MACHINE LEARNING

How to draw an owl

1.

2.

1. Draw some circles

2. Draw the rest of the fucking owl

DEFINITION

ARTIFICAL INTELLIGENCE

MACHINE LEARNING

DEEP LEARNING

"

Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed

Arthur Samuel, 1959

APPLICATIONS

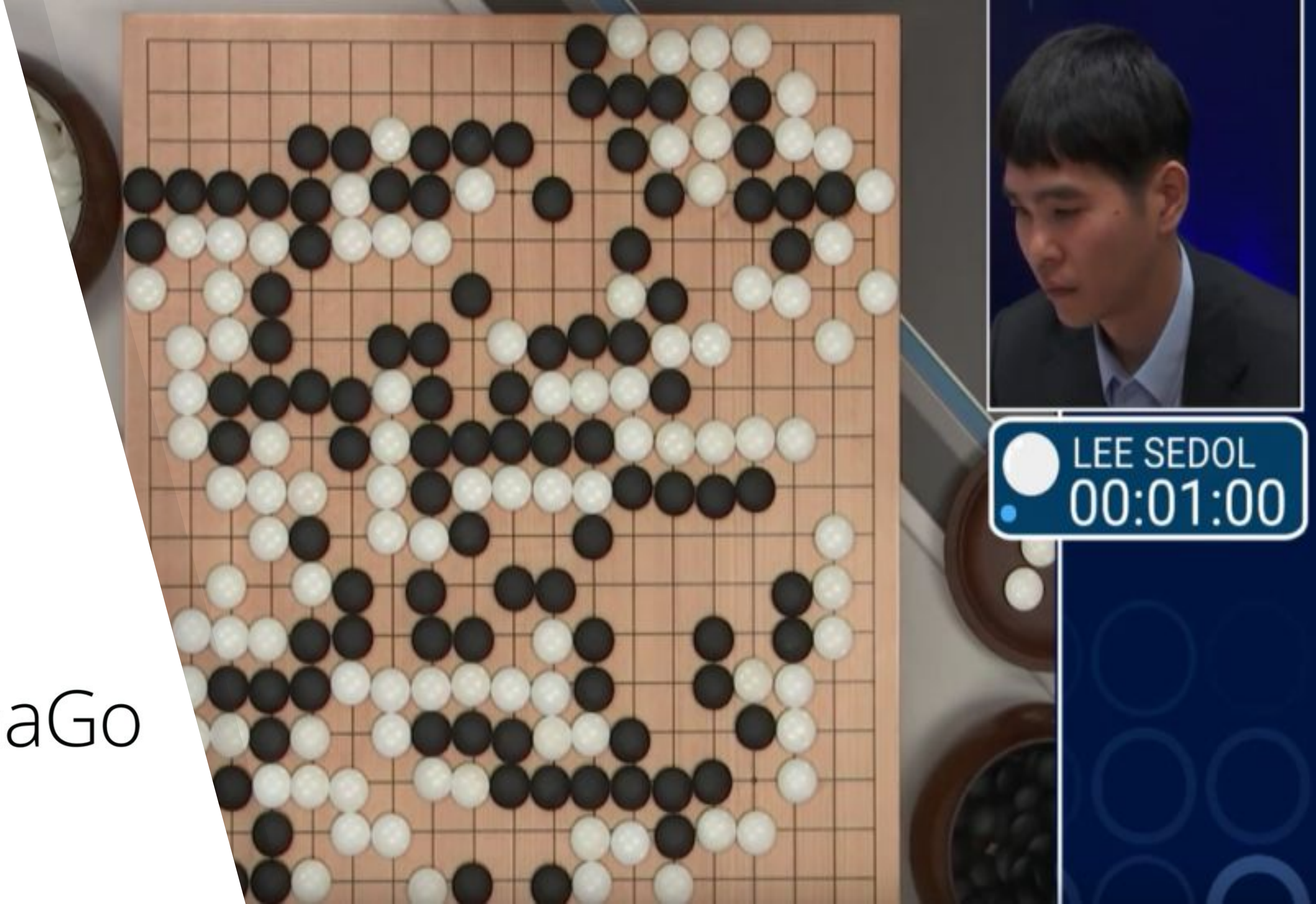
“

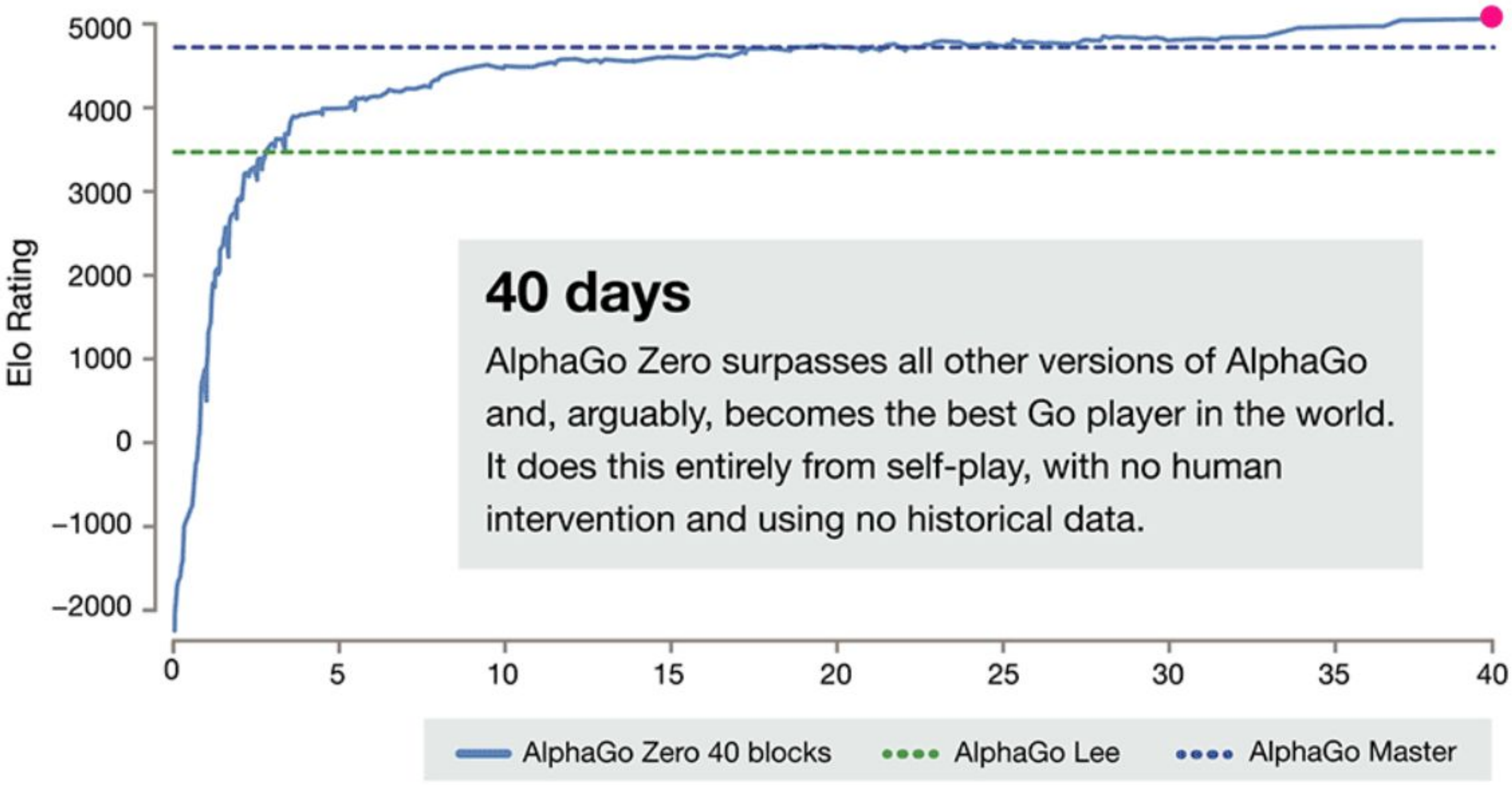
Machine Learning is a core, transformative way by which we’re rethinking how we’re doing everything

Google CEO, Sundar Pichai

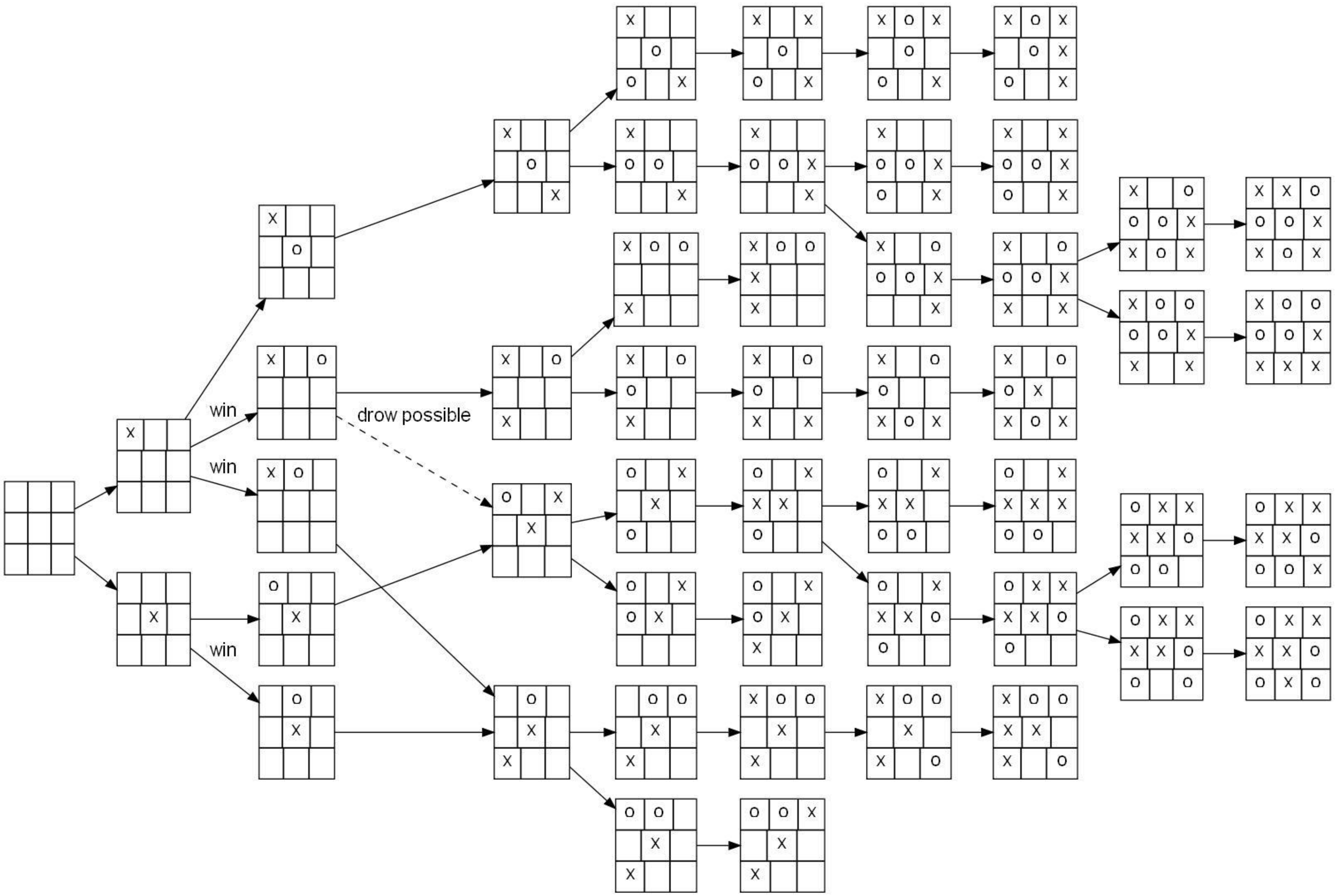
AUTONOMOUS CARS

AlphaGo





Elo Rating
5000
4000
3000
2000
1000
0
−1000
−2000
0
5
10
15
20
25
30
35
40
40 days
AlphaGo Zero surpasses all other versions of AlphaGo and, arguably, becomes the best Go player in the world. It does this entirely from self-play, with no human intervention and using no historical data.
AlphaGo Zero 40 blocks
AlphaGo Lee
AlphaGo Master

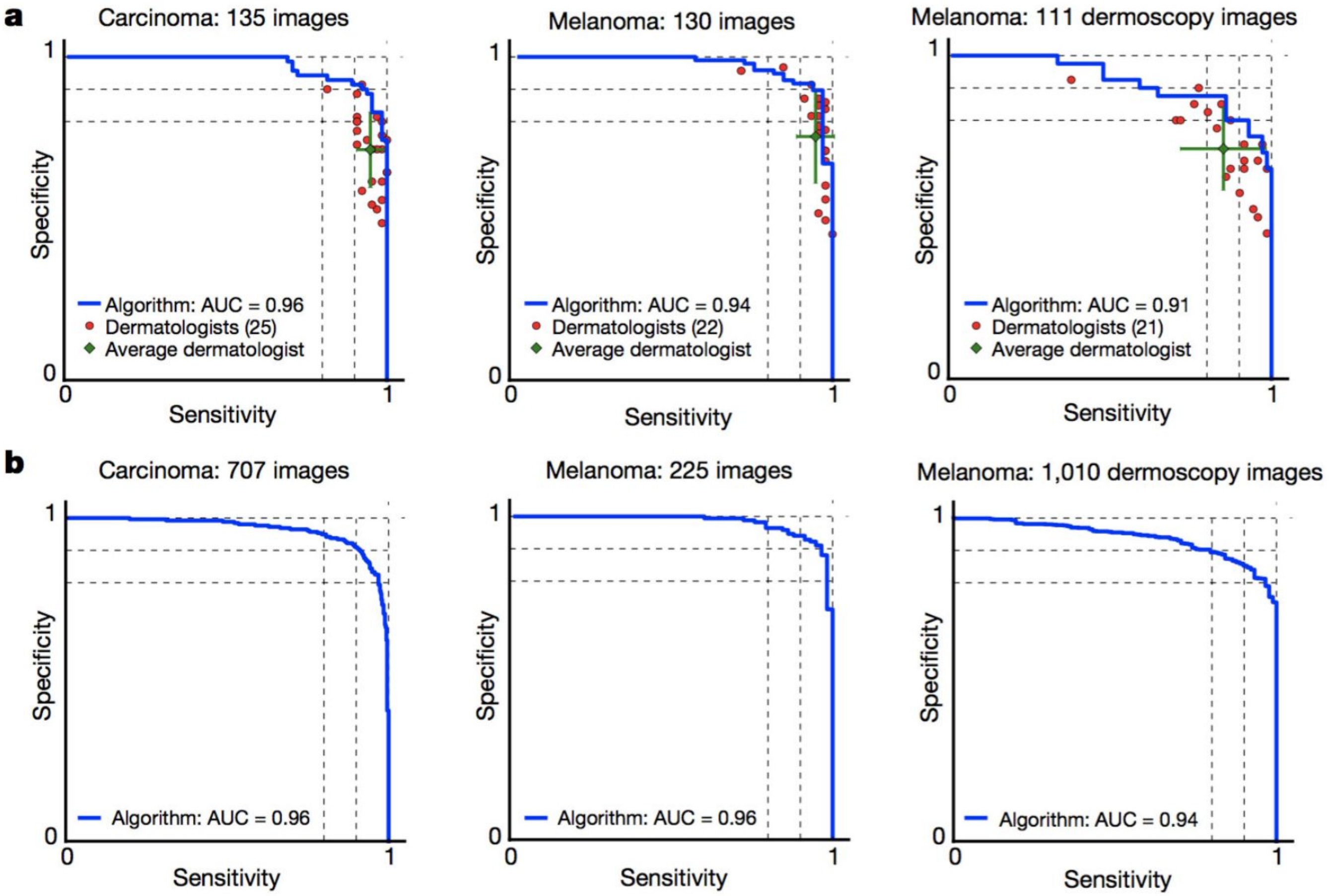


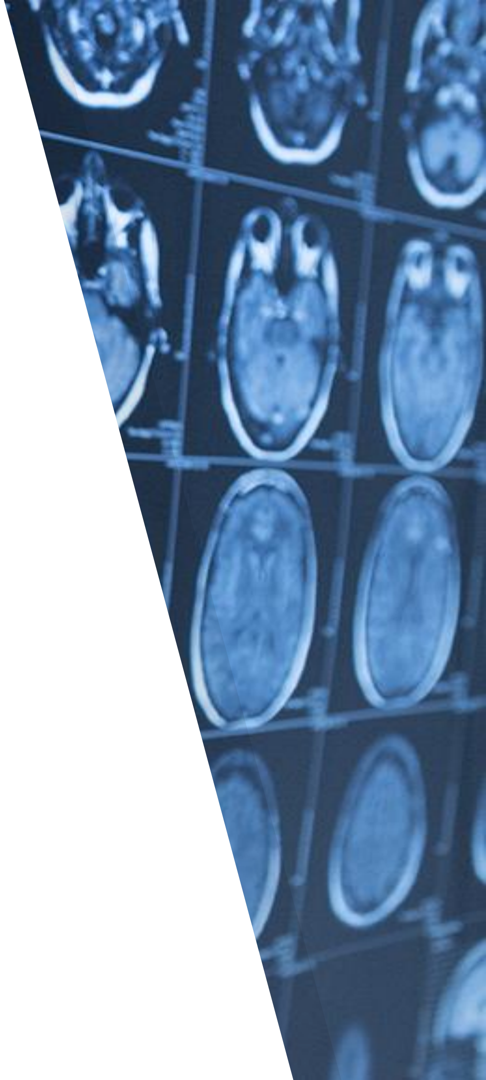
win
win
win
drow possible

JPMorgan Software Does in Seconds What Took Lawyers 360,000 Hours

by **Hugh Son**

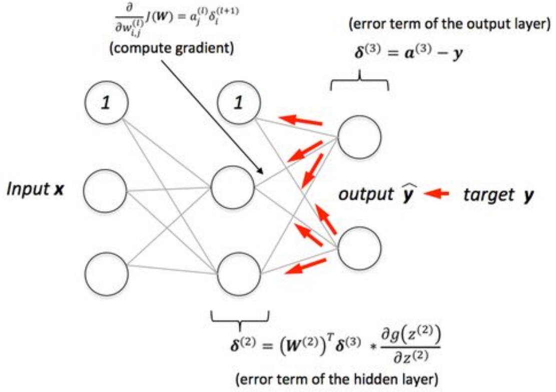
February 27, 2017, 6:31 PM CST *Updated on* February 28, 2017, 6:24 AM CST

