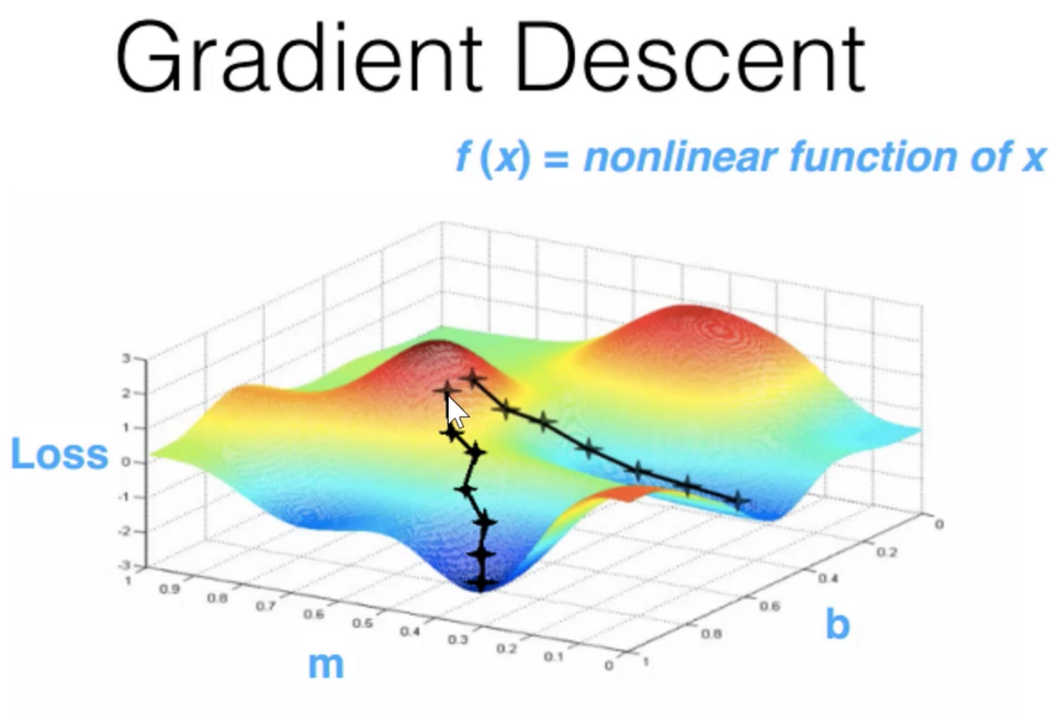
Linear regression with Gradient Descent

$$J = \frac{1}{N}\sum_{i}^{N}((wx_i + b) - y_i)^2$$

Optimization problem

Minimize a loss function using a certain model

Find the parameters which are minimizing the loss function



Linear regression with GD

Find best parameters

$$J(w,b) = \frac{1}{N} \sum_{i=1\ldots N} ((wx_i + b) - y_i)^2$$

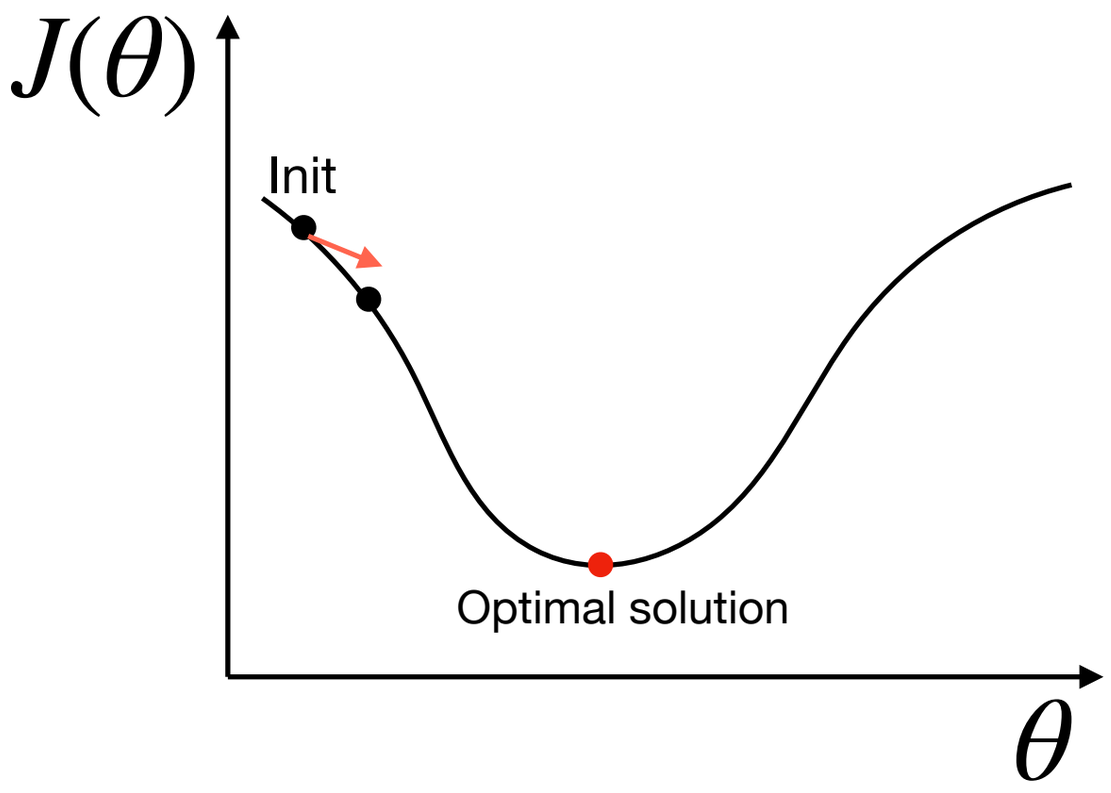
Compute derivatives

$$\frac{\partial J}{\partial w} = \frac{1}{N} \sum_{i=1}^{N} -2x_i(y_i - (wx_i + b))$$

$$\frac{\partial J}{\partial b} = \frac{1}{N} \sum_{i=1}^{N} -2(y_i - (wx_i + b))$$

