Fisica, Tecnologia, Intelligenza artificiale e Machine Learning: applicazioni e sviluppi

F. Conventi, E. Rossi

Machine Learning 4 Nutrition Science Project (ML4N): II Workshop

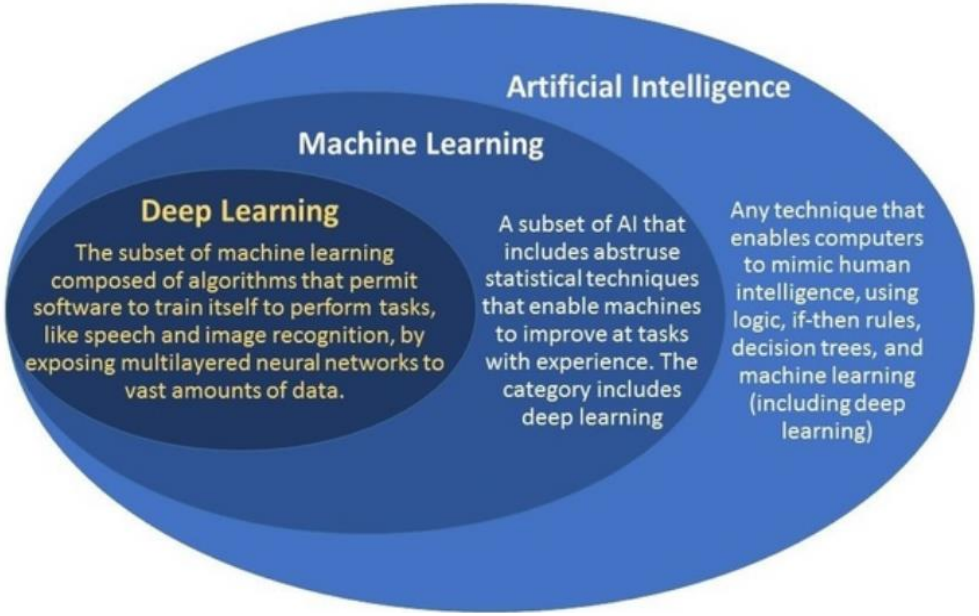
INFN
Istituto Nazionale di Fisica Nucleare

Università degli Studi di Napoli Federico II - Dipartimento di
FISICA "ETTORE PANCINI"

CERN

Università degli Studi di Napoli *Parthenope*
dipartimento ingegneria

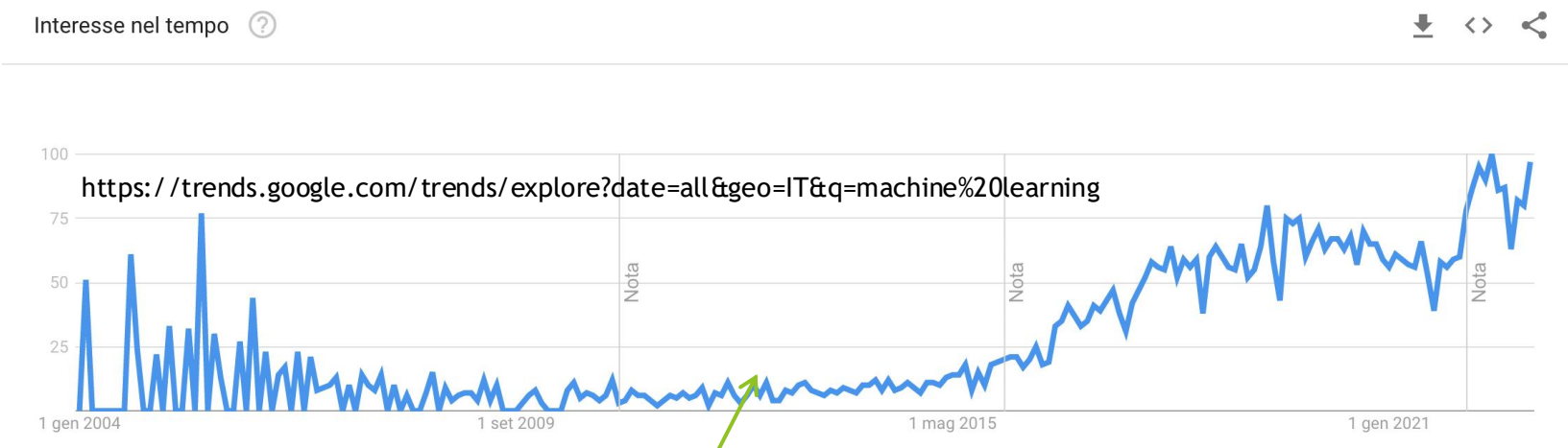
Cosa intendiamo per ML ?



2004 2015 2022

Trends mondiale per ricerche che contengono la parola **ML**

AI and Machine Learning



Trends italiano per ricerche che contengono la parola **ML**

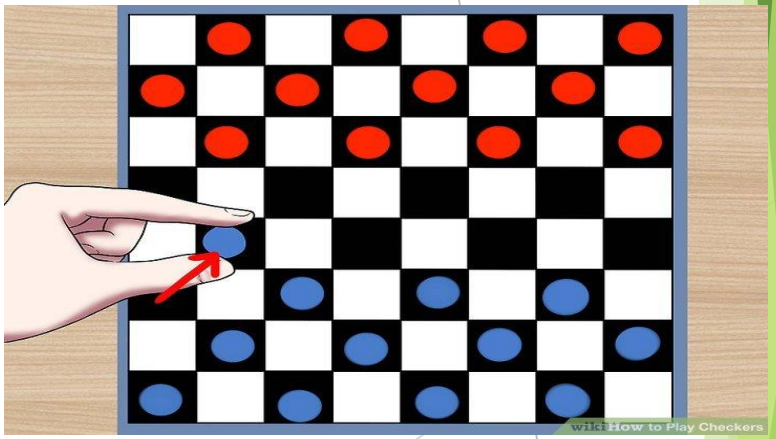
Una possibile definzione del Machine Learnin

Arthur Samuel (1959)

MachineLearning: Campo di ricerca che studia la possibilità di apprendimento di un compito da parte dei computers senza che vengano esplicitamente programmati per quel dato compito

TomMitchell (1998)
Well-posed Learning Problem:

Diremo che dato un certo compito **T** si verifica apprendimento in relazione all'esperienza **E** se la performance **P** migliora con l'esperienza **E**.



https://chessprogramming.wikispaces.com/Arthur+Samuel

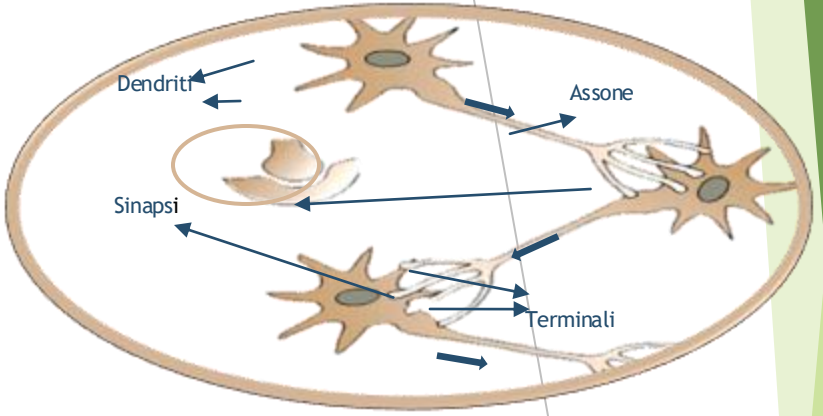
Reti Neurali

Un neurone è formato da un corpo detto soma e da due tipi di diramazioni: i dendriti e l'assone.

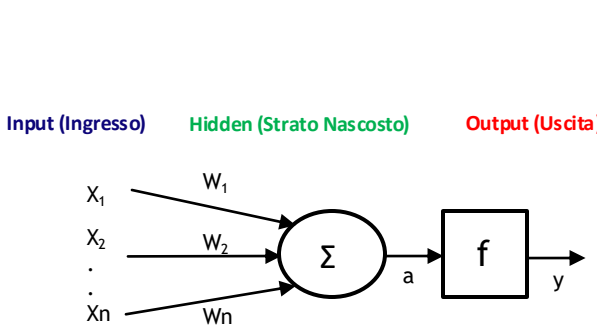
Ogni neurone riceve segnali di **input** da altri neuroni attraverso i **dendriti** e invia segnali di **uscita** attraverso i **terminali dell'assone** collegati a quest'ultimo.

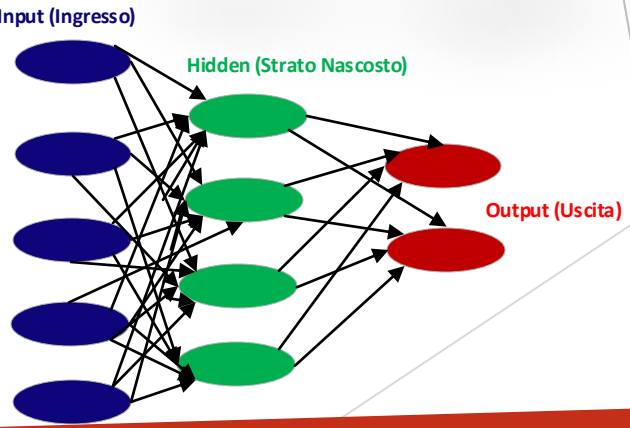
I **neuroni artificiali** hanno una struttura simile a quella dei neuroni umani. Acquisisce informazioni in ingresso, le elabora attribuendo un peso ad ognuno di esse e fornisce una risposta in uscita.

Neuroni Umani



Neurone Artificiale





Apprendimento: Un semplice esempio

Diremo che dato un certo compito **T** si verifica apprendimento in relazione all'esperienza **E** se la performance **P** migliora con l'esperienza **E.**

Il software che usi per legger le e-mail in genere "registra" quali emails vengano "taggate" come spam (mail indesiderate)…
Su questa base apprende come filtrarle al meglio e rimuoverle automaticamente

T → Classificare emails come spam o non-spam.
E →Registrare quali mail vengono scartate o meno dall'utente.
P →Il numero di mail correttamente identificate come spam

Diversi tipi di apprendimento nel Machine learning

Supervised learning

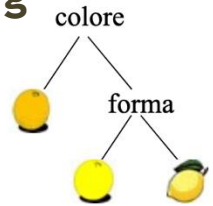
(arancio, rotondo, classe=)

(giallo, lungo, classe=)

(giallo, rotondo, classe=)

(giallo, lungo, classe=)

algoritmo di apprendimento
supervisionato



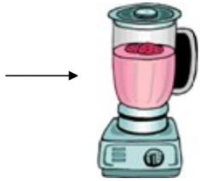
Unsupervised

(arancio, rotondo)

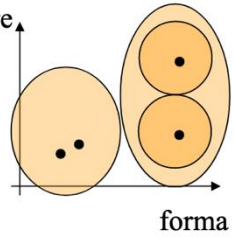
(giallo, rotondo)

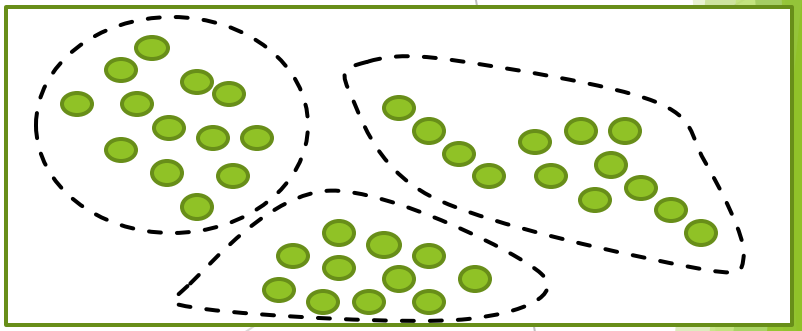
(giallo, rotondo)

(giallo, lungo)

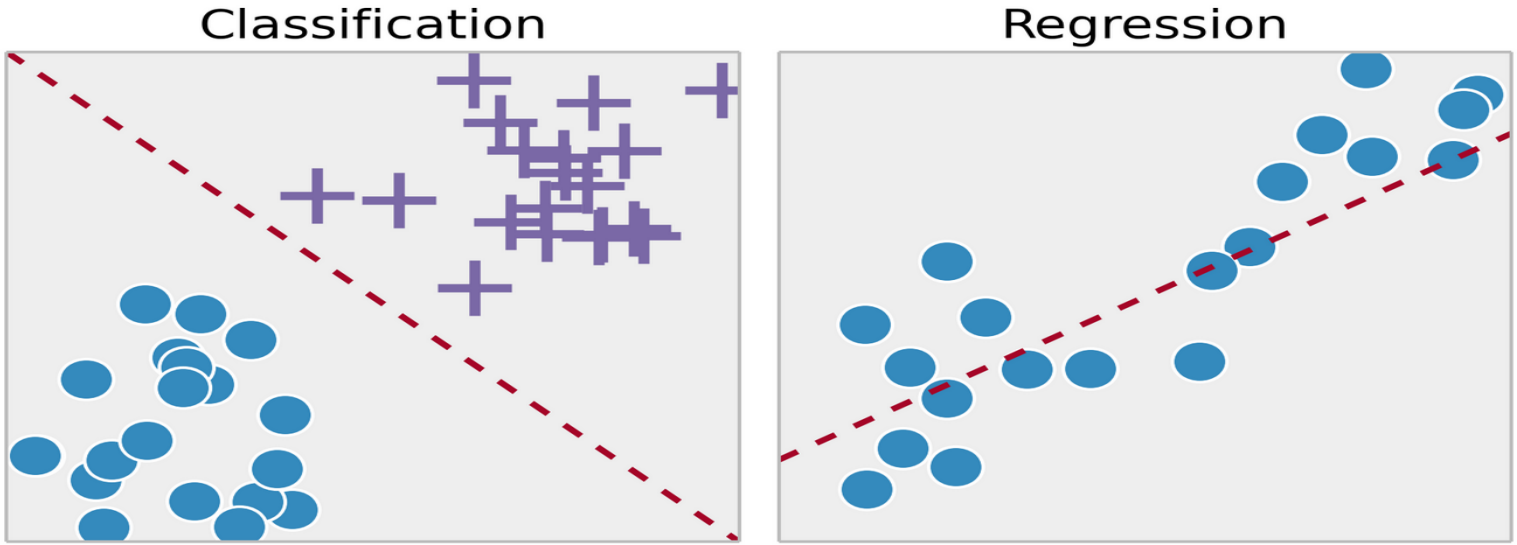


algoritmo di apprendimento
non supervisionato





Algoritmi Supervisionati



Discrete valued output (0 or 1)

Predict continuous valued output

Negli algoritmi supervisionati la risposta “esatta” e’ fornita al computer al fine di permettere l’apprendimento

Algoritmi supervisionati

Regression problems:

Previsioni riguardanti gli andamenti del mercato.
Previsioni sugli andamenti dei prezzi di beni.
Assegnare l'eta' ad una persona partendo da una sua foto.

Classification problems:

Riconoscimento di un testo scritto a mano
Spam filtering per un software di email.
Diagnosi medica a partire da immagini

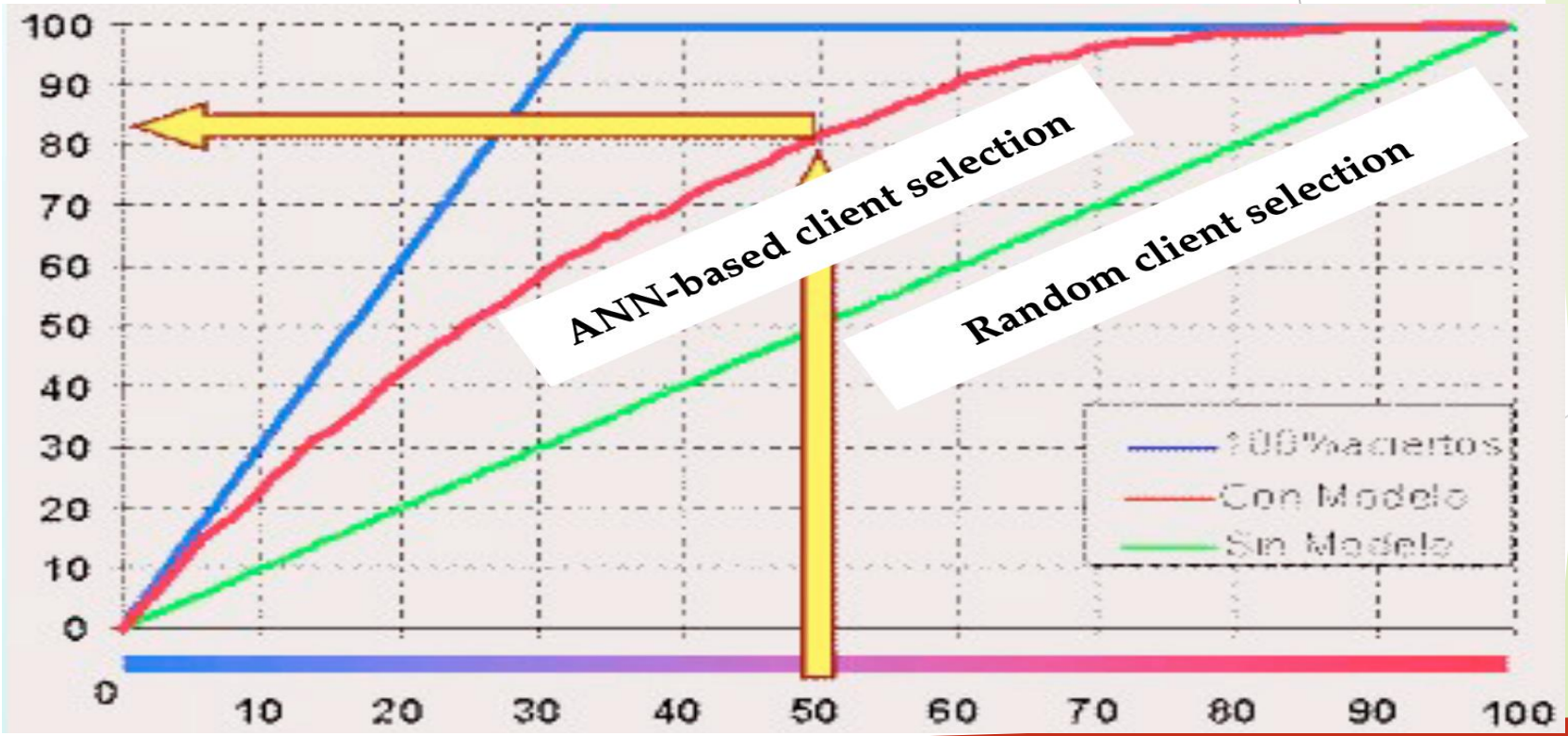
Un esempio dal campo del marketing…

Una banca vuole promuovere una nuova tipologia di carta di credito ai suoi clienti

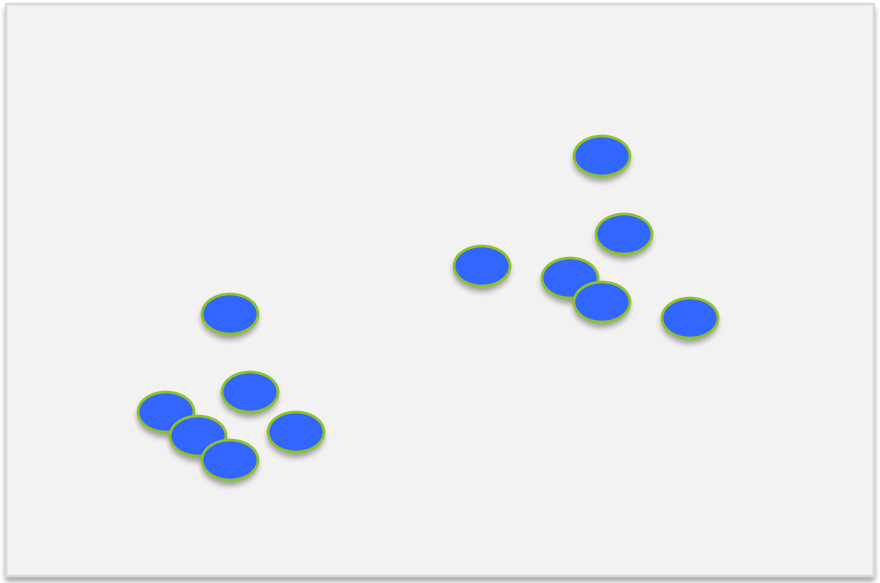
Sicuro ma lento e costoso!!

