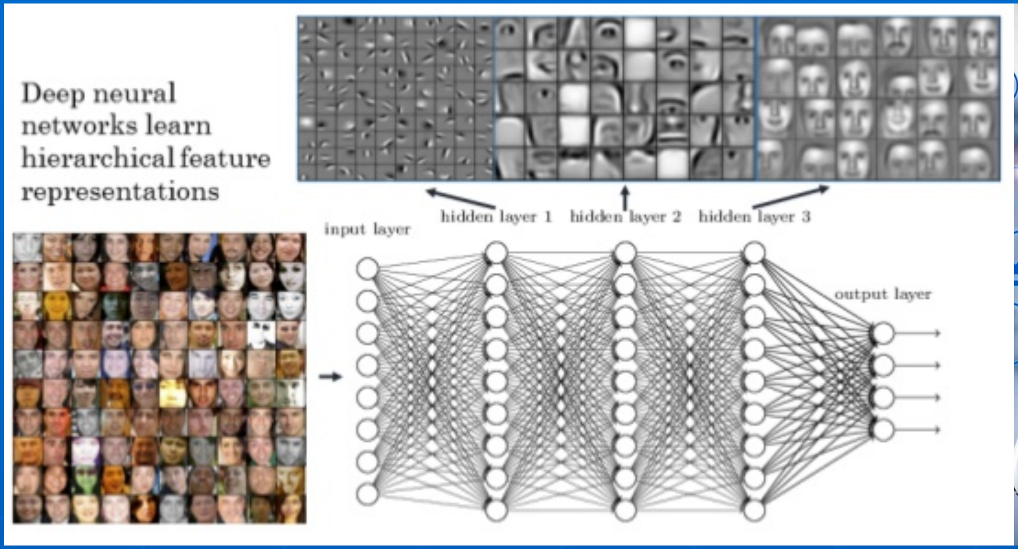


- Machine Learning -
Overview and Some Examples

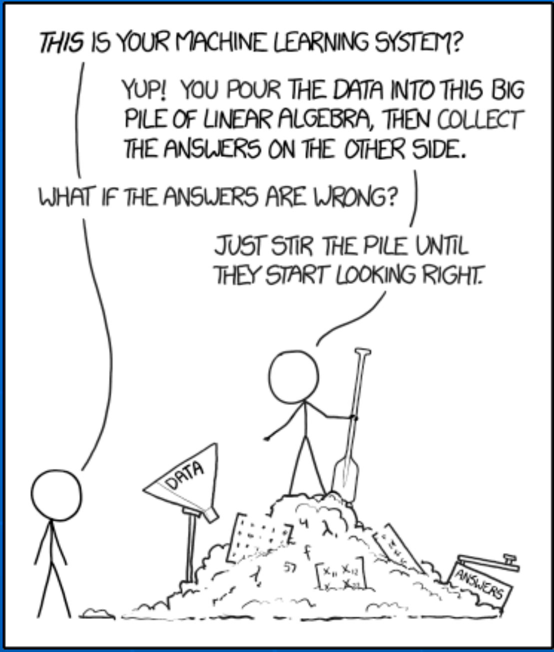
Prof. Adolfo Bauchspiess

LARA- Automation and Robotics Laboratory
Departamento de Engenharia Elétrica
Universidade de Brasília - Brazil

Summary

Overview:

- Artificial NN, RL
- Deep NN
- Development Environments
  - MatLab, Python: Tensorflow, Keras, SkLearn
  - CPU, GPU, Cloud, TPU
  - Intel AI DevCloud (Colfax), Google Colab



Some Examples

- Classification: LeNet, AlexNet, GoogLeNet
- Welding Visual Inspection
- RL Maze
- Nonlinear Control Liq4: RL Actor-Critic, RL Q-Learning