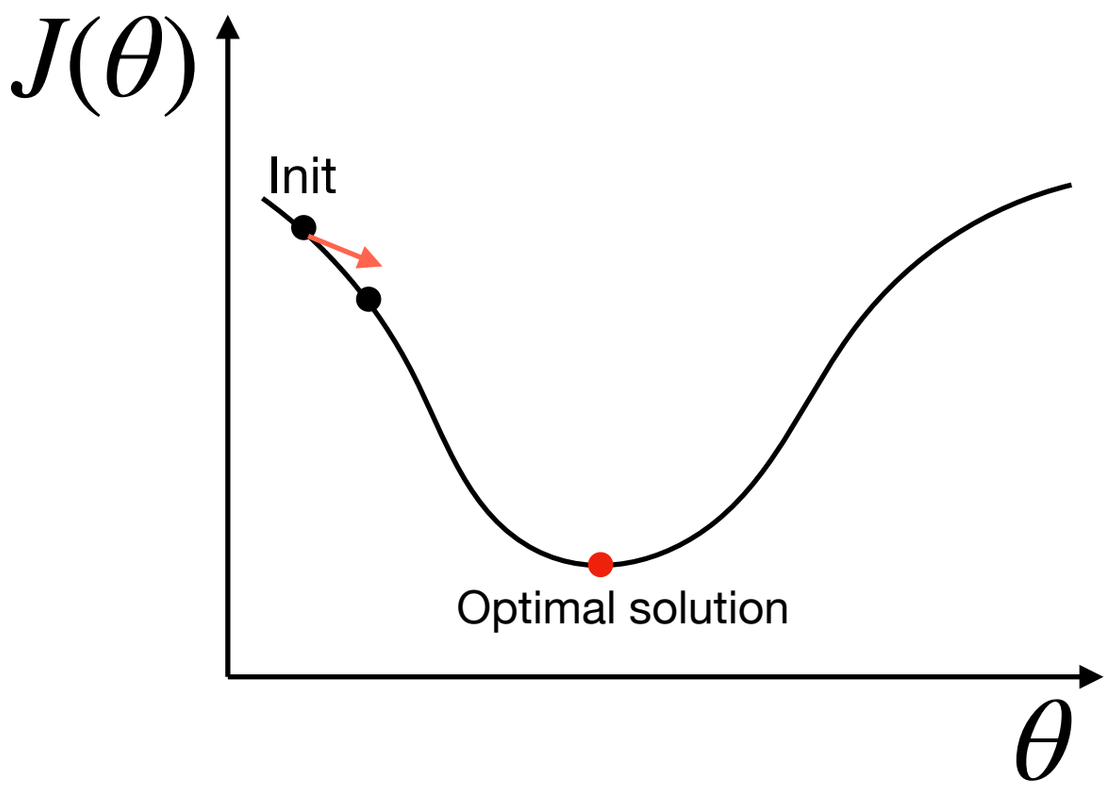
And update parameters iteratively

Gradient Descent



Goal: minimization of a function
Random initialization of parameters
At time t, gradient = slope fo the function
Iterative process
Updating parameters in the positive direction with a learning rate
Repeat until convergence

Gradient Descent



- Init $\hat{\theta}_0$ randomly
- Choose a learning rate η
- for t in range(nb_iter):
  - Update parameter

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta \frac{\partial J(\hat{\theta}_t)}{\partial \theta}$$

Linear Regression with GD