Posso contattare solo 5% dei clienti e da questi dati posso fare apprendere un algoritmo che selezioni i clienti che con maggior probabilità prenderanno la nuova carta

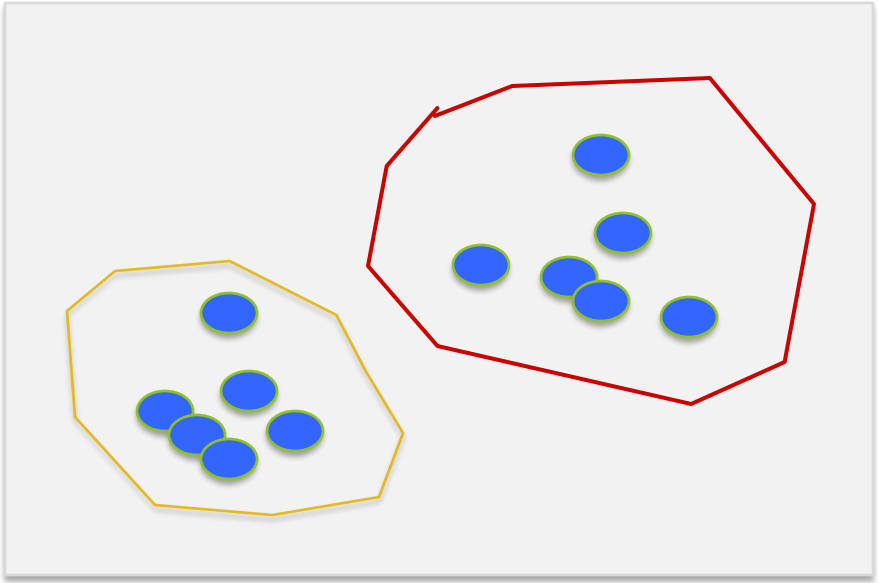
Un esempio dal campo del marketing…



Unsupervised algorithm

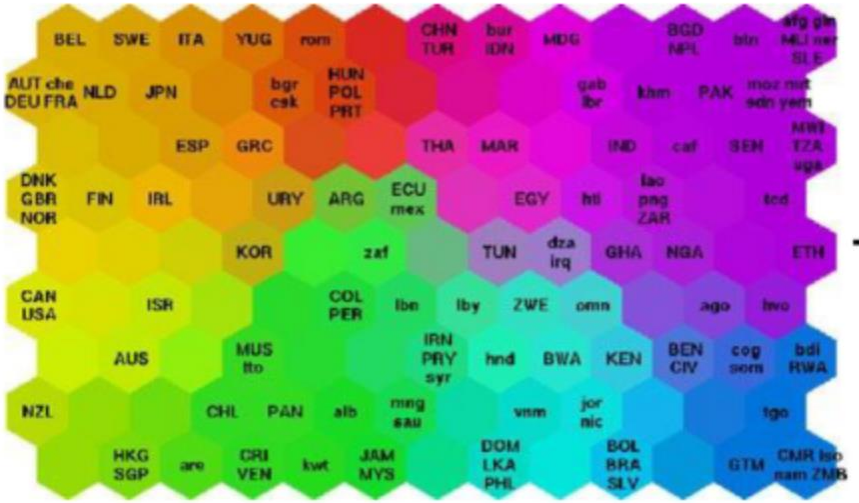


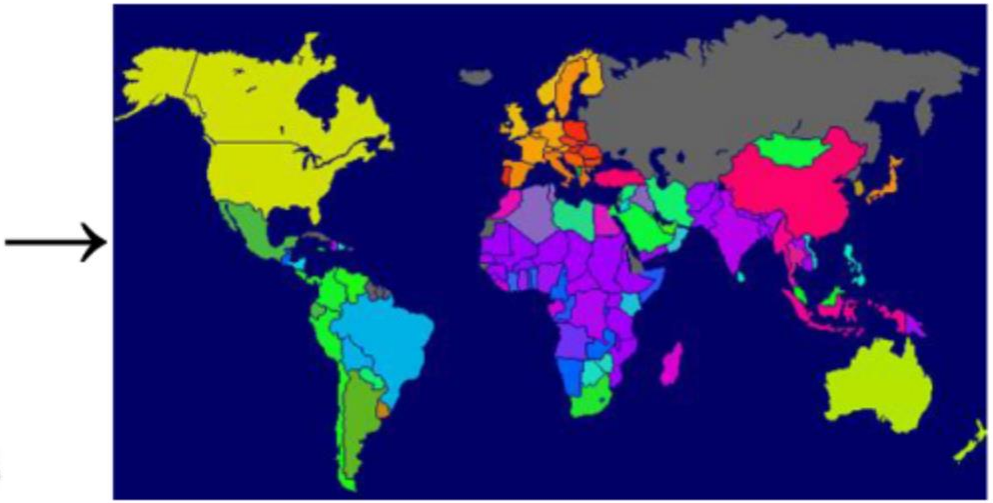
Unsupervised algorithm



- Clustering: Grouping similar points together within clusters.
- **Density estimation**: Estimating a probability density that can explain the distribution of the data points.
- Dimension reduction: Getting a simple representation of high-dimensional data points by projecting them onto a lower-dimensional space. This technique is notably used for data visualization.
- **Manifold learning** (or nonlinear dimension reduction): Finding a low-dimensional manifold containing the data points.

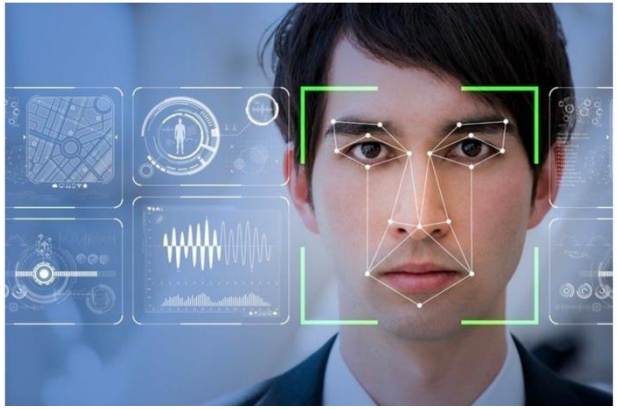
Unsupervised algorithm



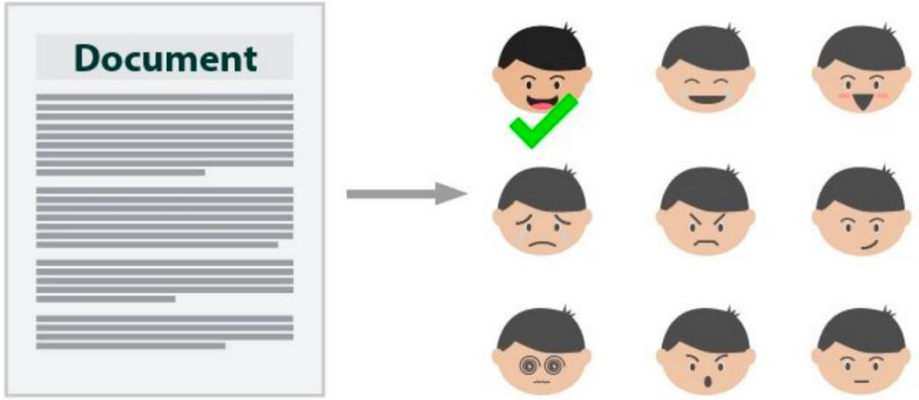


Dove applichiamo il ML?

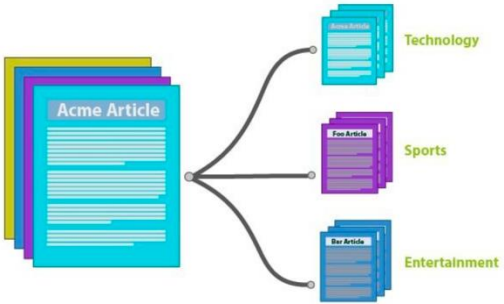
Riconoscimento delle immagini



Riconoscimento delle emozioni



Classificazione delle news



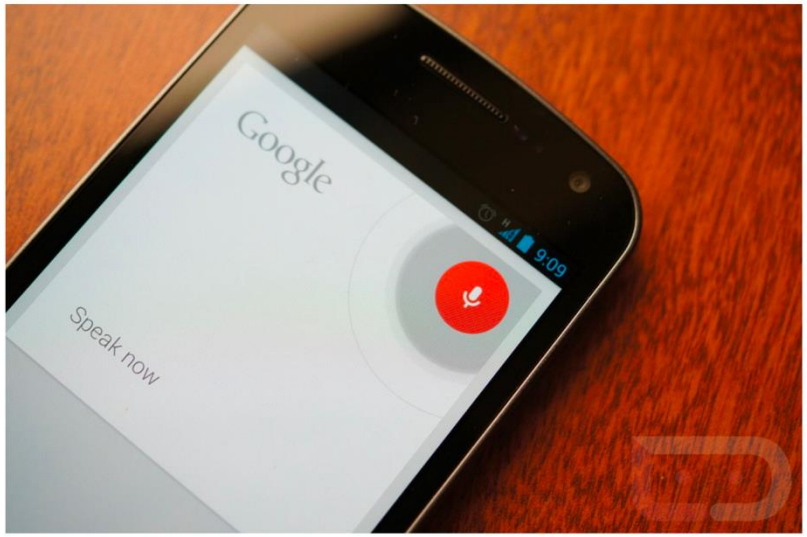
Video sorveglianza

Riconoscimento del parlato

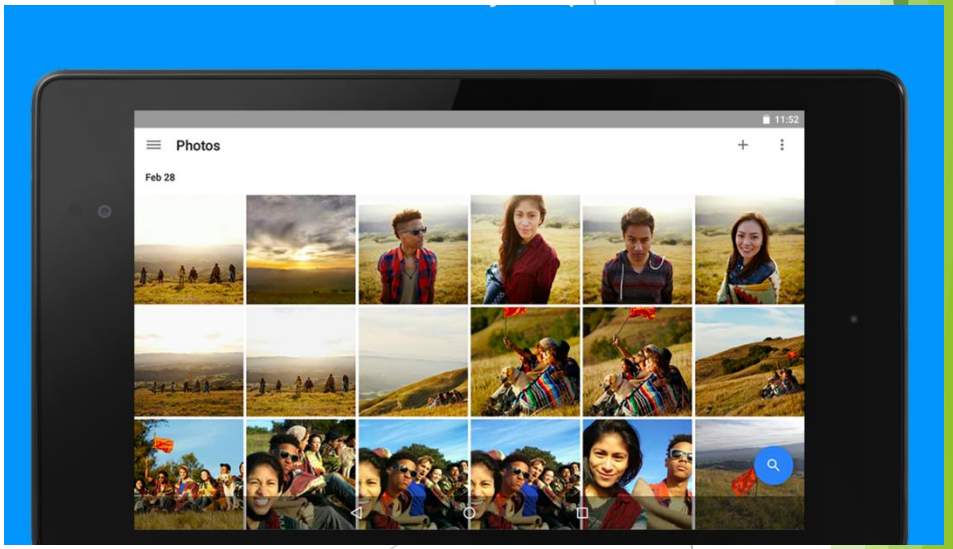
Alcune applicazioni ML di…

Google

Google voice search (→ ehi google)

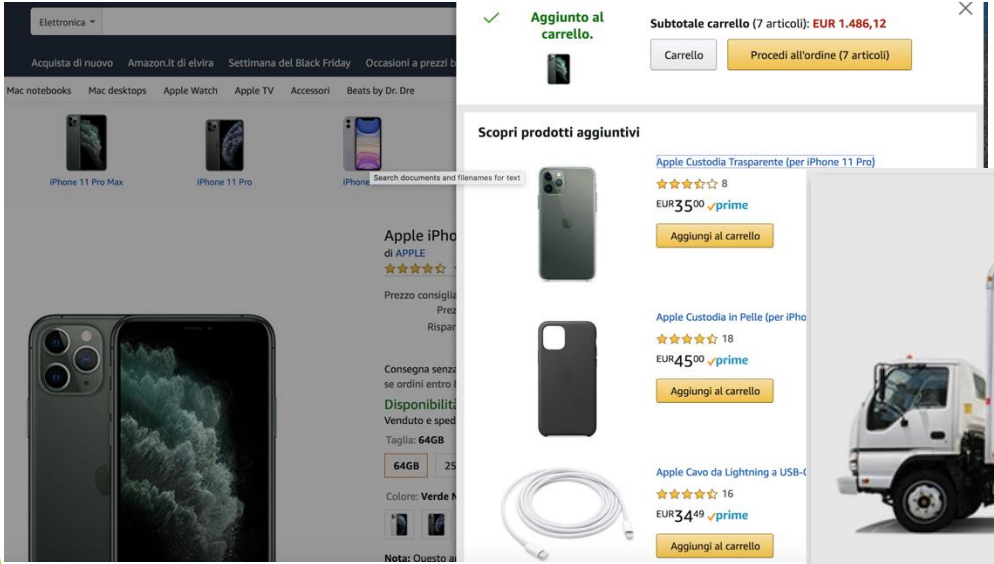


Google Photos



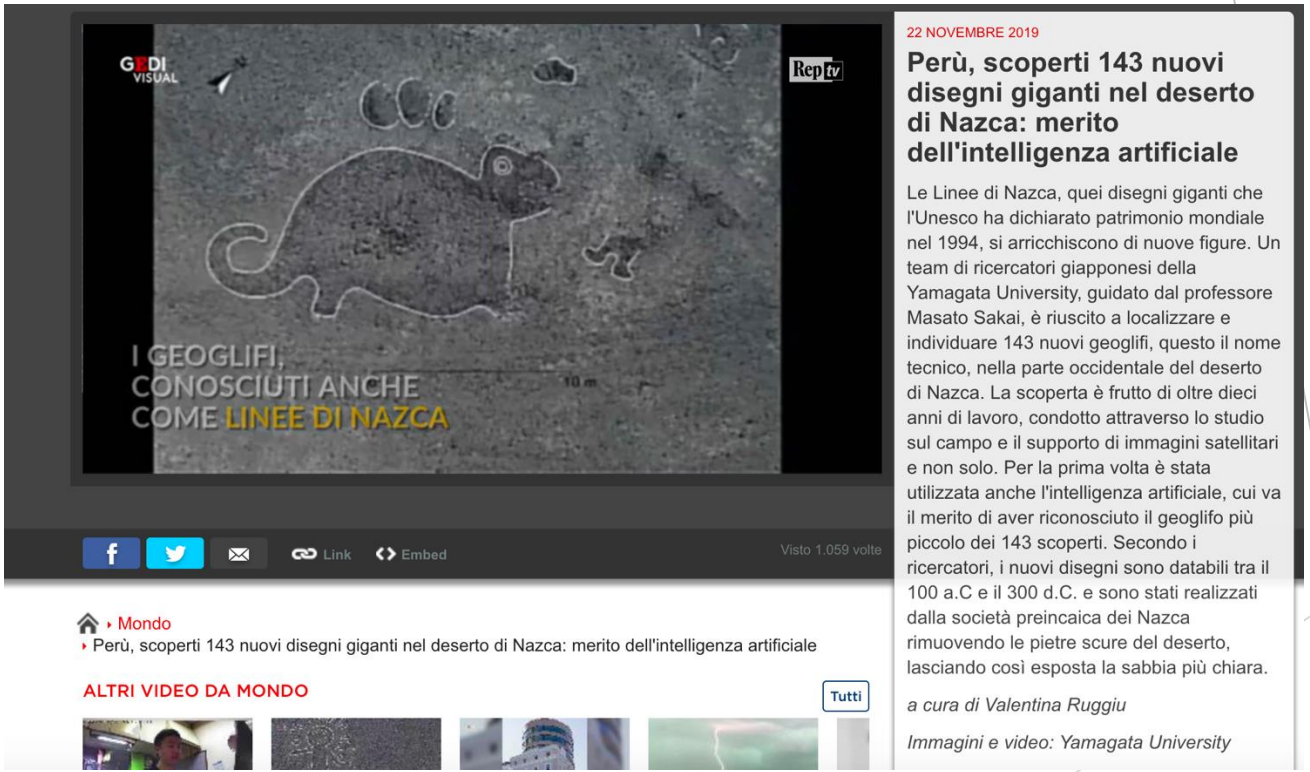
Alcune applicazioni ML di…

amazon



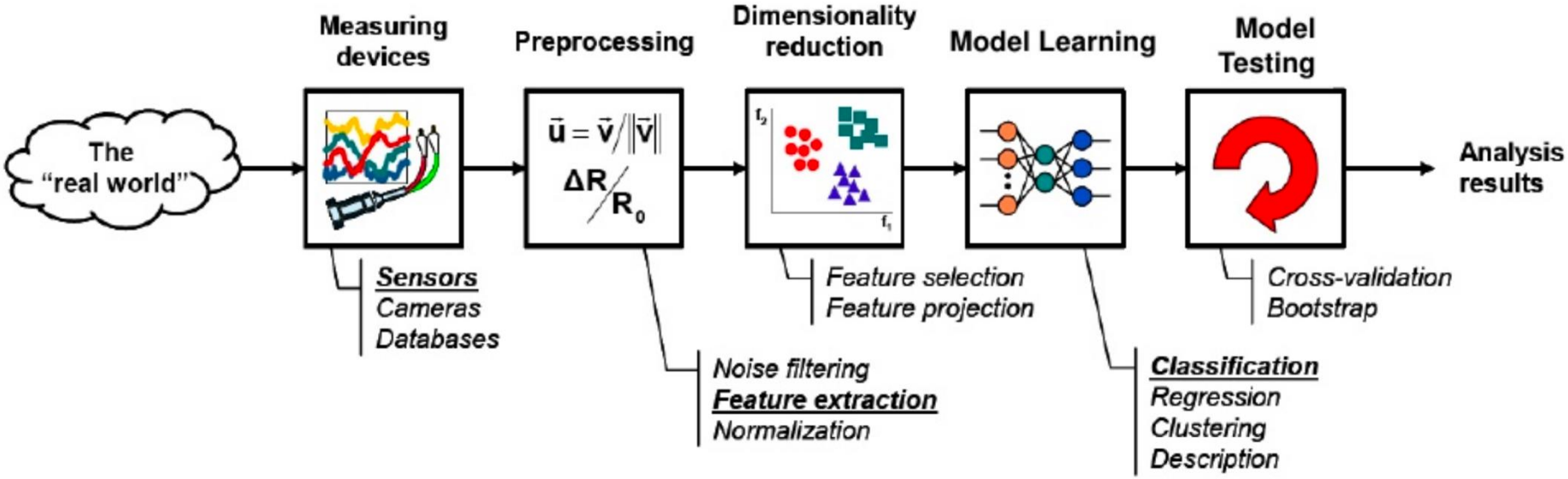
AMAZON knows it !

