# An introduction to Machine Learning

*A (very) gentle Introduction*

**Massimiliano Ruocco, PhD**
*Research Scientist, Telenor Research*
*Adjunct Associate Professor, DART Group, NTNU*

artificial intelligence lab

telenor NTNU

# Outline

---

- Introduction
- Supervised ML
- Unsupervised ML
- Reinforcement Learning
- Deep Learning
- Conclusion

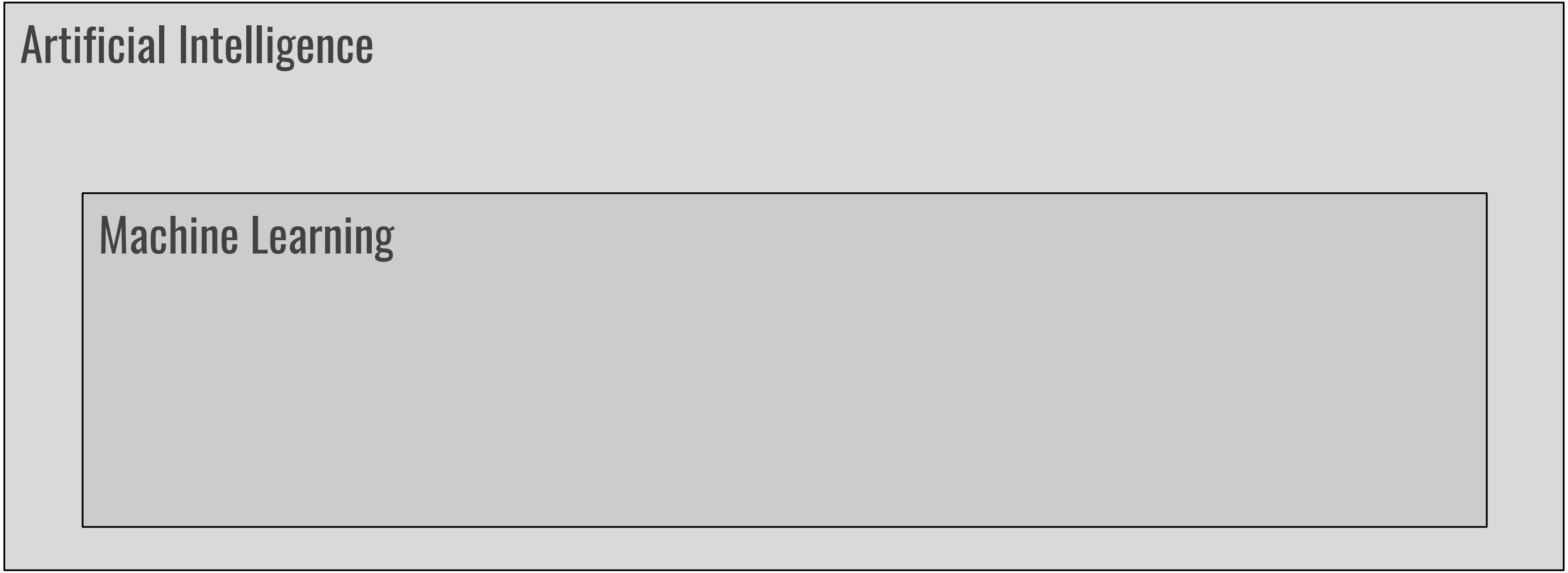
- Exercise!

# Introduction

# What is Machine Learning

Definition

# Taxonomy of Machine Learning



# Taxonomy of Machine Learning

## Artificial Intelligence

### Machine Learning

#### Supervised Learning

- Classification
- Regression

#### Unsupervised Learning

- Clustering

#### Reinforcement Learning

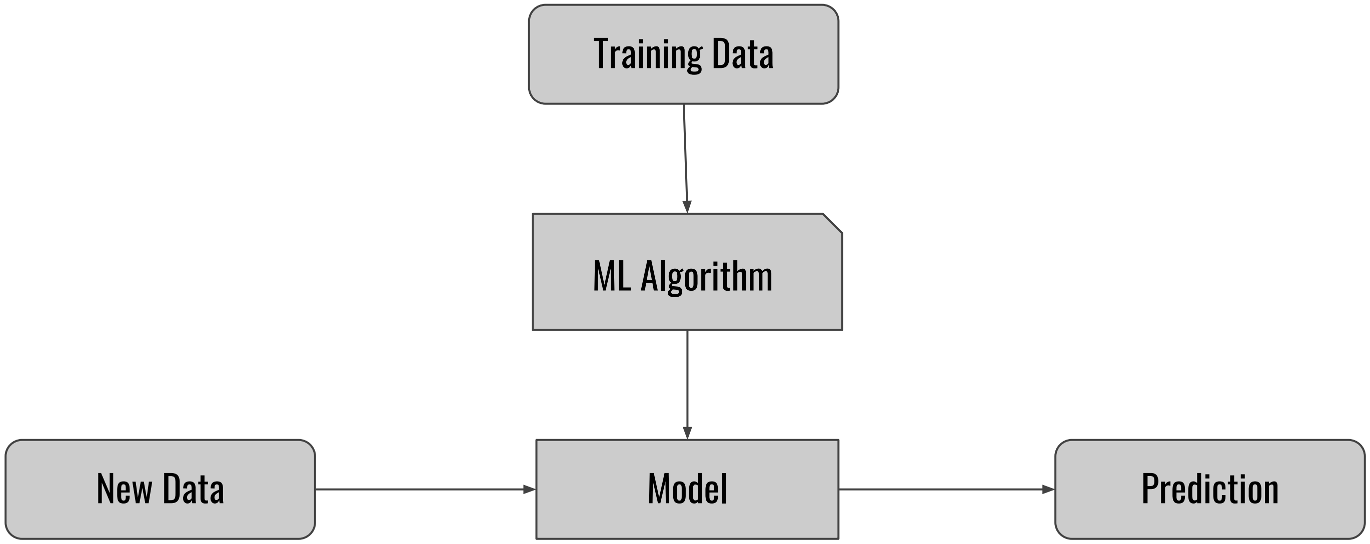
# Why Machine Learning

## Machine Learning Examples

| Retail | Marketing | Healthcare | Telco | Finance |
| --- | --- | --- | --- | --- |
| • Demand forecasting<br>• Supply chain optimization<br>• Pricing optimization<br>• Market segmentation and targeting<br>• Recommendations | • Recommendation engines & targeting<br>• Customer 360<br>• Click-stream analysis<br>• Social media analysis<br>• Ad optimization | • Predicting Patient Disease Risk<br>• Diagnostics and Alerts<br>• Fraud | • Customer churn<br>• System log analysis<br>• Anomaly detection<br>• Preventative maintenance<br>• Smart meter analysis | • Risk Analytics<br>• Customer 360<br>• Fraud<br>• Credit scoring |

# Supervised ML

# Introduction

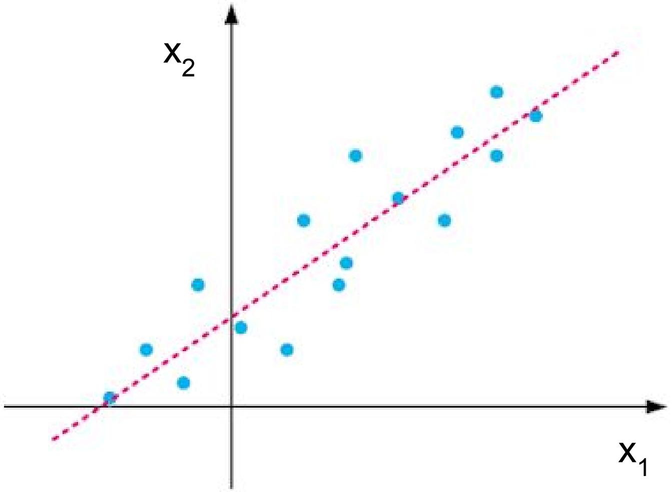


# Classification vs Regression

**Regression**: predict a continuous numerical value.
*How much will be the price of this stock?*

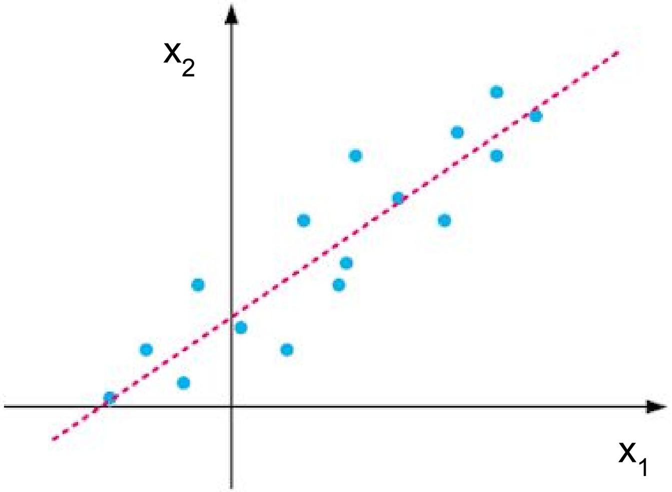
**Classification**: assign a label to a given observation.
*Is this a picture of a car or a motorcycle?*