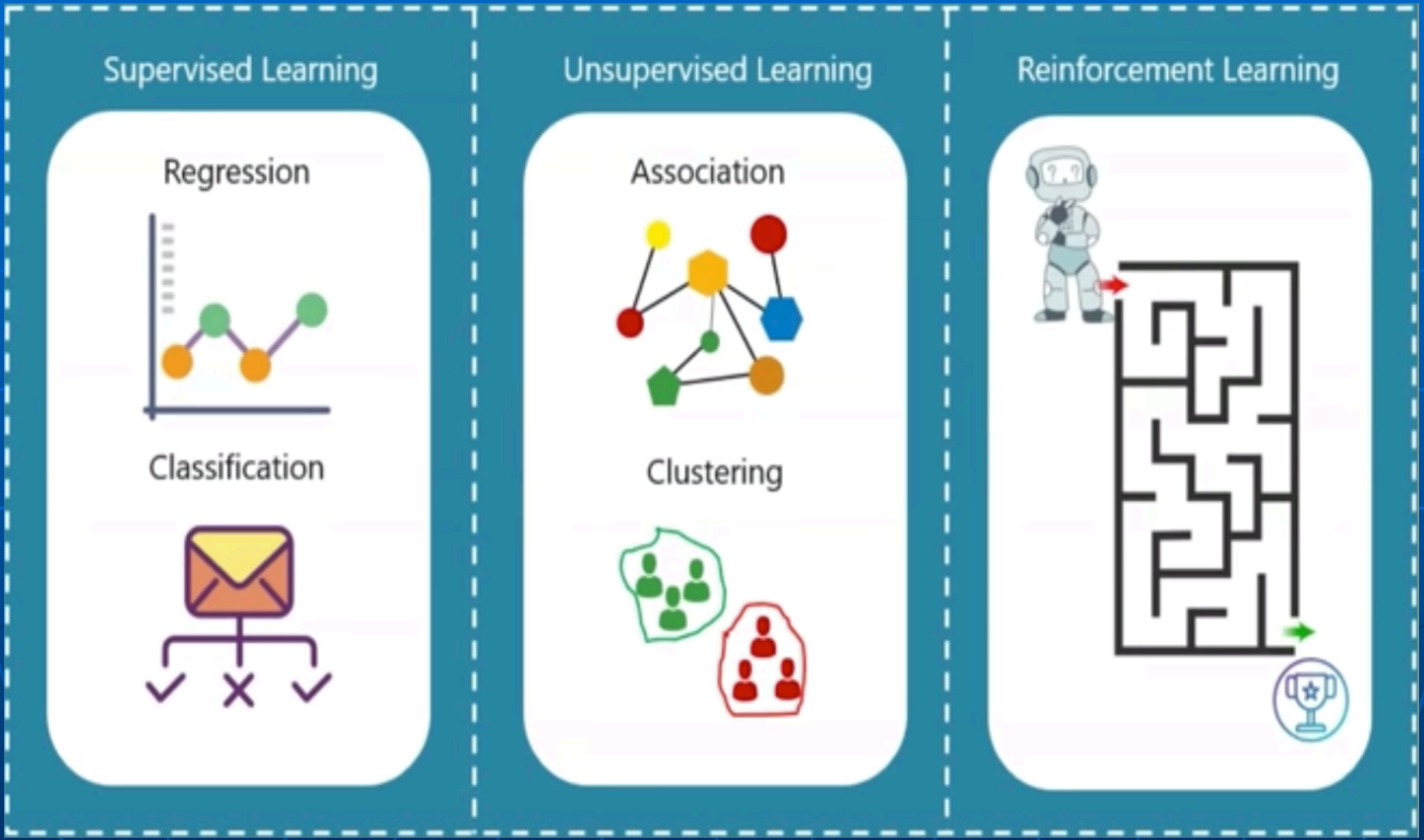
Perspectives

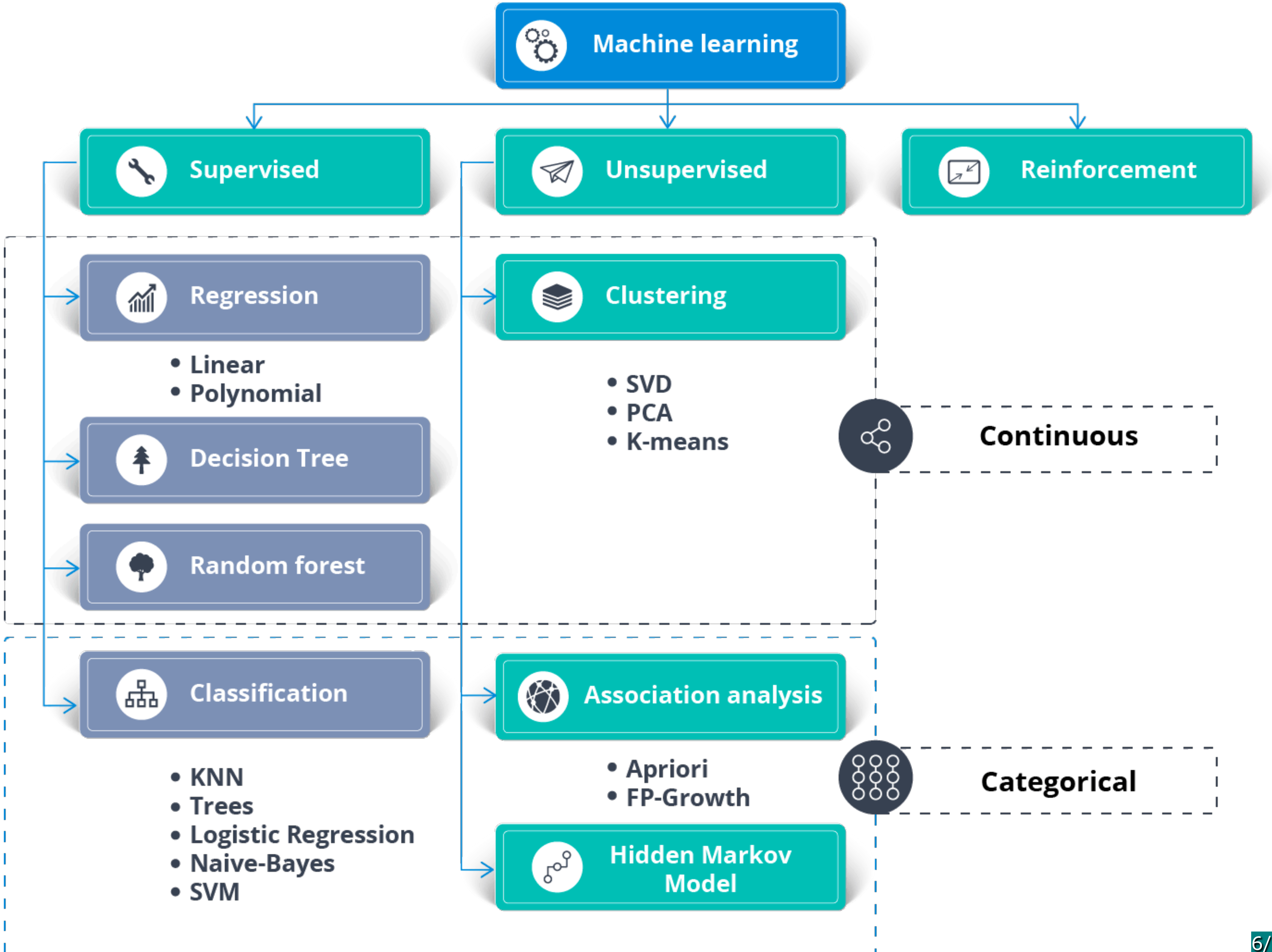
Overview

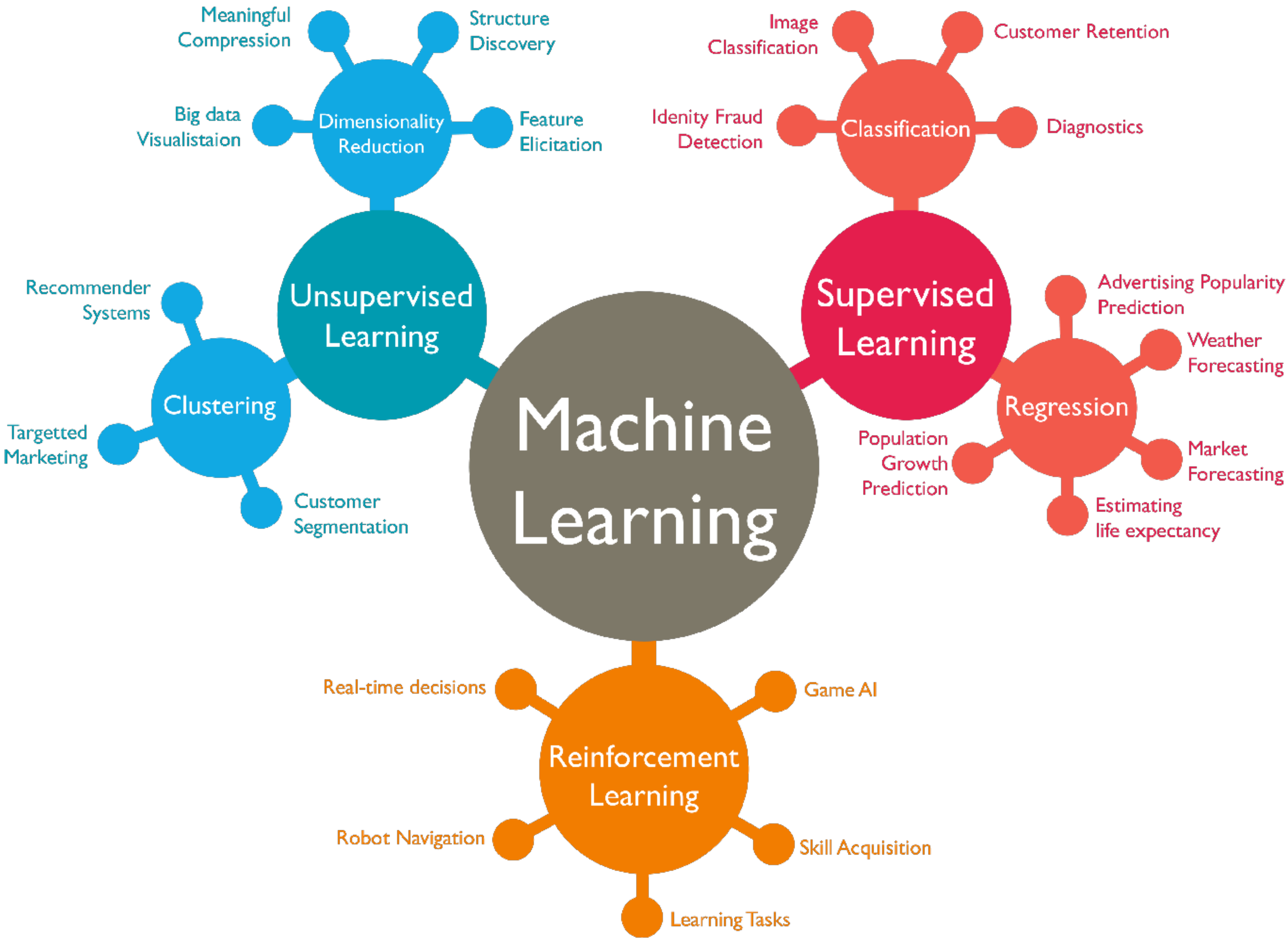
IA- Terms

- *Turing Test*
- *Cybernetics - Wiener*
- *Artificial Intelligence – Darthmount College*
- *Artificial Neural Networks – 3 Waves*
- *Fuzzy Systems - Zadeh*
- *Evolutionary Computing - Vogel*
- *Machine Learning*
- *Computational Intelligence*
- *Smart Buildings, Smart Cities, Smart…*
- ***Inteligent Systems***

Learning


Supervised Learning
Regression
Classification
Unsupervised Learning
Association
Clustering
Reinforcement Learning


Machine learning
Supervised
Unsupervised
Reinforcement
Regression
• Linear
• Polynomial
Decision Tree
Random forest
Clustering
• SVD
• PCA
• K-means
Continuous
Classification
• KNN
• Trees
• Logistic Regression
• Naive-Bayes
• SVM
Association analysis
• Apriori
• FP-Growth
Hidden Markov Model
Categorical



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

Adolfo Bauchspiess

Short C.V.