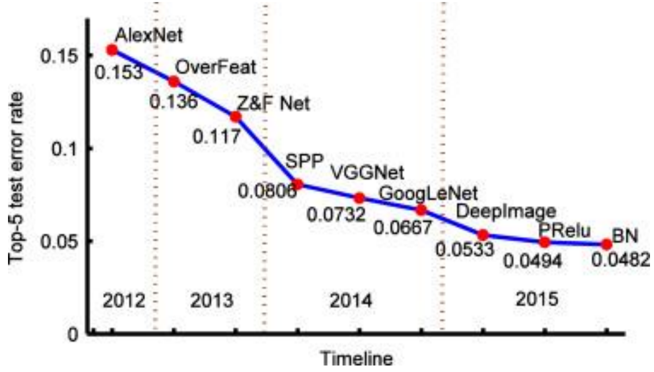

a
Carcinoma: 135 images
Melanoma: 130 images
Melanoma: 111 dermoscopy images
Specificity
Sensitivity
1
0
Algorithm: AUC = 0.96
Dermatologists (25)
Average dermatologist
Algorithm: AUC = 0.94
Dermatologists (22)
Average dermatologist
Algorithm: AUC = 0.91
Dermatologists (21)
Average dermatologist
b
Carcinoma: 707 images
Melanoma: 225 images
Melanoma: 1,010 dermoscopy images
Algorithm: AUC = 0.96
Algorithm: AUC = 0.96
Algorithm: AUC = 0.94



3.

HISTORY





1969
Perceptron
limitations

1974 Backpropagation

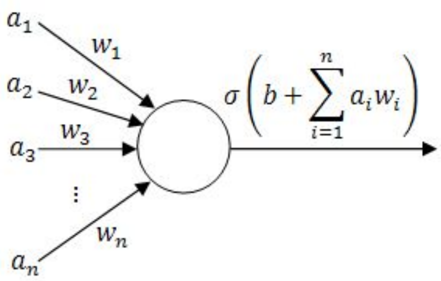
2012 AlexNet

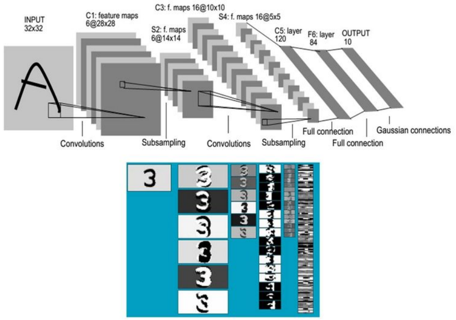
1958 Perceptron

AI Winter

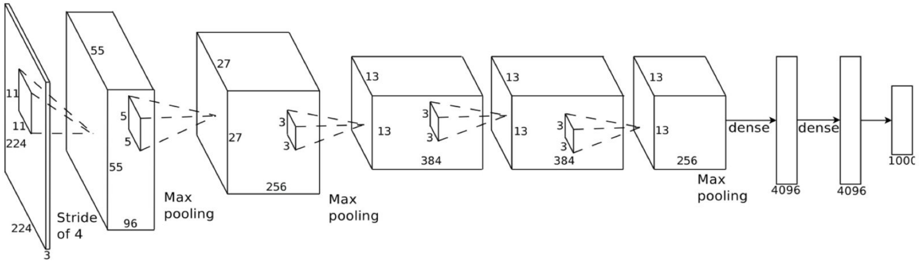
1998 LeNet

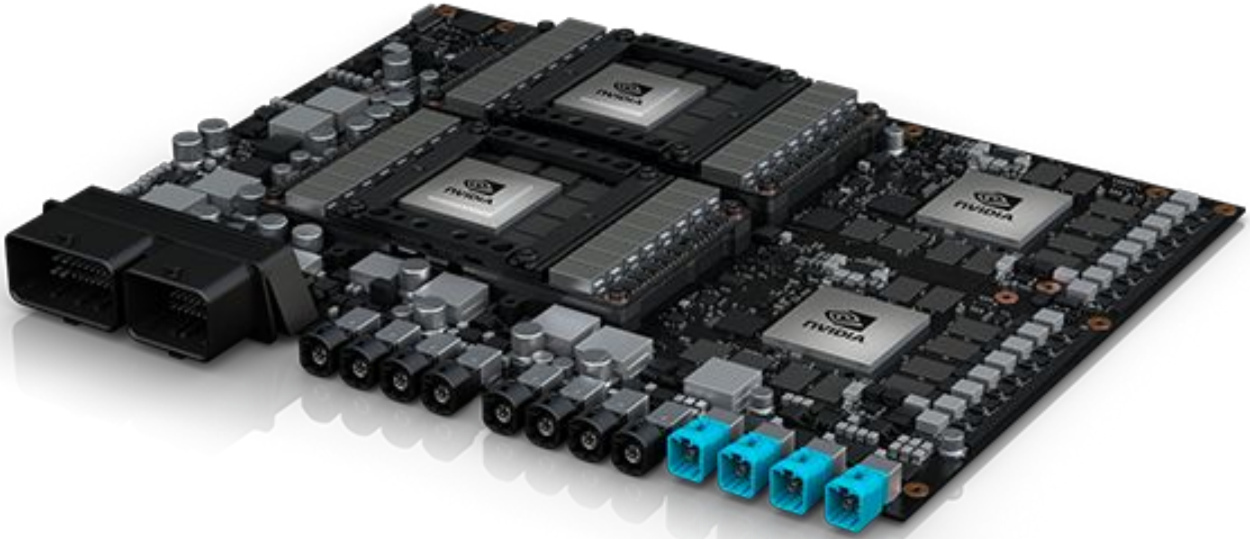
2012 Google Brain
on 16k cores





3 DRIVING FACTORS

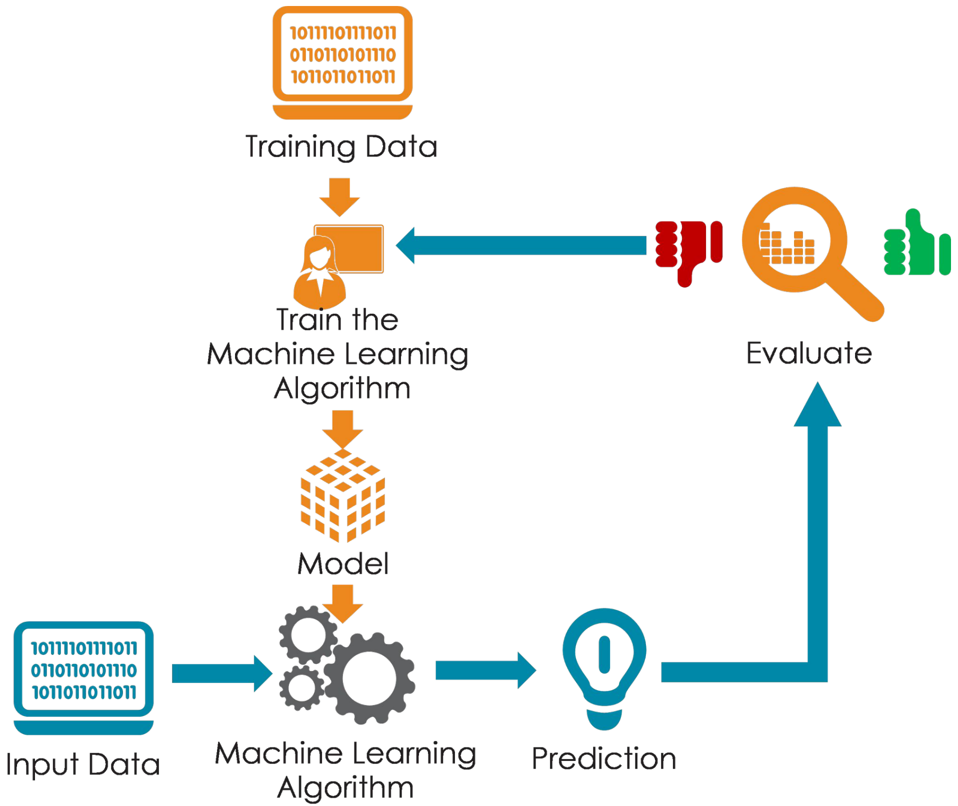




320 TOPS

TYPES

PIPELINE



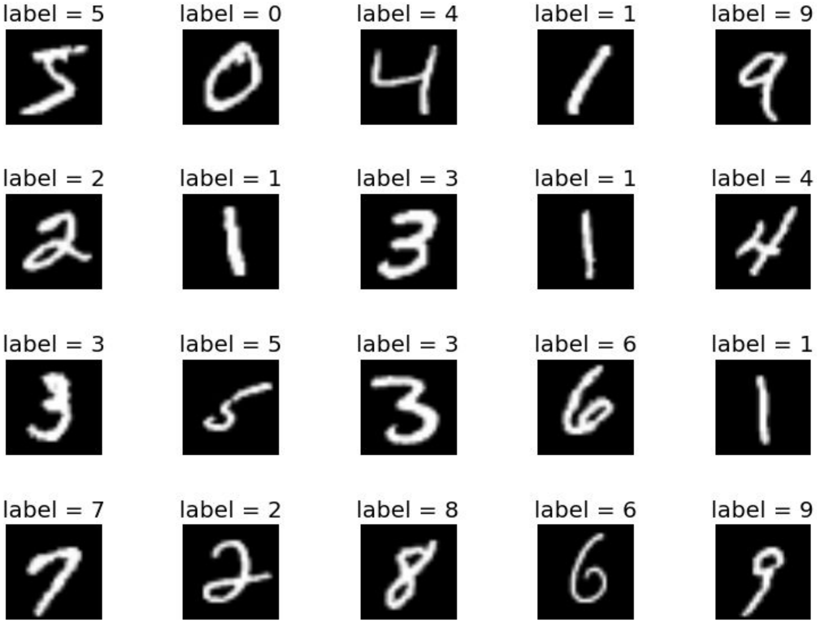
10111101111011
0110110101110
1011011011011
Training Data
Train the
Machine Learning
Algorithm
Evaluate
Model
10111101111011
0110110101110
1011011011011
Input Data
Machine Learning
Algorithm
Prediction

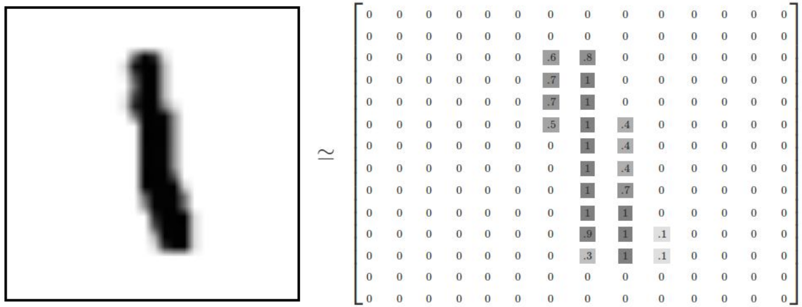
DATASET

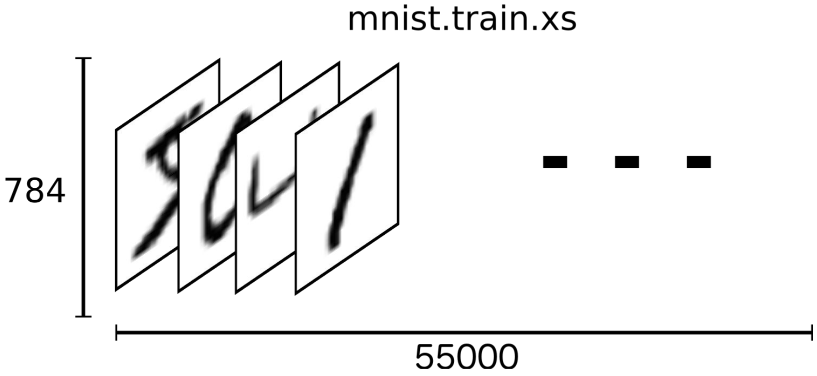
data set, m samples

$$\begin{array}{rcccc} & \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \dots & \mathbf{x}^{(m)} \\ \text{n features} & \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \dots \\ x_n^{(1)} \end{bmatrix} & \begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \\ \dots \\ x_n^{(2)} \end{bmatrix} & & \begin{bmatrix} x_1^{(m)} \\ x_2^{(m)} \\ \dots \\ x_n^{(m)} \end{bmatrix} \\ \hline \text{label} & y^{(1)} & y^{(2)} & & y^{(1)} \end{array}$$