La Repubblica:
scoperti 143 nuovi disegni giganti nel deserto di Nazca con la AI

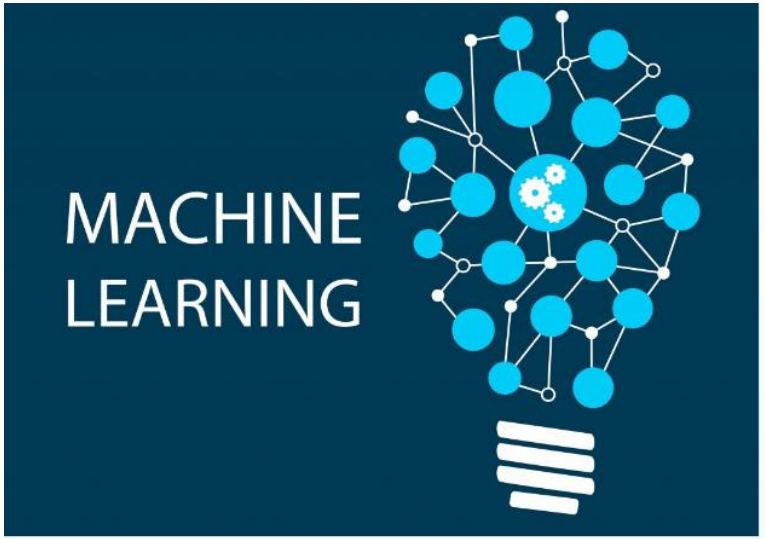


https://video.repubblica.it/mondo/peru-scoperti-143-nuovi-disegni-giganti-nel-deserto-di-nazca-merito-dell-intelligenza-

The Learning Process

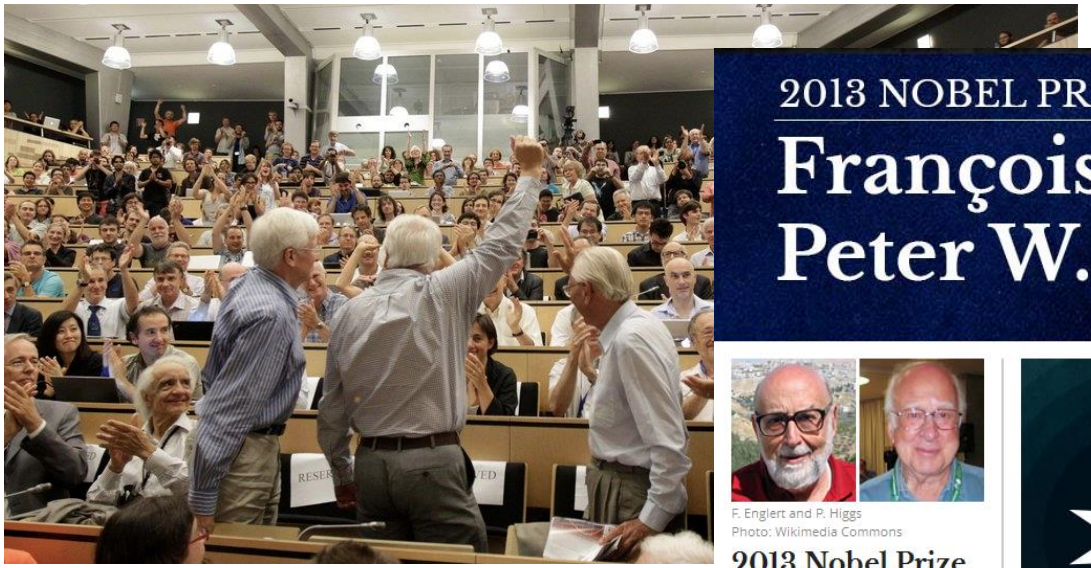


Alcune applicazioni ML di…



Cosa hanno in comune Machine Learning e la fisica delle Particelle?

Abbiamo un problema? Il bosone di Higgs



- È realmente una particella elementare o è composta da altre particelle?
- Interagisce con la materia oscura?

2013 NOBEL PRIZE IN PHYSICS

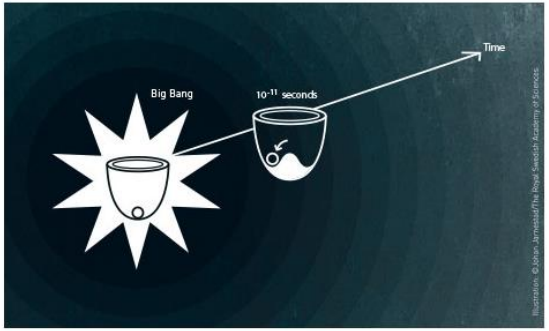
François Englert
Peter W. Higgs

© The Nobel Foundation. Photo: Lovisa Engblom.

F. Englert and P. Higgs
Photo: Wikimedia Commons

2013 Nobel Prize in Physics

The Nobel Prize in Physics 2013 was awarded jointly to François Englert and Peter W. Higgs *"for the theoretical discovery of a mechanism that contributes to our*

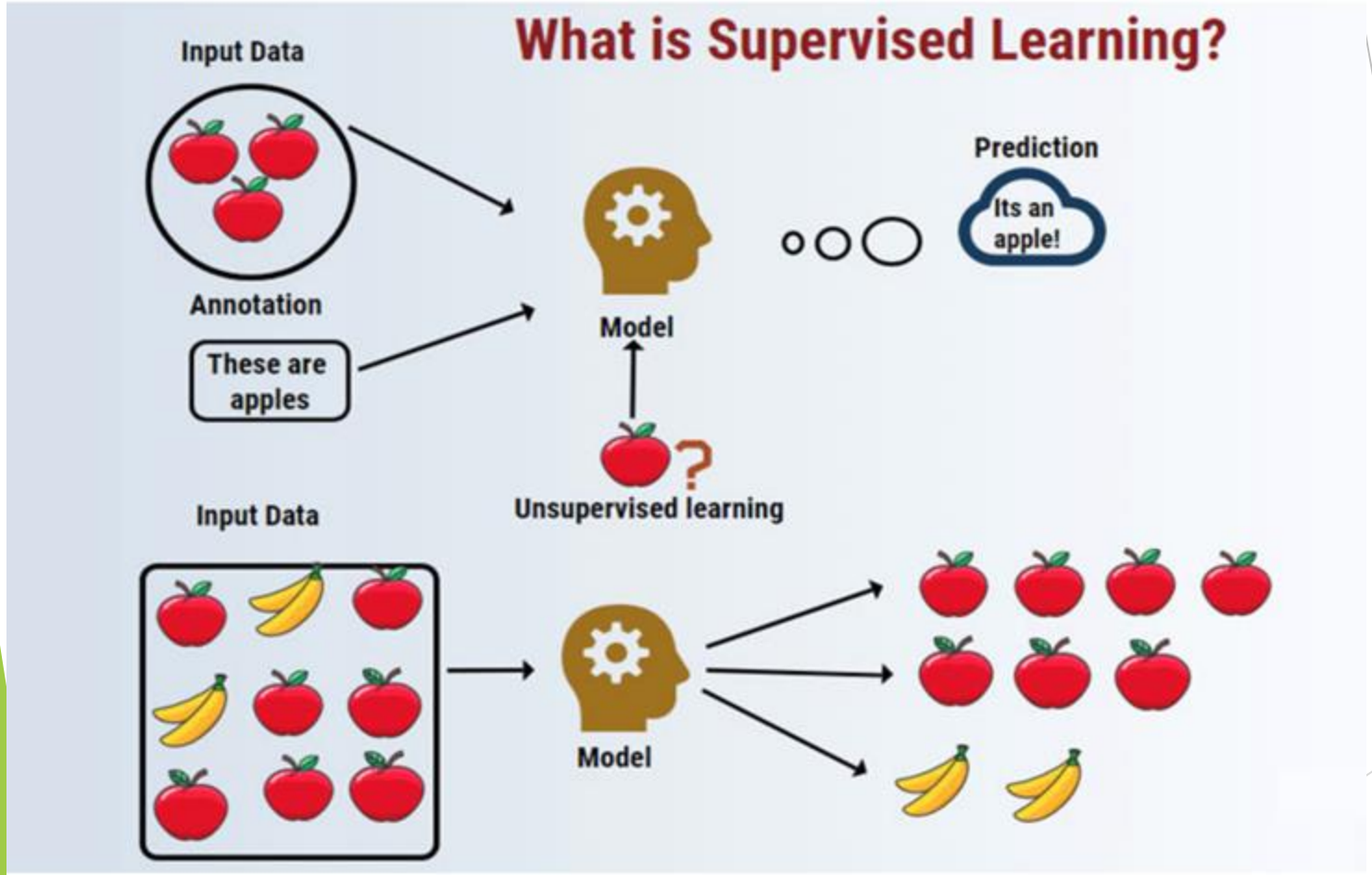


What Happened after the Big Bang?

Announcements of the 2013 Nobel Prizes

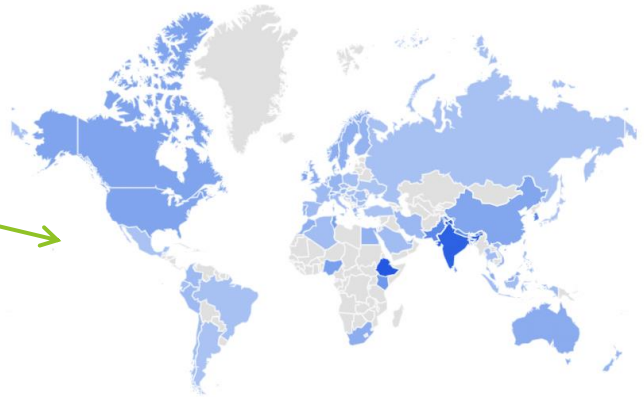
Physiology or Medicine:
Announced Monday 7 October
Physics:
Tuesday 8 October, 11:45 a.m. CET at the earliest
Chemistry:
Wednesday 9 October, 11:45 a.m. CET at the earliest
Literature:
Thursday 10 October 1.00 p.m. CET
Peace:

Backup


What is Supervised Learning?
Input Data
Annotation
These are apples
Model
Prediction
Its an apple!
Unsupervised learning
Input Data
Model

AI and Machine Learning

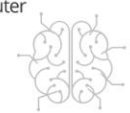
Trends mondiale per ricerche che contengono la parola **ML**



1950

Turing Test

An AI test proposed by Alan M. Turing to determine whether a computer can "think."



1951

Neural Network Machine

Marvin Minsky and Dean Edmonds build the first machine, able to learn.

1955

Computer Gaming Program

IBM's Arthur Samuel develops the program that enables a computer to learn to play to win checkers.

1967

Pattern Recognition

The 'nearest neighbor algorithm' is created by T.M. Cover and P.E. Hart was originally used to map routes.

1959

MIT Computer Science and Artificial Intelligence Lab

John McCarthy and Marvin Minsky found a center for computing research and innovation.

1958

Perceptron

First artificial neural network developed by Frank Rosenblatt for visual recognition tasks while working at the Cornell Aeronautical Laboratory.

1970

Automatic Differentiation (AD)

Seppo Linnainmaa published the general method to numerically evaluate the derivative of a function specified by a computer program.

1975

Genetic Algorithms (GA)

John Henry Holland developed this method to generate high-quality solutions to optimization and search problems.

1979

Mobile Robot

Shakey, the first general-purpose mobile robot able to reason about its own actions, is developed at the Artificial Intelligence Center of Stanford Research Institute (now SRI International).

1990-present

Machine Learning Applications

Scientists begin creating programs to analyze large amounts of data and learn. The fields of computational complexity via neural networks and super-Turing computation begin.

1997

Deep Blue beats Garry Kasparov

IBM develops the computer based on the original chess playing machine Feng-hsiung Hsu began working on in 1985.

2006

Cloud-Computing

Google CEO Eric Schmidt first uses the term in an industry conference.

2014

DeepFace

Facebook introduces a software algorithm that recognizes or verifies individuals in photos.

2012

SIRI

Apple introduces speech recognition and intelligent personal assistant application for its iOS operating system.

2009

Self-Driving Cars

Google begins testing prototypes.

2015

Distributed Machine Learning Toolkit (DMTK)

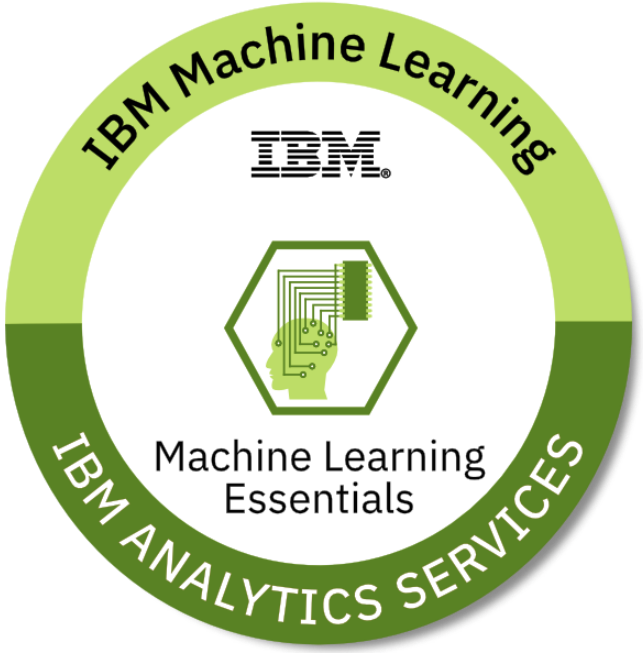
Microsoft provides developers with the underlying machine framework to use big data for big models to generate better accuracy in applications.

2017

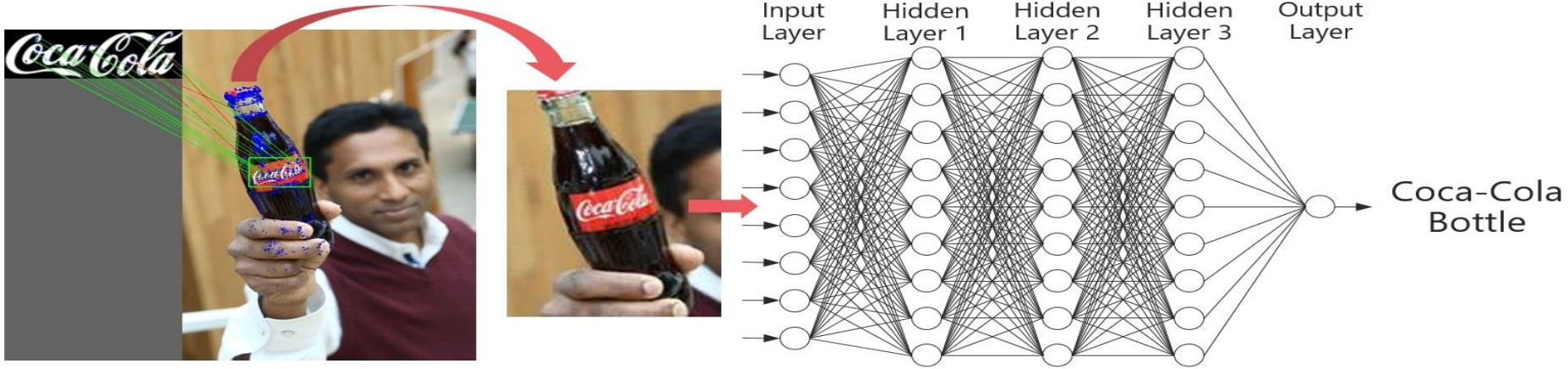
Capsule Neural Networks

Google's Geoffrey Hinton and his team introduce this improved architecture model of hierarchical artificial neural network relationships that impact deep learning.

Alcune applicazioni ML c

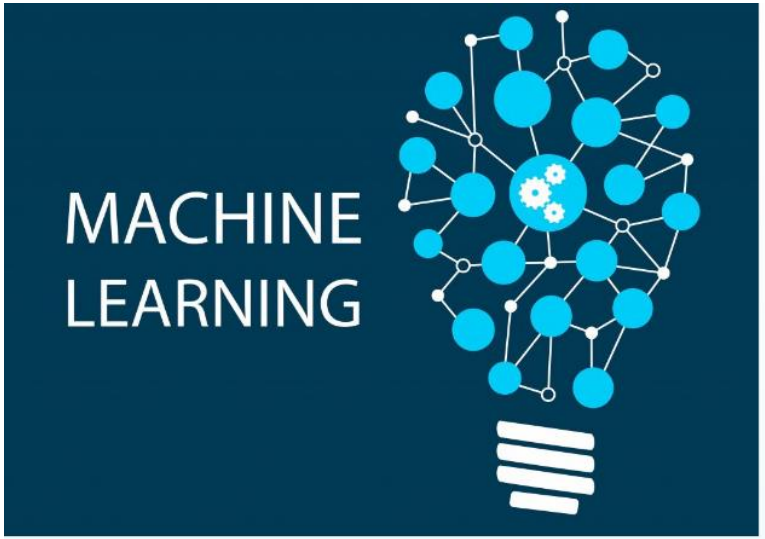


Coca Cola e Deep Learning



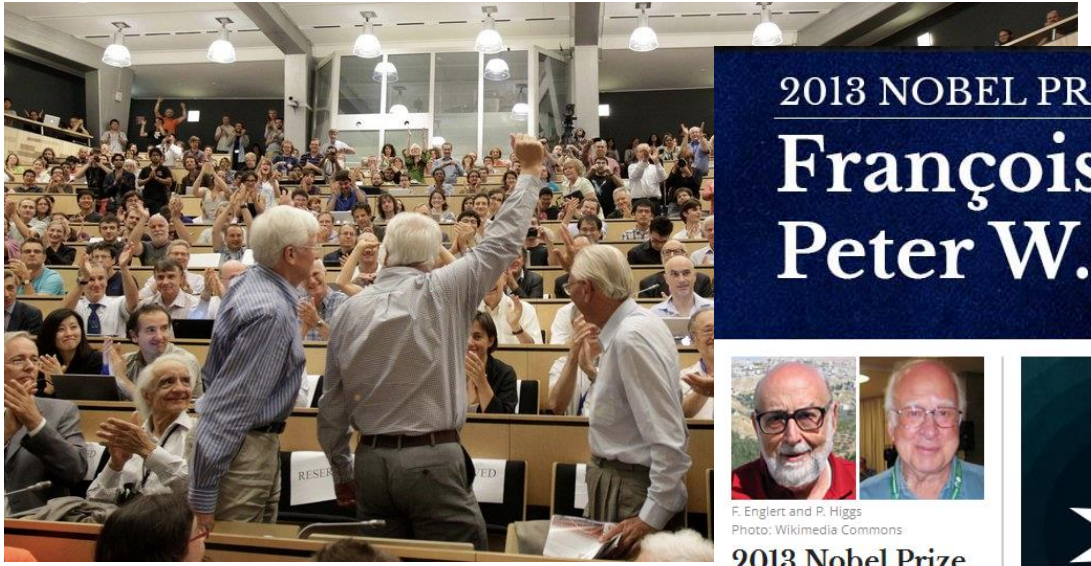
Alcune applicazioni ML di…

Alcune applicazioni ML di…



Cosa hanno in comune Machine Learning e la fisica delle Particelle?

Abbiamo un problema? Il bosone di Higgs



- E’ realmente una particella elementare o e’ composta da altre particelle?
- Interagisce con la materia oscura?

2013 NOBEL PRIZE IN PHYSICS

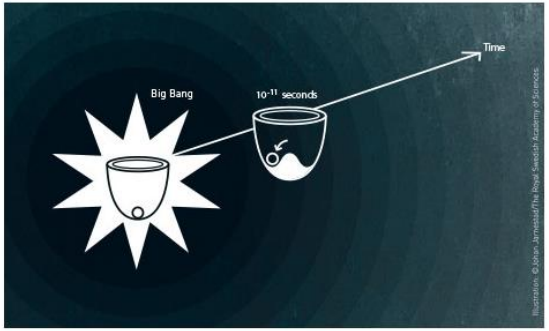
François Englert
Peter W. Higgs

© The Nobel Foundation. Photo: Lovisa Engblom.

F. Englert and P. Higgs
Photo: Wikimedia Commons

2013 Nobel Prize in Physics

The Nobel Prize in Physics 2013 was awarded jointly to François Englert and Peter W. Higgs *"for the theoretical discovery of a mechanism that contributes to our*



What Happened after the Big Bang?