# Classification vs **Regression**




$$X_2 = f(X_1, \beta)$$

# Classification vs **Regression**




Regression function

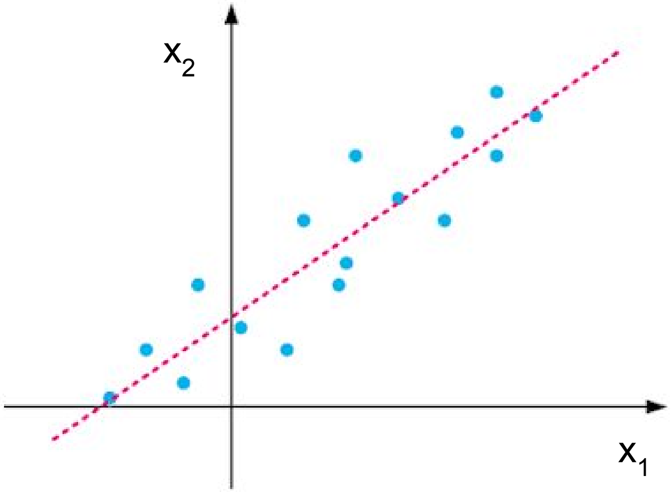
Parameters of the function f

$$X_2 = f(X_1, \beta)$$

Dependent variable

Independent variable

# Classification vs **Regression**




Regression function

Parameters of the function f

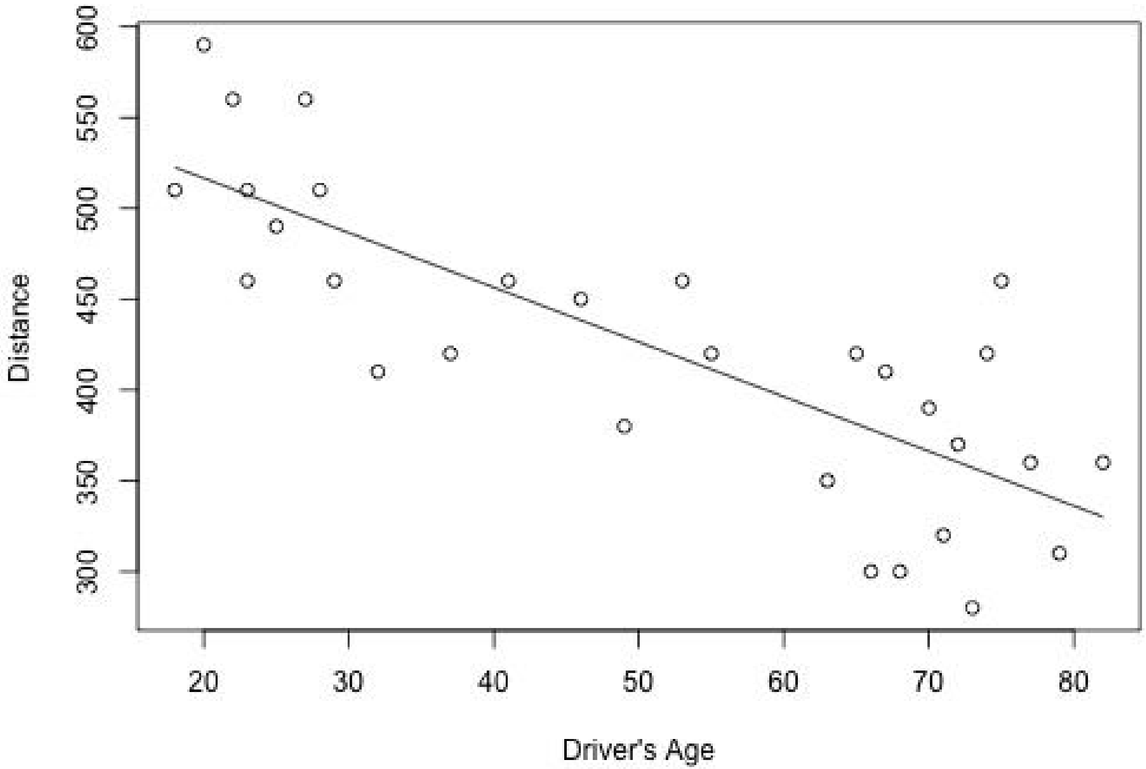
$$X_2 = f(X_1, \beta)$$

Dependent variable

Independent variable

**Main Task:** Estimating *f*, giving a set of training data ***X***

# Classification vs **Regression**: *Example*



# Classification vs **Regression**

---

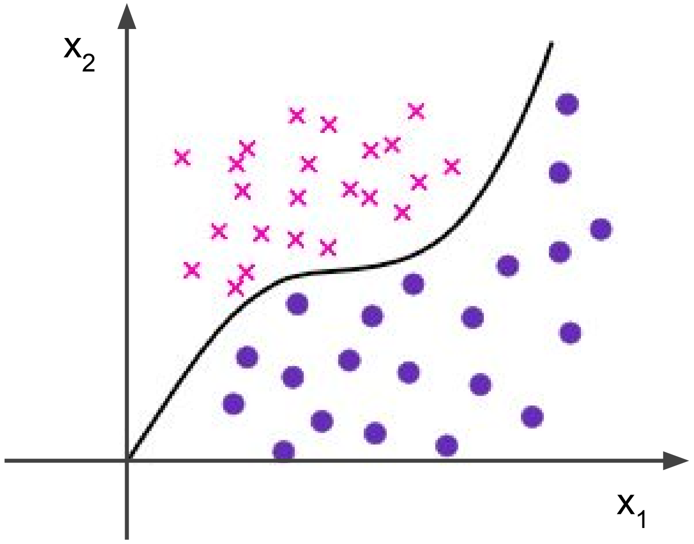
Regression Algorithms

- Logistic Regression
- Linear Regression
- Support Vector Regression
- Regression Trees
- Least Angle Regression
- ...

# Classification vs **Regression**: *Linear Regression*

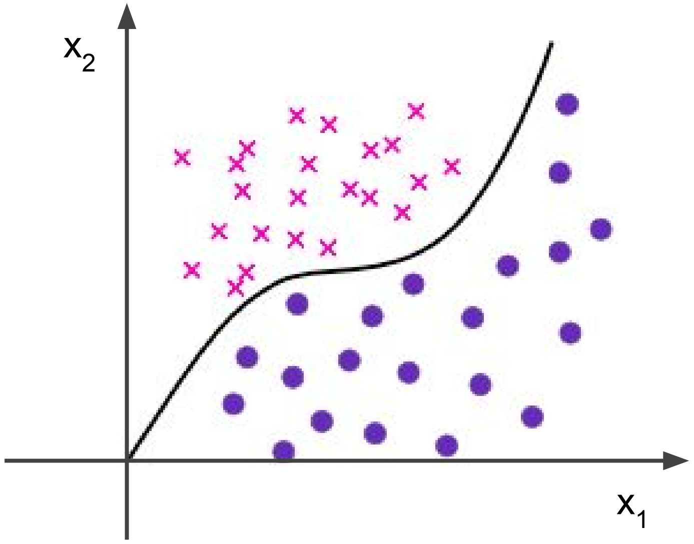
$x_1$

# **Classification** vs Regression




$$Y = f(\mathbf{X})$$

# Classification vs Regression




$f: R^d \rightarrow \{L_1, ..., L_N\}$

$$Y = f(\mathbf{X})$$

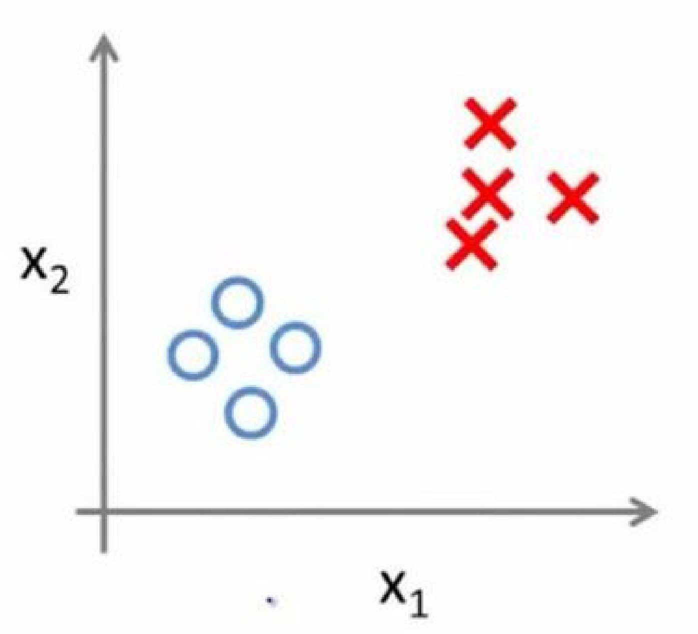
Label

Input

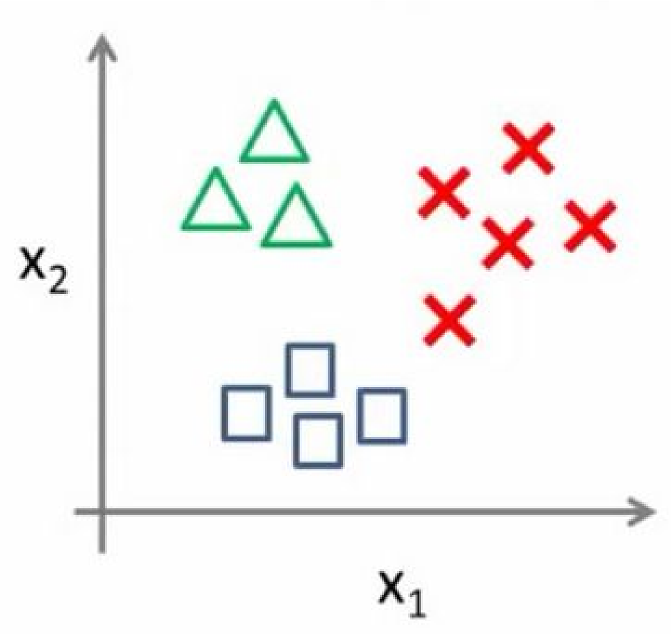
**Main Task:** Estimating *f*, giving a set of training data ***X***