$$J(w,b)_{1:N} = \frac{1}{N}\sum_{i=1}^{N}((wx_i + b) - y_i)^2$$

- Init $\hat{w}_0, \hat{b}_0$ randomly
- Choose a learning rate η
- for t in $1 \ldots T$:
  - Update parameters

$$\hat{w}_{t+1} = \hat{w}_t - \eta\frac{\partial J(\hat{w}_t, \hat{b}_t)_{1:N}}{\partial w}$$

$$\hat{b}_{t+1} = \hat{b}_t - \eta\frac{\partial J(\hat{w}_t, \hat{b}_t)_{1:N}}{\partial b}$$

Linear Regression with Stochastic GD

$$J(w,b)_i = ((wx_i + b) - y_i)^2$$

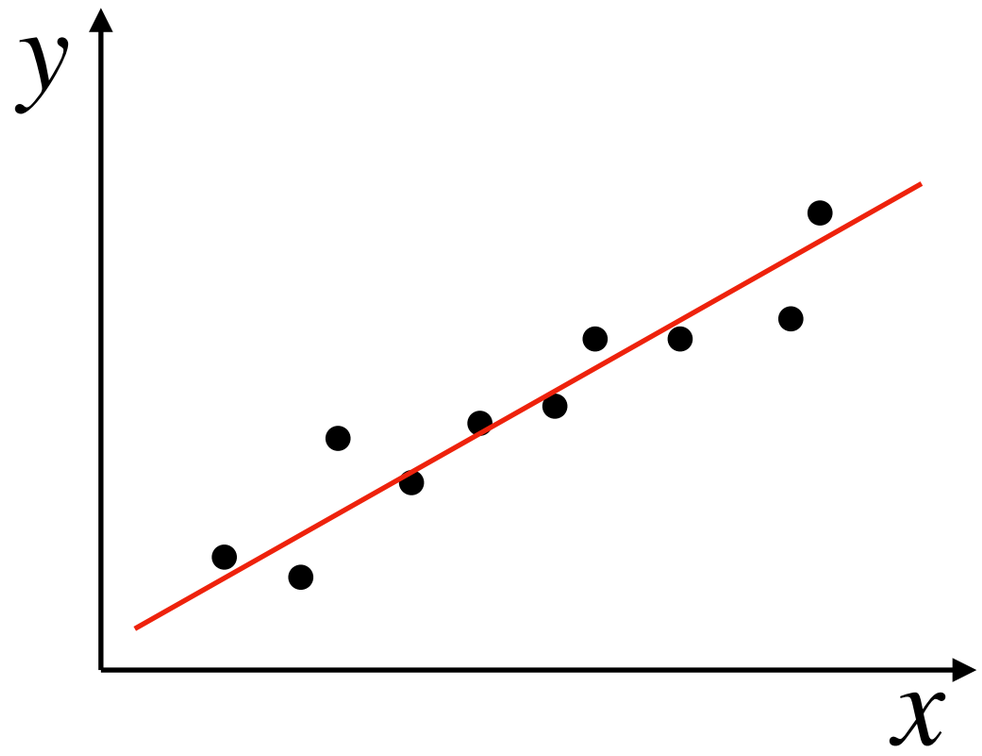
- Init $\hat{w}_0, \hat{b}_0$ randomly
- Choose a learning rate η
- for t in $1 \ldots T$:
  - Sample some points from the data
  - Update parameters