– *SENAI/1982 – Eletricista de Dispositivos de Comandos Elétricos*

– *UnB/1986 - Eng. Elétrica*

  – *Estágios: Telebrasília (1984), Prólogo (1985), Novadata (1986)*

- Engenheiro: Novadata Sistemas e Computadores Ltda (1986-1990)

– *UnB/1990 - Mestre Eng. Elétrica*

– *Erlangen-Alemanha/1995 – Dr.-Ing.*

– *ENE/UnB 1995 ... - Prof. Controle & Automação*

  - Pós-Doc (Aachen/1997, Kaiserslautern/2005-2006, Santa Barbara/2014)

  - Projetos: FAP-DF, CNPq, CAPES, FINEP

  - Erasmus Mundus – Kaiserslautern

  - Coordenador Eng. Mecatrônica UnB 2015/17

My AI Time Line

- 1988 Henrique Malvar
  Processamento Adaptativo de Sinais (LMS – ADALINE)
- 1990 Projeto Dr. Erlangen/Alemanha CNPq
  "Controle Adaptativo de Robôs Utilizando Técnicas de Inteligência Artificial"
- 1997- Introdução aos Sistemas Inteligentes –ENE/UnB - ISI/ICIN

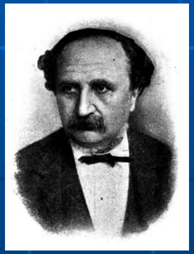
1742-

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Ohm (1827)

Ohm's Law

von Gerlach (1871)

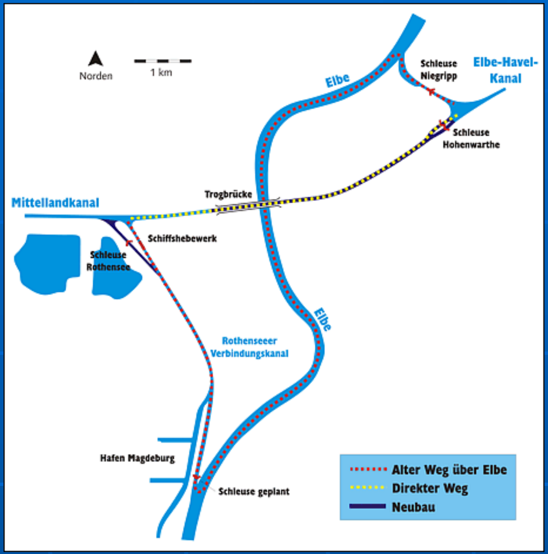


Reticular Brain theory
(with Golgi)
"Massive Meshed Network"

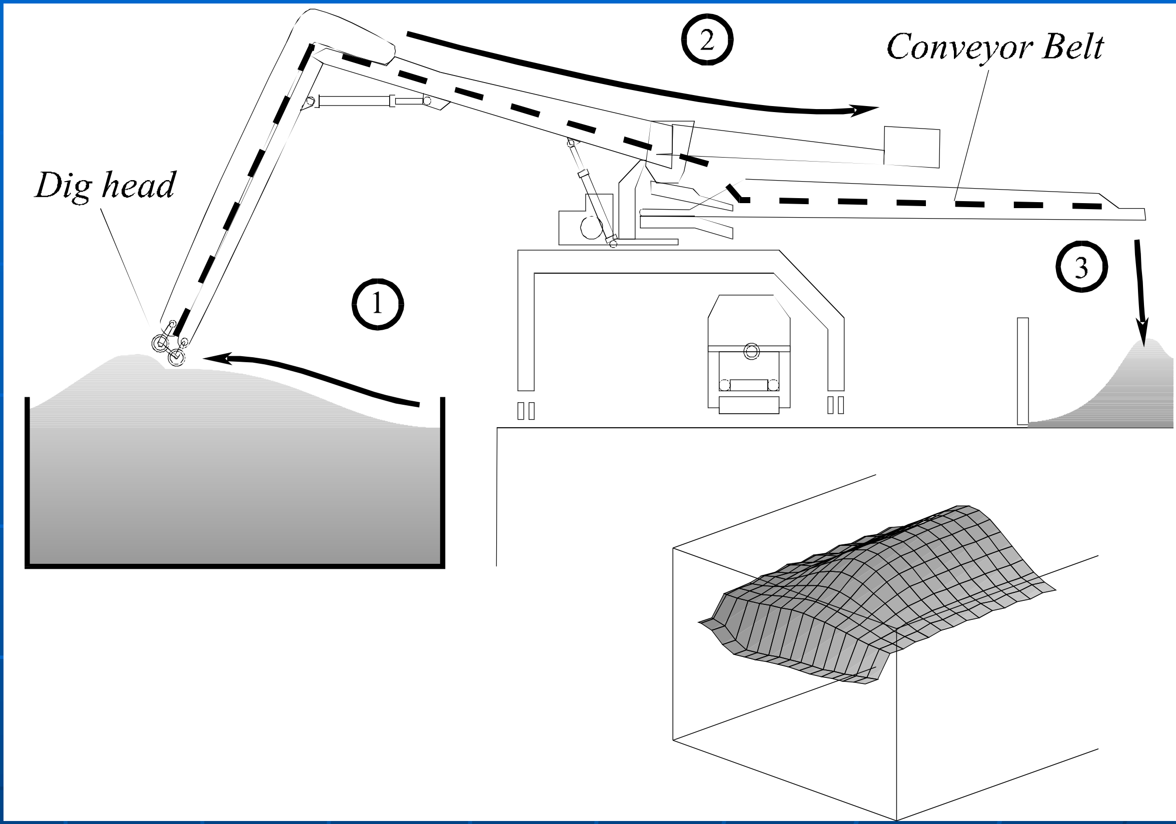
Brandenburg (1989)