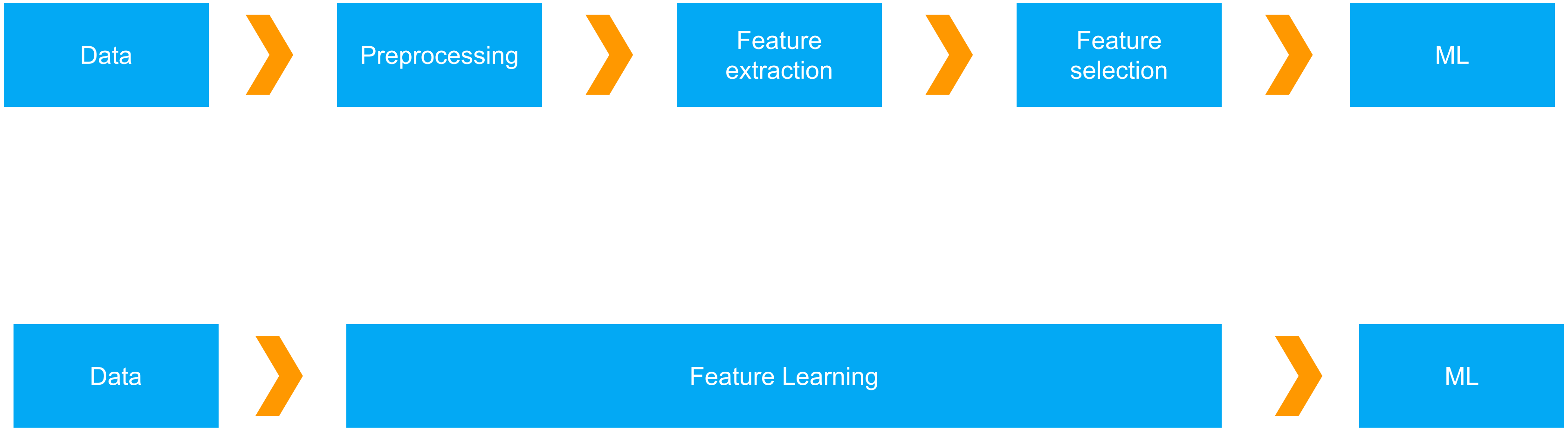
MNIST dataset

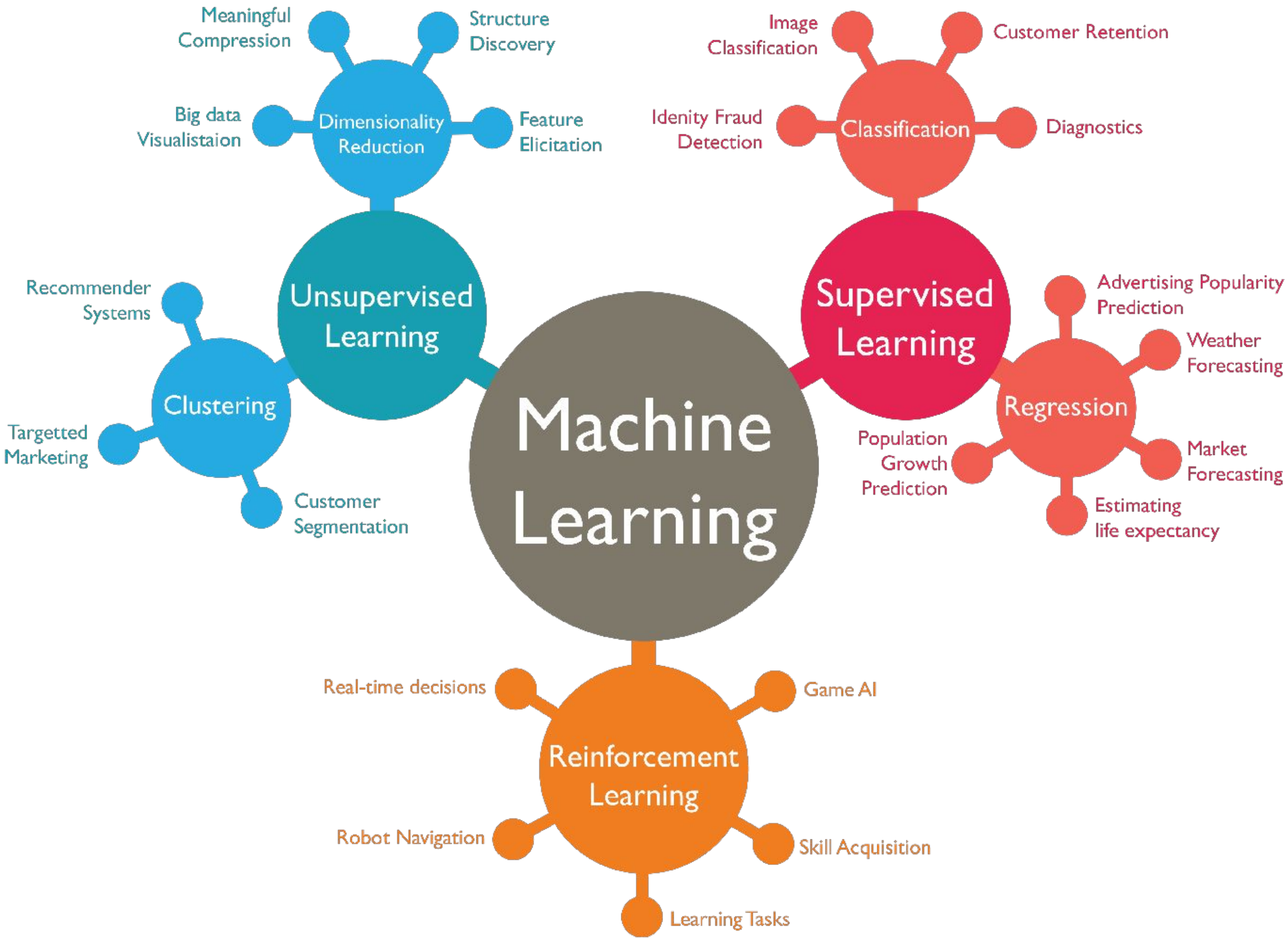




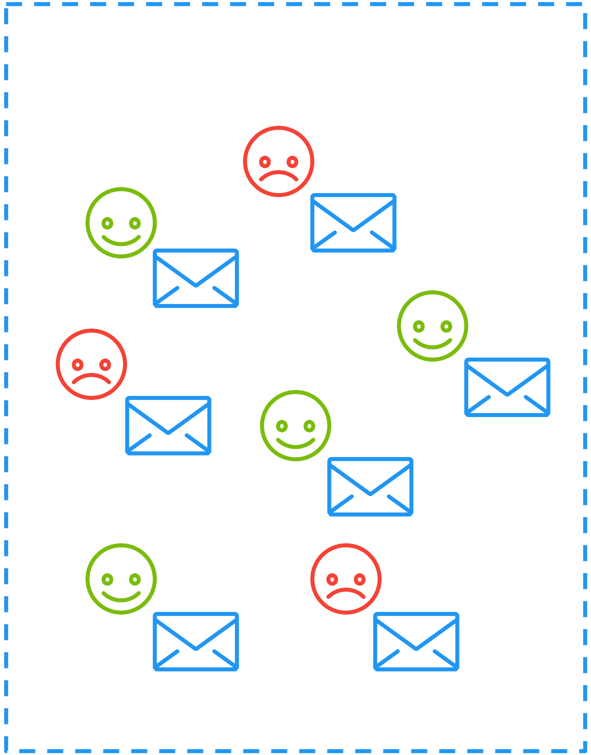


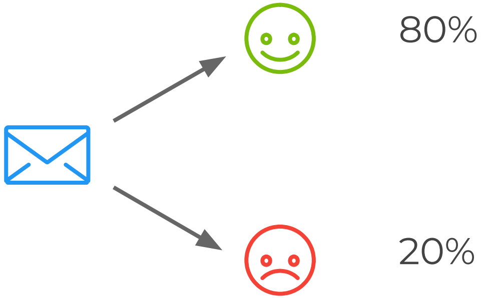
ML WORKFLOW


Data
Preprocessing
Feature extraction
Feature selection
ML
Data
Feature Learning
ML