$$\hat{w}_{t+1} = \hat{w}_t - \eta \frac{\partial J(\hat{w}_t, \hat{b}_t)_{idx}}{\partial w}$$

$$\hat{b}_{t+1} = \hat{b}_t - \eta \frac{\partial J(\hat{w}_t, \hat{b}_t)_{idx}}{\partial b}$$

Exercise

Implementing Gradient Descent
LinearRegression: Closed-form vs GD vs SGD



Supervised Learning

Best Practice

Python library


```
from sklearn.linear_model import LinearRegression
```

INRIA - Telecom ParisTech
Works with NumPy SciPy
Written in Cython

Numerical and Categorical Variable

One-hot-encoding:
From categorical to numerical

$$red = [1, 0, 0]$$
$$yellow = [0, 1, 0]$$
$$green = [0, 0, 1]$$

Binning:
From numerical to categorical

Normalization - Standardization

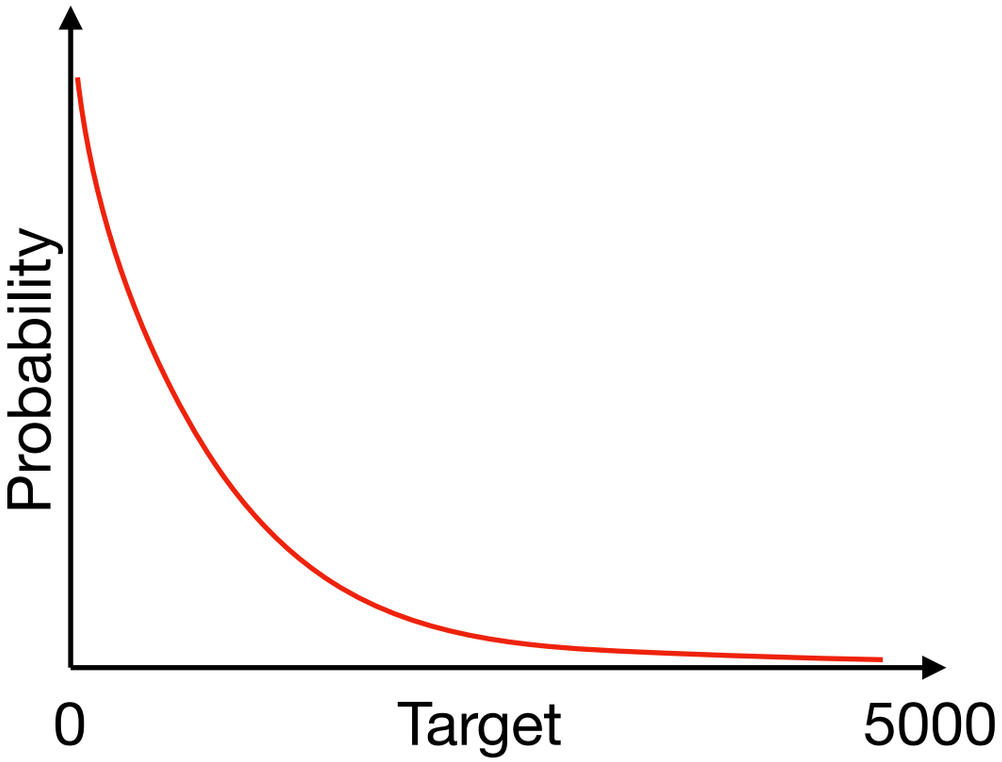
Normalization (0-1)