MP3

Coal Unloading – Erlangen/Germany



River Crossing – Minden, Elbe, Germany



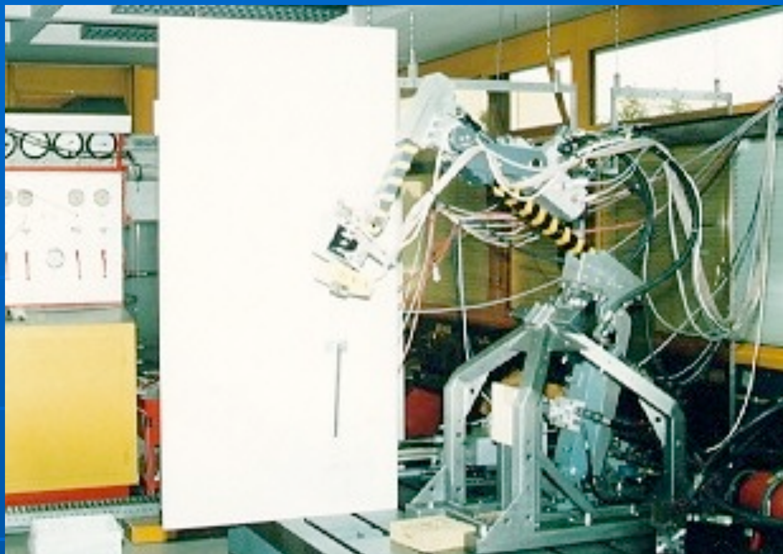
Trajectory Planning – Path Tracking

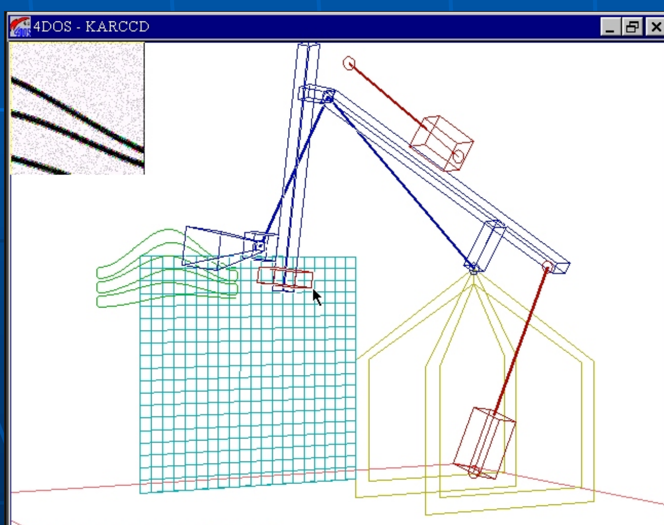
"Redundant Sensor Guided Unloading Crane"