# **Classification** vs Regression

Binary Classification




Multiclass Classification



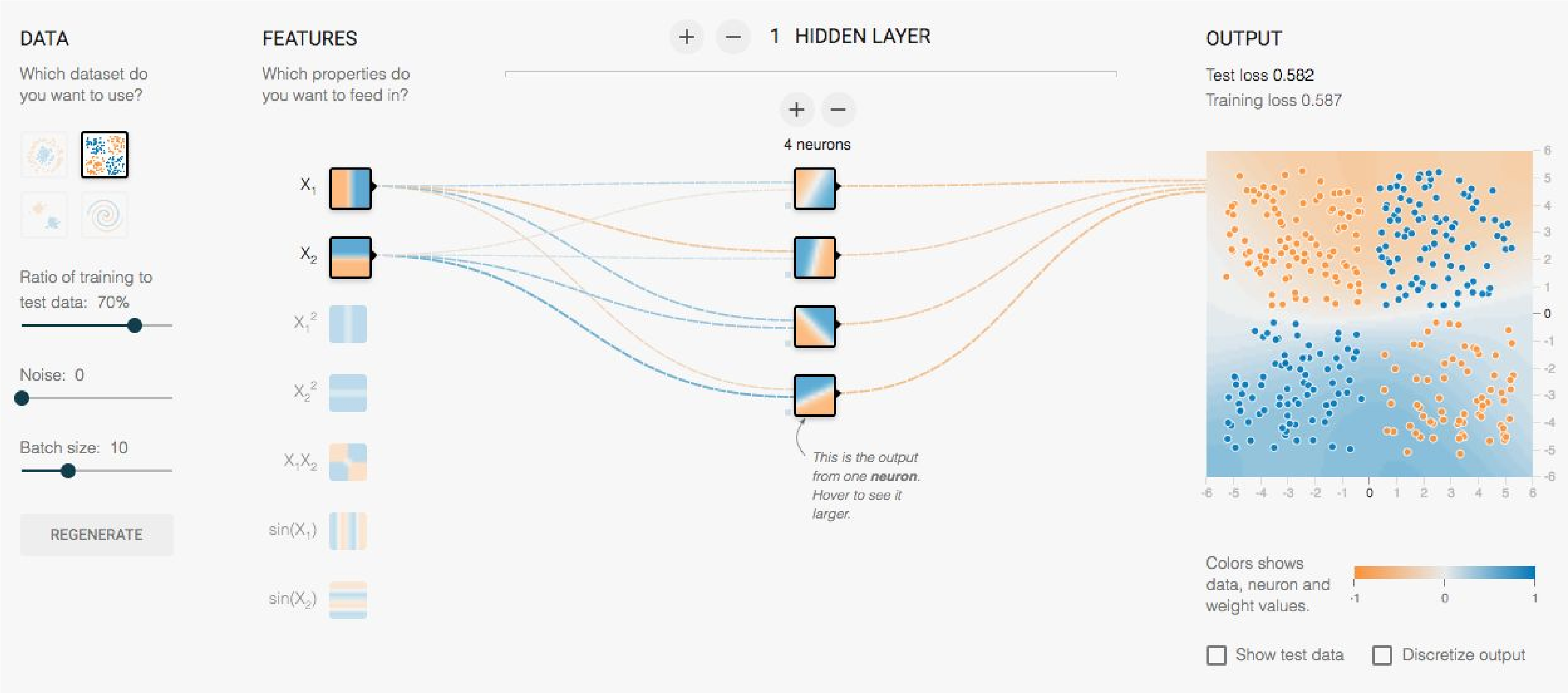
# <u>Classification</u> vs Regression

---

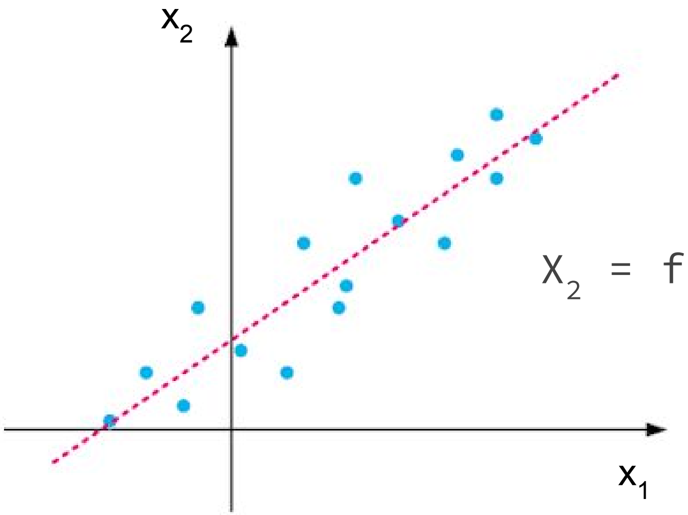
## Classification Algorithms

- Support Vector Machines (SVM)
- Neural Networks
- Naive Bayes Classifier
- Nearest Neighbors (kNN)
- ...

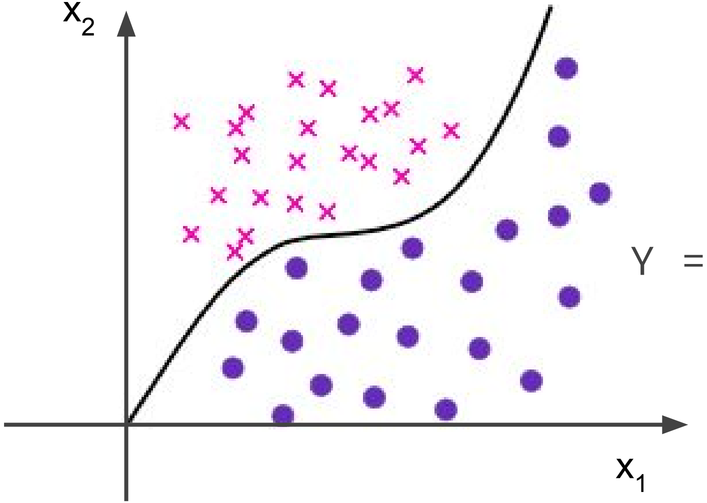
# Classification vs Regression: *Neural Network*



# Classification vs Regression: **Summary**




$X_2 = f(X_1, \beta)$



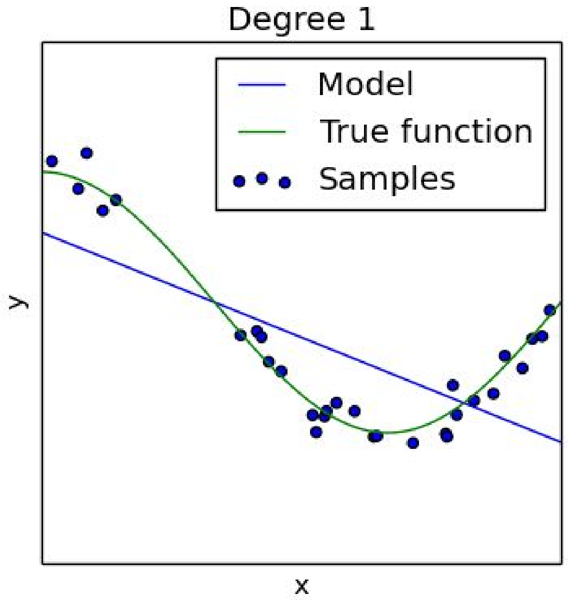

$Y = f(\mathbf{X})$

**Input** - Continuous
**Model** - Best fit Line
**Evaluation** - Sum of squared error
**Inference** - Predict the value of $X_2$ given the value of $X_1$

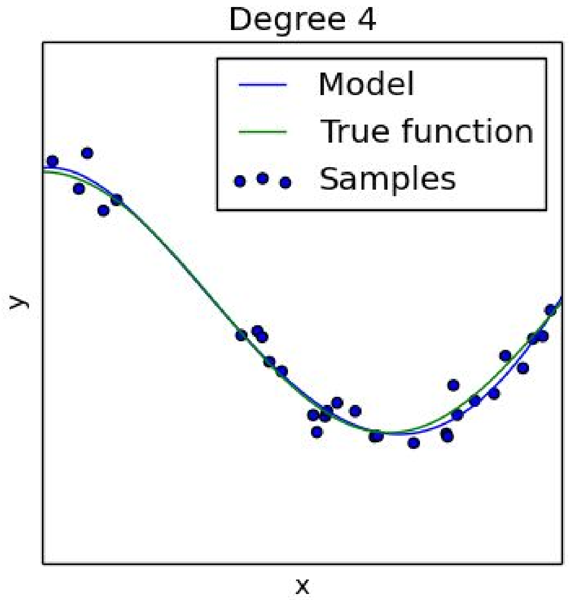
**Input** - Discrete
**Model** - Decision Boundary
**Evaluation** - Accuracy
**Inference** - Predict the label of a given **X**

# Overfitting and Underfitting

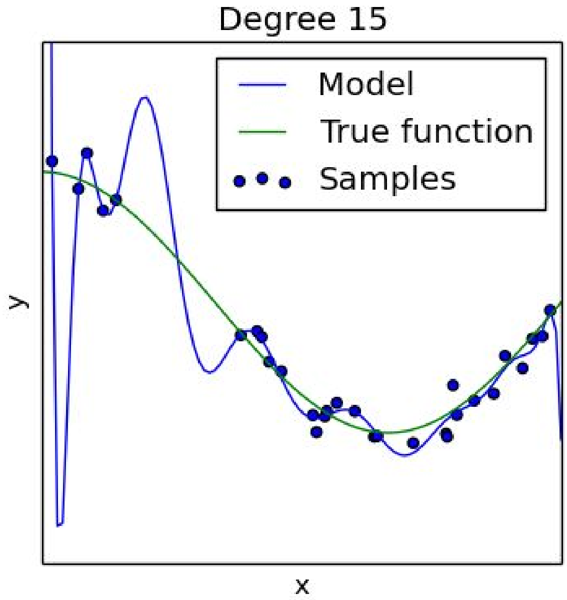
# Overfitting and Underfitting




Underfitting




Optimal




Overfitting

# Parametric vs Nonparametric Methods

Mapping Function

$$Y = f(X)$$

Output variables

Input Variables

# Parametric vs Nonparametric Methods

Mapping Function

$$Y = \boxed{f}(X)$$

Output variables

Input Variables

**Which kind of function is better is better to use for approximating f?**

# Parametric vs Nonparametric Methods

---

- **Assumption** on the underlying **function**
- **Fixed** Number of Parameters to learn
- I.e.: *Linear Regression*

$$x_2 = \boldsymbol{b}x_1 + \boldsymbol{a}$$

## Example

- Logistic Regression
- Linear Discriminant Analysis
- Perceptron
- Naive Bayes
- Simple Neural Network
- Linear SVM

## Pros

- Faster
- Simpler
- Less data
- Easy on memory

## Cons

- Constrained
- Limited Complexity
- Poor Fit

# Parametric vs **Nonparametric** Methods

---

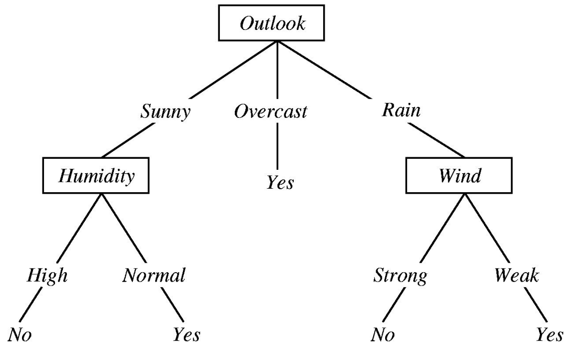
- **No Assumption** on the underlying mapping **function**

## Pros

- Flexibile
- Better Performance

## Example

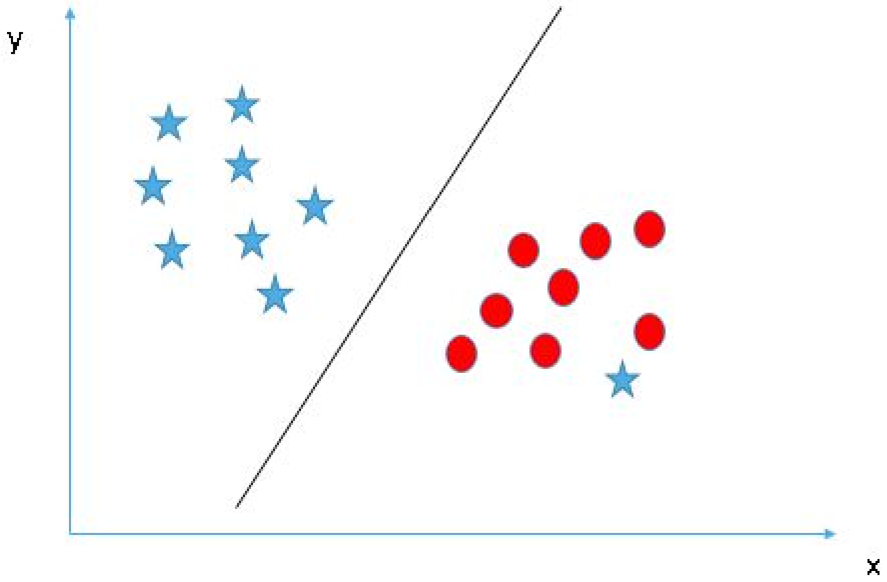
- k-NN
- Decision Trees
- RBF-Kernel SVM