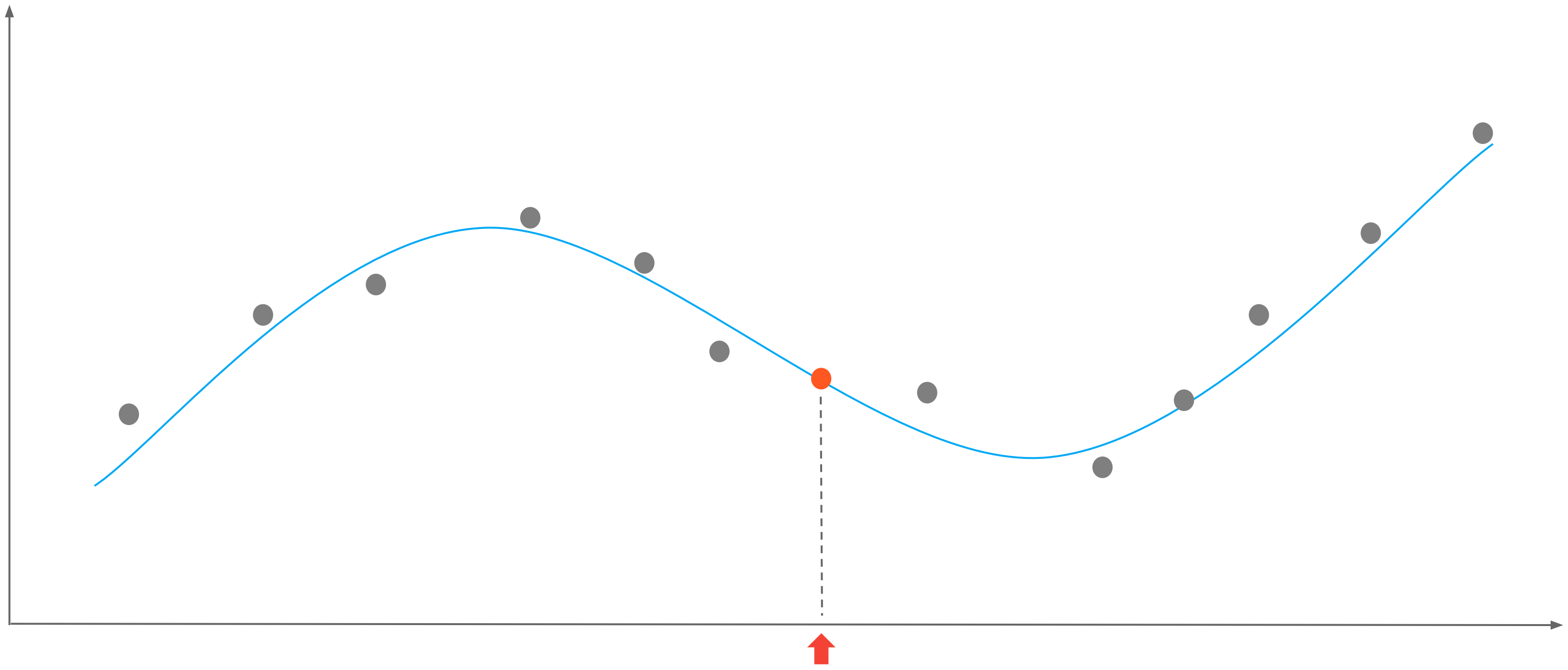

Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

CLASSIFICATION

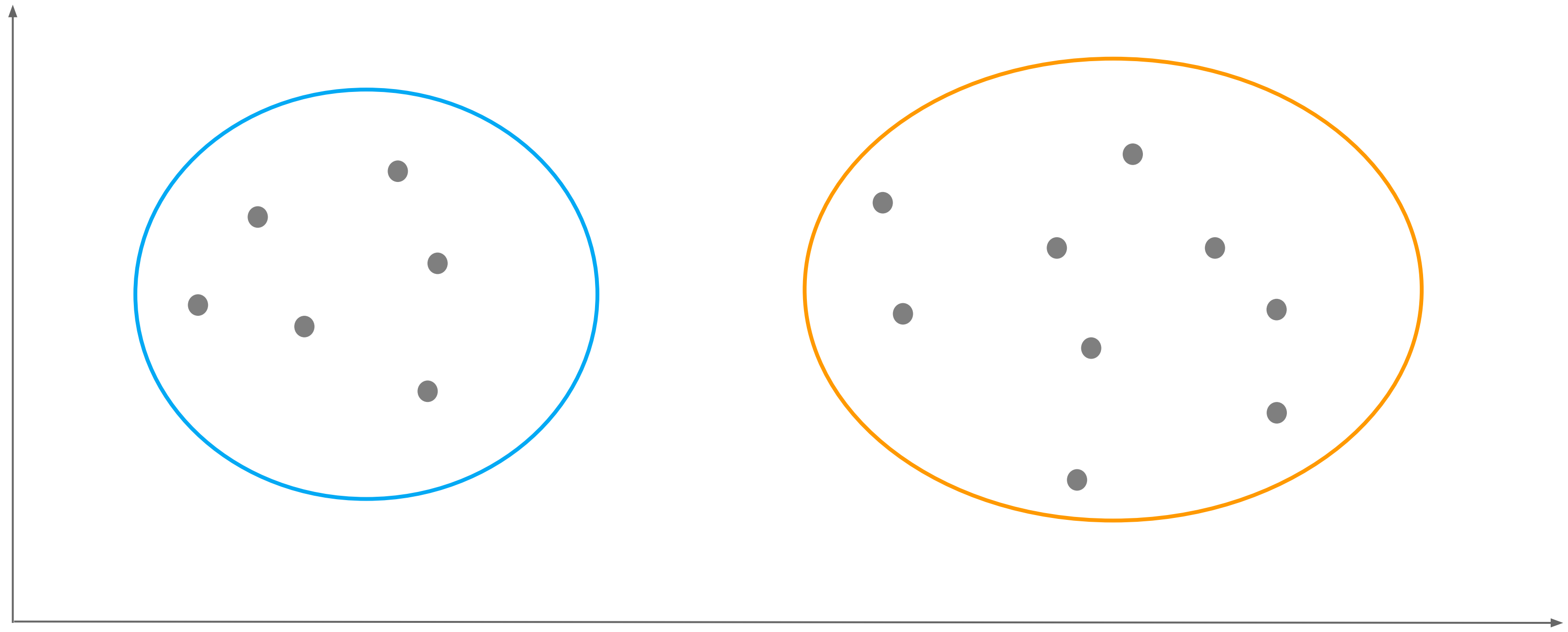




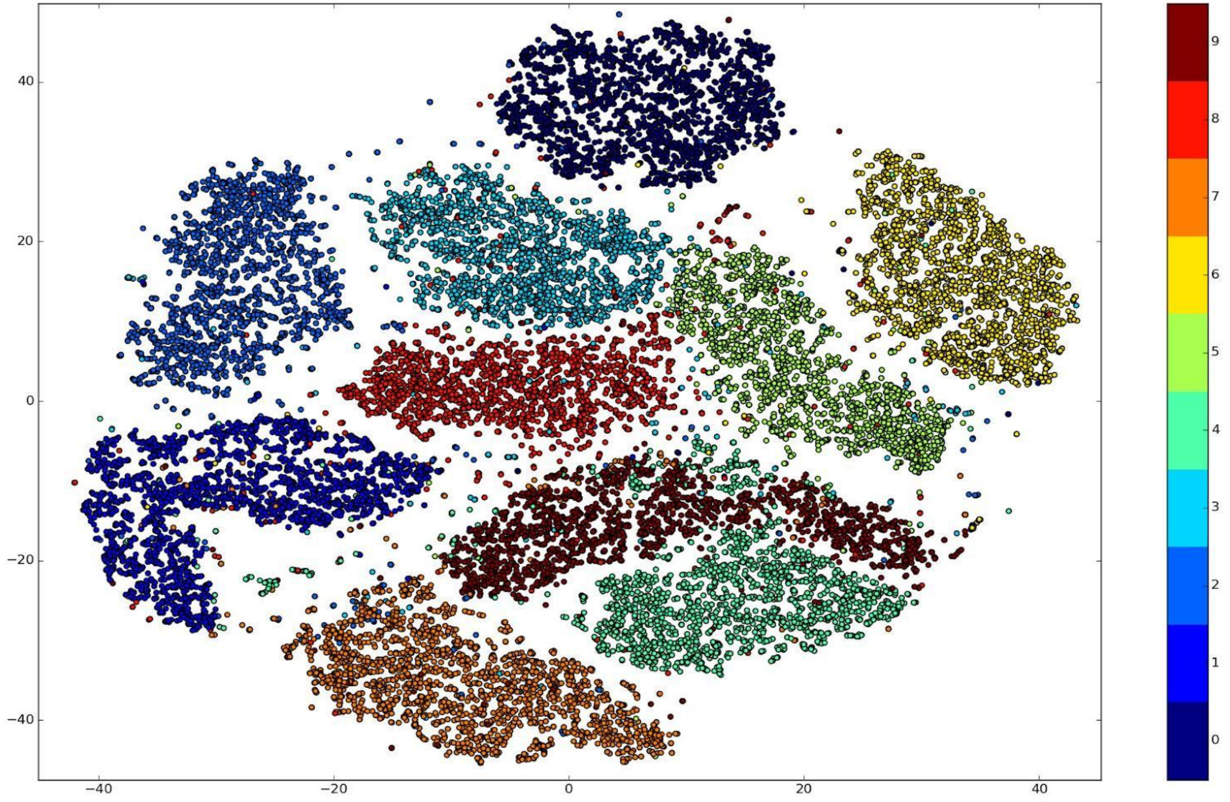
REGRESSION



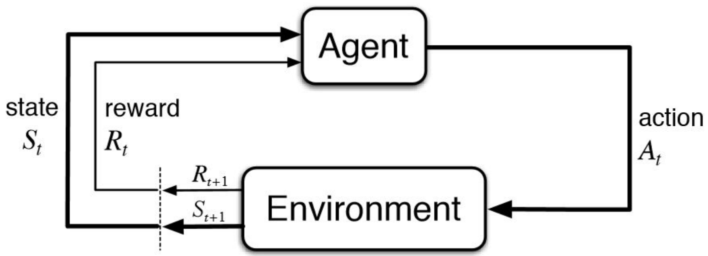
CLUSTERING

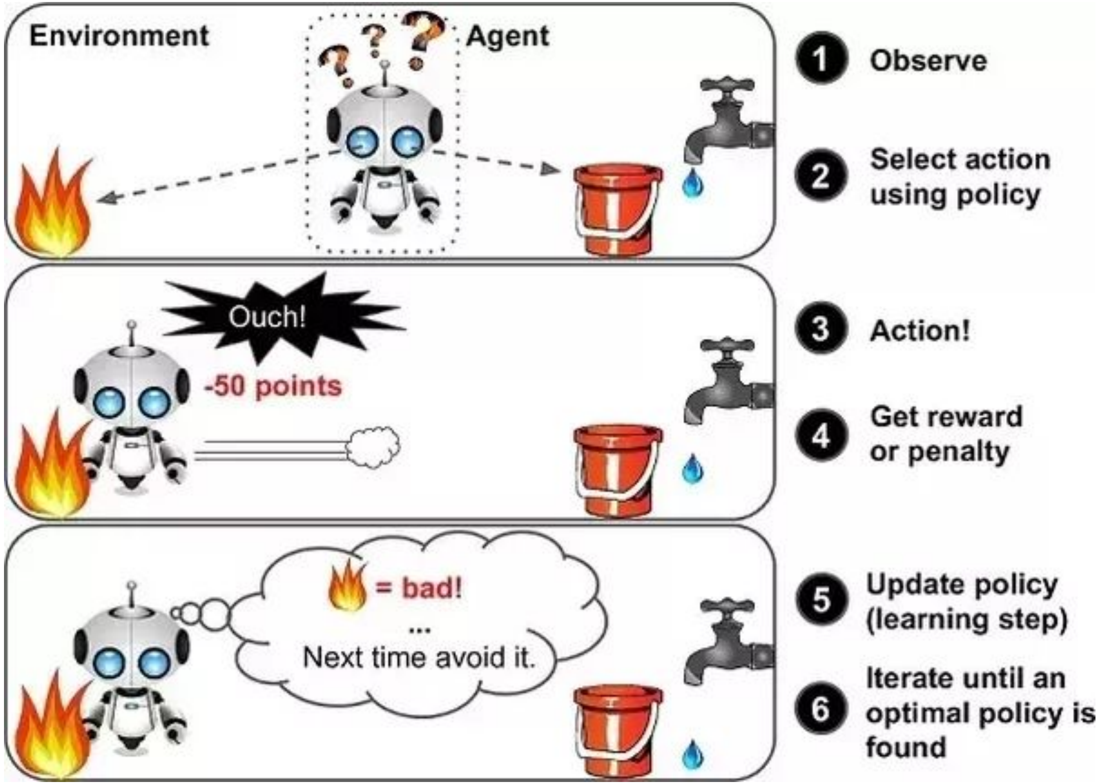


DIMENSIONALITY REDUCTION



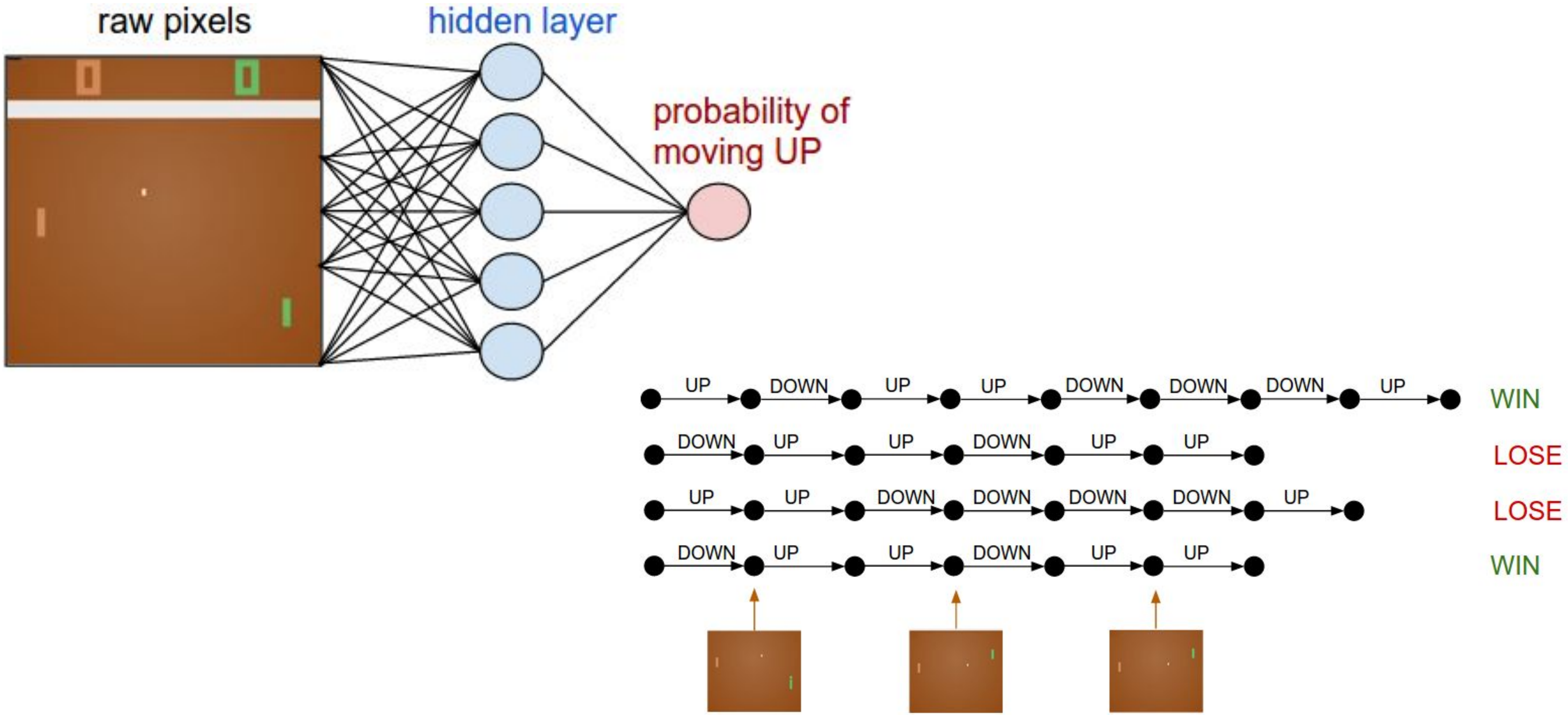
REINFORCEMENT LEARNING





DEMO
REINFORCEMENT

Atari 2600 Pong



raw pixels
hidden layer
probability of moving UP
UP DOWN UP UP DOWN DOWN DOWN UP
WIN
DOWN UP UP DOWN UP UP
LOSE
UP UP DOWN DOWN DOWN DOWN UP
LOSE
DOWN UP UP DOWN UP UP
WIN

DEMO
REGRESSION

House price prediction

PRICE ESTIMATION

size

rooms

bedrooms

schools

sea proximity

area

```
def estimate_house_sales_price(
num_of_bedrooms, sqft, neighborhood):


    price = 78427

    price += num_of_bedrooms * 31.45678
    price += size * 953.764231
    price += neighborhood * 132.42341421

    return price
```

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

COST FUNCTION

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

All samples

Computed result

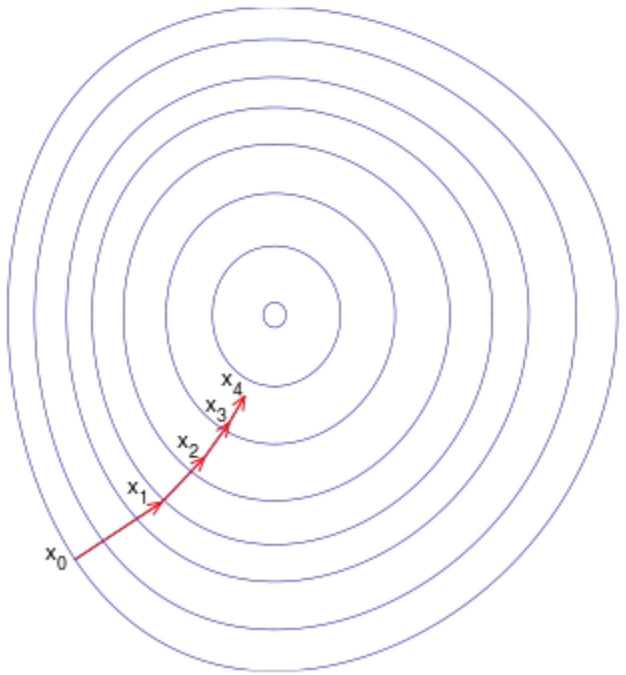
Expected result

GRADIENT DESCENT

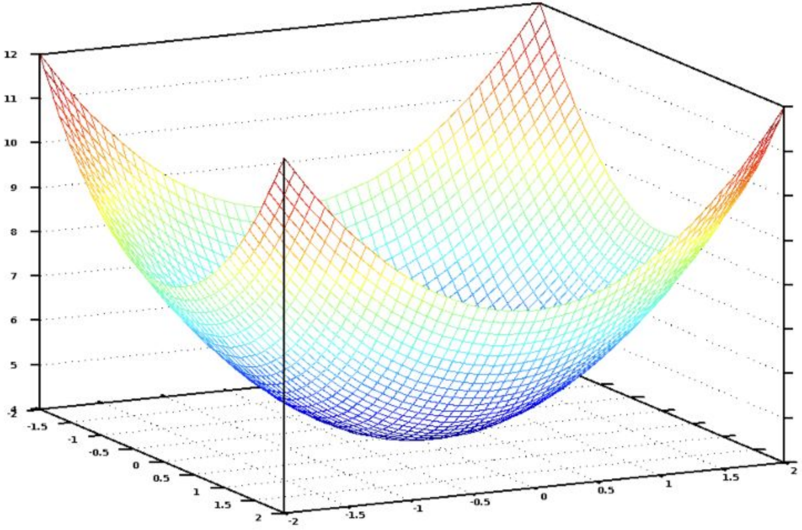
Repeat until convergence {

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

}



GRADIENT DESCENT



Cost Function – "One Half Mean Squared Error":

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Objective:

$$\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$$

Update rules:

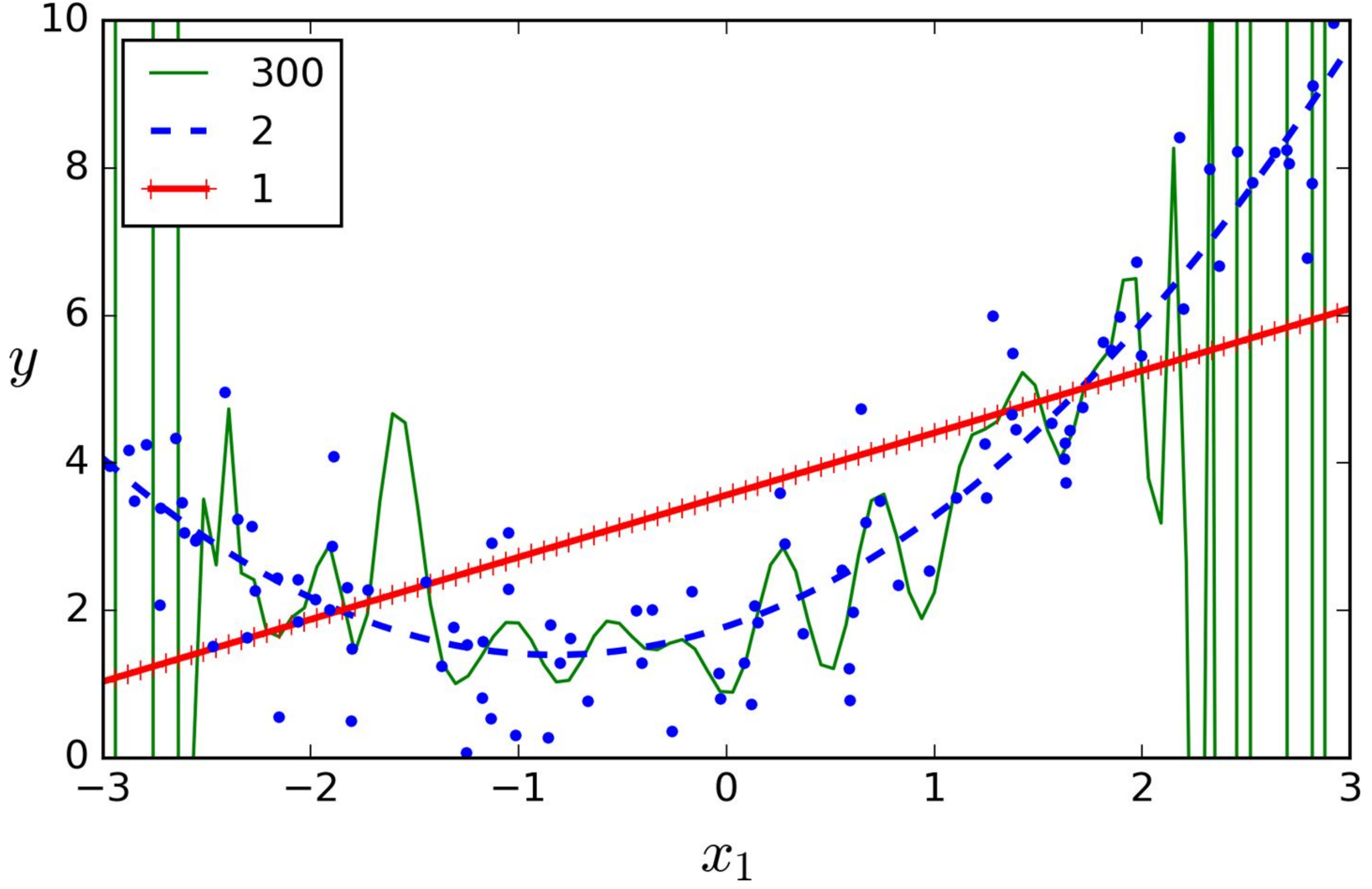
$$\theta_0 := \theta_0 - \alpha \frac{d}{d\theta_0} J(\theta_0, \theta_1)$$

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_0, \theta_1)$$

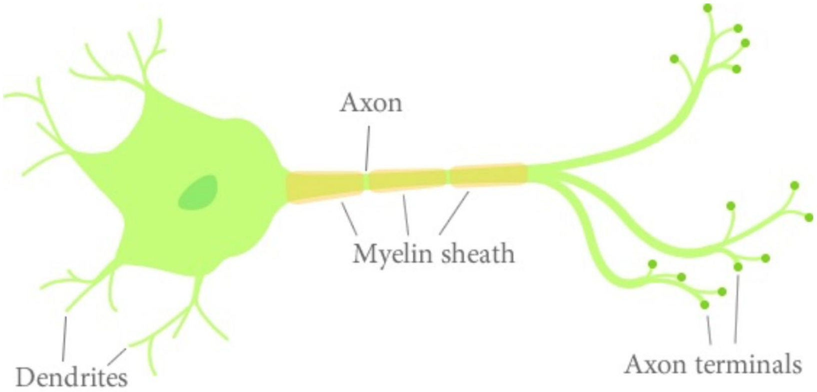
Derivatives:

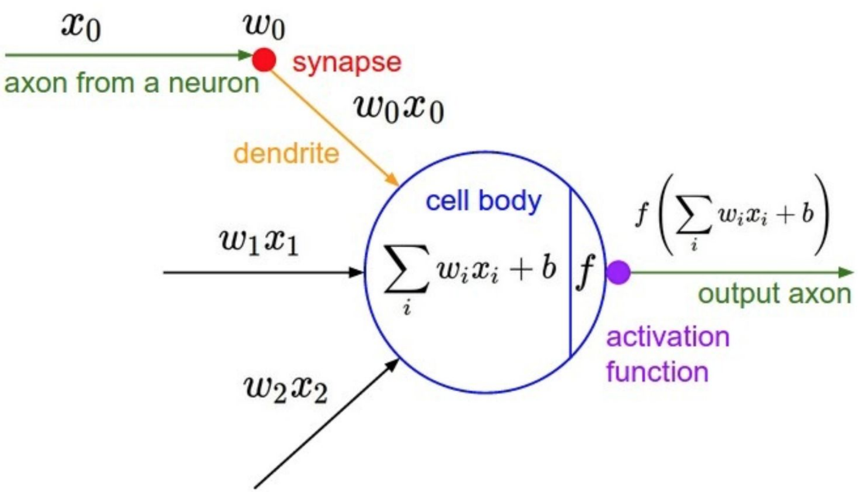
$$\frac{d}{d\theta_0} J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