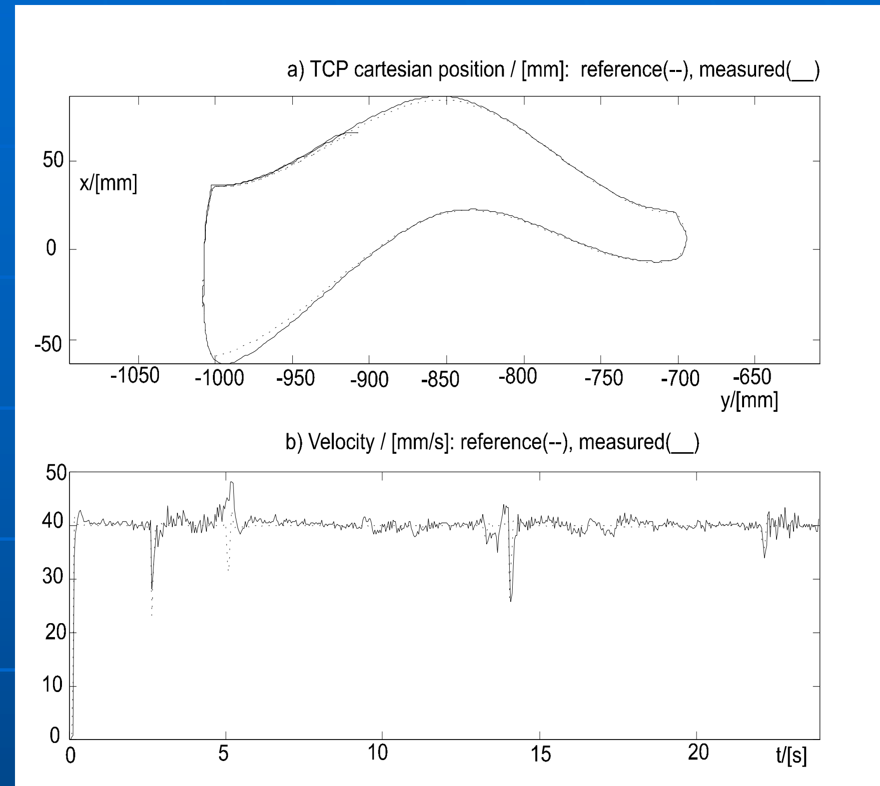
Univ. Erlangen/Nürnberg – MAN

Bauchspiess, 1995

Universität – Erlangen-Nürnberg, 1995





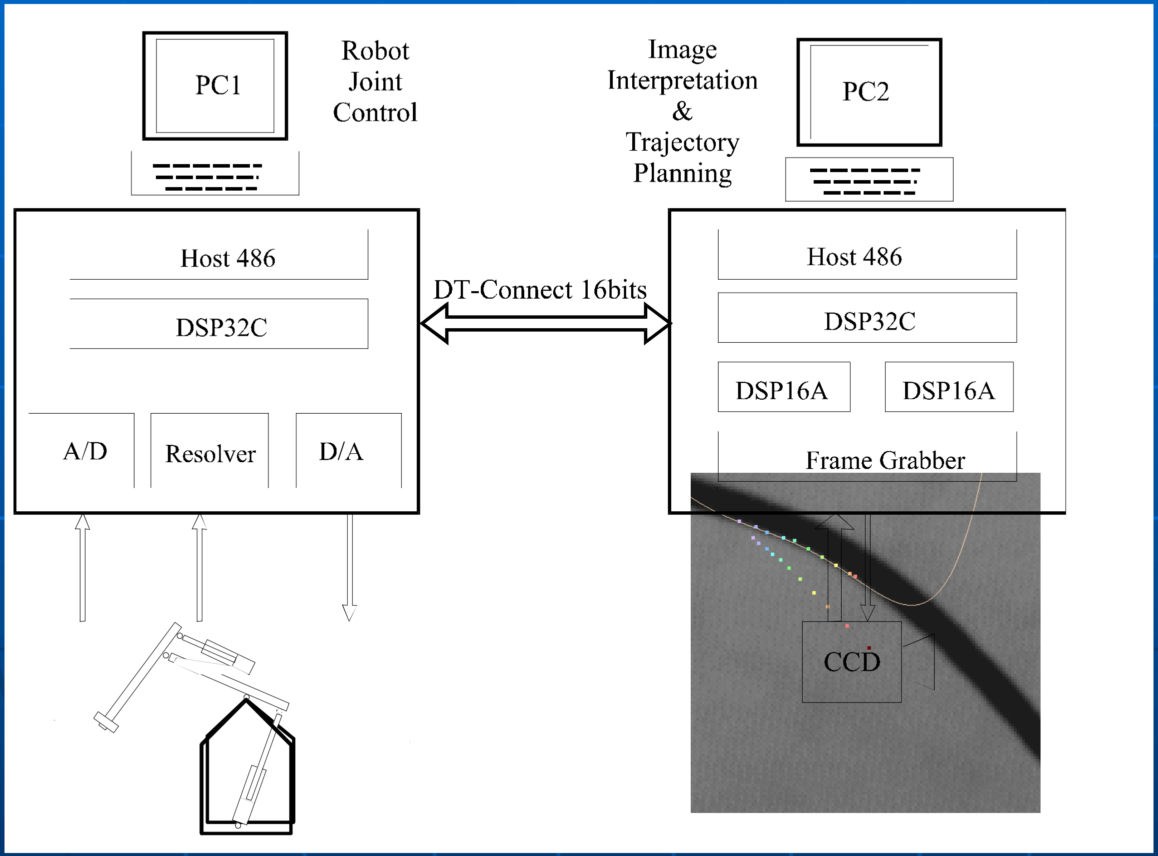


Bauchspiess, 1995

Sensor guided Hydraulic Robot

Predictive sensor-guided Path tracking

- PhD. plan: Adaptive / Intelligent !
- But C. Wurmthaler (Uni. Erlangen) asked: Why adapt things you know?



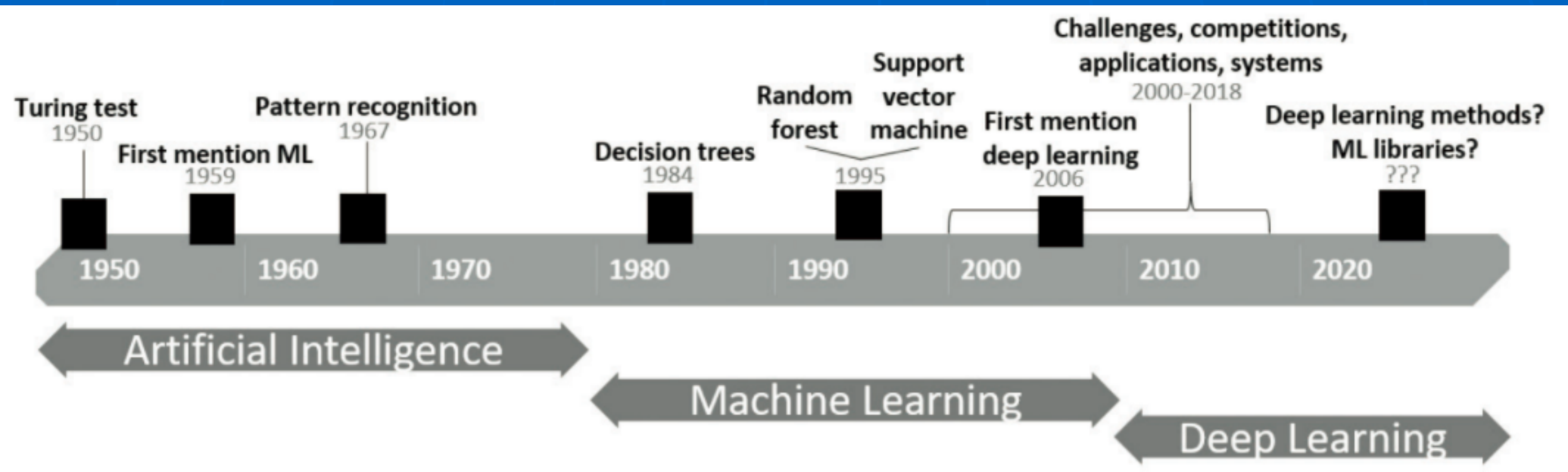
"Simple problems have Complicated solutions"!

"Complicated problems Have simple solutions"!