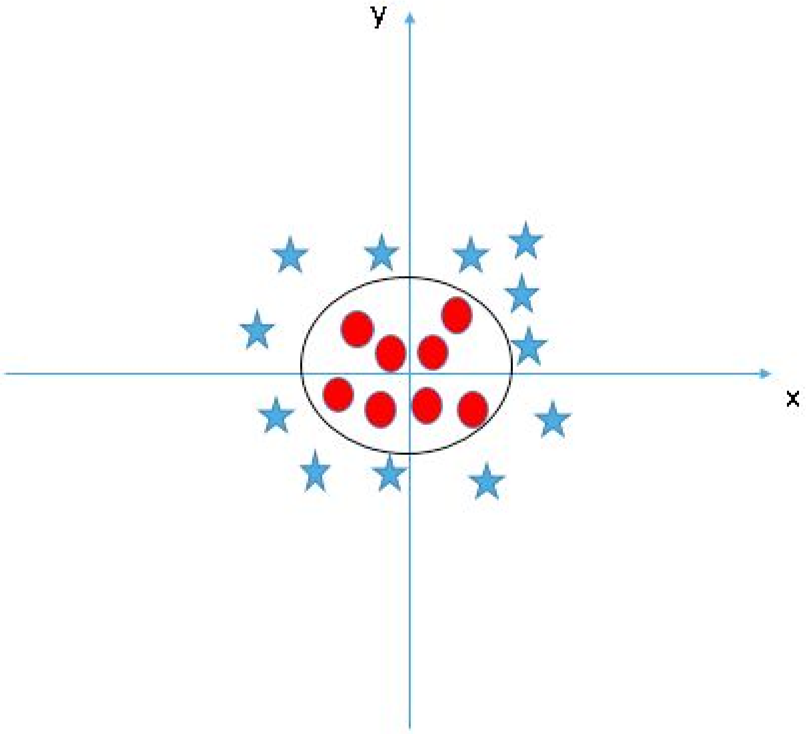

## Cons

- More Data
- Slower
- Overfitting

# Linear vs Non-Linear




Linearly Separable Data



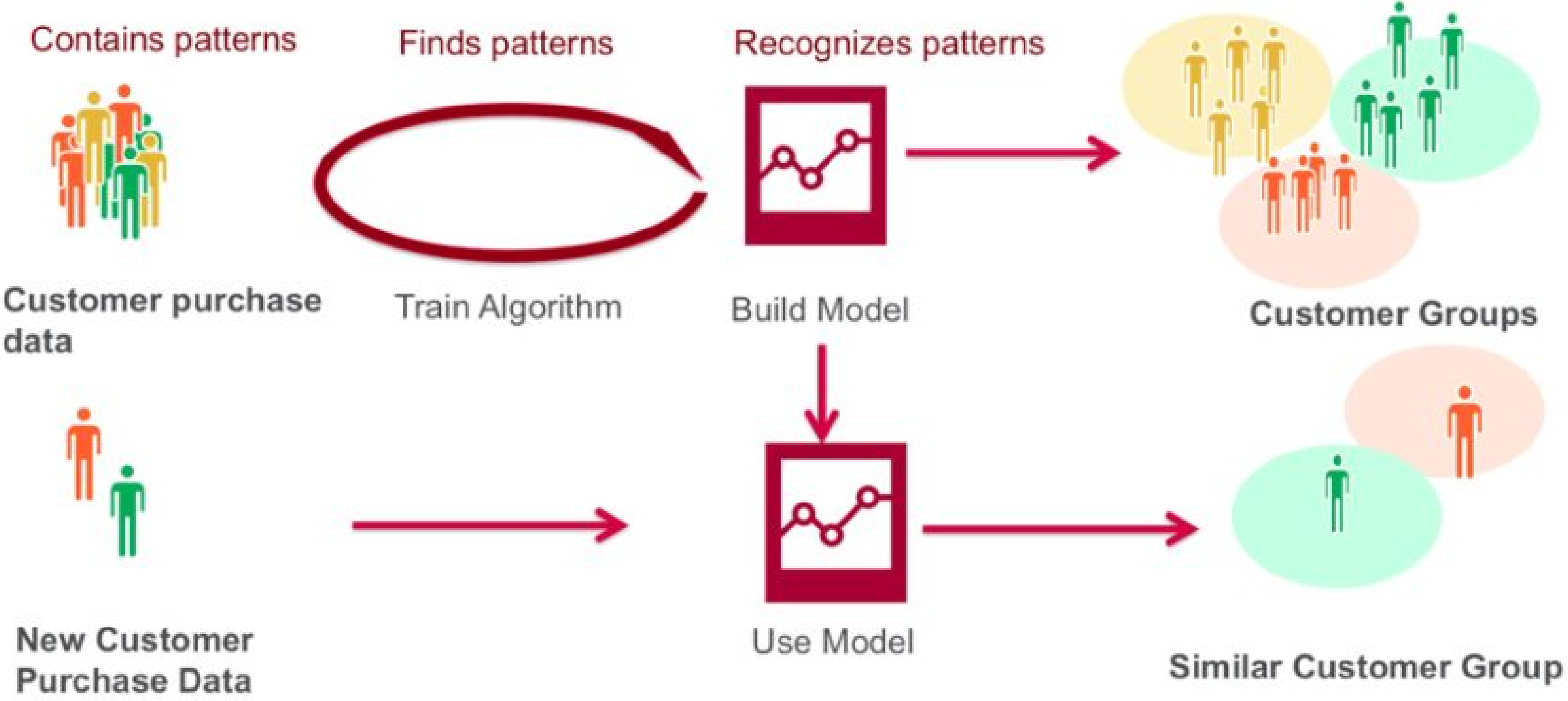

Non Linearly Separable Data

# Applications

# Unsupervised ML

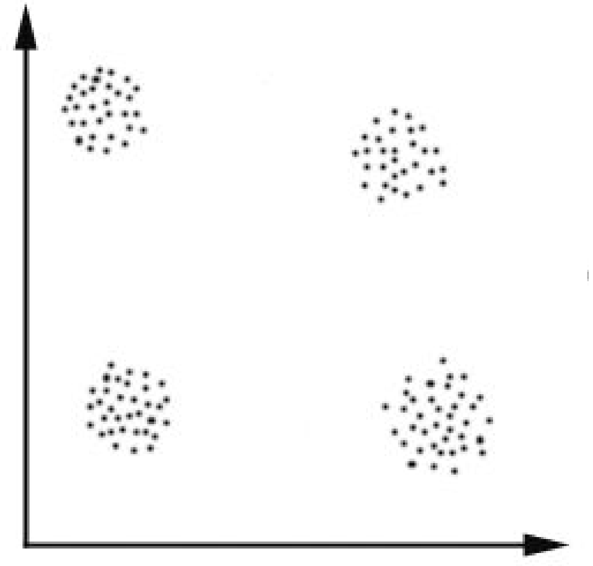
# Introduction

# Introduction: *Example*

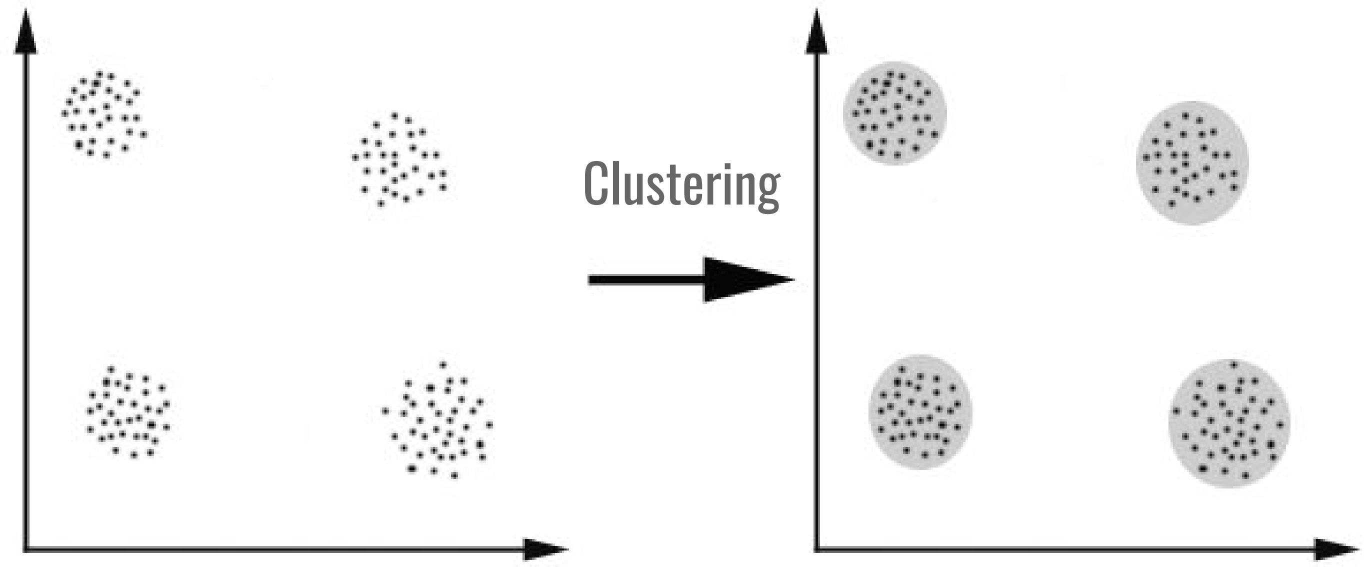


# Clustering

# Clustering



# Clustering



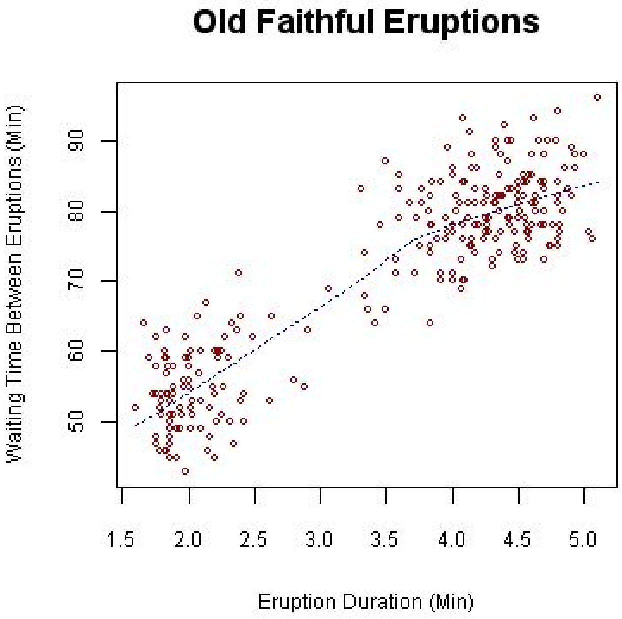
# Clustering: *Applications*

- **Biology**: find similar entities/organisms
- **Information Retrieval**: Search results grouping.
- **Marketing**: Grouping similar customers.
- **Climate**: Find pattern of weather behaviour
- **Document/Text** categorization
- **Network security:** anomaly detection (finds what is not similar, the outliers from clusters).