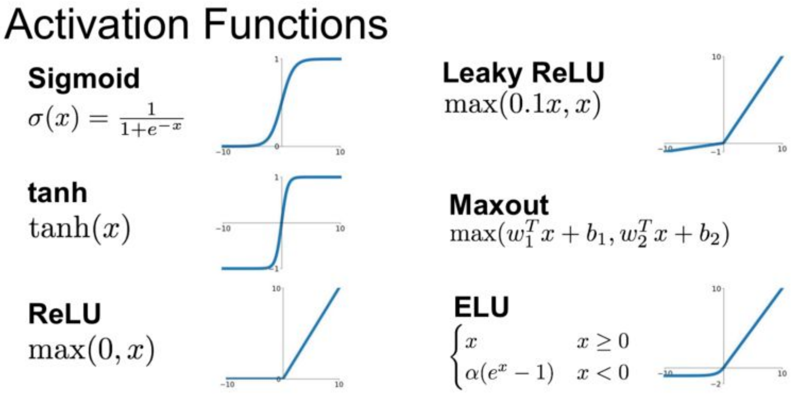
$$\frac{d}{d\theta_1} J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x^{(i)}$$


300
2
1
10
8
6
4
2
0
y
−3
−2
−1
0
1
2
3
x_1

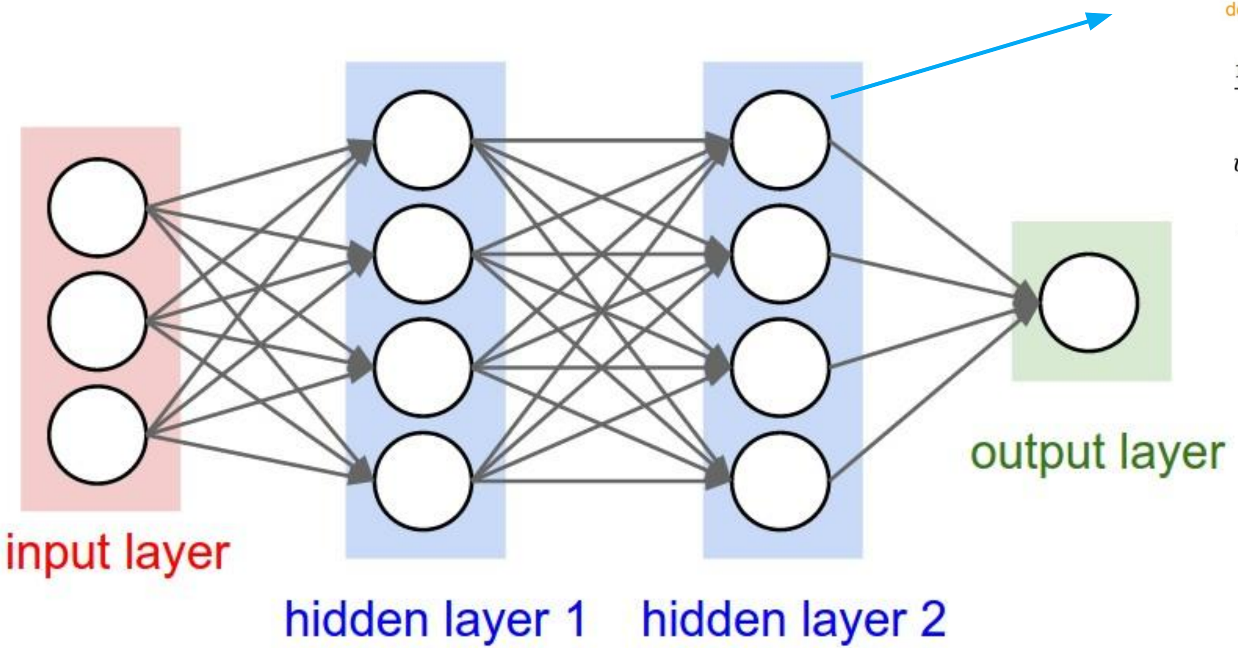
NEURONS

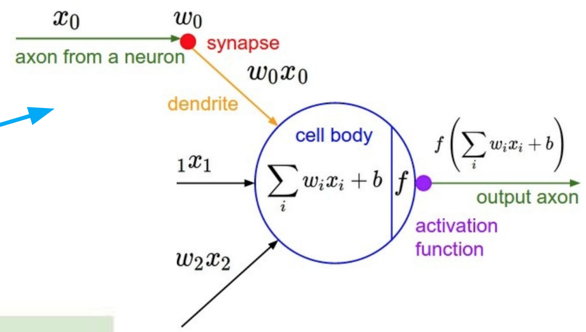

Axon
Myelin sheath
Dendrites
Axon terminals


x_0
w_0
synapse
axon from a neuron
$w_0 x_0$
dendrite
cell body
$f\left(\sum_i w_i x_i + b\right)$
$w_1 x_1$
$\sum_i w_i x_i + b$
f
output axon
activation function
$w_2 x_2$


Activation Functions
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$
Leaky ReLU
$\max(0.1x, x)$
tanh
$\tanh(x)$
Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$
ReLU
$\max(0, x)$
ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

NETWORK





$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} \left[y_k^{(i)} \log((h_\Theta(x^{(i)}))_k) + (1 - y_k^{(i)}) \log(1 - (h_\Theta(x^{(i)}))_k) \right] + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (\Theta_{j,i}^{(l)})^2$$

Summary: the equations of backpropagation

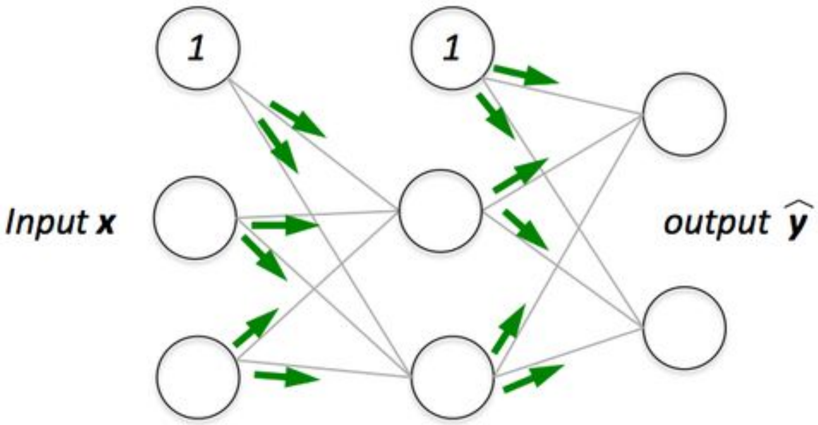
$$\delta^L = \nabla_a C \odot \sigma'(z^L) \tag{BP1}$$

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \tag{BP2}$$

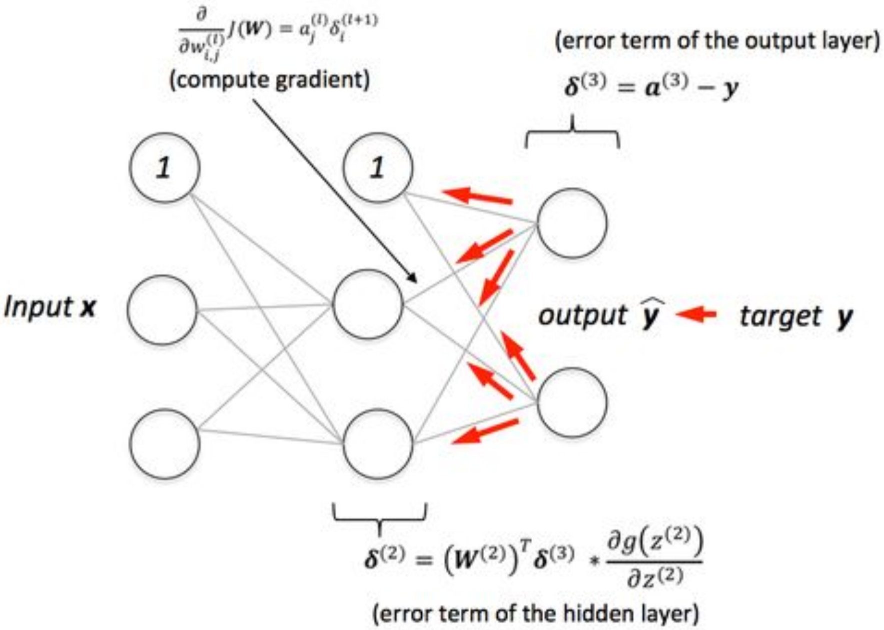
$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \tag{BP3}$$

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \tag{BP4}$$

1. FORWARD PROPAGATION



2. BACKPROPAGATION



WHY DEEP LEARNING ?

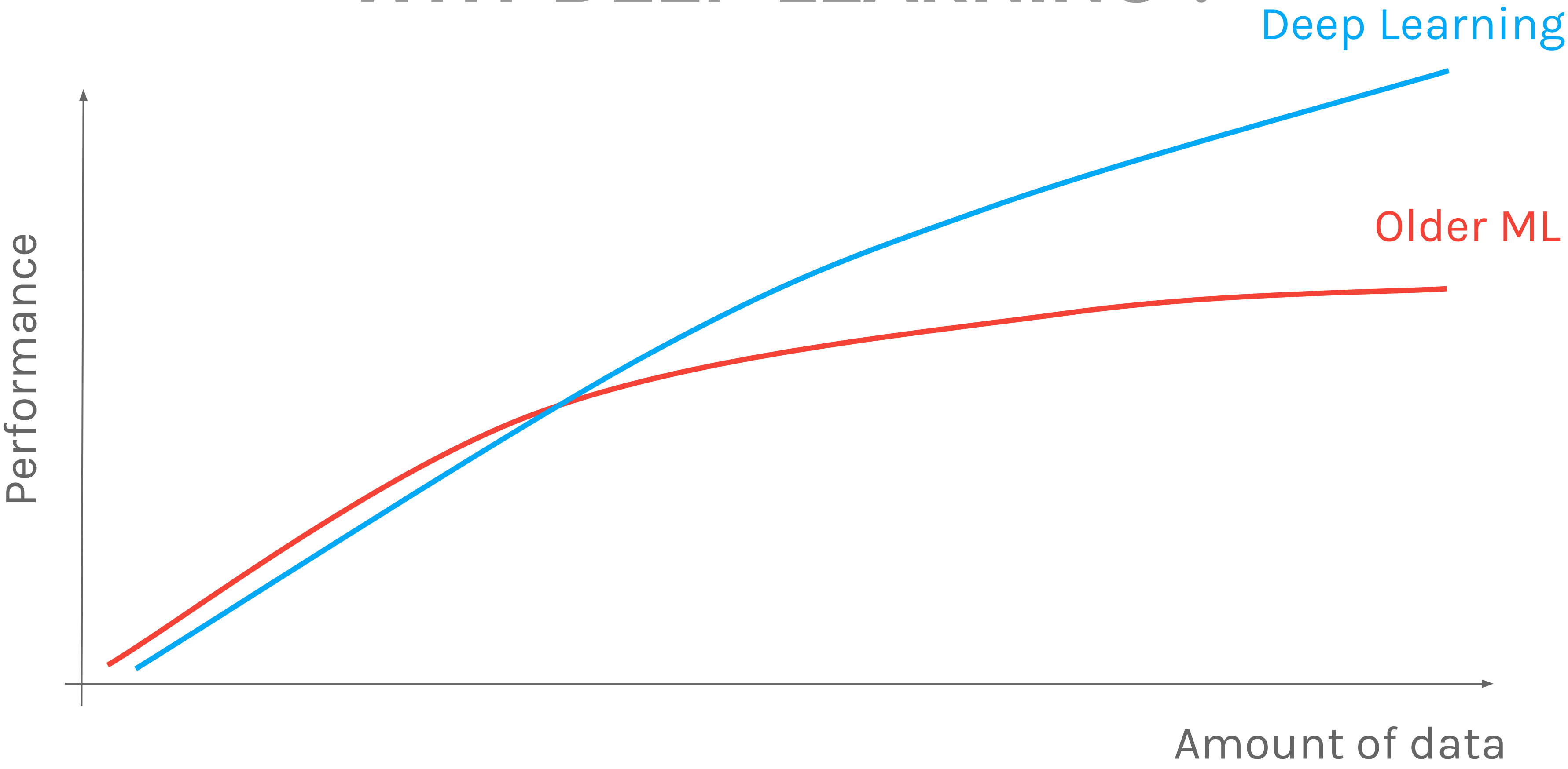
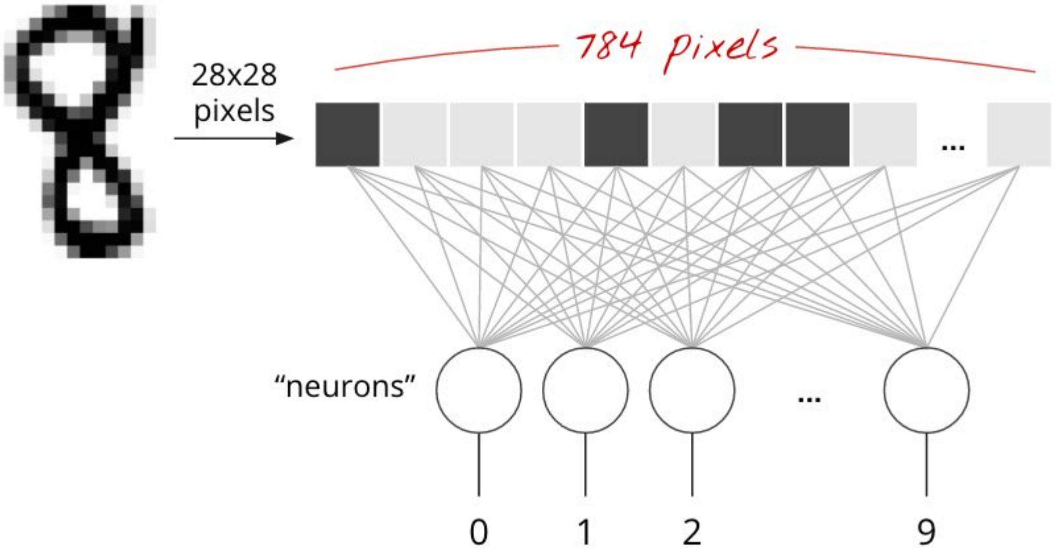
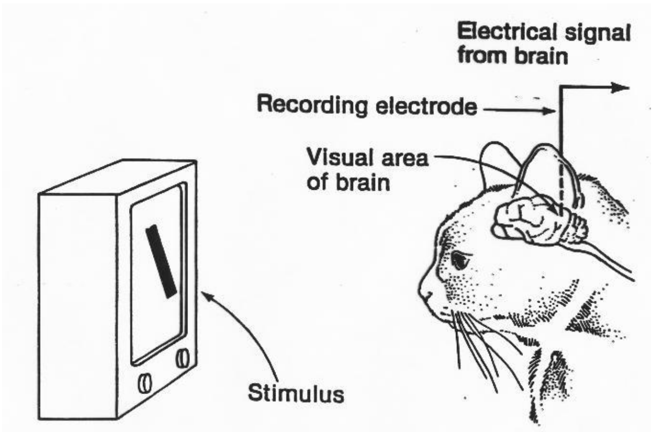
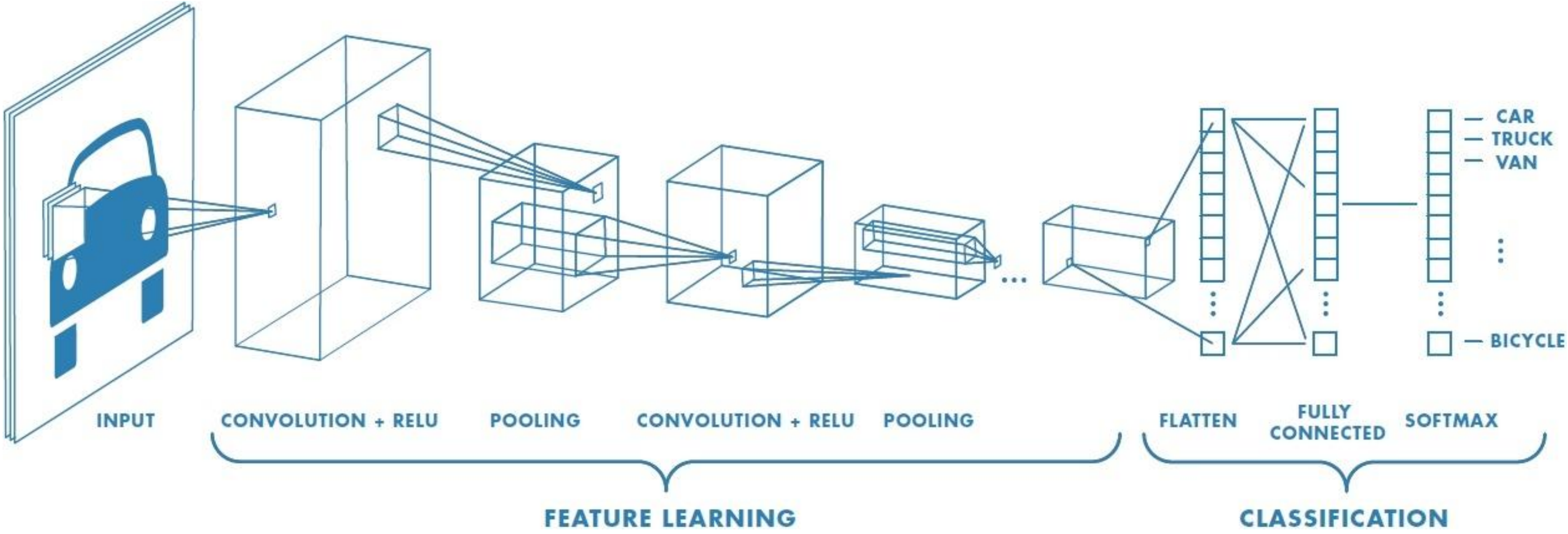