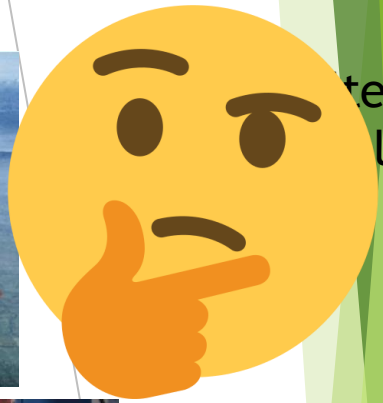
Announcements of the 2013 Nobel Prizes

Physiology or Medicine:
Announced Monday 7 October
Physics:
Tuesday 8 October, 11:45 a.m. CET at the earliest
Chemistry:
Wednesday 9 October, 11:45 a.m. CET at the earliest
Literature:
Thursday 10 October 1.00 p.m. CET
Peace:

Abbiamo un problema? La materia oscura

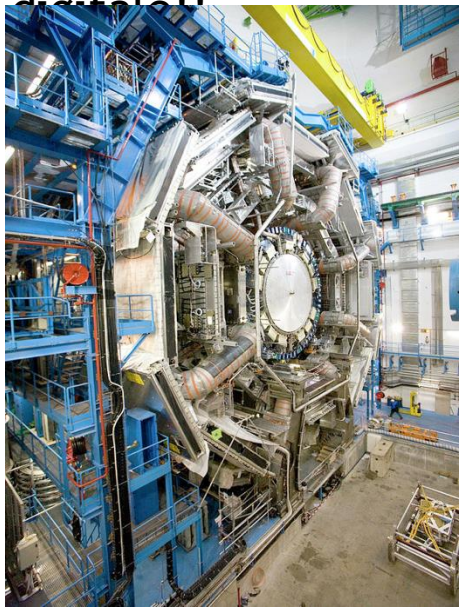
Aggiungere il video sulla dark matter dell'INFN

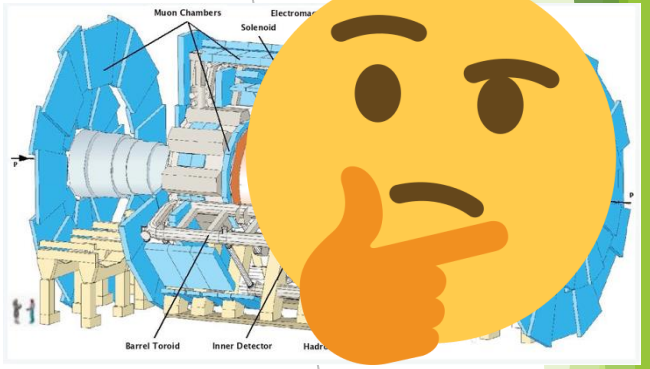
E da dove prendiamo i dati??



E da dove prendiamo i dati??

Atlas: Una grande fotocamera digitale!!





Quanto e' grande una "foto" fatta da Atlas??
2 Mbyte → tutto ok ??

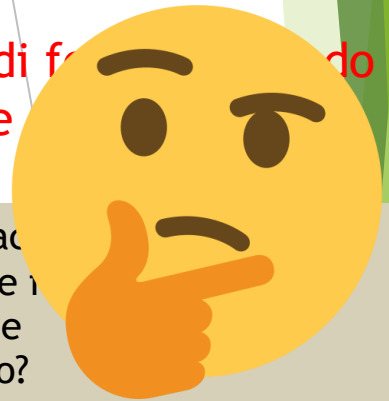
40 milioni di foto al secondo
→ 80 TByte al secondo!!!

…non possiamo conservare tutte le foto!!

Il filtro in tempo reale di Atlas

40 milioni di f do
→ 80 TByte

Sembra fac
ma quante
conservare
al secondo?



100 foto su 40 milioni

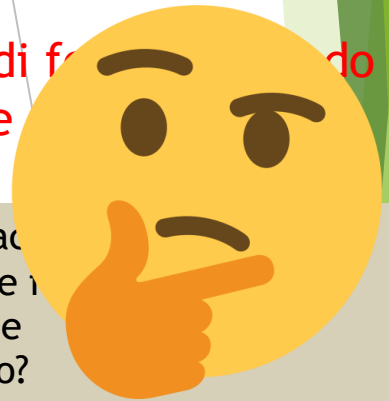
Una ogni 10 milioni!!

…non possiamo conservare tutte le foto!!

Il filtro in tempo reale di Atlas

40 milioni di f do
→ 80 TByte

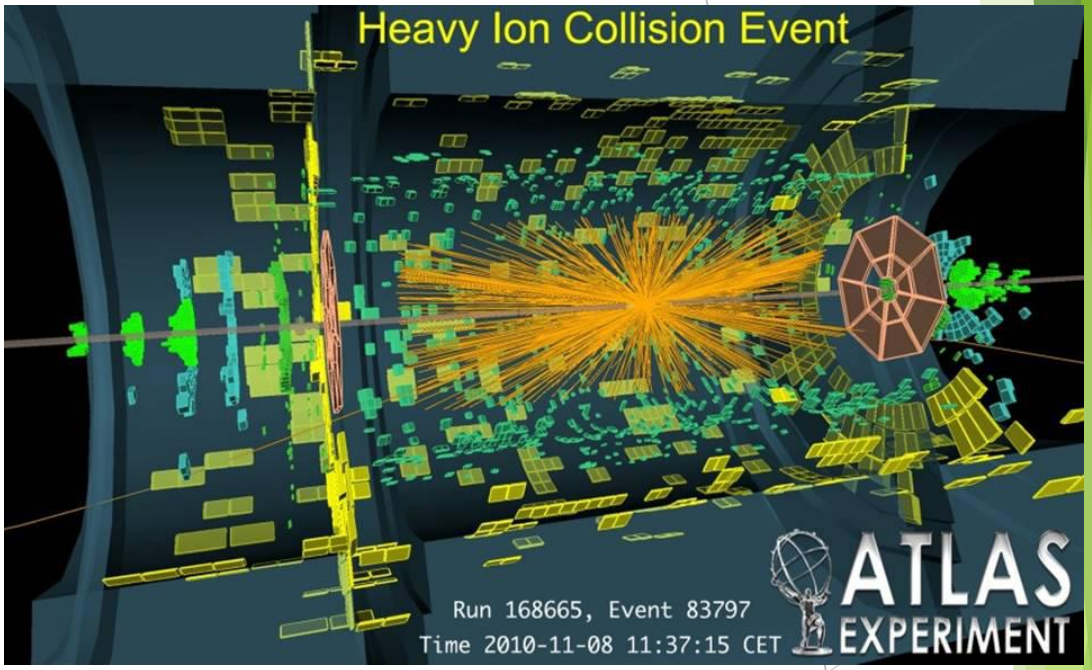
Sembra fac
ma quante
conservare
al secondo?



100 foto su 40 milioni

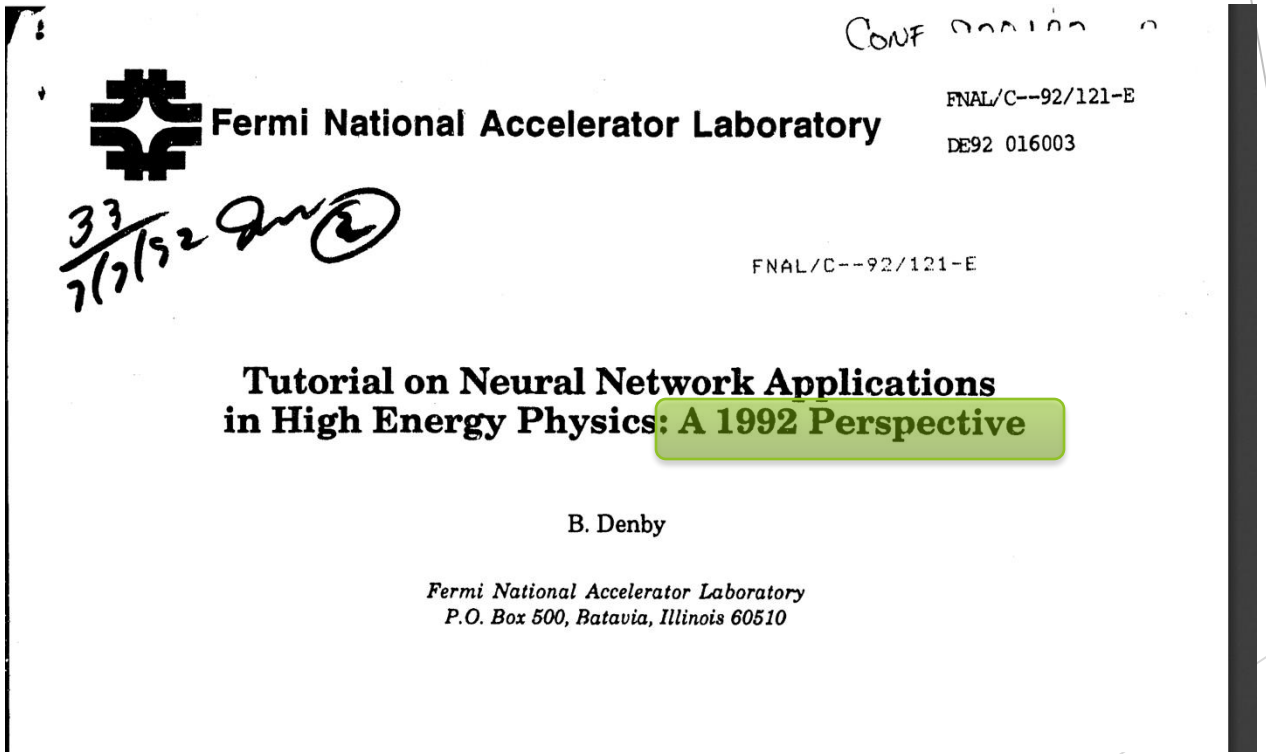
Una ogni 10 milioni!!

Come visualizziamo le foto??



Inserire filmato su data processing at LHC

Il Machine Learning e' una novita'?



CONF

Fermi National Accelerator Laboratory

FNAL/C--92/121-E

DE92 016003

33
7/7/92

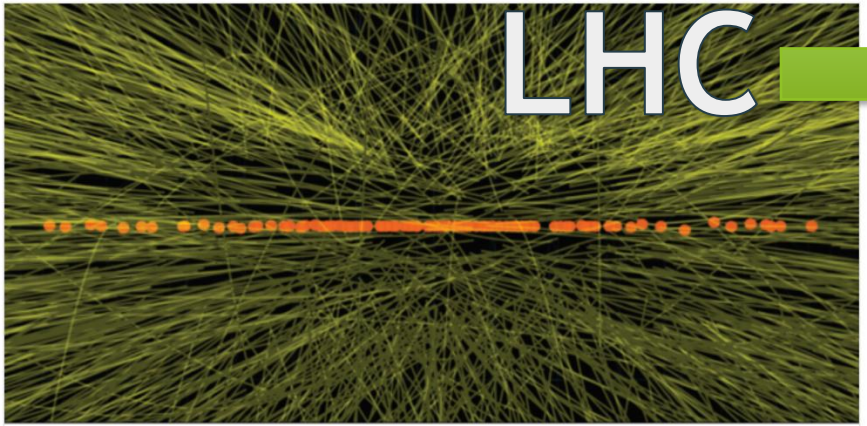
FNAL/C--92/121-E

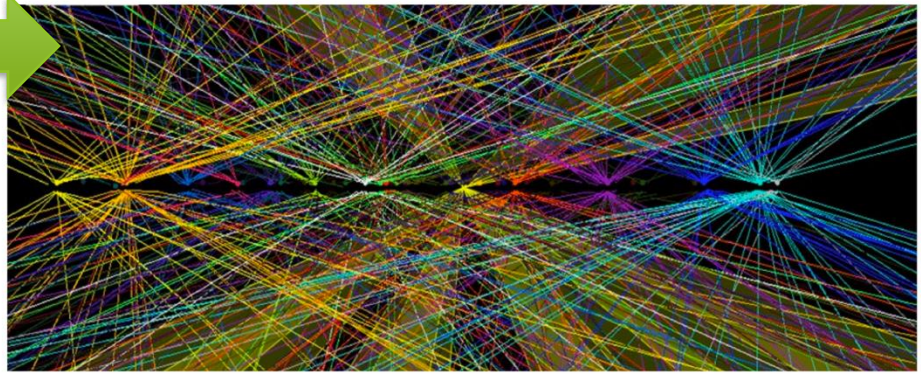
Tutorial on Neural Network Applications in High Energy Physics: A 1992 Perspective

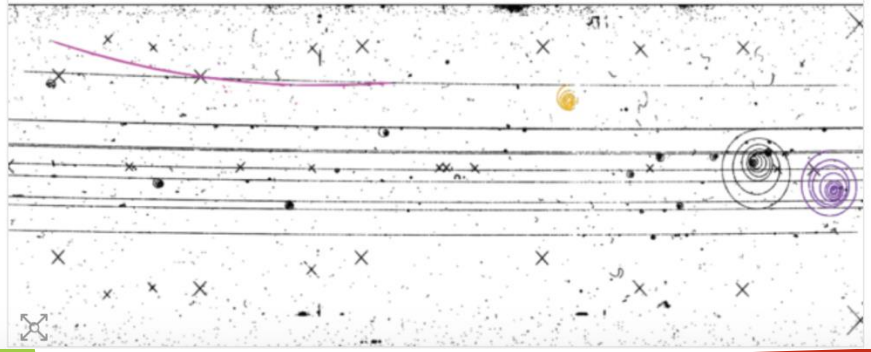
B. Denby

Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

..e allora cosa c'e' di nuovo?

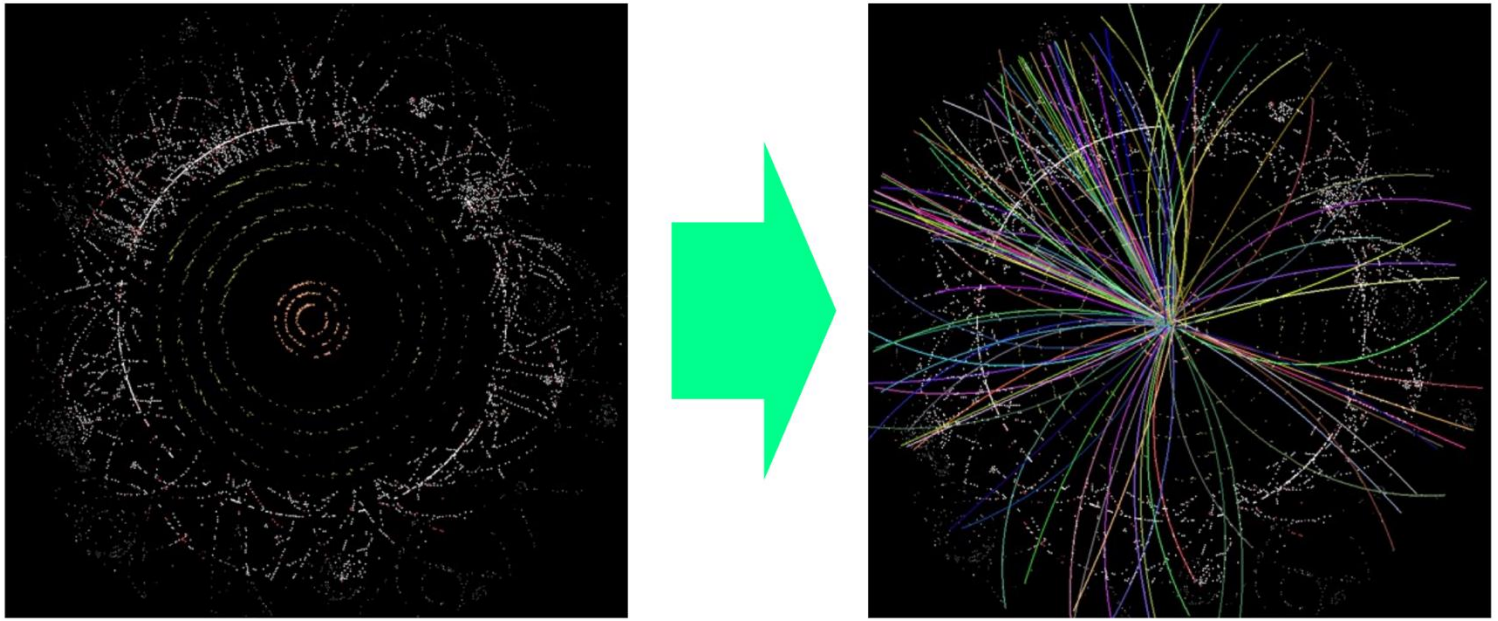






Un evento da un esperimento degli anni '60!

..e allora cosa c'e' di nuovo?



From hits ...

... to trajectory parameters

Un esempio dalla computer vision

Zagoruyko et al, https://arxiv.org/pdf/1604.02135.pdf

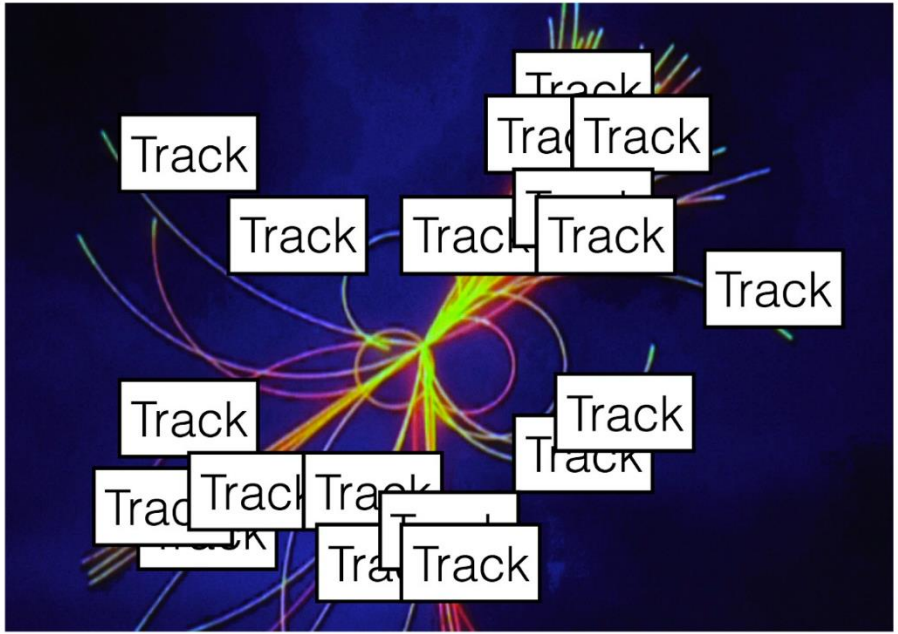
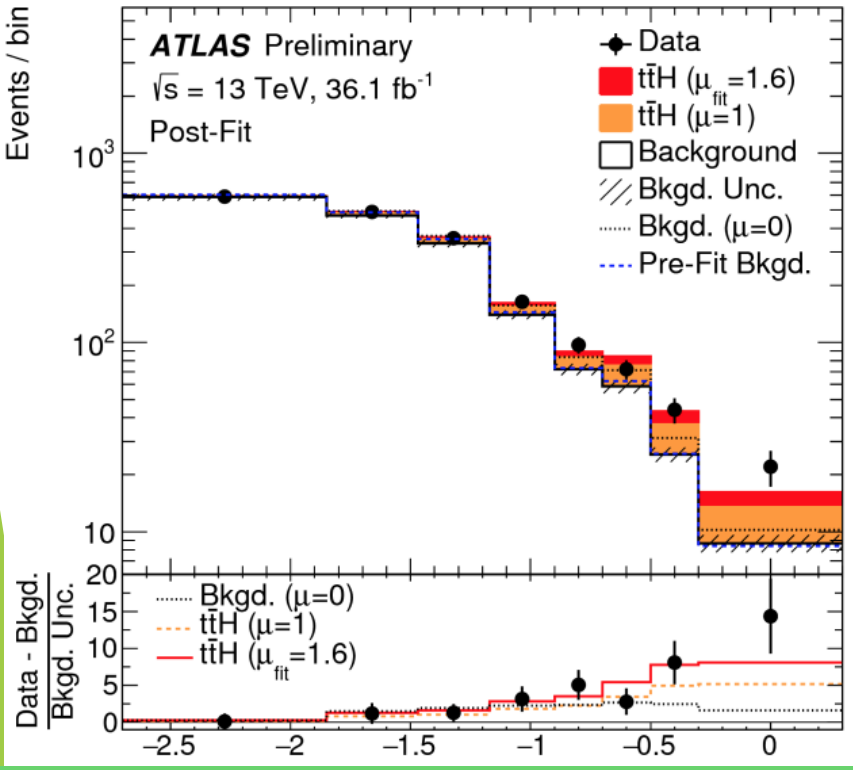
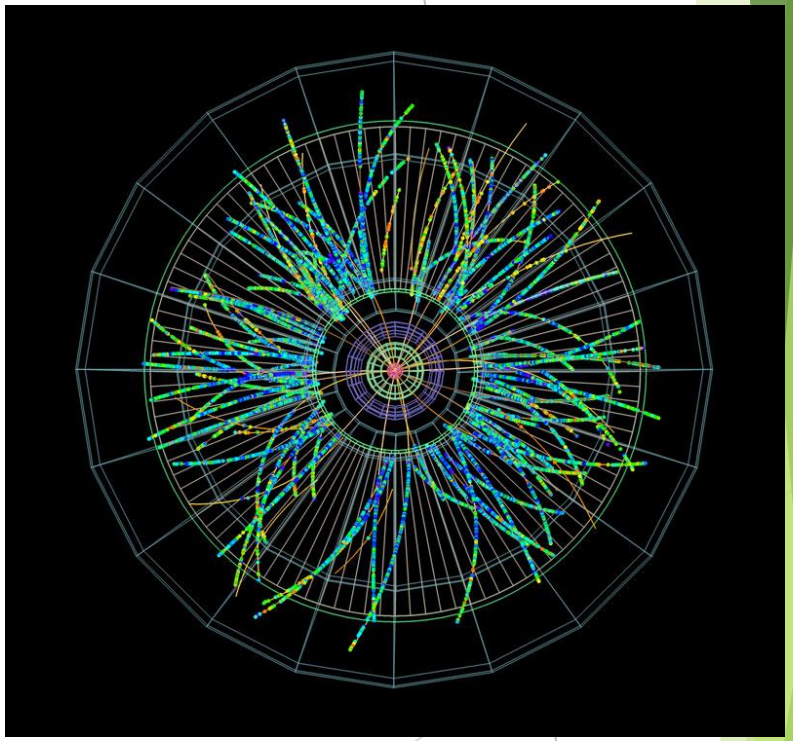