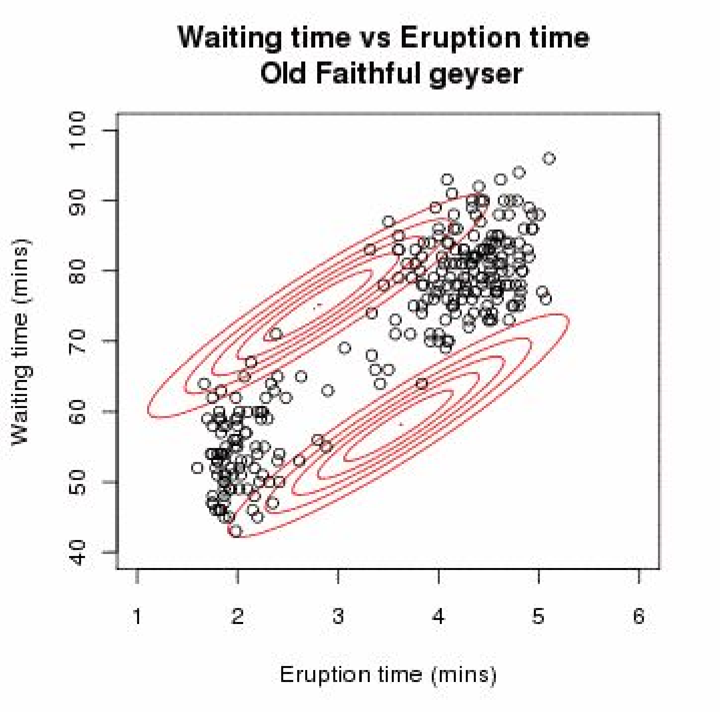
# Clustering: *Methods*

- **K-means Clustering**
- **Principal Component Analysis (PCA)**
- **Gaussian Mixture Model**
- **Self-Organizing Map (SOM)**
- **Hidden Markov Models (HMM)**
- ...

# Clustering - *GMM demo*

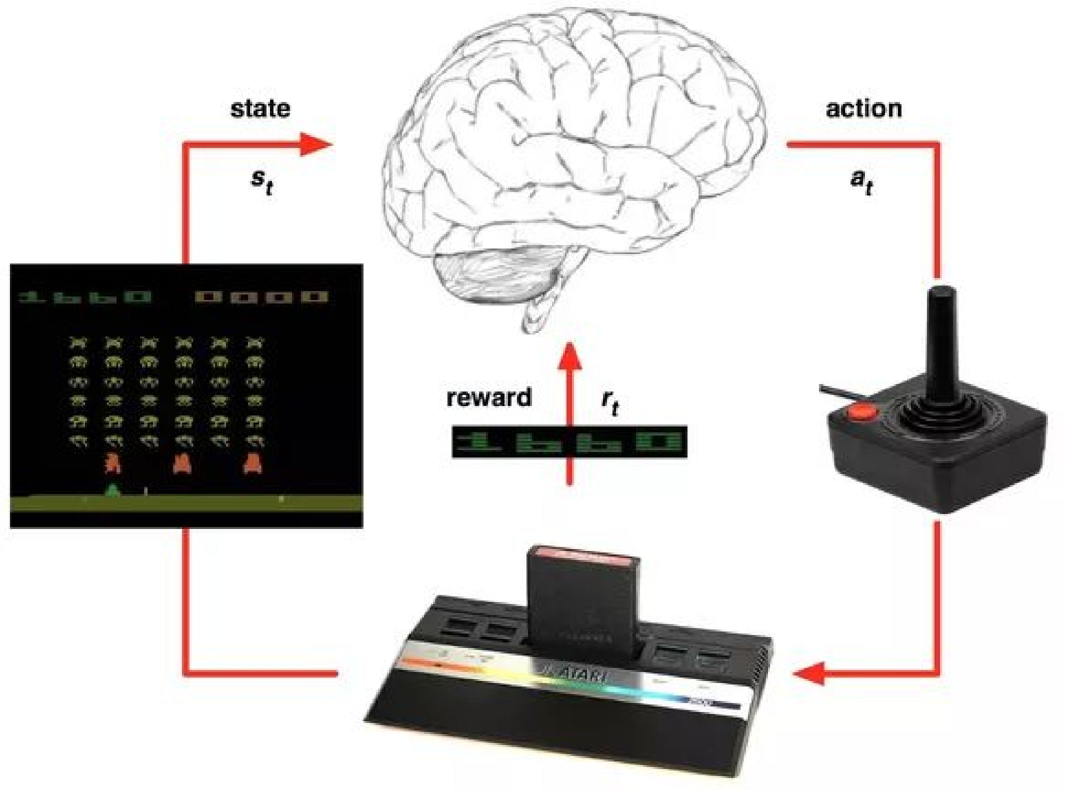


# Clustering - *GMM demo*



# Reinforcement Learning

# Introduction

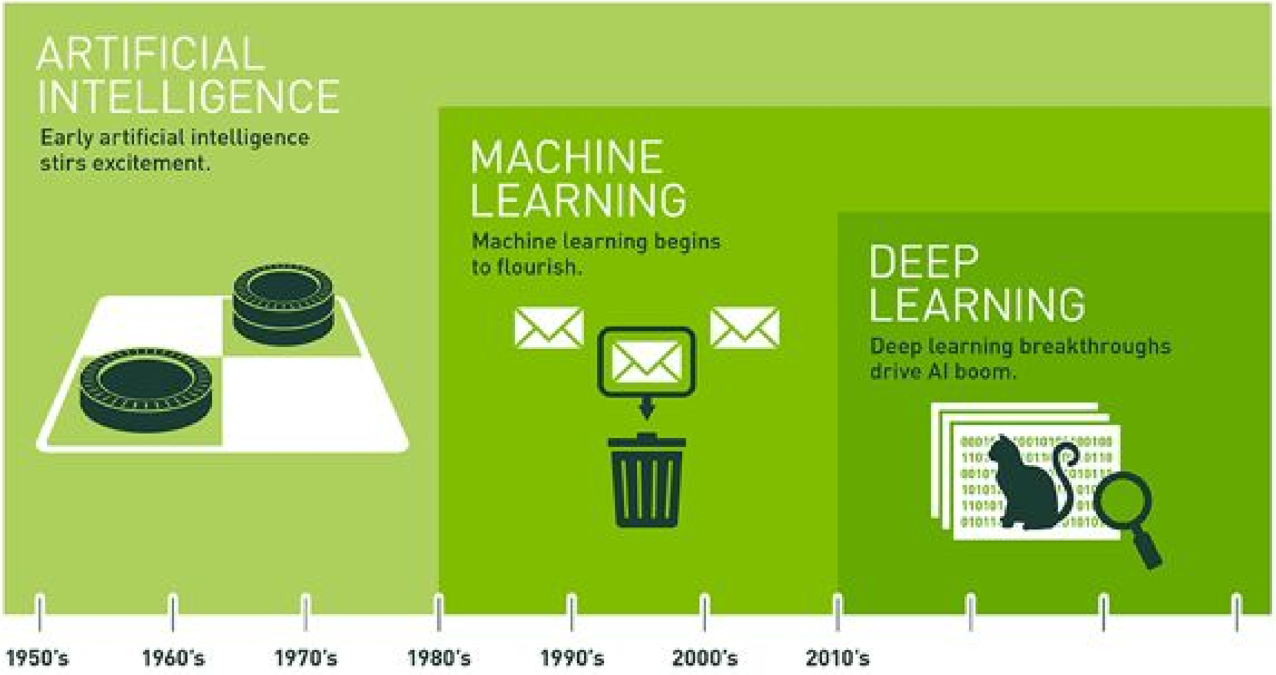


# Application: *Facts on Alpha Go Zero*

- Google Deep Mind **AlphaGo Zero** beated 100-0 the previous **AlphaGo**
- **AlphaGo** needed 100K go games to be trained
- **AlphaGo Zero** only programmed with the basic rules of Go
- **AlphaGo** beated **18 times** the world champions **Lee Se-dol**
- **AlphaGo Zero** staerted beating (90%) the strongest Alpha Go after 40 days of training