IMAGE RECOGNITION ISSUES

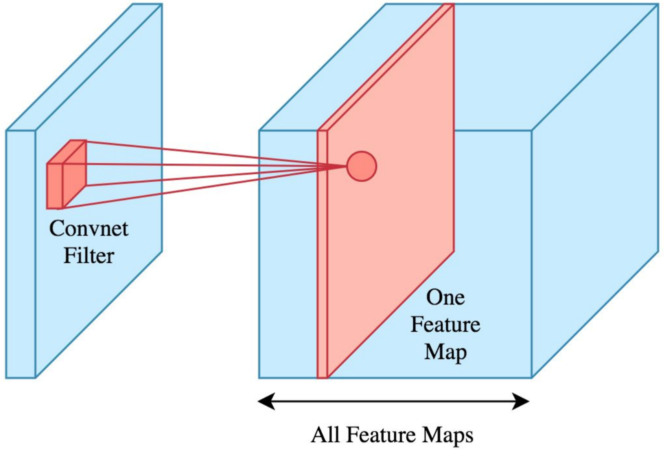


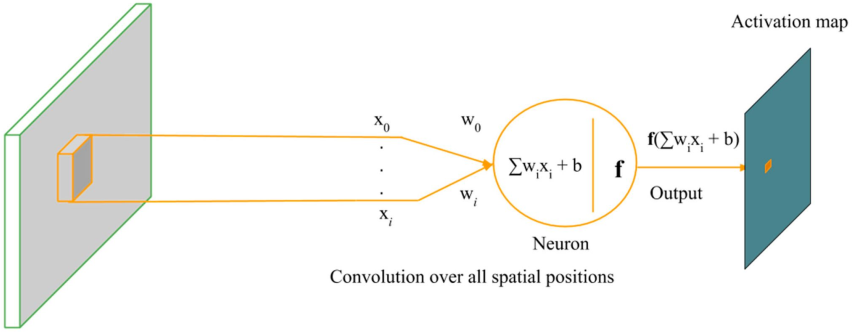


CONVOLUTIONAL NEURAL NETWORK

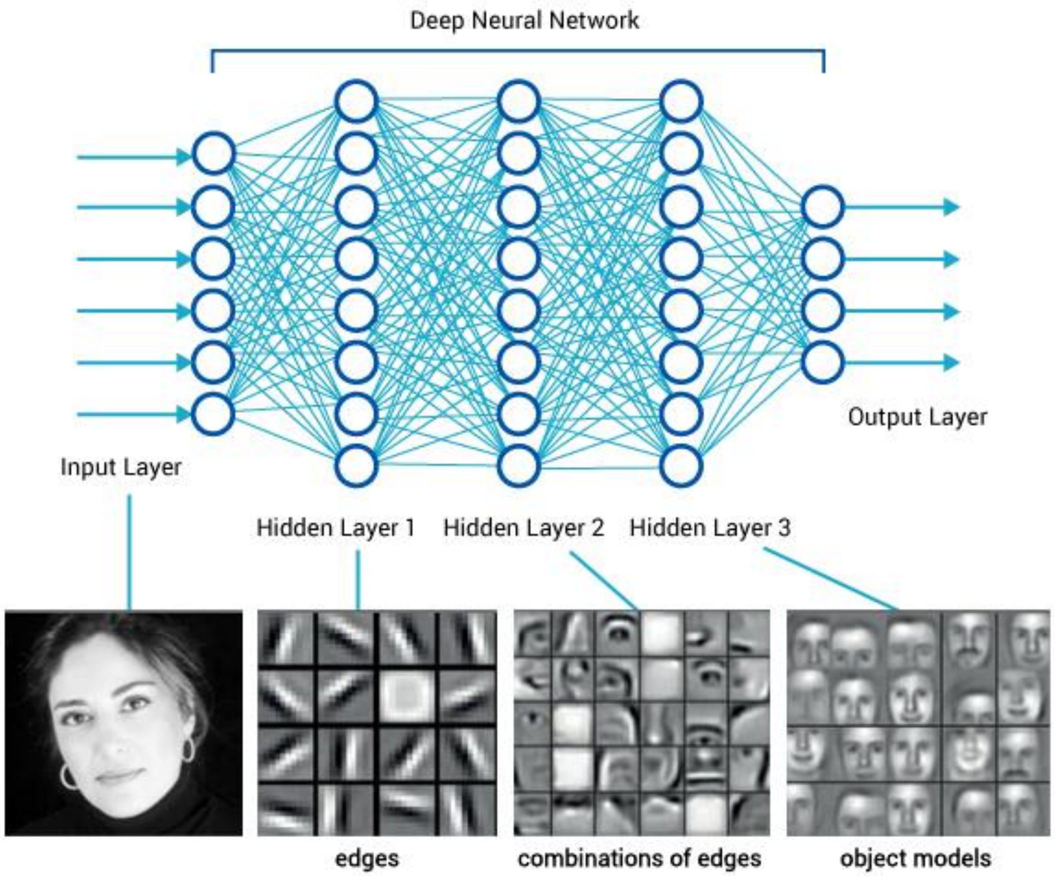


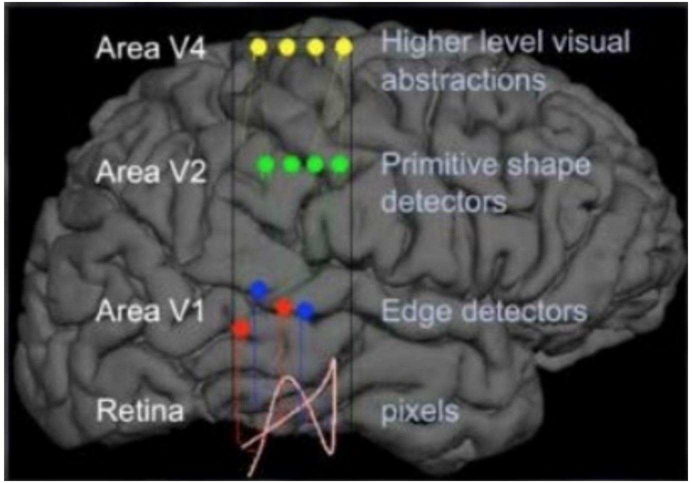
CONVOLUTION OPERATION



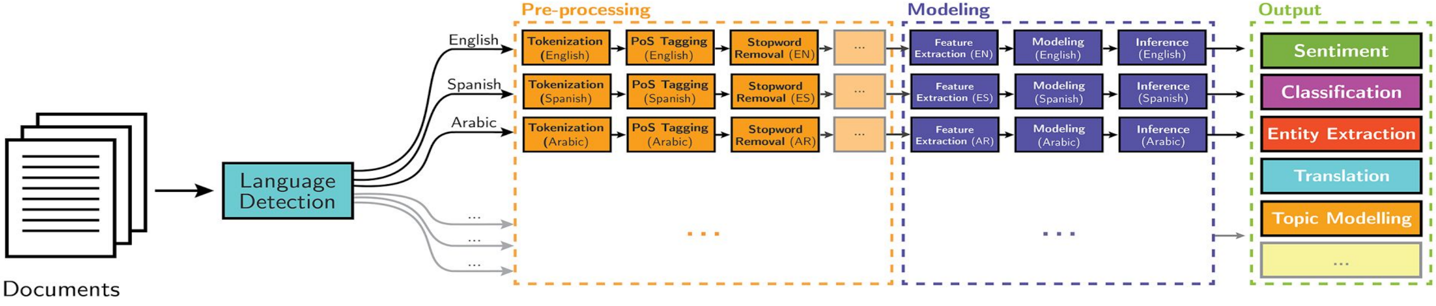


- Locally Receptive Fields
- Shared Weights
- Spatial or temporal sub-sampling

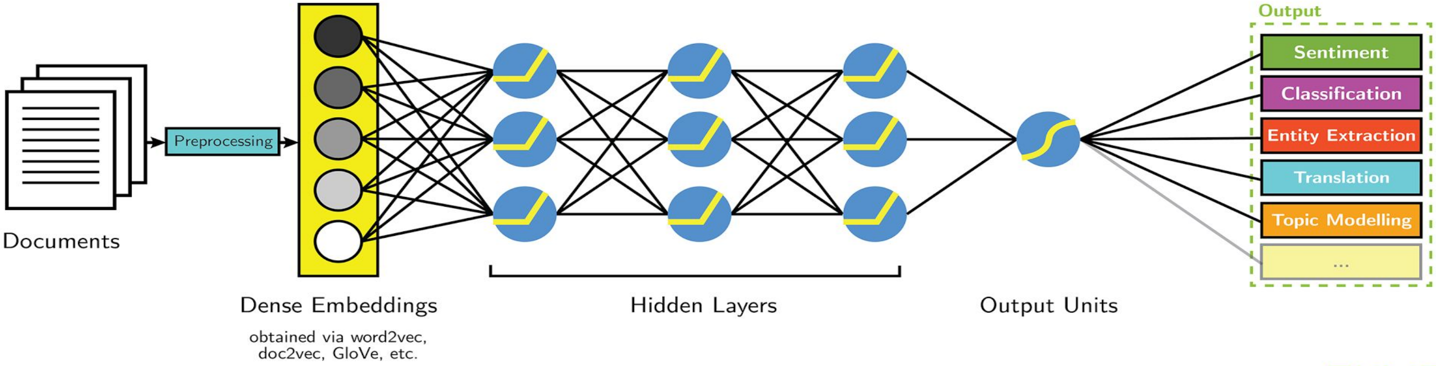

Deep Neural Network
Input Layer
Hidden Layer 1
Hidden Layer 2
Hidden Layer 3
Output Layer
edges
combinations of edges
object models


Area V4
Higher level visual abstractions
Area V2
Primitive shape detectors
Area V1
Edge detectors
Retina
pixels

Classical NLP



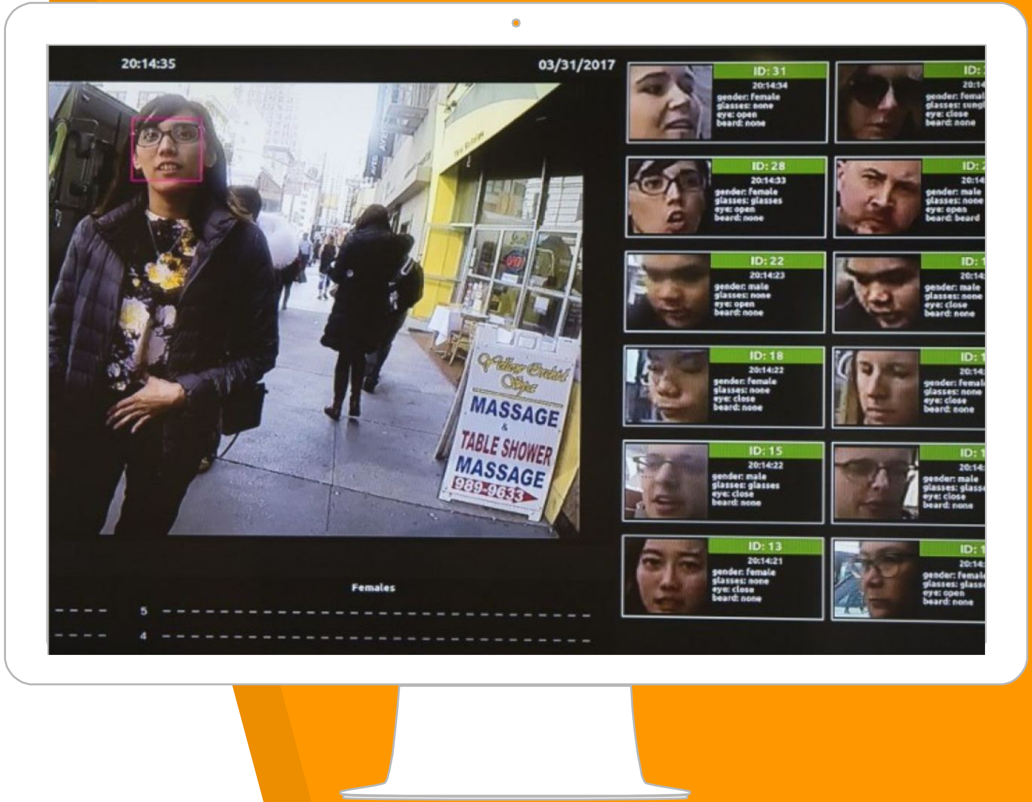
Deep Learning-based NLP



AYLIEN

DEMO
NEURAL NETWORK

Age and gender recognition

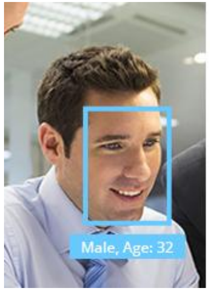


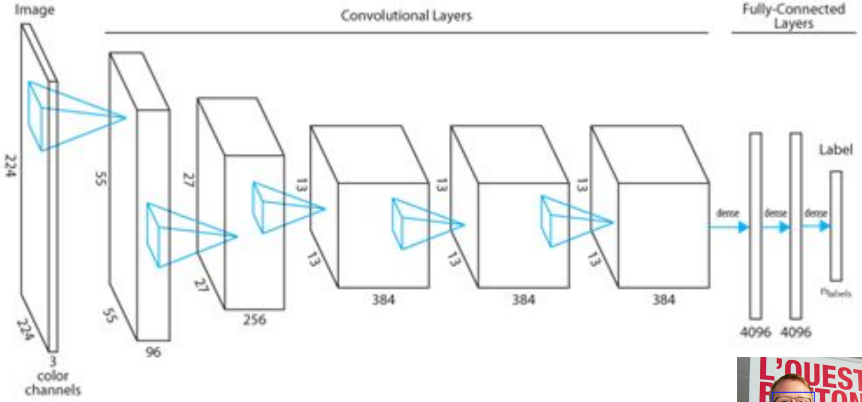
Step 1: Training
(in Data Center - Over Hours/days/Weeks)

Lots of labeled input data

Create DNN model

Output: Trained Model





Step 2: Inference
(Endpoint or Data Center - Instantaneous)