Photo by Pier Marco Tacca/Getty Images

ML and High Energy Physics





Physics Briefing plots: ATLAS finds evidence of the Higgs boson produced in association with a pair of top quarks (26 Oct 2017)

Link utili

SIRI: https://www.macitynet.it/com-funziona-ehi-siri-una-rete-neurale-dietro-al-sistema/

https://medium.com/marketing-and-entrepreneurship/10-companies-using-machine-learning-in-cool-ways-887c25f913c3

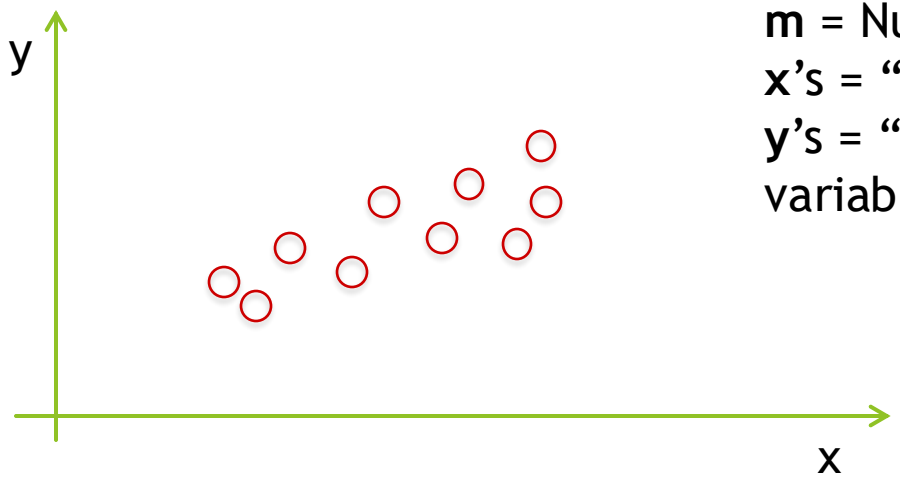
https://www.slideshare.net/clevertap/how-machine-learning-is-transforming-app-marketing

Fatta molto bene con l'esempio dello spam nelle email:

https://www.slideshare.net/liorrokach/introduction-to-machine-learning-13809045

Backup

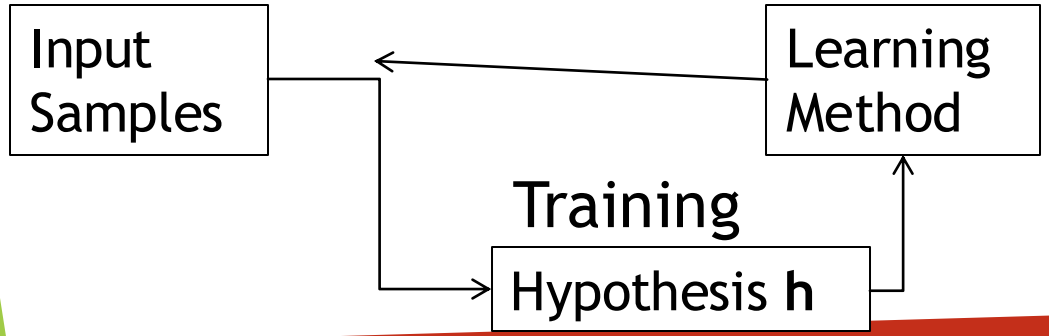
ML and Linear regression



m = Number of training examples
x's = "input" variable → features
y's = "output" variable → "target" variable

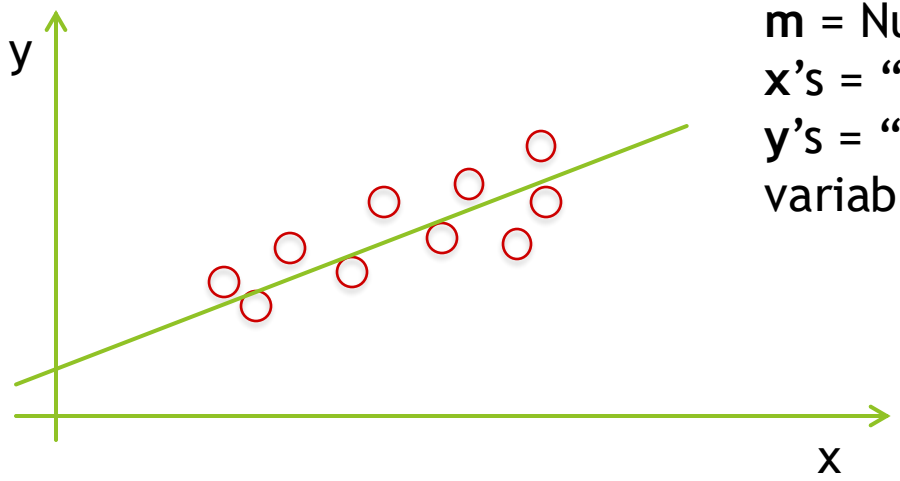
Our model is embedded in the hypothesis h function!!

$$h_\theta(x) = \theta_0 + \theta_1 x$$



How to choose the parameters?

ML and Linear regression



m = Number of training examples
x's = "input" variable → features
y's = "output" variable → "target" variable

Our model is embedded in the hypothesis h function!!

$$h_\theta(x) = \theta_0 + \theta_1 x$$

The **COST function:**

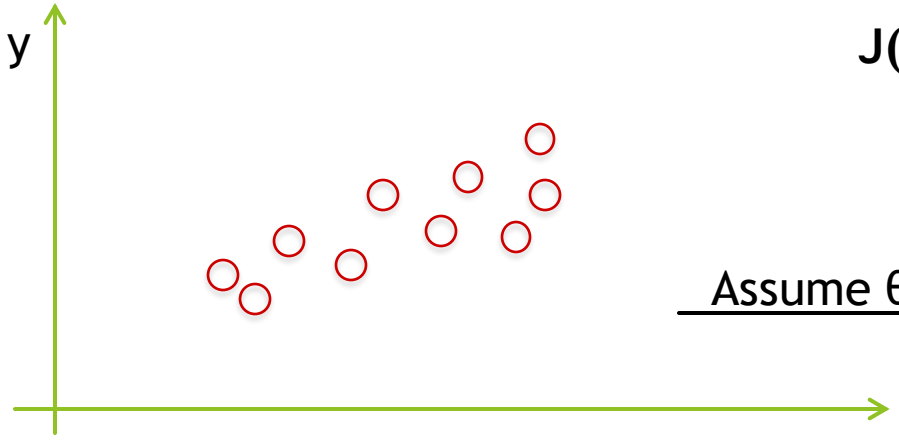
$$J(\theta_0,\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

How to choose the parameters?

Minimize the COST function

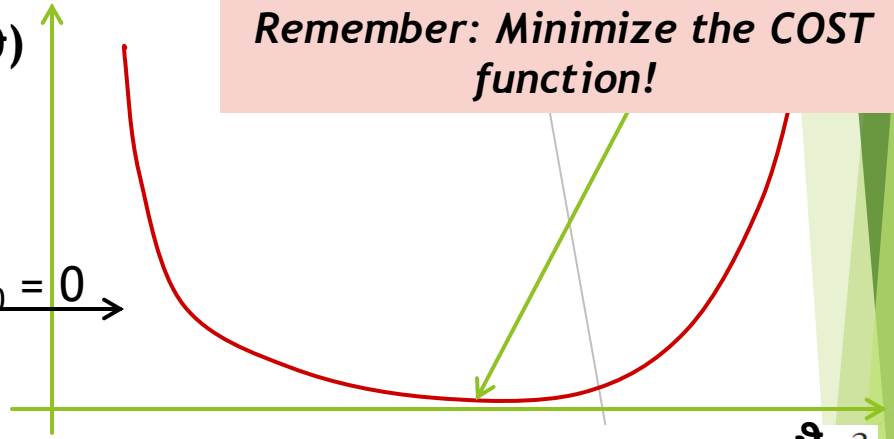
This is supervised ;-)

Linear regression: Where is the learning step??



$$h_\theta(x) = \theta_0 + \theta_1 x$$

Assume $\theta_0 = 0$

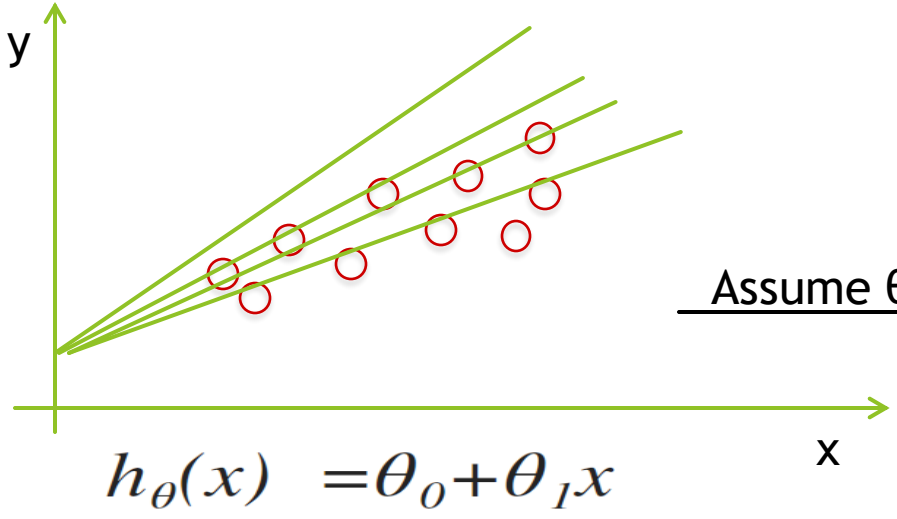


Remember: Minimize the COST function!

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

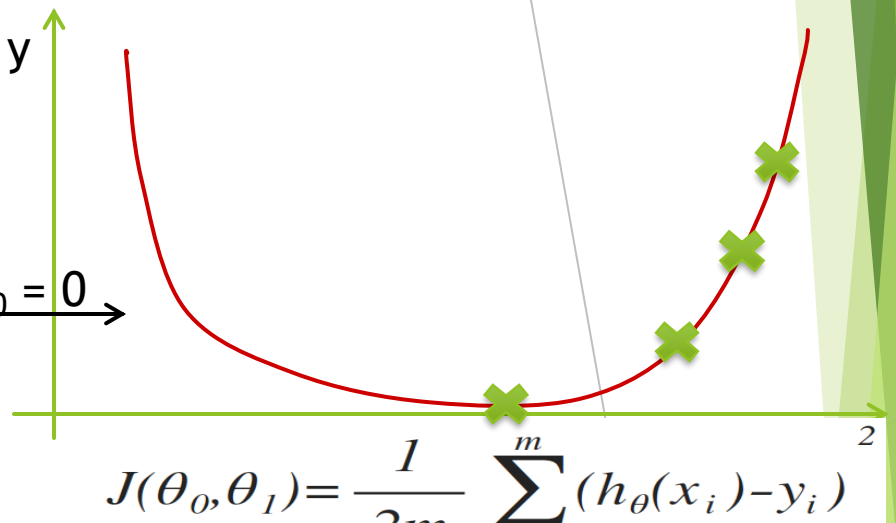
General Idea:
Start with some θ_0 and θ_1 and keep changing θ_0, θ_1 to reduce the cost function until we end up at a minimum

Linear regression: Where is the learning step??



$$h_\theta(x) = \theta_0 + \theta_1 x$$

Assume $\theta_0 = 0$



$$J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)^2$$

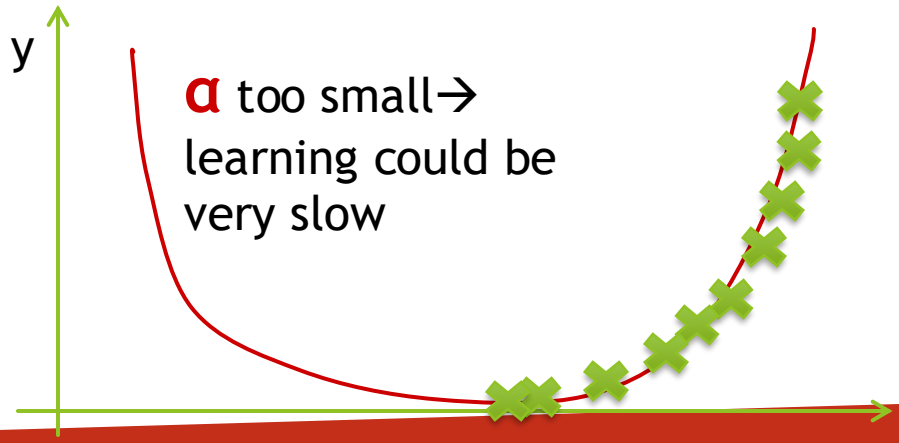
General Idea:
Start with some θ_0 and θ_1 and keep changing θ_0, θ_1 to reduce the cost function until we end up at a minimum

Gradient descent: Tools and tips (I)

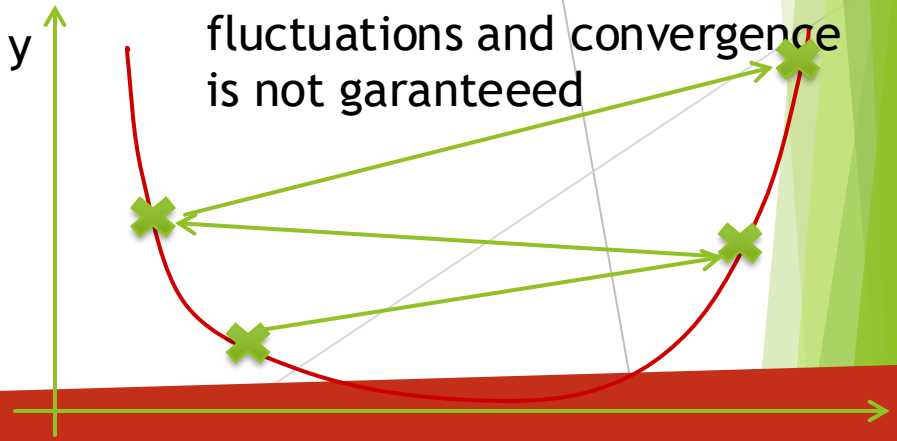
$$\theta_j := \theta_j - \alpha \frac{\partial J}{\partial \theta}(\theta)$$

Update both j=0 and j=1

Learning rate

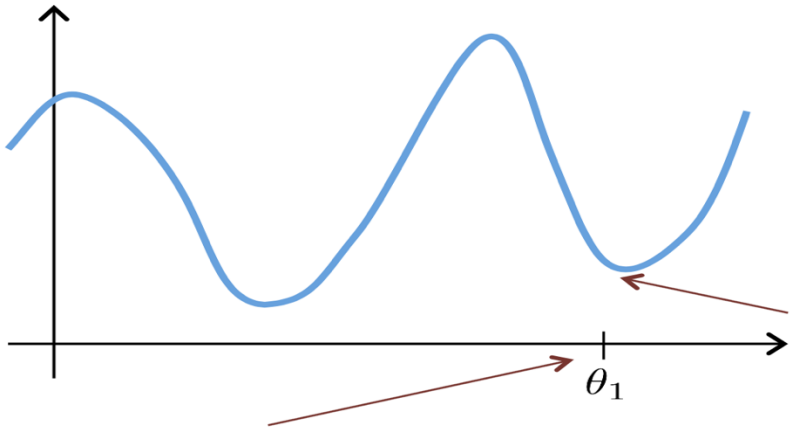


α too small→ learning could be very slow



α too large→ large fluctuations and convergence is not garanteeed

Gradient descent: Tools and tips (I)



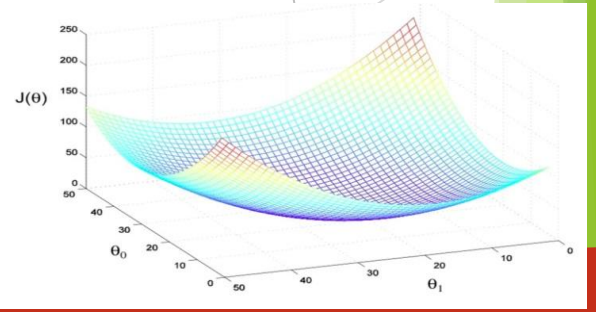
Is J(ϑ) a convex function ?

Gradient discent for linear regression

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

$$\theta_0 := \theta_0 - \alpha \sum_{i=1}^{m} (h_\vartheta(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha \sum_{i=1}^{m} (h_\vartheta(x_i) - y_i)\, x_i$$



Linear regression with >1 features

$$h_\theta = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \dots$$

m = Number of training examples
N = Number of features
X_j s = "input" variable → features #j
y's = "output" variable → "target"

$$h_\theta(X) = \Theta^T X$$

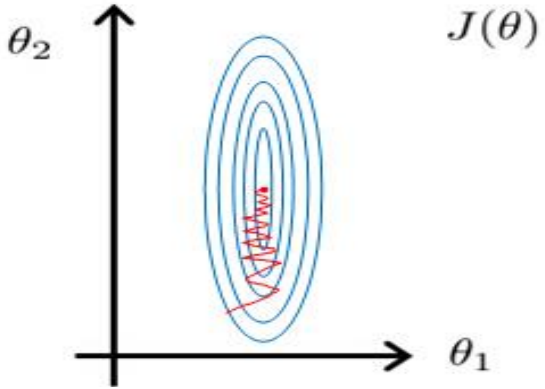
$$(\theta_0\ \theta_1\ \theta_2 \dots \theta_n) \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

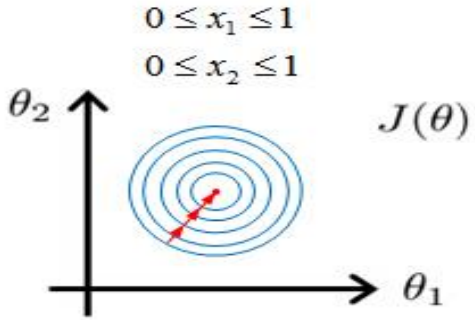
Gradient discent
for linear regression with
multiple features doesn't
change!!!

$$q_j := q_j - a \frac{1}{m} \sum_{i=1}^{m} \left(h_q\left(x_j^{(i)}\right) - y^{(i)} \right) x_j^{(i)}$$

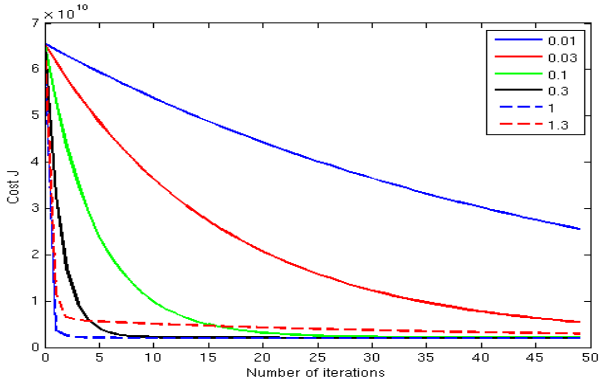
Gradient discent : Tools and tips (II)

If you have N features make sure are on similar range of values!!