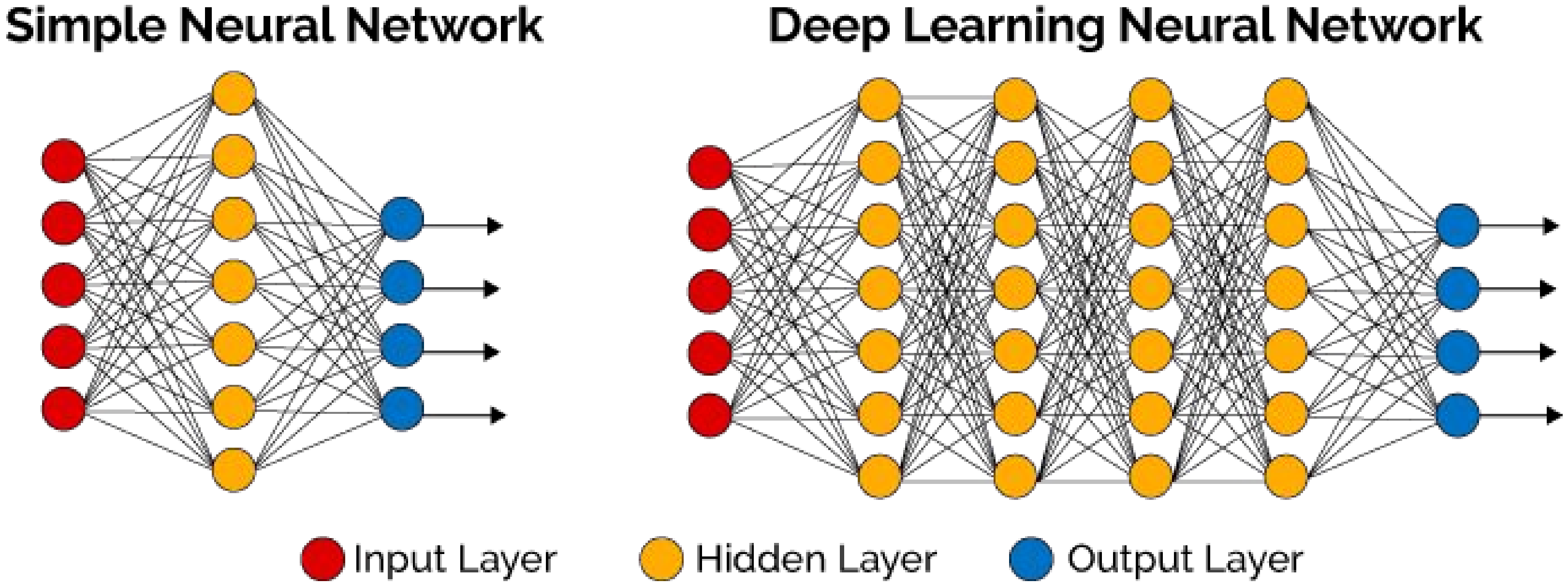
# Deep Learning

# Introduction: *AI vs ML vs DL*

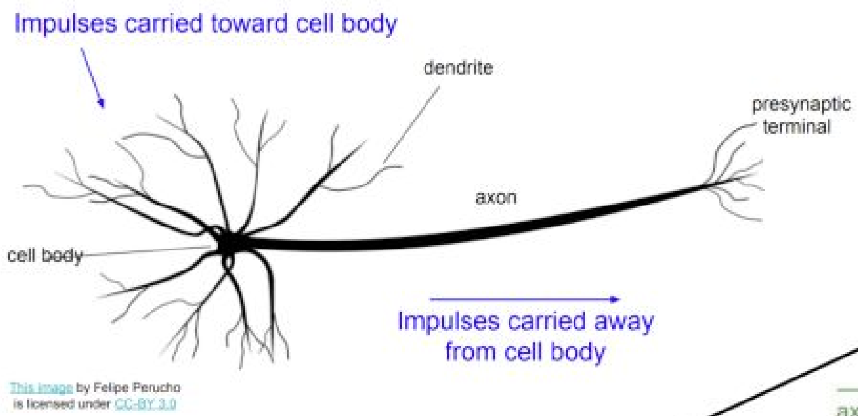


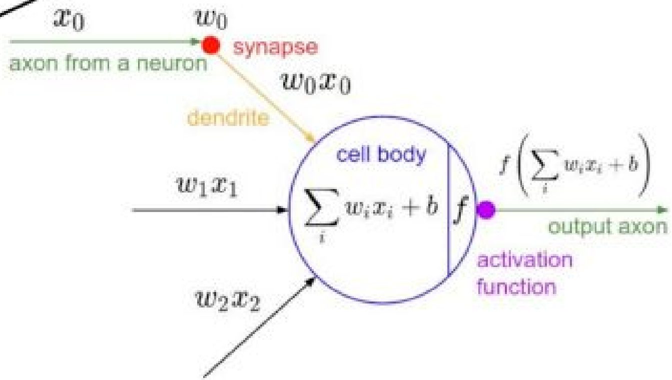

Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

# Introduction: *Deep Network*

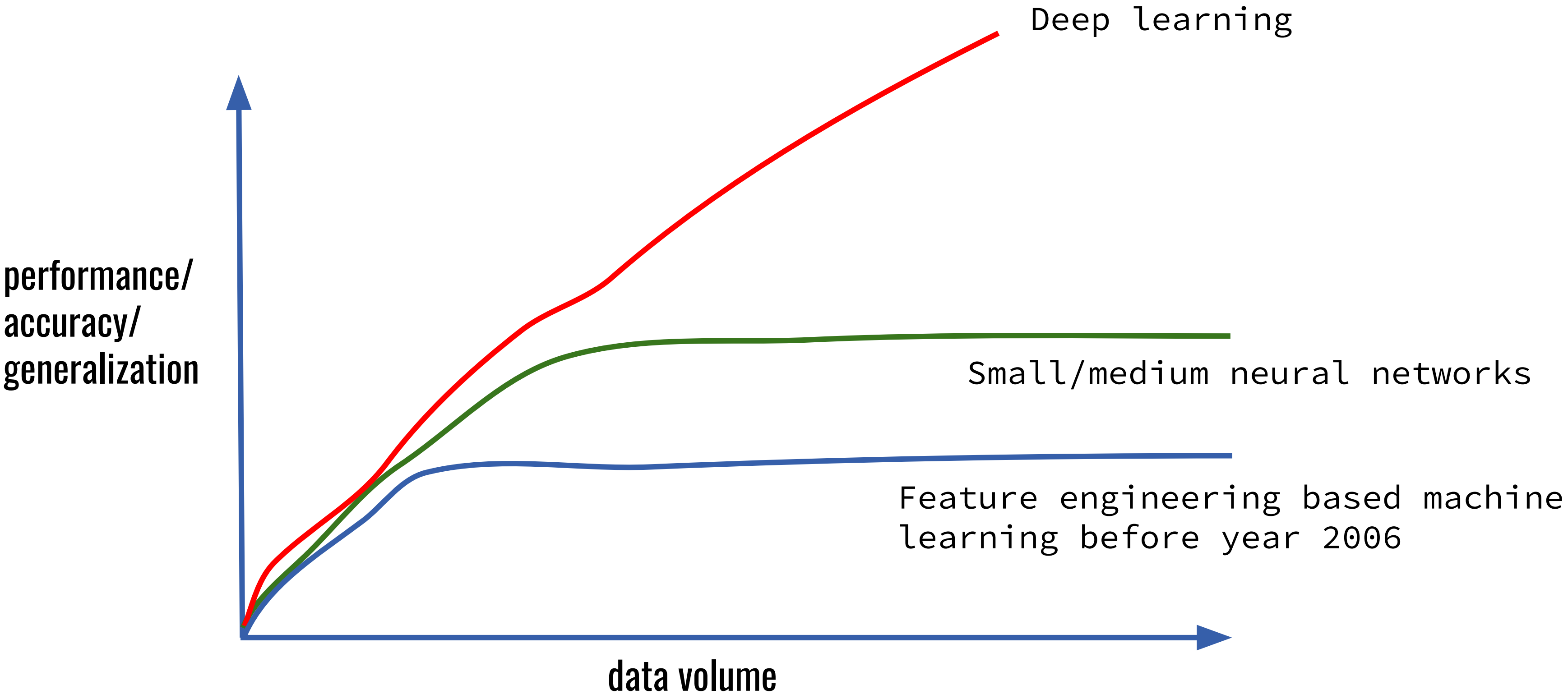


# Introduction: *Biological Inspiration*



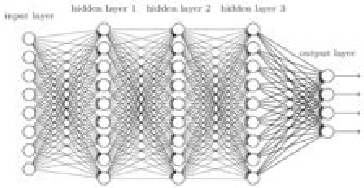
# Introduction: *Motivation on Data Volume*



# Introduction: *Main Architectures*

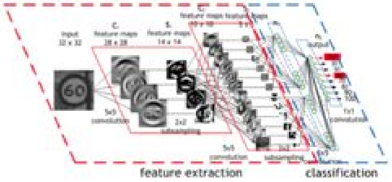
providing lift for classification and forecasting models

*Deep* Neural Networks



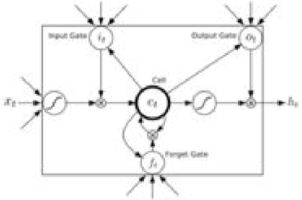

feature extraction and classification of images

*Convolutional* Neural Networks




for sequence of events, language models, time series, etc.

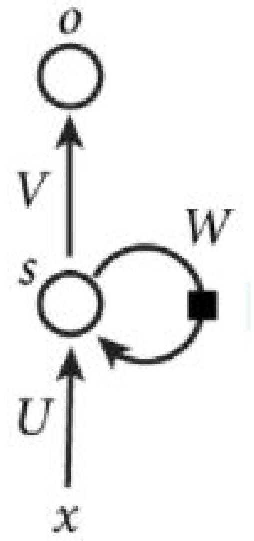
*Recurrent* Neural Networks



# Recurrent Neural Network: Intro

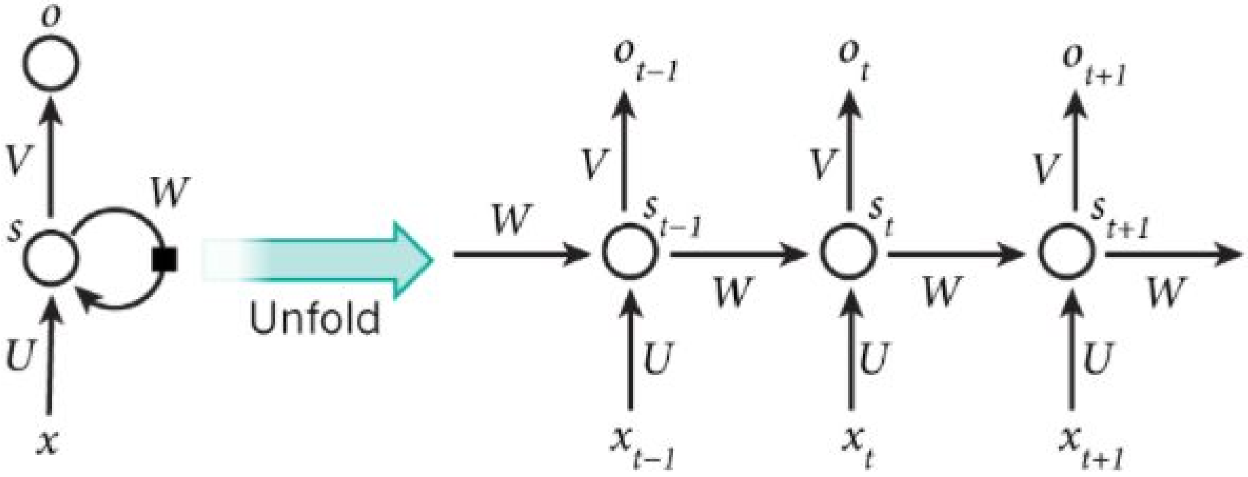
---

- RNN: family of NN for processing **sequential data**
- **Example**: *predicting the next word of a sentence*
- **Recurrent**: performing the **same task** for every element of the sequence
- **Output**: dependent on previous computation
- RNN have **memory**



# Recurrent Neural Network: Intro

- Unfolding RNN



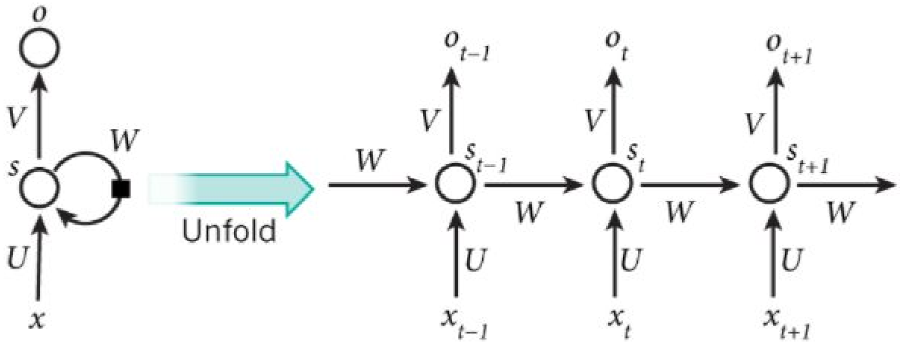
# Recurrent Neural Network: Intro

- xt : **input** at timestamp t
- st : **hidden** state

$$s_t = f(Ux_t + Ws_{t-1})$$

- ot : **output** at timestamp t

$$o_t = \mathrm{softmax}(Vs_t)$$

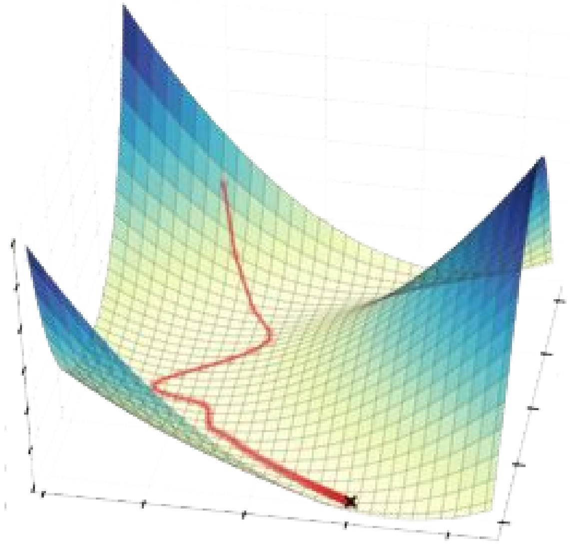


# Recurrent Neural Network: Training

- **Learning** the **parameters**: *U,V,W*
- **SGD**: Stochastic Gradient Descent
  - Minimizing the **total loss** of the training data
  - **Iterative** process
  - Nudge the parameters in the **directions** of the **gradients**

$$\frac{\partial L}{\partial U}, \frac{\partial L}{\partial V}, \frac{\partial L}{\partial W}.$$