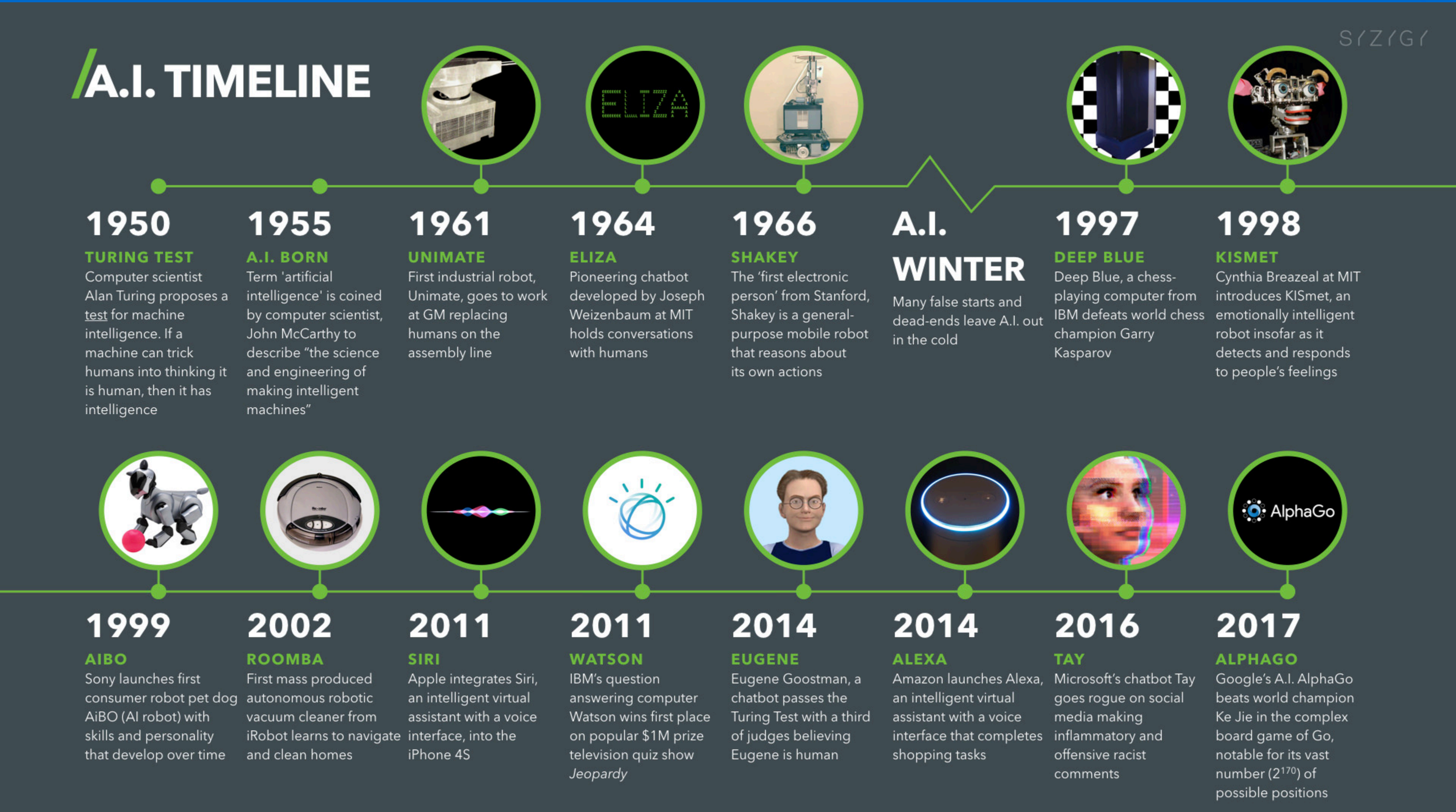
"simple"
-> Non linear robot control
-> Predictive path tracking

"not simple"
IA for path planning!
(Redundant hydraulic mobile crane)