Try with mean normalization:
x' = (x- μ)/σ



Making sure gradient descent is working correctly.

For **α** sufficiently small, **J** should decrease on every iterationon. But if **α** is too small, gradient descent can be slow to converge.

ML and Classification

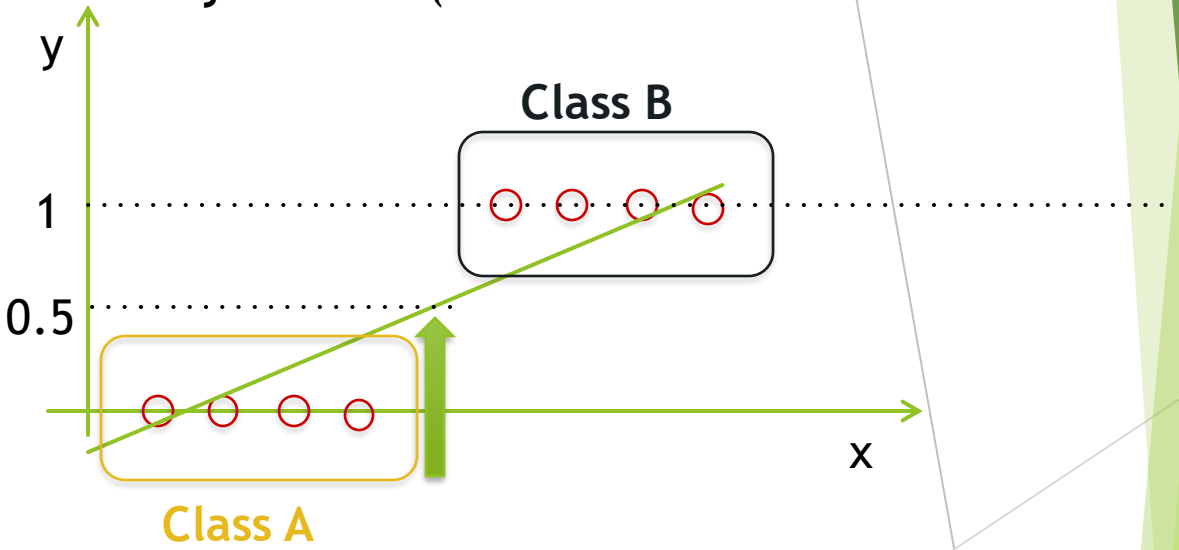
In classification problem our dataset **y** is 0 or 1 (or a discrete finite number 0,1,2….N)

$$h_\theta(X) = \Theta^T X$$

In linear regression a possible solution is:

Y= 1 if h_θ > 0.5

Y= 0 if h_θ < 0.5



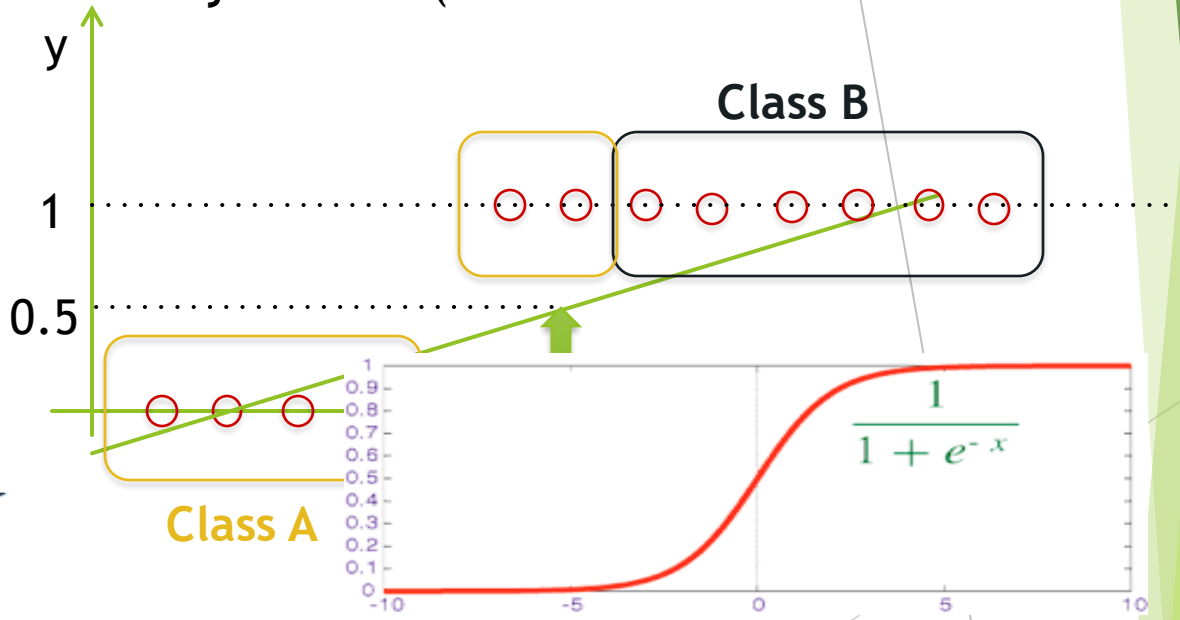
ML and Classification

In classification problem our dataset **y** is 0 or 1 (or a discrete finite number 0,1,2….N)

In linear regression a possible solution is:

Y= 1 if $h_\theta > 0.5$

Y= 0 if $h_\theta < 0.5$



$$h_\theta(X) = \Theta^T X$$

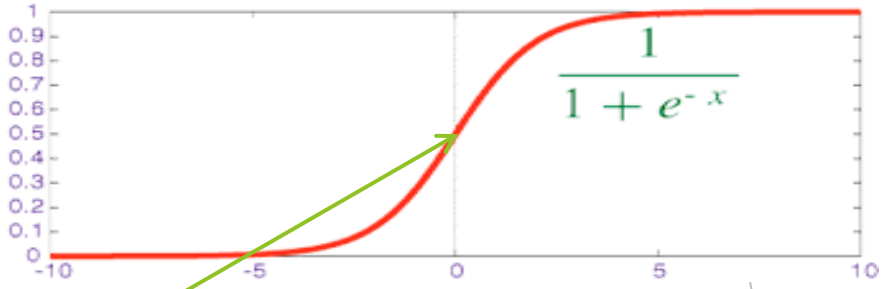
$$h_\vartheta = g(\Theta^T X)$$

$$g(z) = \frac{1}{1+e^{-z}}$$

Classification: Logistic regression

$$h_\vartheta = g(\Theta^T X)$$

$$g(z) = \frac{1}{1+e^{-z}}$$



In logistic regression a possible solution is:

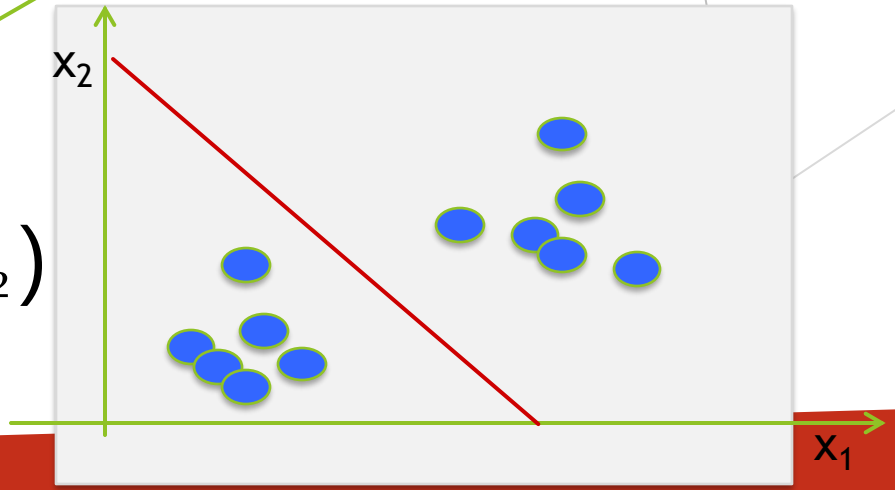
Y= 1 if h_θ > 0.5→ Z >=0

Y= 0 if h_θ < 0.5→ Z <0

$$h_q(x) = g(q_0 + q_1 x_1 + q_2 x_2)$$

$$\Theta = \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix}$$

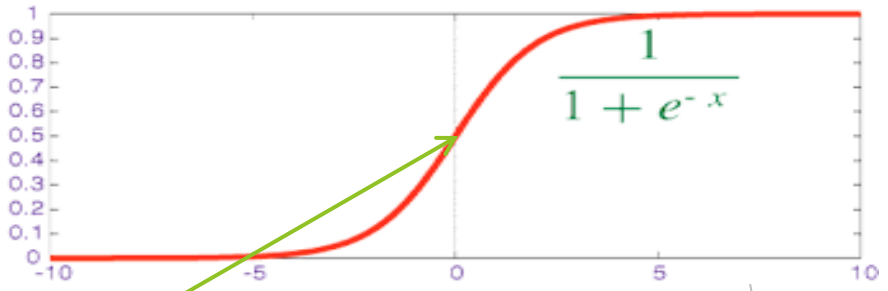
→ $-3+x_1+x_2 > 0$



Classification: Logistic regression

$$h_\vartheta = g(\Theta^T X)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$



In logistic regression a possible solution is:

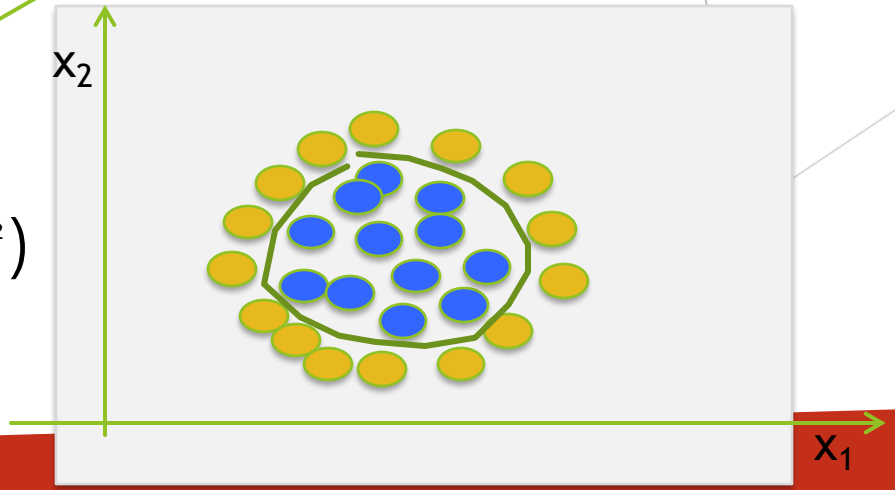
Y= 1 if h_θ > 0.5→ Z >=0

Y= 0 if h_θ < 0.5→ Z <0

$$h_q(x) = g(q_0 + q_1 x_1 + q_2 x_2 + q_3 x_1^2 + q_4 x_2^2)$$

$$\Theta = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

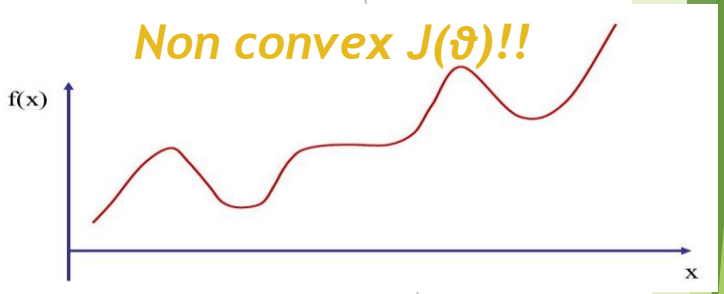
→ $x_1^2 + x_2^2 > 1$



Logistic regression: Cost function

Linear Cost function applied to logistic regression

$$J(\theta_0,\theta_1)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)^2$$



$$\text{Cost}\left(h_q(x),y\right)=\begin{cases}-\log\left(h_q(x)\right) & \text{if } y=1\\ -\log\left(1-h_q(x)\right) & \text{if } y=0\end{cases}$$

If y=1 AND $h_\vartheta(x)$=1 → cost = -log(1) = **0** → you pay no cost for correct answer!

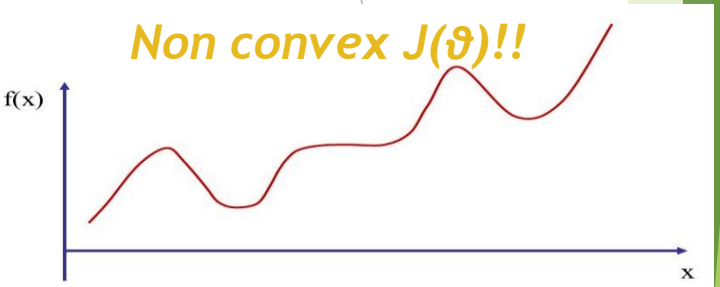
but….

If y=1 AND $h_\vartheta(x)$=0 → cost = -log(0) ≈ ∞ → you pay very large cost for incorrect answer!

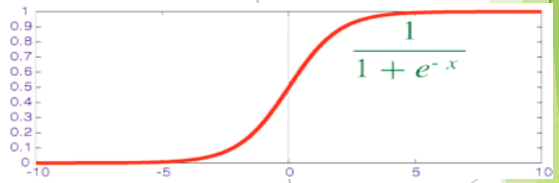
Logistic regression: Cost function

Linear Cost function applied to logistic regression

$$J(\theta_0,\theta_1)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)^2$$



$$J(q)=-\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\log h_q\left(x^{(i)}\right)+\left(1-y^{(i)}\right)\log\left(1-h_q\left(x^{(i)}\right)\right)\right]$$



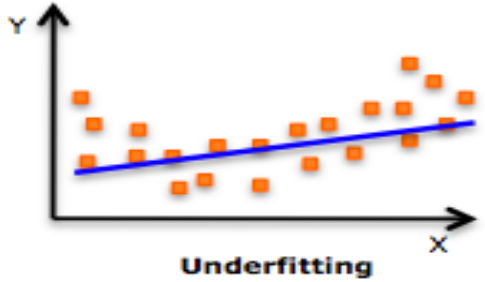
Gradient discent

$$q_j := q_j - a\frac{1}{m}\sum_{i=1}^{m}\left(h_q\left(x_j^{(i)}\right)-y^{(i)}\right)x_j^{(i)}$$

The same upgrade algorithm used for linear regression!

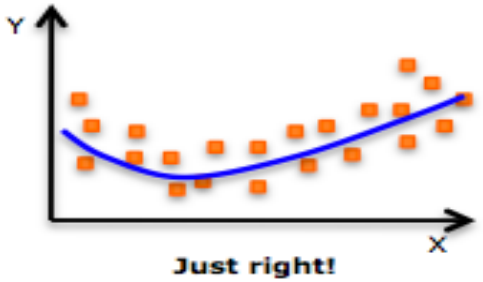
Overfitting the model

Linear regression



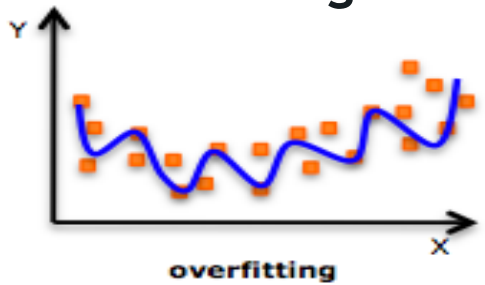
Underfitting

$$\theta_0+\theta_1 x_1$$



Just right!

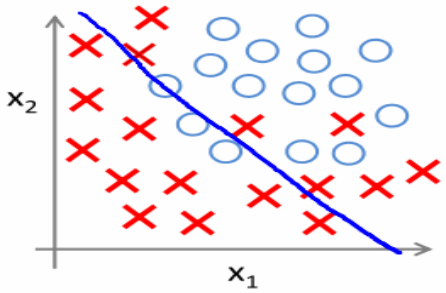
$$\theta_0+\theta_1 x_1+\theta_2 x_1^2$$



overfitting

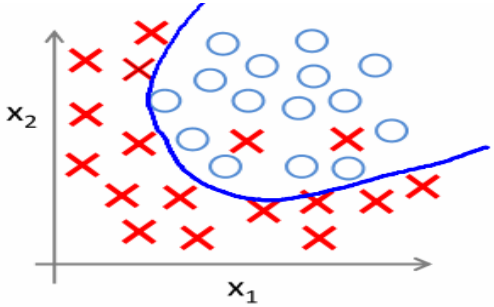
$$\theta_0+\theta_1 x_1+\theta_2 x_1^2+\theta_3 x_1^3+\theta_4 x_1^3$$

Logistic regression

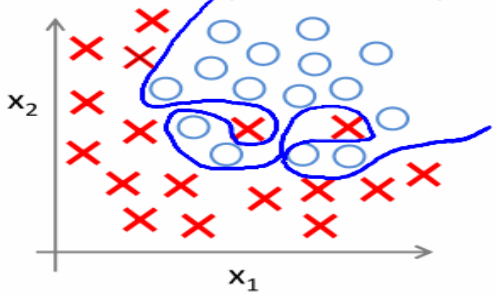


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

(g = sigmoid function)



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

Overfitting the model

In overfitting regime the h(θ) learn to fit the training example very well but (may) fail to adapte well to a new set of data (lack of generalization)

How to deal with overfitting??

- Reduce number of features
- Keep all the features, but reduce magnitude/values of some parameters **θ**

Regularization

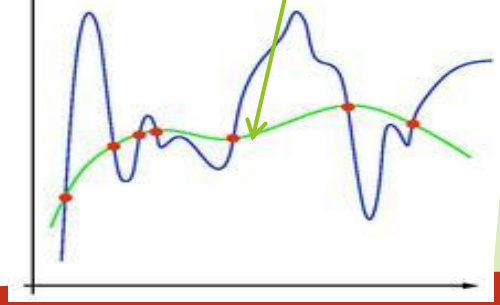
Extra-price to pay for some θ

Linear regression

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{m}\theta_j^2\right]$$

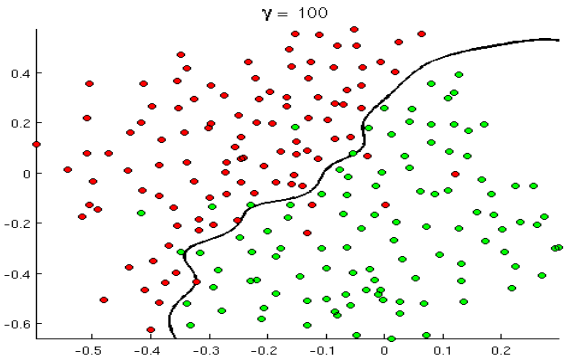
Logistic regression

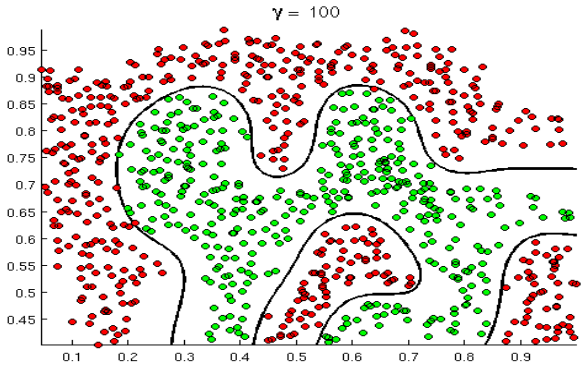
$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right) + \lambda \sum_{j=1}^{m}\theta_j^2\right]$$



Neural Networks

A (highly) **non linear** classification problem with **2 features**





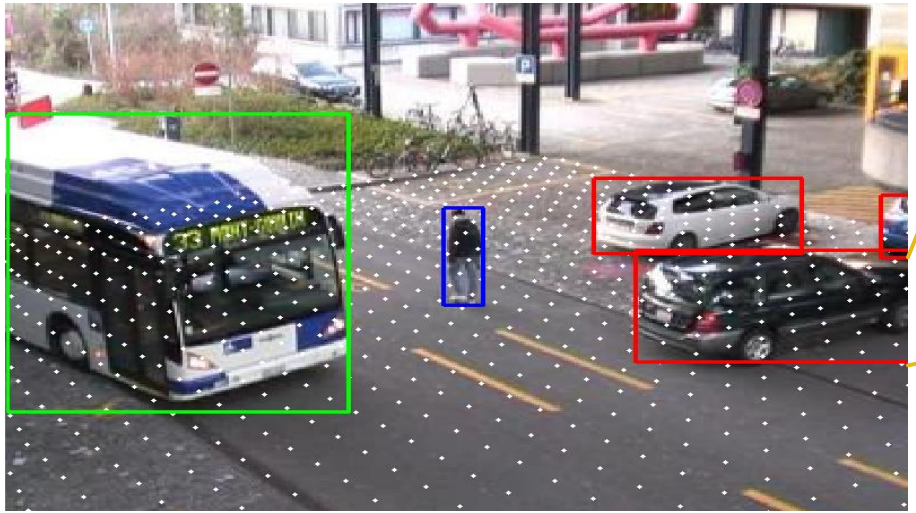
$$g\left(q_0 + q_1 x_1 + q_2 x_2 + q_3 x_1 x_2 + q_4 x_1^2 x_2 + q_5 x_1^3 x_2 + q_6 x_1 x_2^2 + \ldots\right)$$

What about high dimension of features??