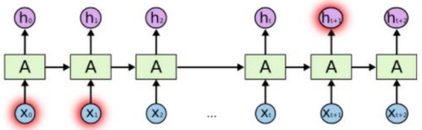
- **BPTT:** Backpropagation Through Time
  - **Modified** version of **backpropagation algorithm** for **computing** the **gradients**



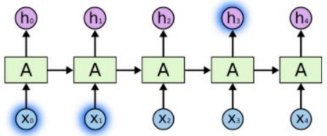
# Long-Term Dependency Problem

---

- **Example:** *Prediction of next word*
  - *"The <u>clouds</u> are in the" → ? ["Sky"]*



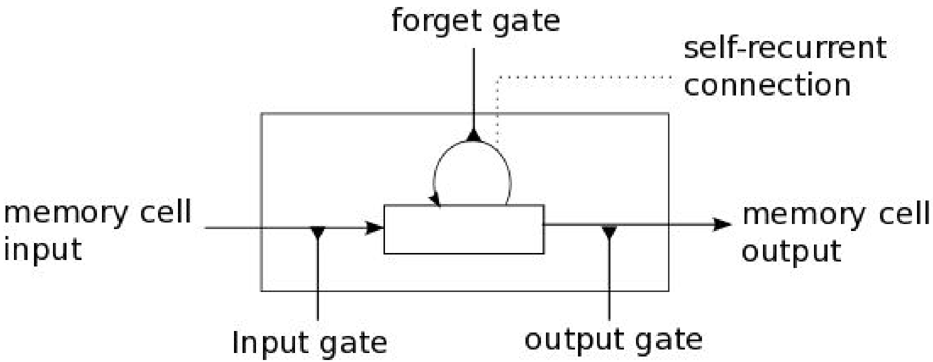

  - *"I grew up in <u>Italy</u> (...) I speak fluent" → ? ["Italian"]*




- **Vanishing Gradient Problem**
  - Gradients become too large or too small during the iterative process of parameter learning

# LSTM - Long Short-Term Model

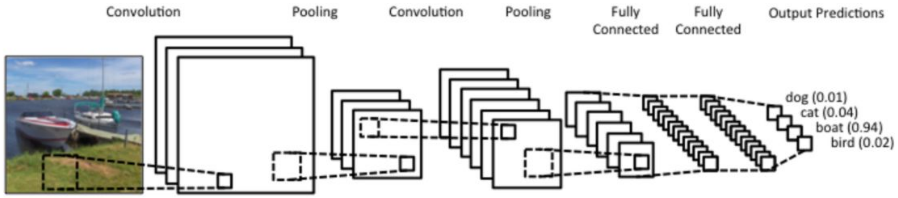
- Designed to **handle long-term** dependency
- **Memory cell** unit
  - *Forget Gate*: information to throw away (in the cell state)
  - *Input Gate*: information to store (in the cell state)
  - *Output Gate:* what to output



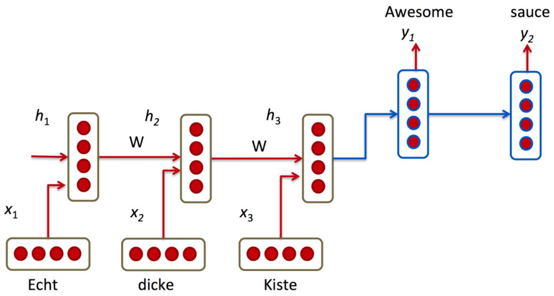
# RNN vs CNN

---

- **CNN:** Neural network able to recognize patterns across the space (i.e.: component of an image)




- **RNN:** NN able to capture pattern from sequential data



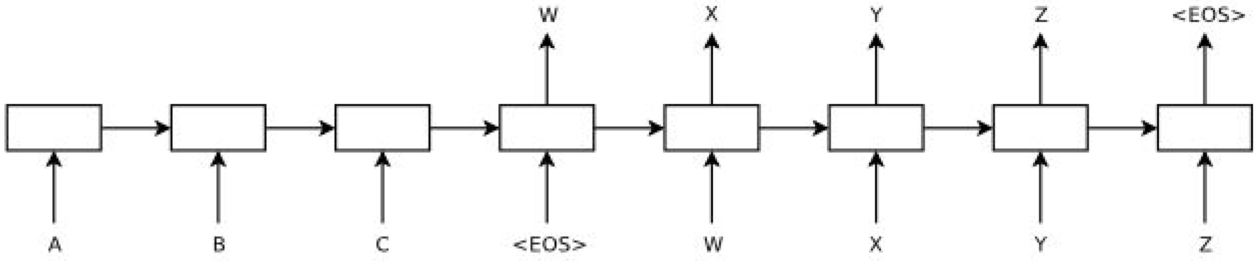

- CNN + RNN in joint architectures!

# Recurrent Neural Network: Applications

- Sentence Modelling
- Click Prediction
- Location Prediction
- Language Translation
- Sentiment Analysis
- Image Captioning and Description
- Speech Recognition
- Question/Answering Systems
- Text Generation

# RNN Applications: Machine Translation

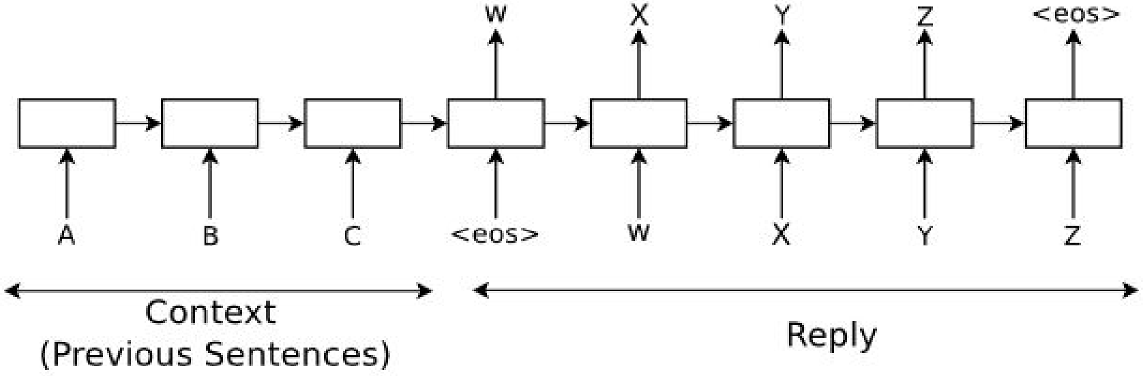
- Sequence to Sequence Model* *[Sutskever, Vinyals, Le, 2014]*
    - LSTM based model




*__Sequence to Sequence Learning with Neural Networks__, *I.Sutskever, O.Vinyals, Q.V.Le (2014)*

# RNN Applications: Conversational Systems

- Neural Conversational Model*
  *[Sutskever, Vinyals, Le, 2014]*
  - LSTM based model

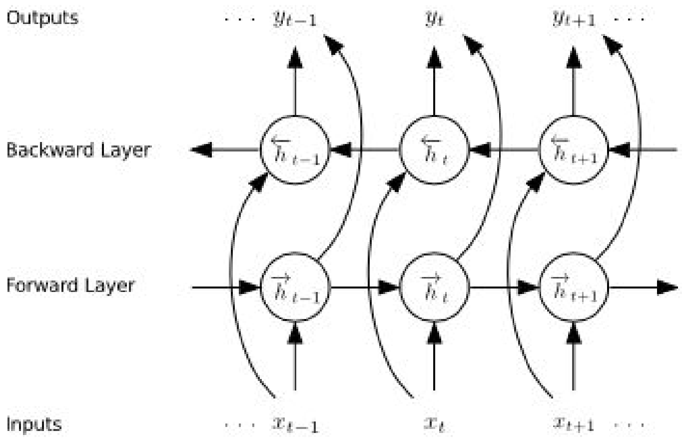



**Machine:** *hi*
**Human:** *hello*
**Machine:** *could you please let me know what are the operating systems you are using ?*
**Human:** *i am using linux*
**Machine:** *ok it was in the ssl vpn on which you stated it was broken*
**Machine:** *please check if you are logged in to corp network*
**Machine:** *what is the error that you are running please*
**Human:** *i am seeing an error related to vpn*
**Machine:** *what is the error message that you are getting when connecting to vpn using network connect ?*
**Human:** *connection refused or something like that*
**Machine:** *may i know the version of network connect you connect ?*
**Human:** *i am not sure i know that*
**Machine:** *is the network connect prompting for any user certificate*
**Machine:** *?*
**Human:** *yes*
**Machine:** *are you at home or at work at the moment ?*
**Machine:** *office ?*
**Human:** *i am at home*

*A Neural Conversational Model**, *O.Vinyals, Q.V.Le (2015)*

# RNN Applications: Speech Recognition

- End to End Speech Recognition*
  *[Graves, Jaitly, 2014]*
  - Bi-LSTM based model

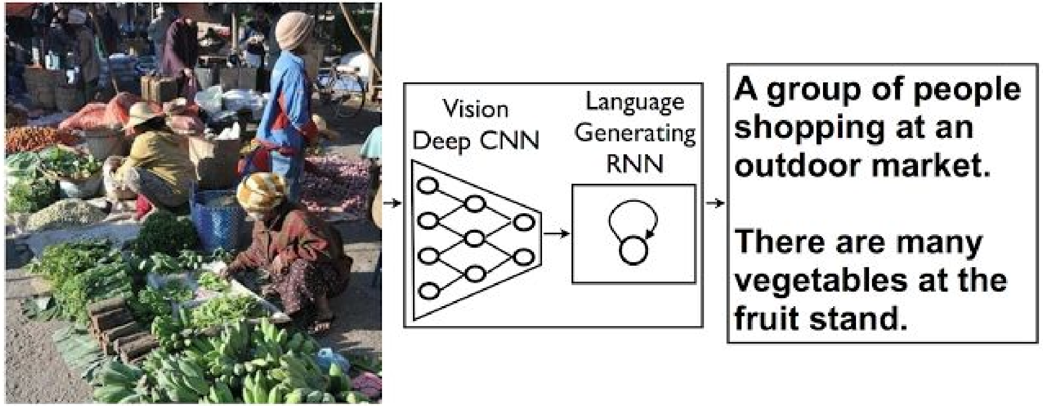



*__Towards End-to-End Speech Recognition with Recurrent Neural Networks__, *A.Graves, N.Jaitly (2014)*

# RNN Applications: Image Captioning

---

- Generating Captions describing an Image *[O.Vinyals et al, 2015]*
  - RNN + CNN based model





*Show and Tell: A Neural Image Caption Generator, *O.Vinyals (2015)*

# RNN Applications: Demo

- Generating Polyphonic Music
- Sentence Generation from Picture
- Handwriting Generation Demo
- Sentiment Analysis