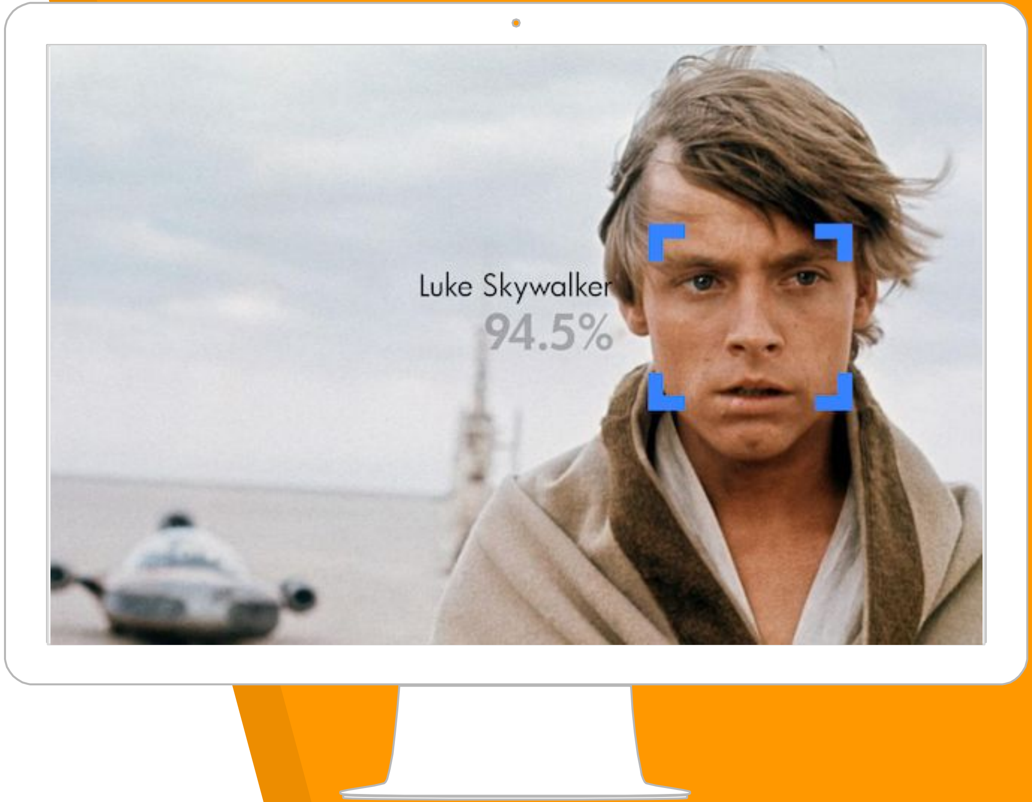
Input from camera

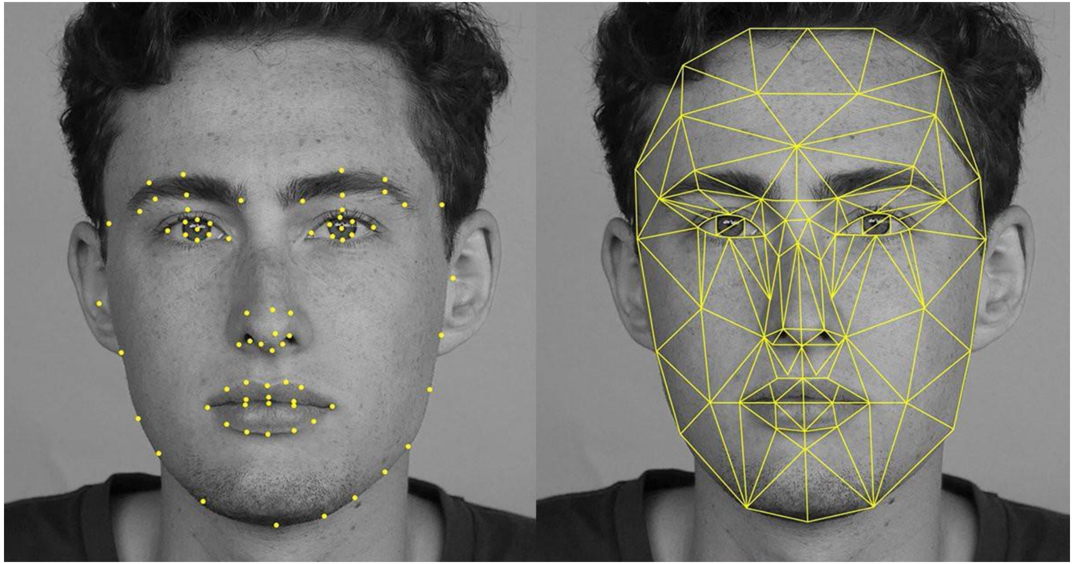
Trained neural model

Output: Classification

DEMO
NEURAL NETWORK

Face recognition

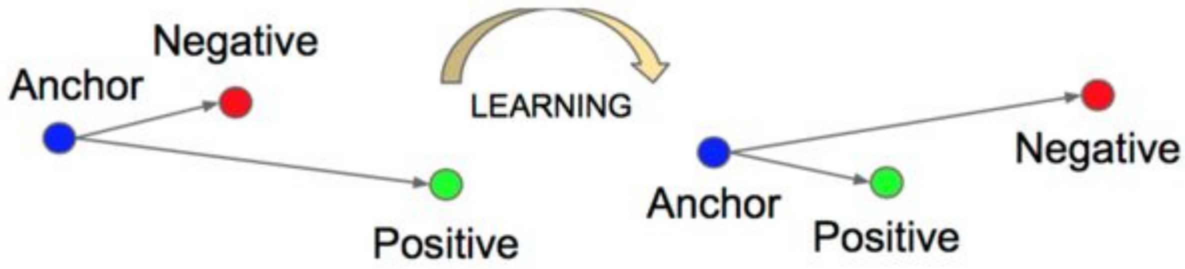




1960, Woodrow Bledsoe

Technique involving marking the coordinates of prominent features of a face (hairline, eyes, nose ...)

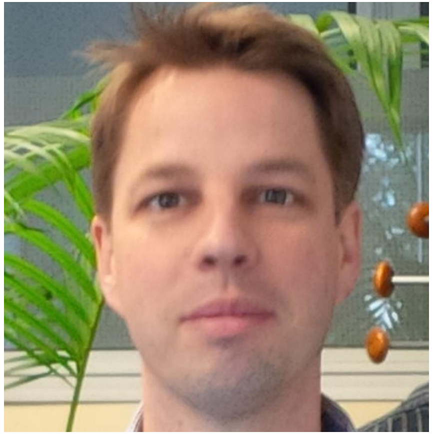
2015 GOOGLE FACENET



Triplet Loss Function

$$\sum_{i}^{N}\left[\|f(x_i^a)-f(x_i^p)\|_2^2-\|f(x_i^a)-f(x_i^n)\|_2^2+\alpha\right]_+$$

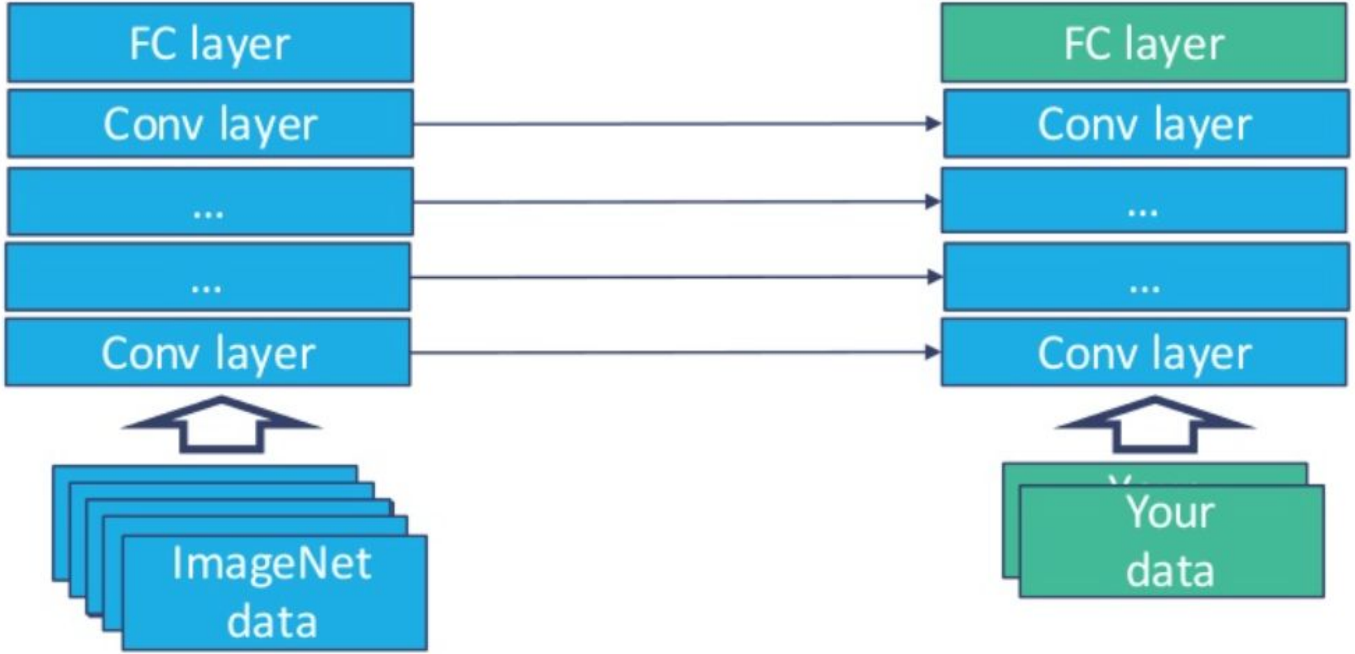
Vector embeddings



128 Measurements Generated from Image

0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

TRANSFER LEARNING

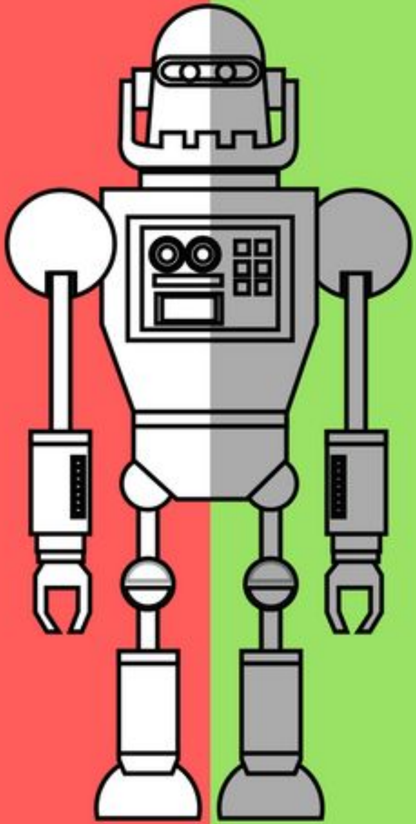


DEEP LEARNING QUESTIONS

SKYNET

NEURAL NET-BASED ARTIFICIAL INTELLIGENCE

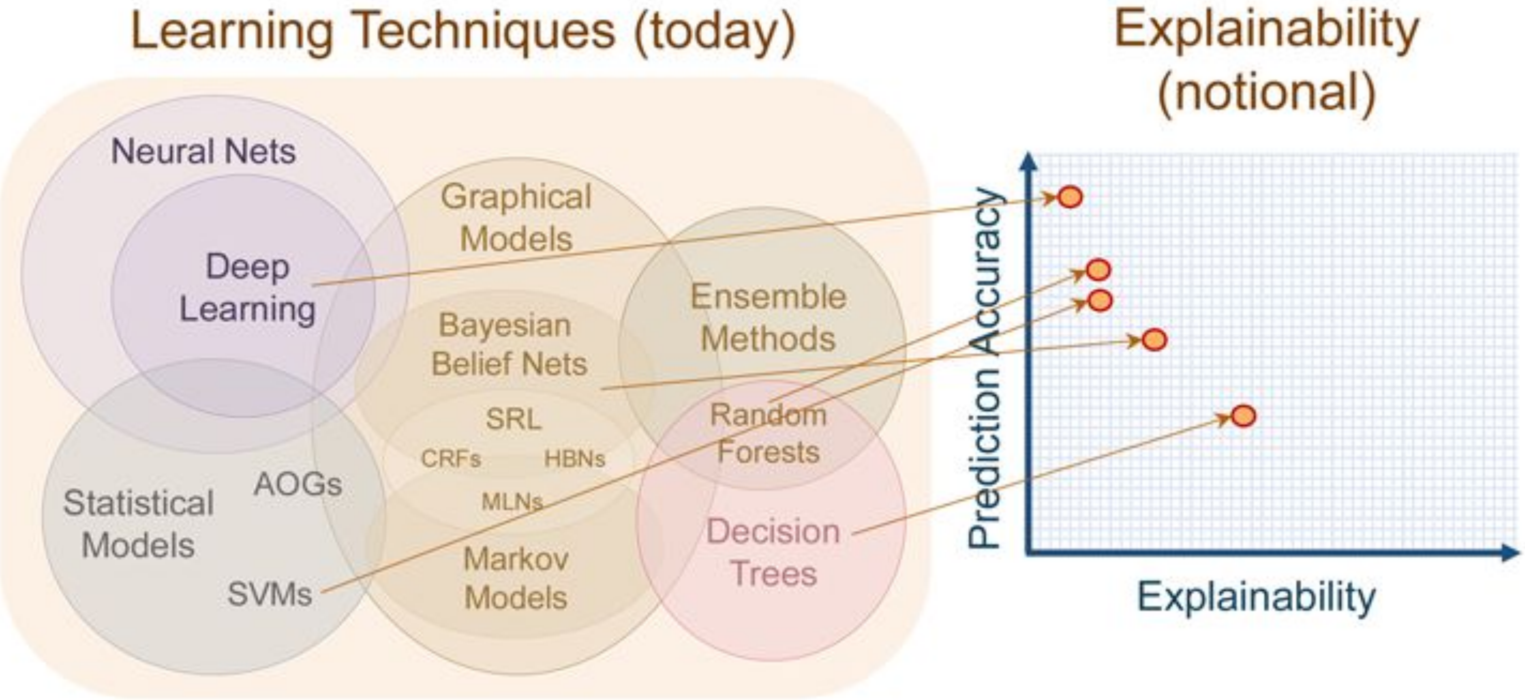
Cognitive technologies such as robots, artificial intelligence (AI), machine learning, and automation will replace 7% of US jobs by 2025.



However, while these jobs will be replaced by automation, 8.9 million new jobs will be created in new fields such as robot monitoring, data science, automation, and content curation will be created.

- Forrester, 2016

BLACK BOX ?



Learning Techniques (today)
Neural Nets
Deep Learning
Graphical Models
Bayesian Belief Nets
SRL
CRFs
HBNs
MLNs
AOGs
Statistical Models
SVMs
Markov Models
Ensemble Methods
Random Forests
Decision Trees
Explainability (notional)
Prediction Accuracy
Explainability

THANKS!

Any questions?

You can find me at leseney@gmail.com

CAS D’USAGES

Smartbuilding

Cas d’usage: Nantes Métropole

Des problématiques

- Un parc de bâtiment énorme (>600)
- Des usages disparates (bureaux, sports, associations, salles festives, …)
- Des ressources humaines limitées
- Des solutions de maitrise énergétique peu évolutives, complexes et souvent onéreuses

1) Détection automatique des plages d'utilisation

- récupération des données de consommation des compteurs Linky

- apprentissage sur la base de données d'occupation récupérées par capteurs

➡ Détection non intrusive (suppression des capteurs après apprentissage) des plages d'occupation du bâtiment

2) Amélioration des plages de chauffe

Caractérisation d'un bâtiment en fonction de son environnement : température extérieure, ensoleillement, orientation, occupation, courbe de chauffe, …

➡ Va permettre de déterminer la température de réduit ainsi que le l'heure de démarrage de la période de chauffe

➡ Réduction de la consommation

Impact sur le produit

- Adaptation de l’architecture logicielle pour la collecte massive et le stockage de données

- Intégration de frameworks de traitement de données (normalisation, …) et de deep learning

Smarthome

Que faire de plus dans l’habitat ?