- *If you stop at second order → $O(n^2)$ terms*
- *With 100 input features at third order you get ≈ 200.000 terms..*

But you really need high number of input features??

Neural networks



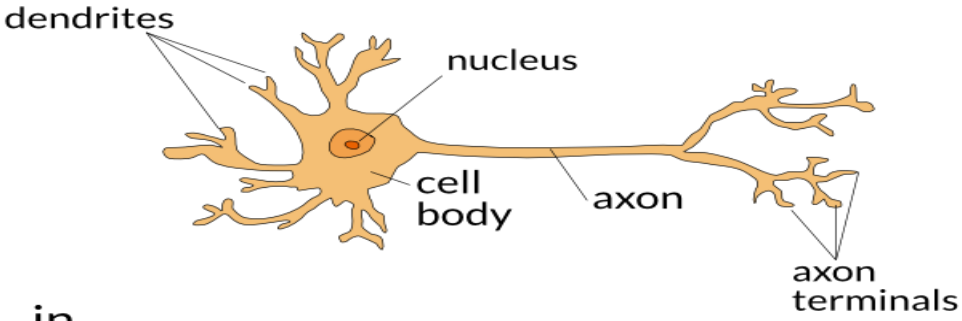
But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

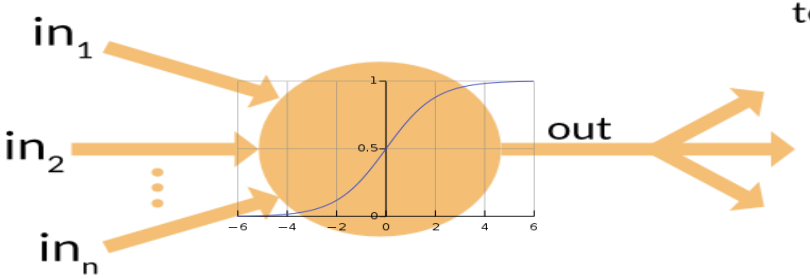
Take a low resolution image of this car 50x50 pixels → **n = 2500 features**

With only second order we end up with ≈ ***3 millions of terms***

(Artificial) Neural networks (ANN)



$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

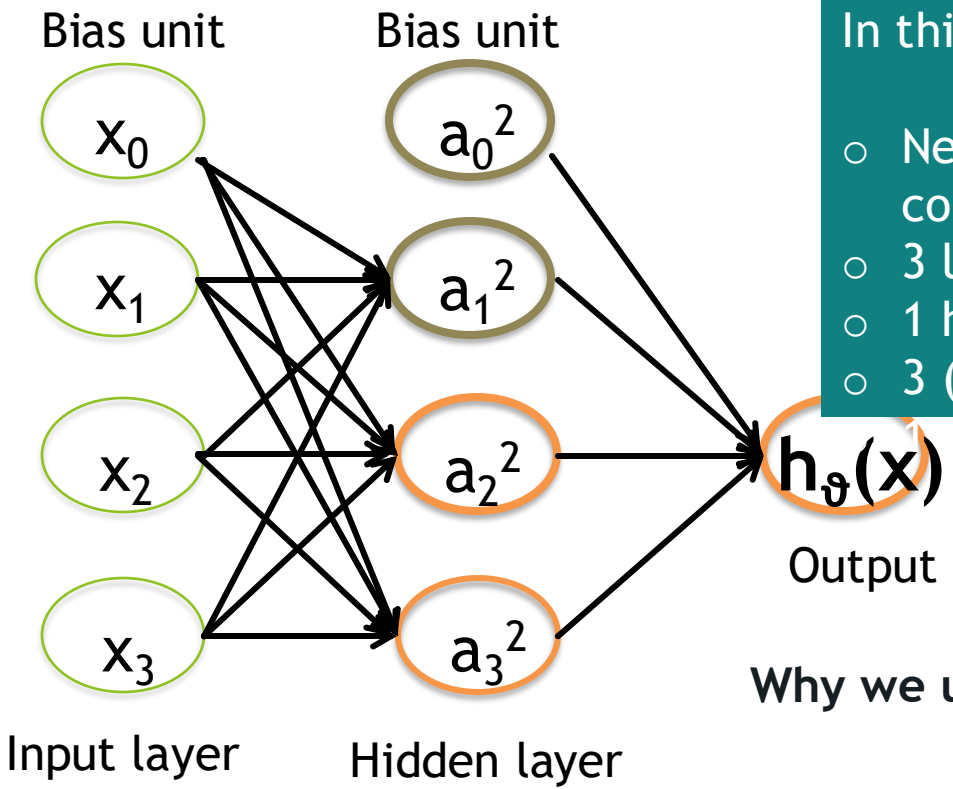


$h_\vartheta(x)$

$$h_\vartheta = g(\Theta^T X) \qquad g(z) = \frac{1}{1+e^{-z}}$$

Logistic units are the building blocks of our ANN!!

Artificial Neural Networks: Architecture

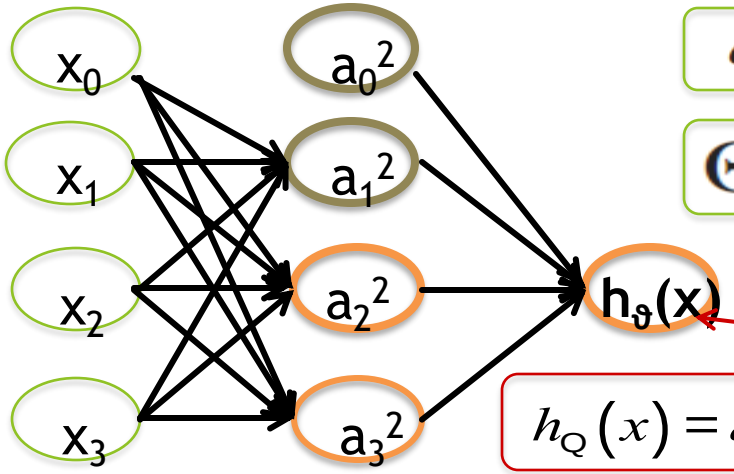


In this example:

- Neural Network fully connected
- 3 layers
- 1 hidden layer
- 3 (+1) input features

Why we use this representation for ANN?

Artificial Neural Networks: Architecture



a_i^j **Activation unit i in layer j**

Θ_{lm}^j **Weight Matrix from layer j to layer j+1**

$$g(z) = \frac{1}{1+e^{-z}}$$

$$h_\Theta(x) = a_1^{(3)} = g\left(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)}\right)$$

$$a_1^{(2)} = g\left(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(2)} x_2 + \Theta_{13}^{(2)} x_3\right)$$

$$a_2^{(2)} = g\left(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(2)} x_2 + \Theta_{23}^{(2)} x_3\right)$$

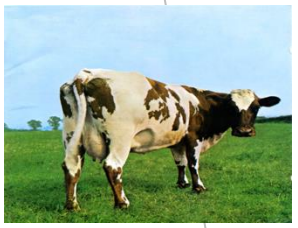
$$a_3^{(2)} = g\left(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(2)} x_2 + \Theta_{33}^{(2)} x_3\right)$$

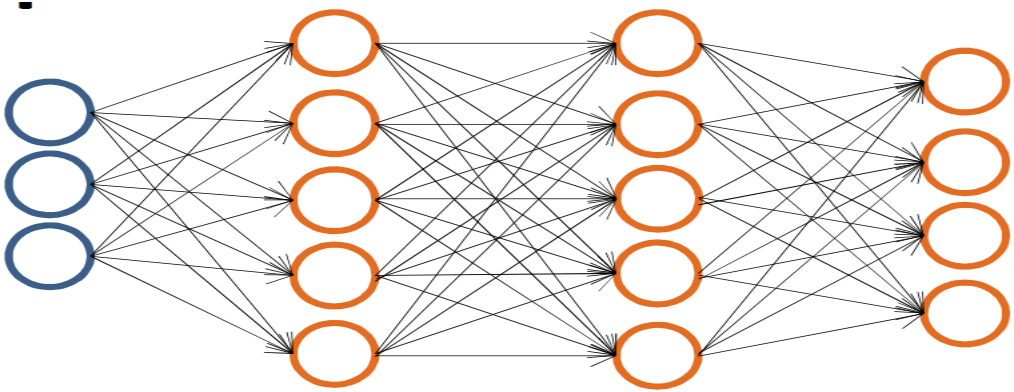
Θ_{lm}^j

For j=1 is a 3x4 matrix!!

Artificial Neural Networks: Architecture





$h_\vartheta(x) \rightarrow R^4$

$$h_\theta(x) \cong \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

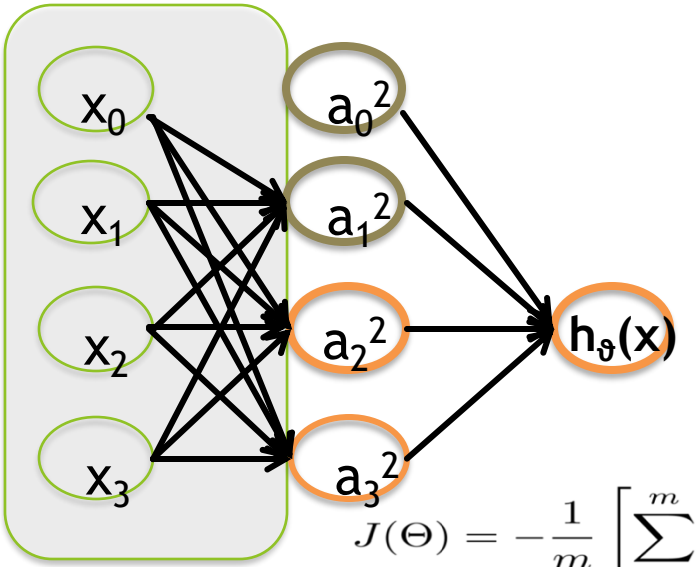
$$h_\theta(x) \cong \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

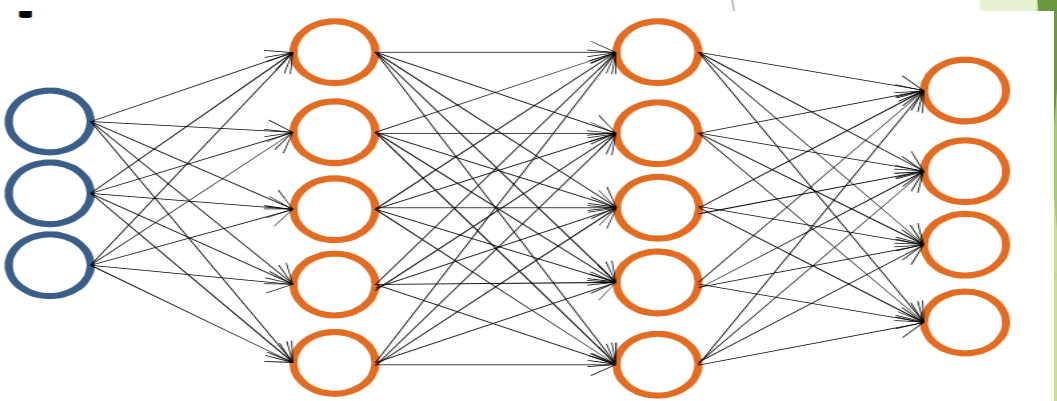
and so on…

Artificial Neural Networks: Cost Function

Logistic regression

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right] + \lambda \sum_{j=1}^{m} \theta_j^2$$





$$h_\Theta(x) \in \mathbb{R}^K \quad (h_\Theta(x))_i = i^{th} \text{ output}$$

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log(h_\Theta(x^{(i)}))_k + (1 - y_k^{(i)}) \log(1 - (h_\Theta(x^{(i)}))_k)\right]$$

$$+\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(\Theta_{ji}^{(l)})^2$$

Artificial Neural Networks:Back Propagation

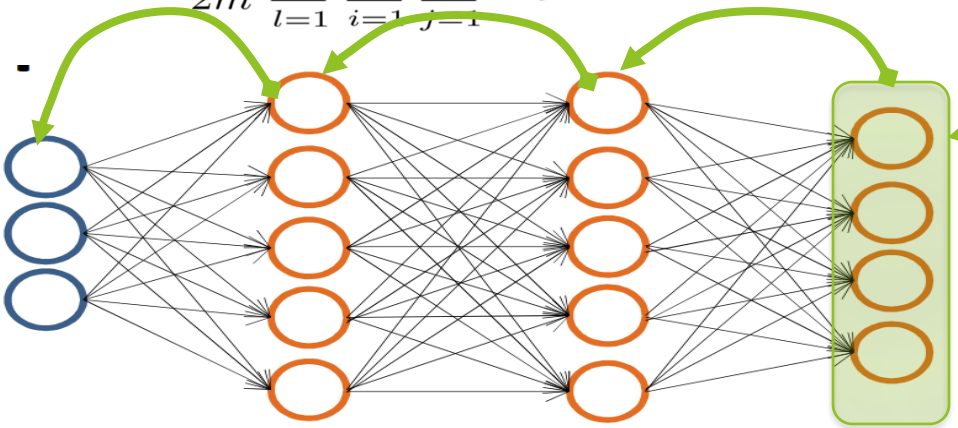
Artificial Neural Networks: Back Propagation

$$h_\Theta(x) \in \mathbb{R}^K \quad (h_\Theta(x))_i = i^{th} \text{ output}$$

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log(h_\Theta(x^{(i)}))_k + (1 - y_k^{(i)}) \log(1 - (h_\Theta(x^{(i)}))_k)\right]$$

$$+\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(\Theta_{ji}^{(l)})^2$$

-



Layer #4

Compute error on output layer as

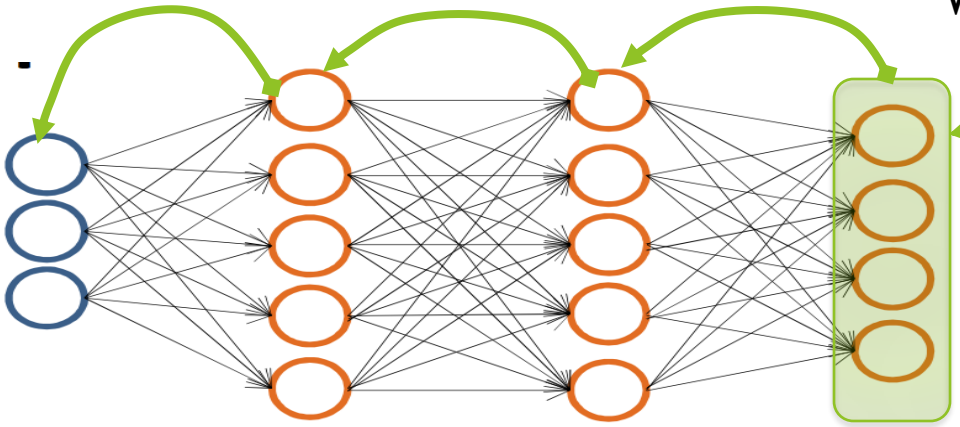
$$\delta_j^4 = a_j^4 - y_j$$

We want to minimize J(ϑ) ⇨

$$Q_{ij}^{(l)} := Q_{ij}^{(l)} - a\frac{\P}{\P Q_{ij}^{(l)}} J(q)$$

Artificial Neural Networks: Back Propagation

-



We want to minimize J(ϑ)→

Layer #4

Compute error on output layer as

$$\delta_j^4 = a_j^4 - y_j$$

$$\theta_j := \theta_j - \alpha \frac{\partial J}{\partial \theta}(\theta)$$

$$\delta^{(3)} = \left(\Theta^{(3)}\right)^T \delta^{(4)} .* g'\left(z^{(3)}\right)$$

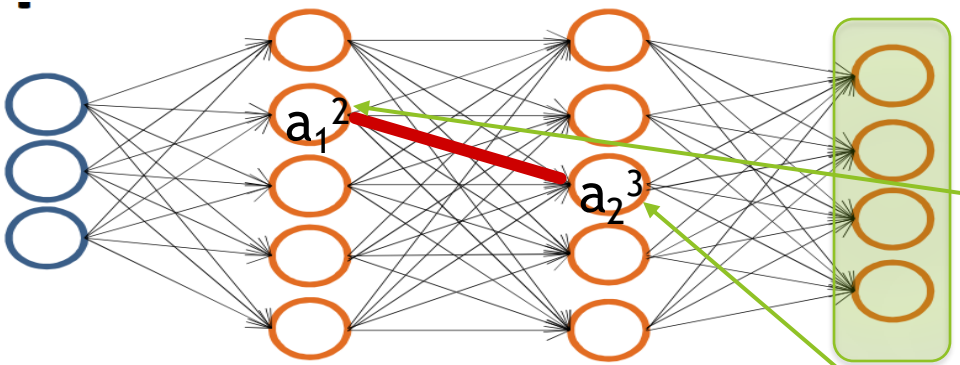
$$\delta^{(2)} = \left(\Theta^{(2)}\right)^T \delta^{(3)} .* g'\left(z^{(2)}\right)$$

$$\Theta_{ij}^{(l)} := \Theta_{ij}^{(l)} - \alpha \frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta)$$

$$\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta) = a_j^{(l)} \delta_i^{(l+1)}$$

Artificial Neural Networks: Back Propagation

-



$$\frac{\partial}{\partial Q_{21}^{(2)}} J(Q) = a_1^{(2)} d_2^{(3)}$$

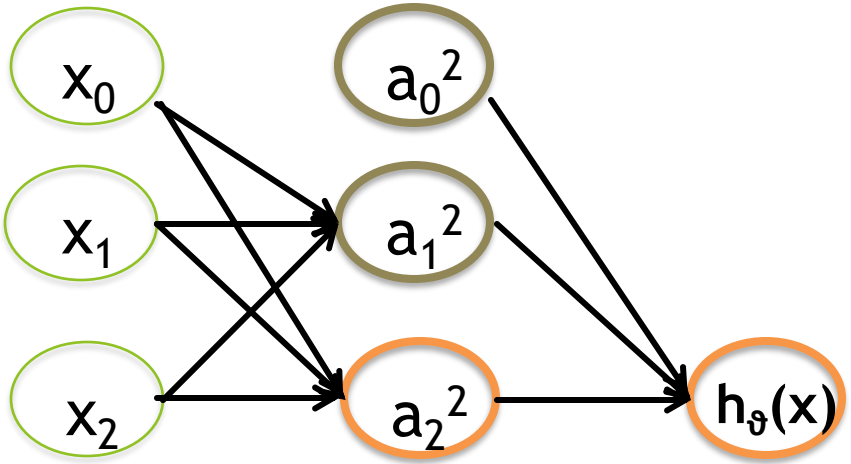
$$\frac{\partial}{\partial Q_{21}^{(2)}} J(Q) = a_1^{(2)} d_2^{(3)}$$

$$\theta_j := \theta_j - \alpha \frac{\partial J}{\partial \theta}(\theta)$$

$$Q_{ij}^{(l)} := Q_{ij}^{(l)} - a \frac{\partial}{\partial Q_{ij}^{(l)}} J(q)$$

l (layer) = 2, i = 2 ,j = 1

Neural Networks: What about Initialization???