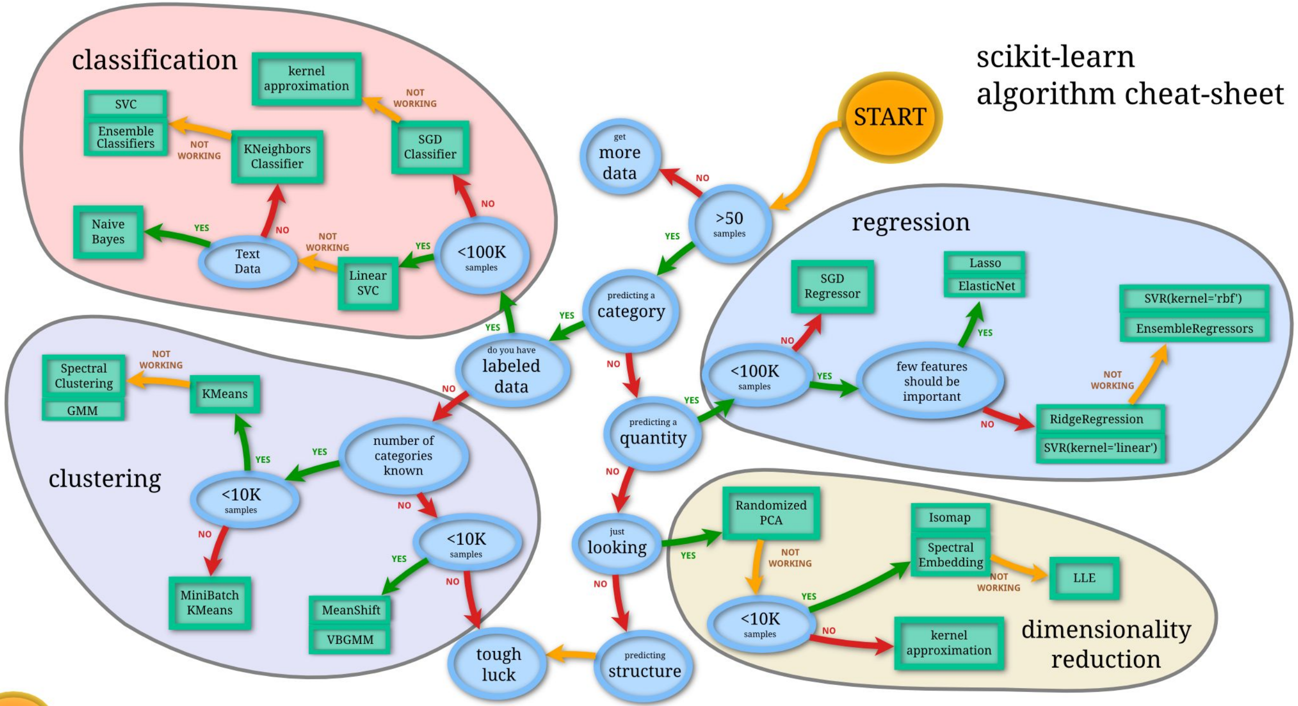
# Conclusion

---

- Recurrent Neural are able to model sequences
- Training RNN is hard because the vanishing problem
- LSTM tackle the Long-Term Dependency Problem
- Mostly useful in NLP related problems

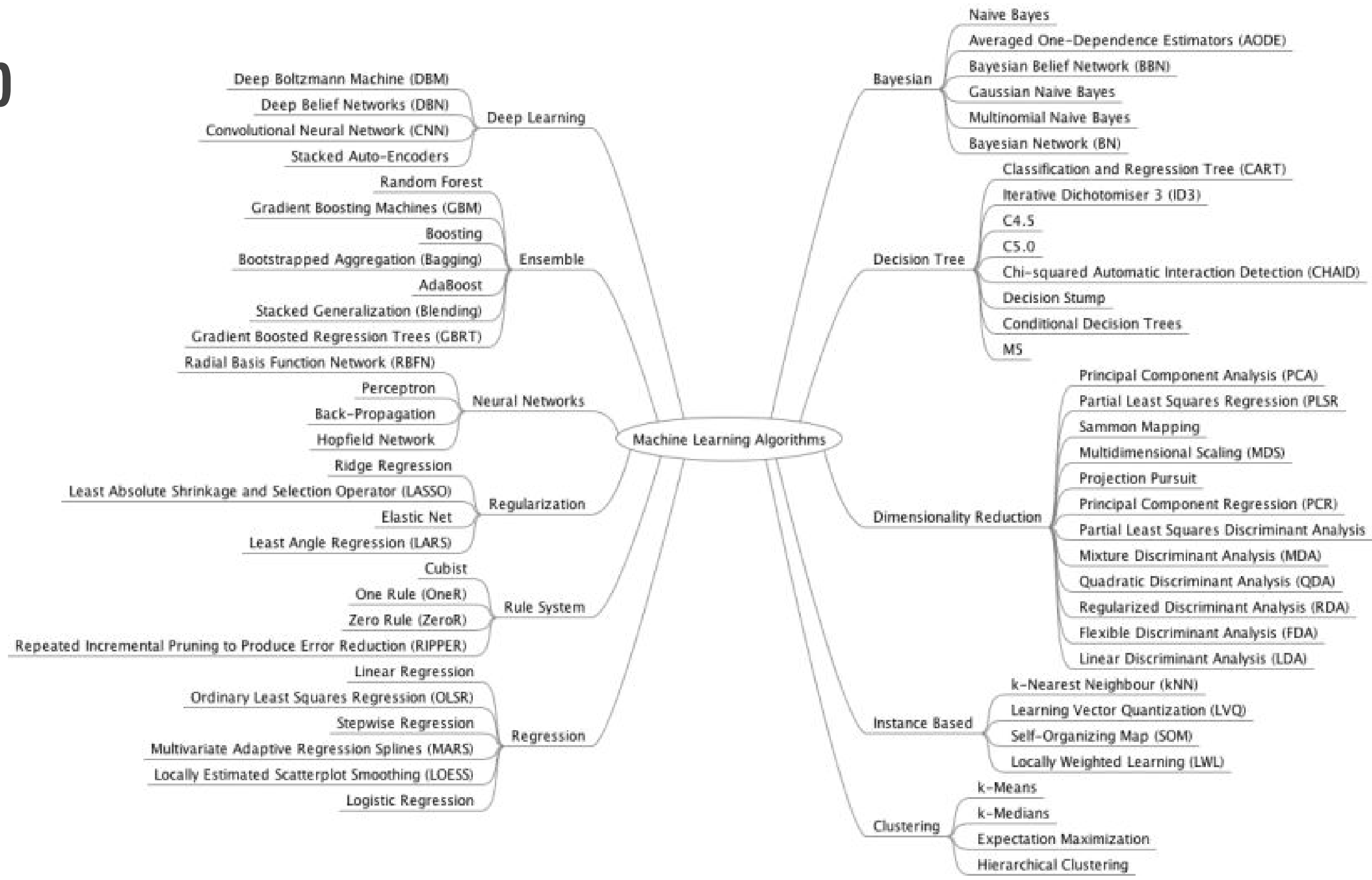
# Useful Resources

# ML Cheat-Sheet



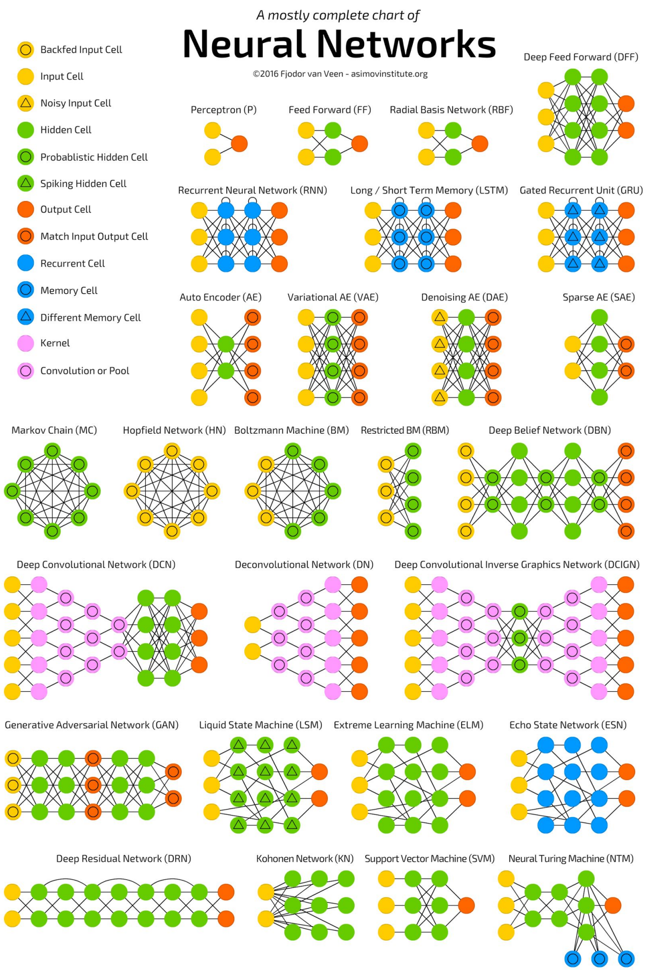

Back

# ML Mind Map



Machine Learning Algorithms
Deep Learning
Deep Boltzmann Machine (DBM)
Deep Belief Networks (DBN)
Convolutional Neural Network (CNN)
Stacked Auto-Encoders
Ensemble
Random Forest
Gradient Boosting Machines (GBM)
Boosting
Bootstrapped Aggregation (Bagging)
AdaBoost
Stacked Generalization (Blending)
Gradient Boosted Regression Trees (GBRT)
Neural Networks
Radial Basis Function Network (RBFN)
Perceptron
Back-Propagation
Hopfield Network
Regularization
Ridge Regression
Least Absolute Shrinkage and Selection Operator (LASSO)
Elastic Net
Least Angle Regression (LARS)
Rule System
Cubist
One Rule (OneR)
Zero Rule (ZeroR)
Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
Regression
Linear Regression
Ordinary Least Squares Regression (OLSR)
Stepwise Regression
Multivariate Adaptive Regression Splines (MARS)
Locally Estimated Scatterplot Smoothing (LOESS)
Logistic Regression
Bayesian
Naive Bayes
Averaged One-Dependence Estimators (AODE)
Bayesian Belief Network (BBN)
Gaussian Naive Bayes
Multinomial Naive Bayes
Bayesian Network (BN)
Decision Tree
Classification and Regression Tree (CART)
Iterative Dichotomiser 3 (ID3)
C4.5
C5.0
Chi-squared Automatic Interaction Detection (CHAID)
Decision Stump
Conditional Decision Trees
M5
Dimensionality Reduction
Principal Component Analysis (PCA)
Partial Least Squares Regression (PLSR
Sammon Mapping
Multidimensional Scaling (MDS)
Projection Pursuit
Principal Component Regression (PCR)
Partial Least Squares Discriminant Analysis
Mixture Discriminant Analysis (MDA)
Quadratic Discriminant Analysis (QDA)
Regularized Discriminant Analysis (RDA)
Flexible Discriminant Analysis (FDA)
Linear Discriminant Analysis (LDA)
Instance Based
k-Nearest Neighbour (kNN)
Learning Vector Quantization (LVQ)
Self-Organizing Map (SOM)
Locally Weighted Learning (LWL)
Clustering
k-Means
k-Medians
Expectation Maximization
Hierarchical Clustering

# The Neural NW Zoo



# Thanks for the Attention

**Massimiliano Ruocco, PhD**
*Research Scientist, Telenor Research*
*Adjunct Associate Professor, DART Group, NTNU*

artificial intelligence lab

telenor NTNU

# Exercises

# Exercise

---

https://github.com/ruoccoma/ml_talk_hioa