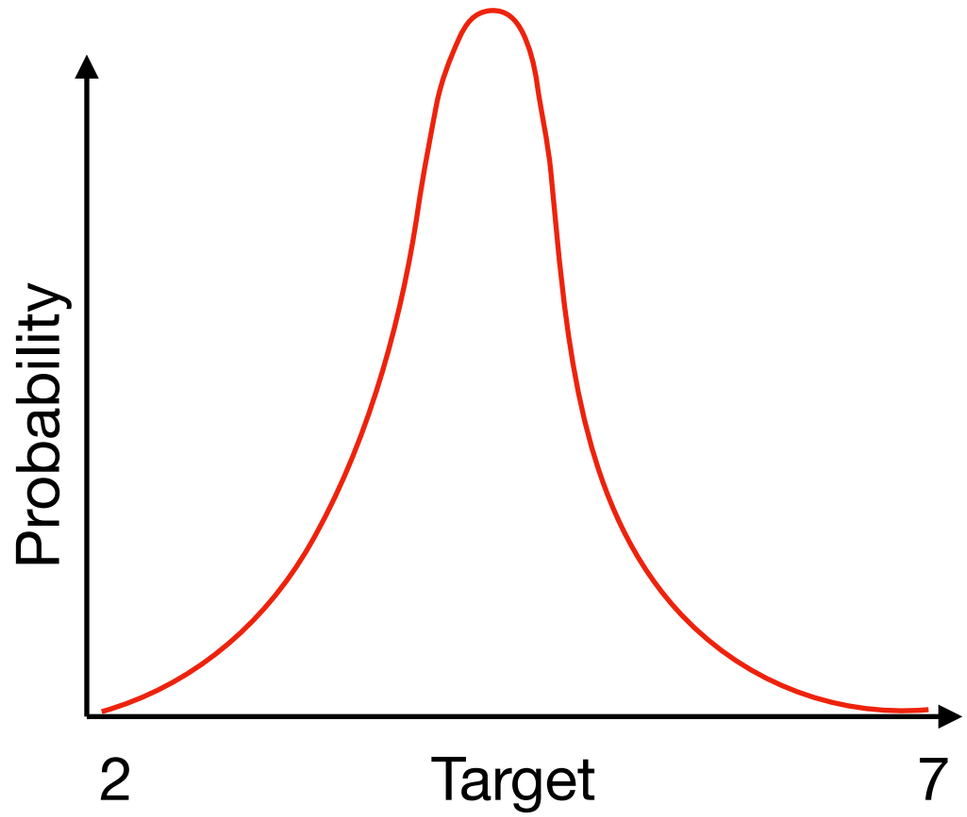
$$\bar{x}^{(j)} = \frac{x^{(j)} - min^{(j)}}{max^{(j)} - min^{(j)}},$$

Z-score

$$\hat{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}}.$$

Target transformation





Log rescaling

Interactions between variables

Inductive bias with expert knowledge

Indicator Variables:
Threshold (e.g. age >=21)