$$Q_{ij}^{(l)} = \text{constant for all i, j, l}$$

$a_1^2 = a_2^2 \rightarrow \delta_1^2 = \delta_2^2$

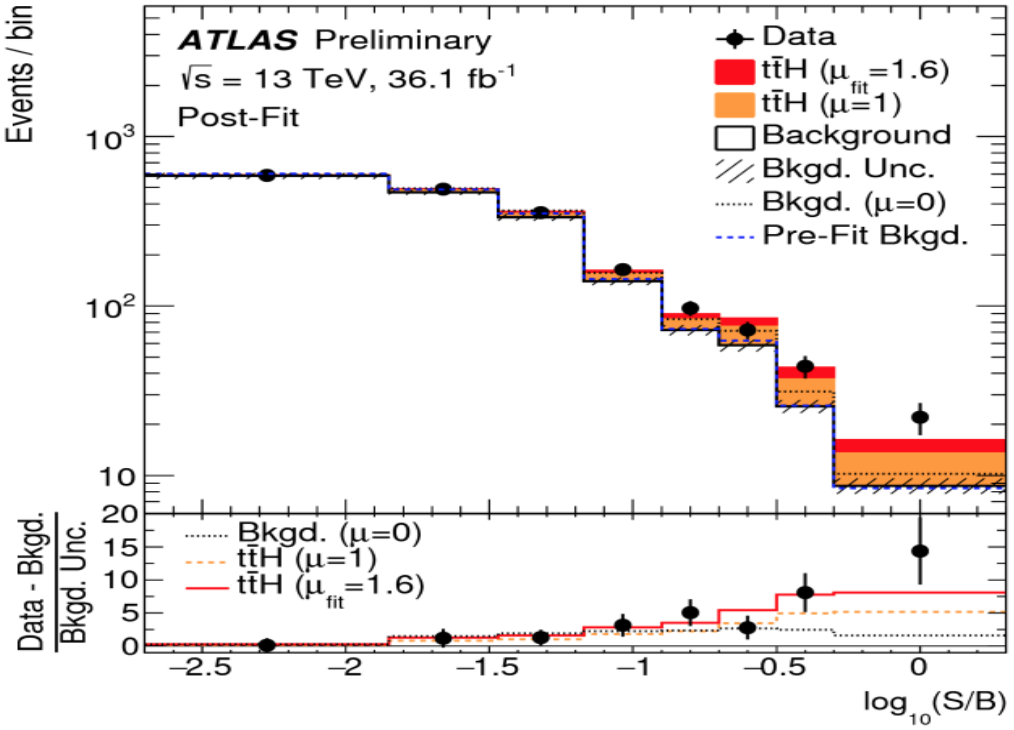
Even if you apply back propagation algorithm the parameters Θ are all equals → $a_1^2 = a_2^2$

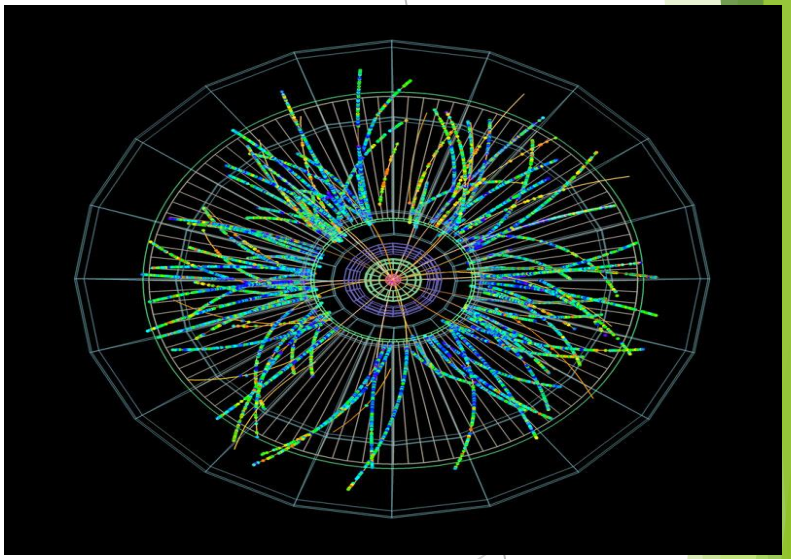
You need random initialization of parameters

→ simmetry breaking:

Initialize Θ_{ij}^{l} randomly in a given range [-a,a]

NN and High Energy Physics



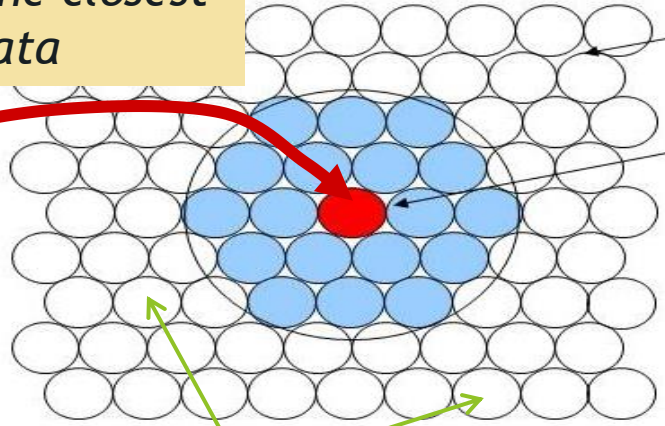


Physics Briefing plots: ATLAS finds evidence of the Higgs boson produced in association with a pair of top quarks (26 Oct 2017)

SOM Neural Networks and Unsupervised learning

*For each data you may define the **BestMatchingUnit** as the closest neuron to the data*

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$



You have **MxN** neurons on a 2 Dimensional grid

Of course you start with random

Each neuron is a R^n number: W_k (**n**= number of features)

You have a metrics in the grid since you can compute the distance between any two neurons

$$\vec{a} = (a_1, a_2, \dots, a_N)$$

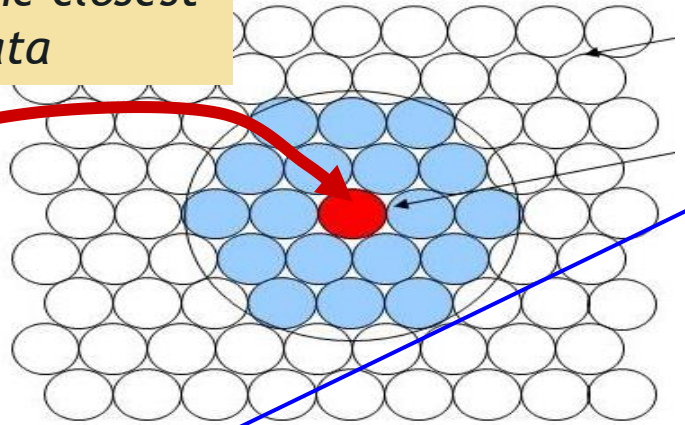
$$\vec{b} = (b_1, b_2, \dots, b_N)$$

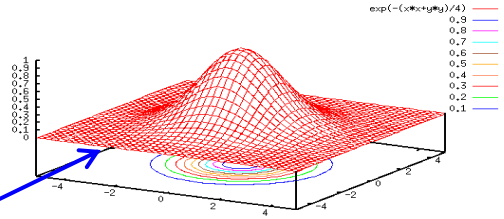
$$d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots (a_N - b_N)^2} = \sqrt{\sum_{i=1}^{N} (a_i - b_i)^2}$$

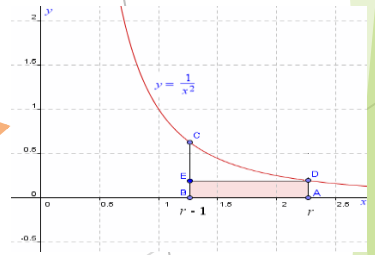
SOM Neural Networks and Unsupervised learning

*For each data you may define the **BestMatchingUnit** as the closest neuron to the data*

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$







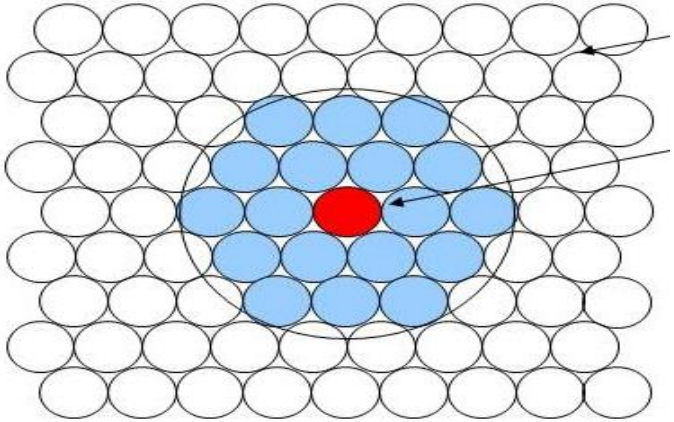
Learning with modified gradient discent

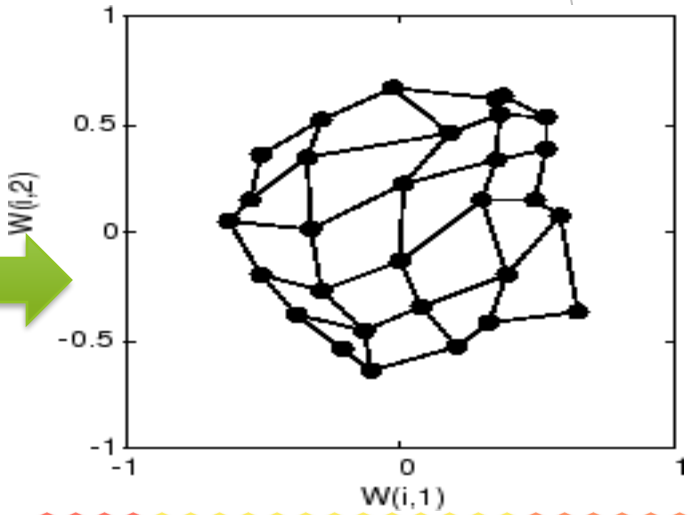
$W_k := W_k + \Theta(n,t)\ \alpha(t)\ [D(t) - W_k]$

Θ(n,t) is a gaussian neighbourhood function;

α(t) is the learning rate;

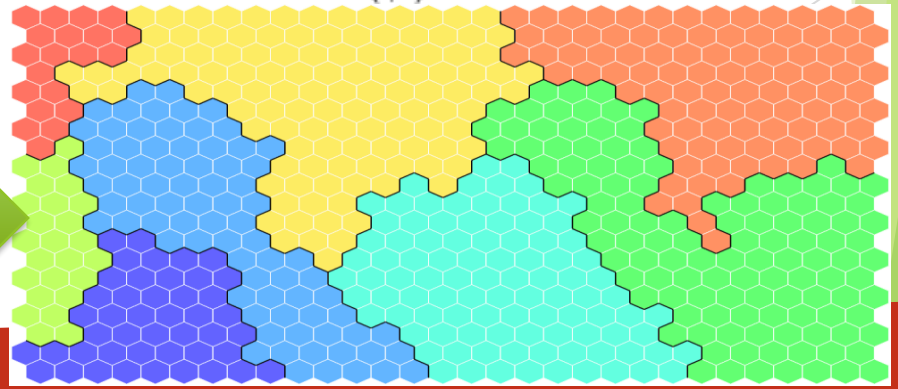
SOM Neural Networks and Unsupervised learning



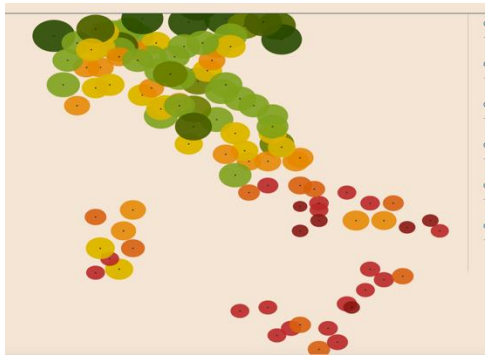


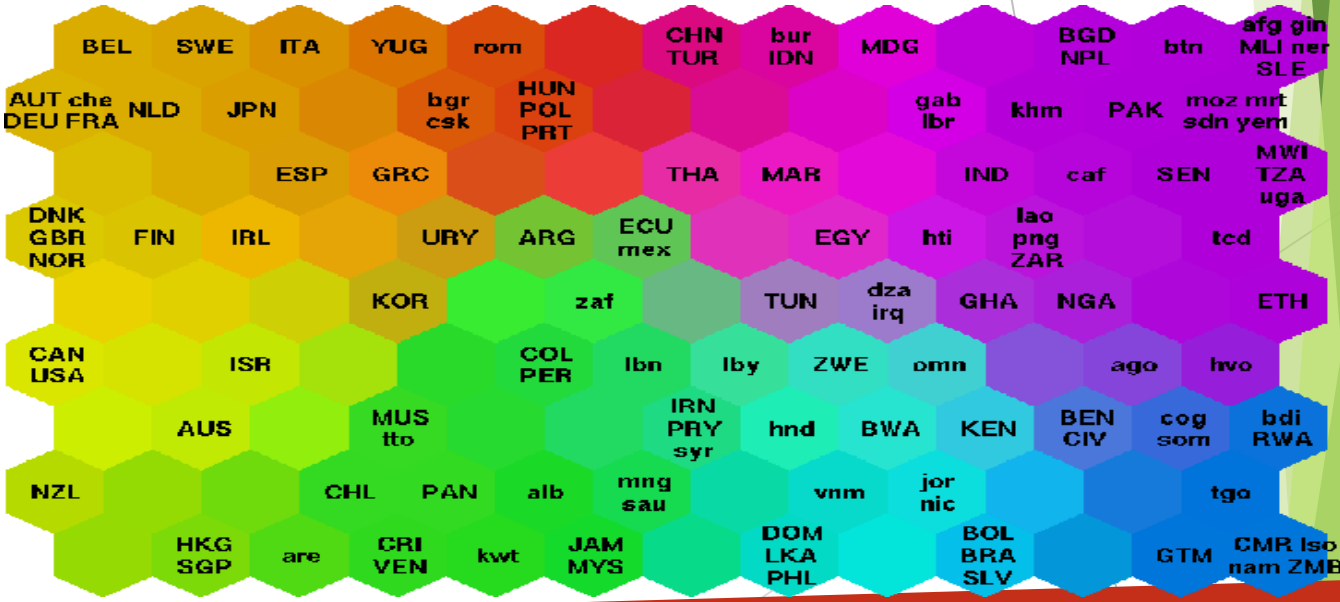
After the training the metrics in the grid has been changed!! Two neurons in the grid that are close at the init step should be very far at the end

Define different clusters of close-neurons

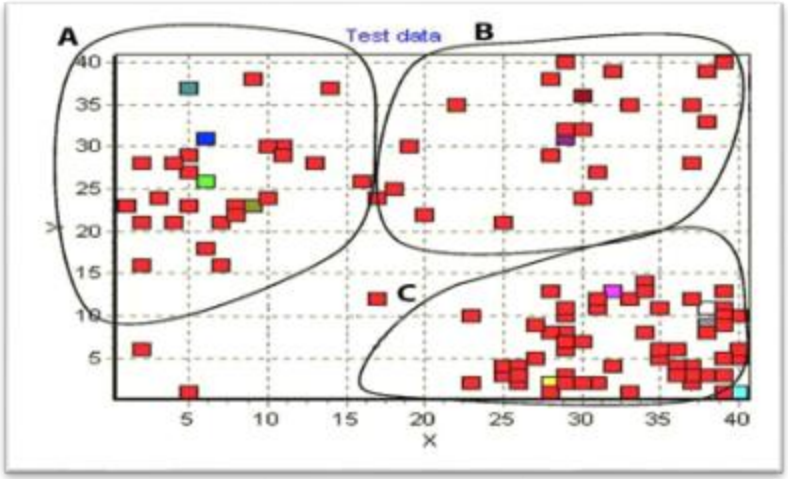


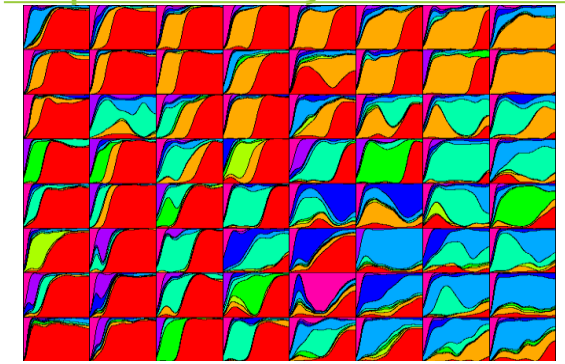
SOM Neural Networks and Unsupervised learning

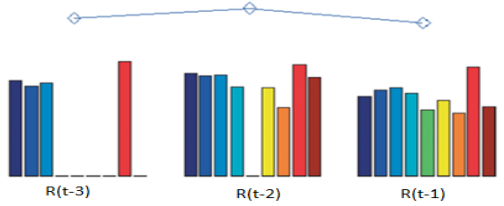




https://www.youtube.com/watch?v=aQkIg69ZAXs



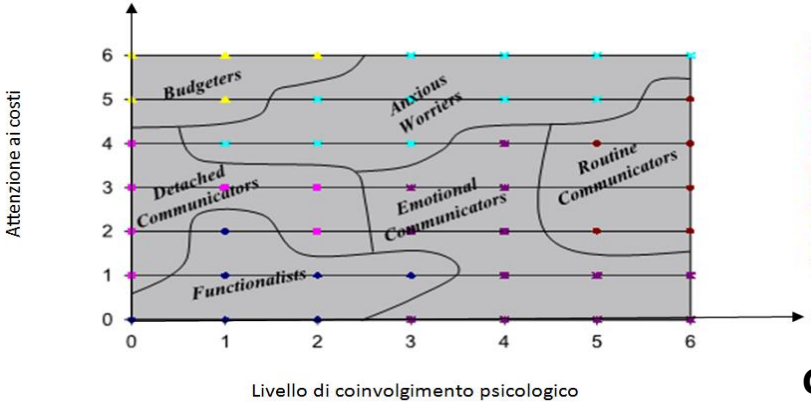




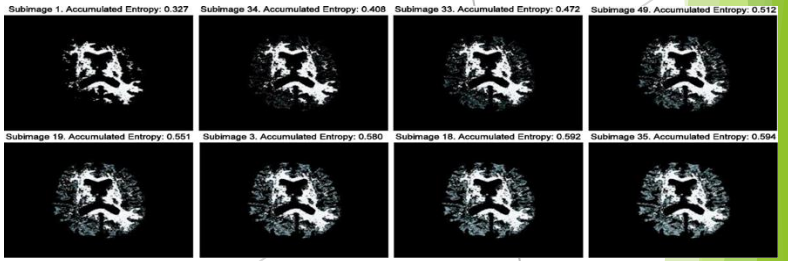
Applicazione di una rete SOM nella previsione dei fenomeni di piena fluvia

ANALISI DEI PERCORSI DI CARRIERA

Market segmentation



Gestione del rischio sismico



ANALISI DI IMMAGINI DA RISONANZA MAGNETICA MEDIANTE RETI NEURALI SOM

Why “Learn”?

Machine learning is programming computers to optimize a performance criterion using example data or past experience.

Learning is used when:

- Human expertise does not exist (navigating on Mars),
- Humans are unable to explain their expertise (speech recognition)
- Solution changes in time (routing on a computer network)
- Solution needs to be adapted to particular cases (user biometrics)

Insert your title here

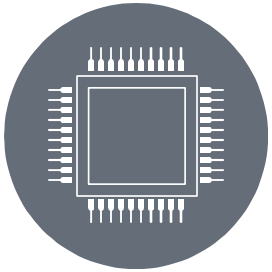
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Insert your title here

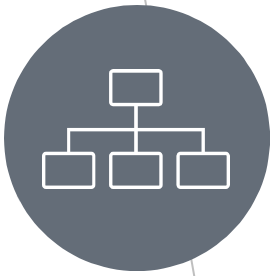
Your text here

This is a sample text. Insert your desired text here.



Your text here

This is a sample text. Insert your desired text here.



Your text here

This is a sample text. Insert your desired text here.

Insert your title here

Insert your text here

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Insert your title here

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.