-> détection des usages répétitifs

Simplification de la solution par limitation de la configuration initiale

-> détection de signatures des équipements électriques dans les courbes de consommation

Détection d’anomalies, prédiction de consommation

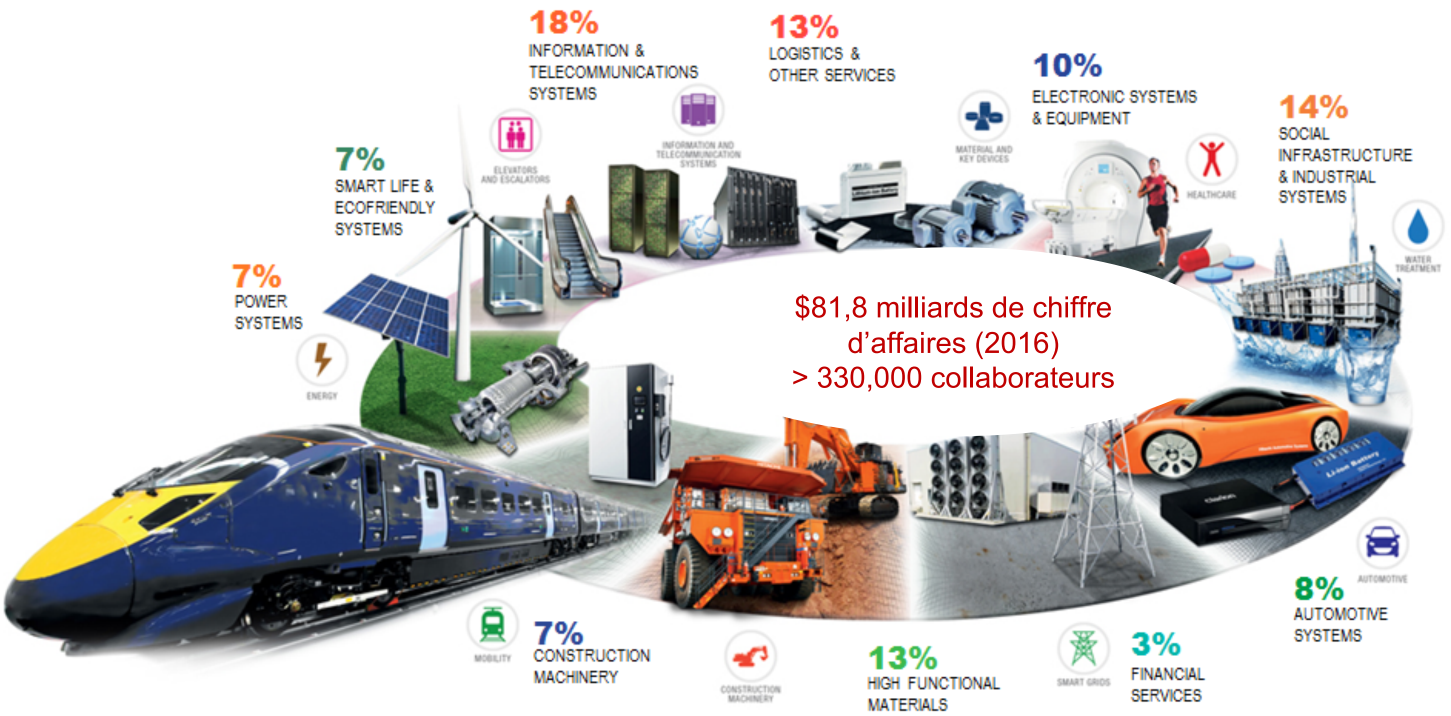
Cas d’usage / Artificial Intelligence

January, 2018

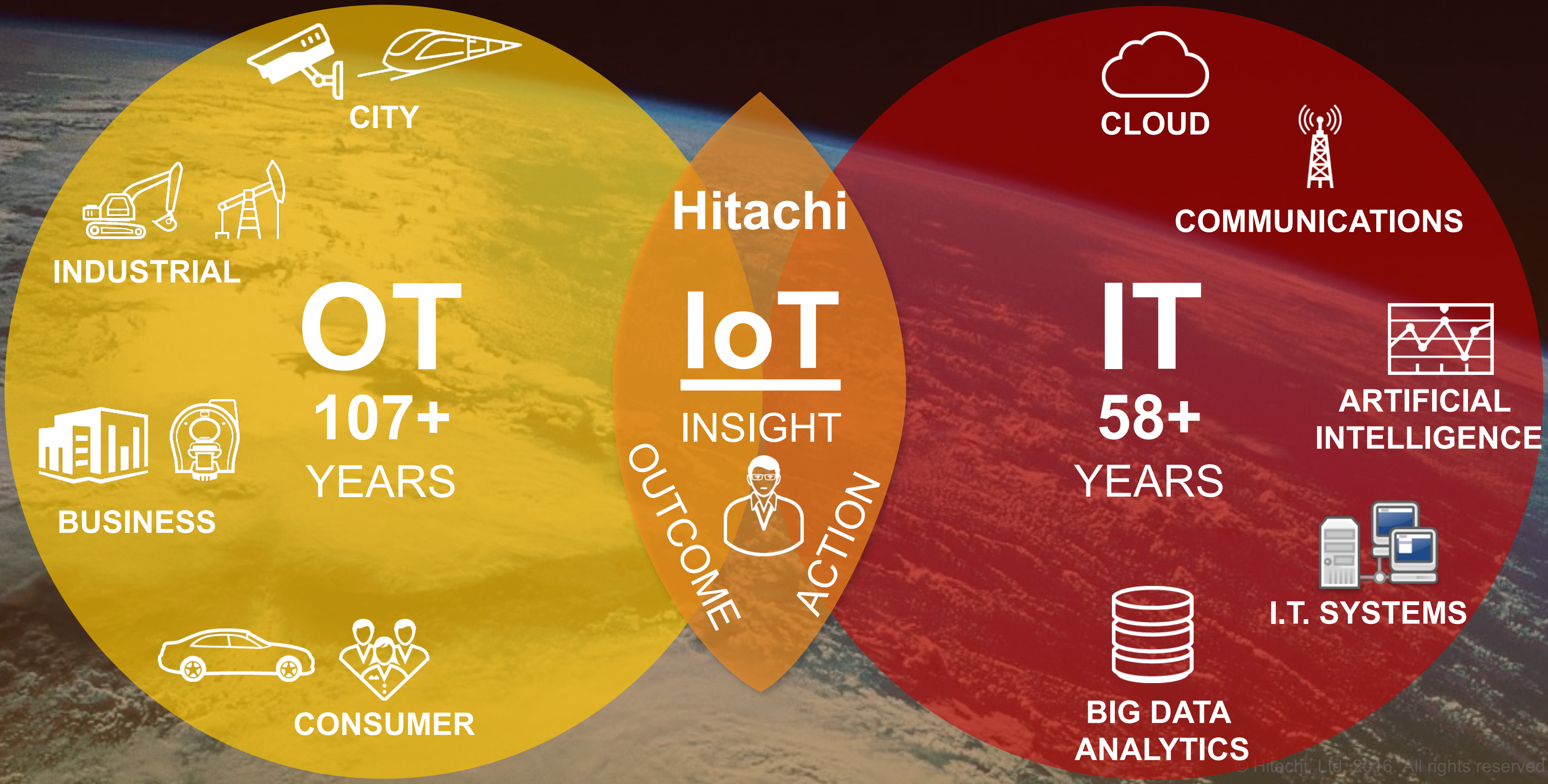
David Le Goff

Hitachi en un coup d'oeil

HITACHI
Inspire the Next



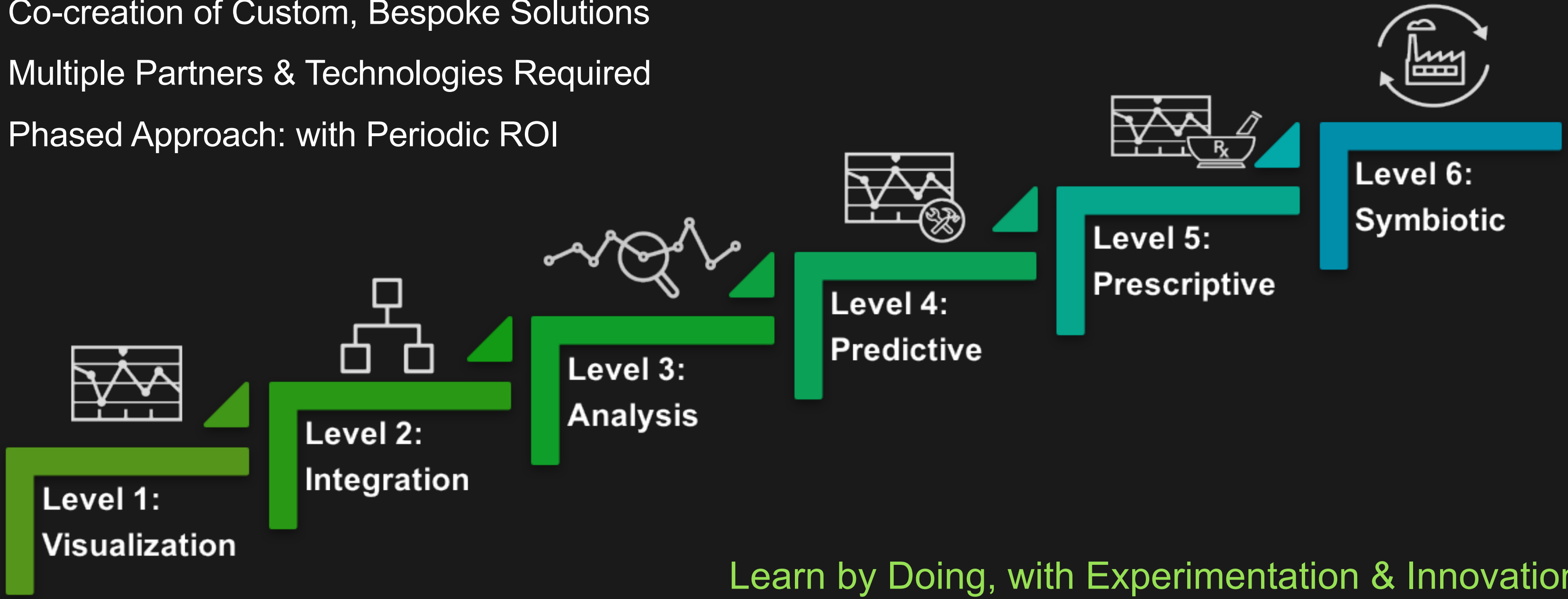
Hitachi Is Uniquely Positioned To Drive Digital Transformation



Digital Manufacturing: Transformation Roadmap

HITACHI
Inspire the Next

- 5 to 10 Year Journey
- Co-creation of Custom, Bespoke Solutions
- Multiple Partners & Technologies Required
- Phased Approach: with Periodic ROI



Learn by Doing, with Experimentation & Innovation.

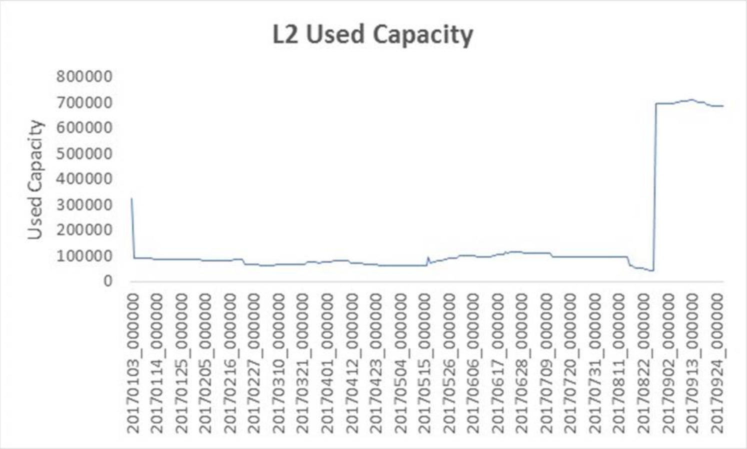
Predictive Capacity

HITACHI
Inspire the Next

Multi-Step Ahead Time-Series model

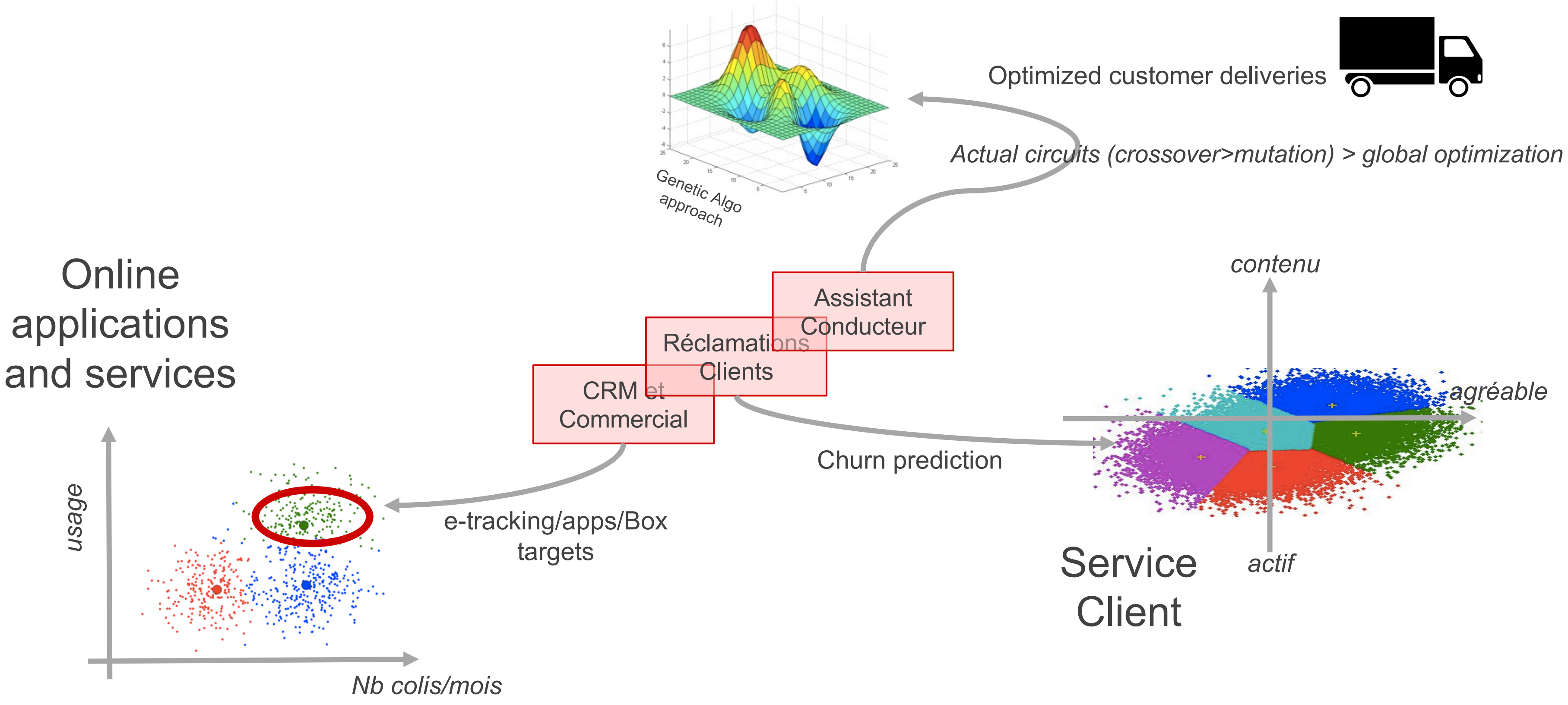
this is a combination of Ordinary Least Squares, but in the form of a General Linear Model (i.e. Y= XB+E), with time-series data.

We used such a model because this model is able to accept data that doesn't show the usual periodicity in the data that usual time-series data shows.



Service, CRM & Optimized Circuit delivery

HITACHI
Inspire the Next

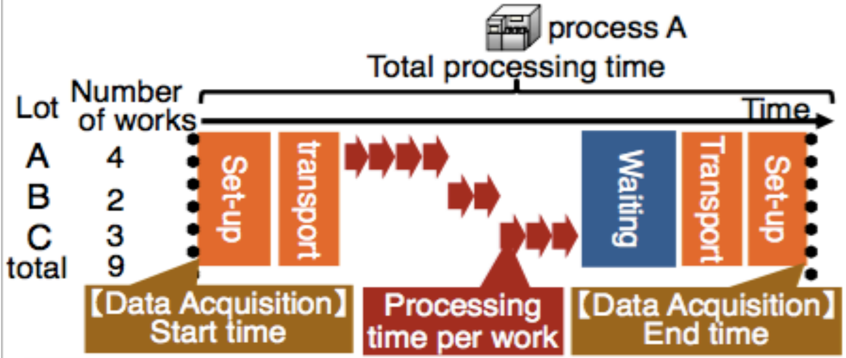


CONFIDENTIAL – For use by Hitachi, Ltd. employees and other audiences under NDA only.

2-5. Scheduling
(b-1)Statistical Model Learning

Lv.5 Visualize/Connect

- Getting accurate processing time per work from shop flor is important for production planning
- Total processing time from shop flor has a lot of noise such as "Waiting", "Transport", "Set-up"
- Statistical model learning gets accurate processing time per work using distribution profile



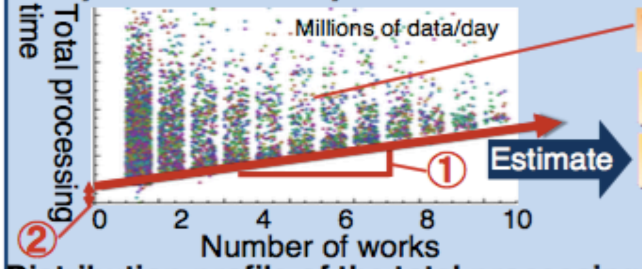
General method：Average based

Processing time per work

$$= \frac{\text{Total processing time}}{\text{Number of works}}$$

Problem：Large margin errors due to set-up, transport, and waiting time

Statistical model learning :Getting the accurate processing time from slope and other part without waiting time from intercept



Variability of data：Impact of waiting time

①Slope：Processing time per work

②Intercept：Set-up and transportation time

Y = ax +b

Distribution profile of the total processing time

Thank You