Interaction Features:
Sum
Difference
Product
Quotient

Three sets

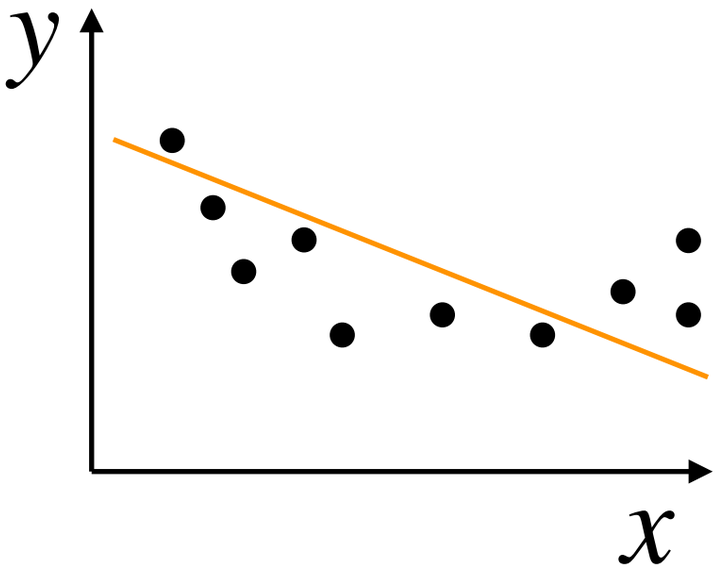
1) Training set
2) Validation set
3) Test set

Shuffle dataset
70 % - 15 % - 15 %
Hyper parameters on val while training on train

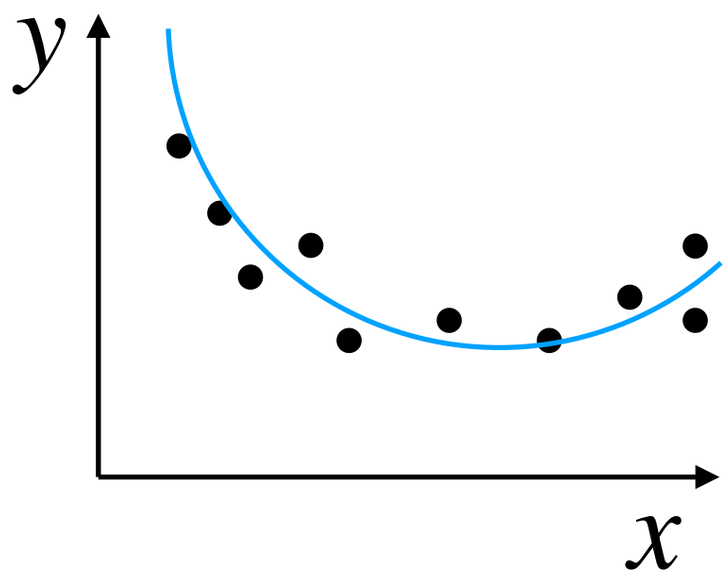
Generalization !

k-cross fold validation

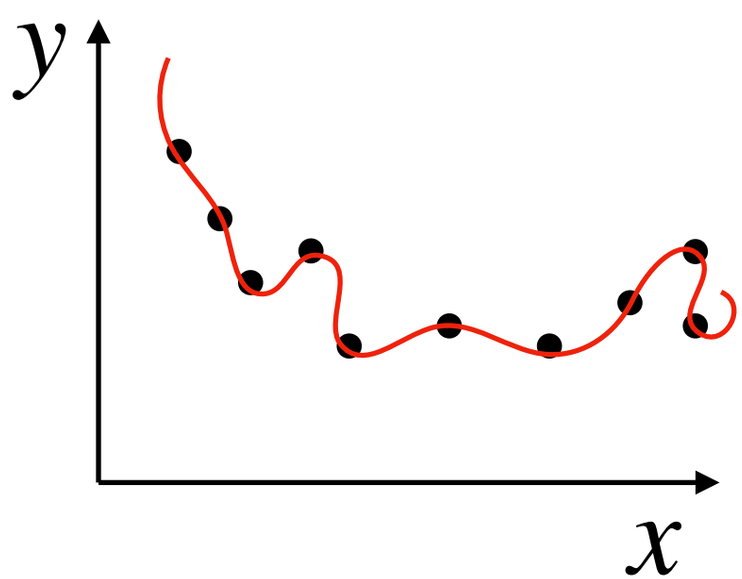
Underfitting and Overfitting



Underfitting



Good fit



Overfitting

Regularization

Build less complex model

L1 normalization

$$\min_{w,b} \frac{1}{N} \sum_{i=1}^{N} ((wx_i + b) - y_i)^2 + C|w|$$

L2 normalization

$$\min_{w,b} \frac{1}{N} \sum_{i=1}^{N} ((wx_i + b) - y_i)^2 + C||w||^2$$

L1 + L2 = Elastic Net

Example on real data

Boston dataset

Train + Val set = 1st 300 examples !
Kaggle exercise: Find the best model for the test set

Supervised Learning

Classification

Logistic Regression

$$D = \{(x_i, y_i)\}_{i=1}^{N} \qquad y \in \{0,1\}$$

Mapping we want to learn

$$y = f(x)$$

Our modeling

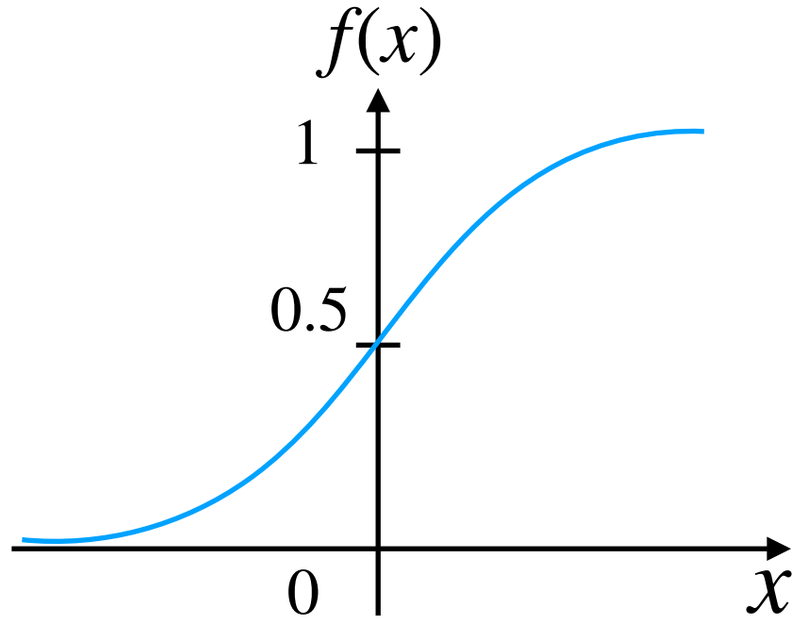
$$y = f_{w,b}(x)$$

How to predict 0 or 1 ?

Logistic Regression

Sigmoid function

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



Model $$f_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}}$$

0.5 is the threshold value !

Loss function

Model

$$f_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}} = p_{w,b}(x)$$

Loss function

$$J(w,b) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p_i) + (1 - y_i)\log(1 - p_i)$$

Minimization of the negative maximum likelihood

Fully differentiable and parameters can be estimated by SGD

?

Confusion Matrix

		Actual Class 1	0
Predicted Class	1	True Positive (TP)	False Negative (FN)
	0	False Positive (FP)	True Negative (TN)

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

$$\textbf{F-mesure} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Exercise

Breast Cancer

Train + Val = 1st 300 examples

Find best model for the test set

Recent Algorithm

SVM - kernel methods (gaussian kernel)
Gradient Boosting
Neural Network