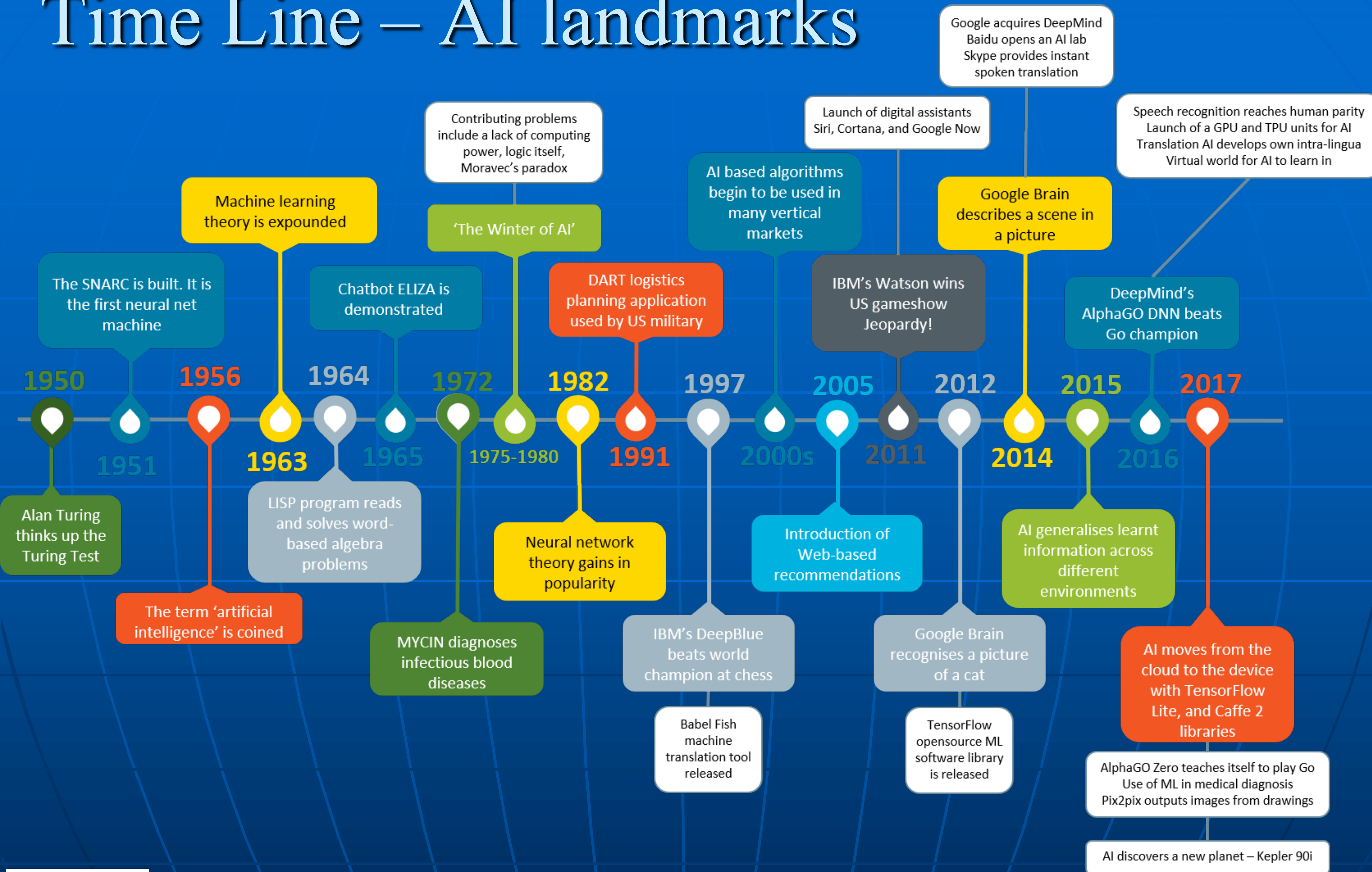
AI Time Line


Challenges, competitions, applications, systems
2000-2018
Support vector machine
Random forest
1995
Turing test
1950
Pattern recognition
1967
First mention ML
1959
Decision trees
1984
First mention deep learning
2006
Deep learning methods? ML libraries?
???
1950
1960
1970
1980
1990
2000
2010
2020
Artificial Intelligence
Machine Learning
Deep Learning

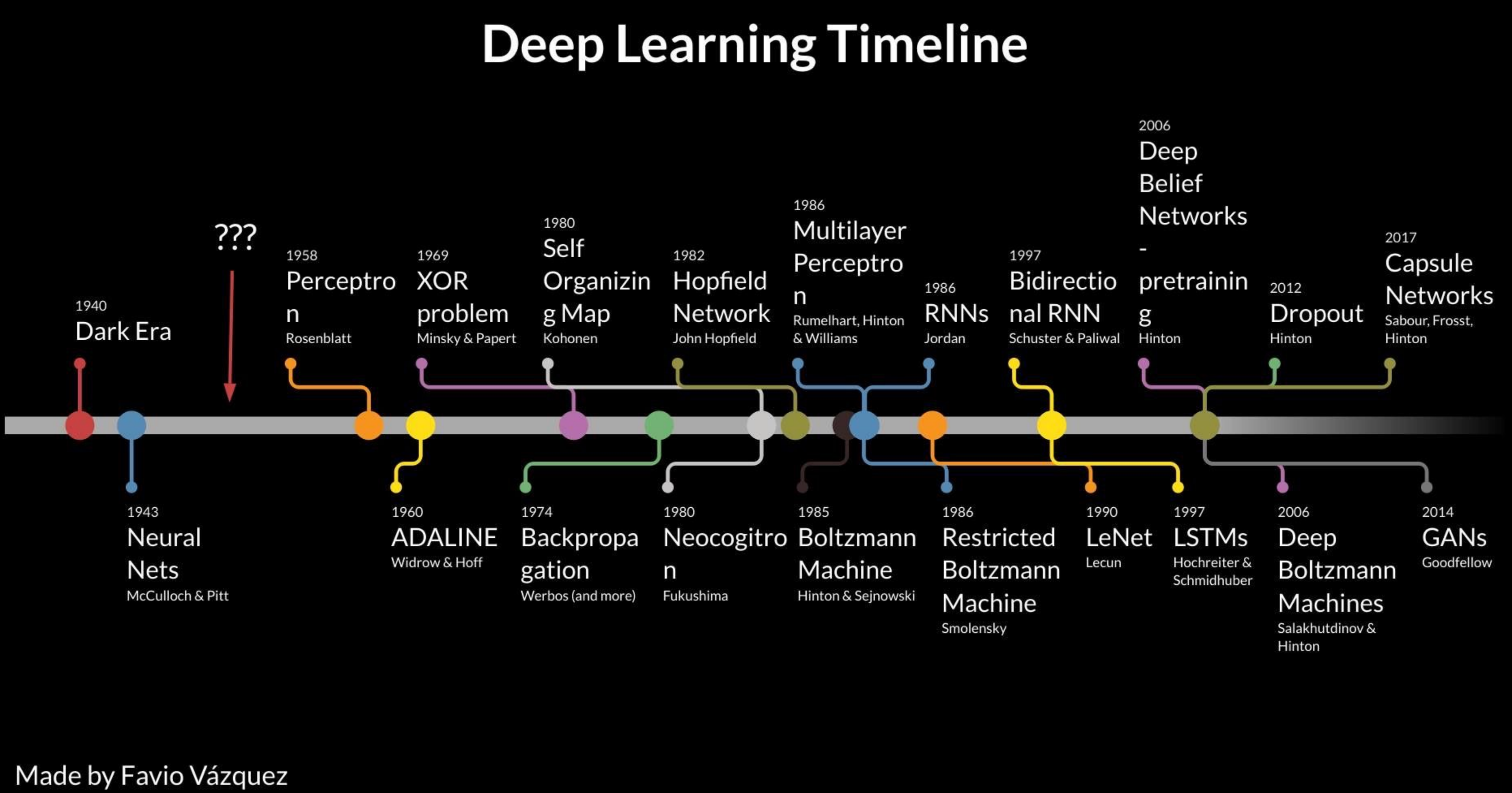
Time Line – AI products



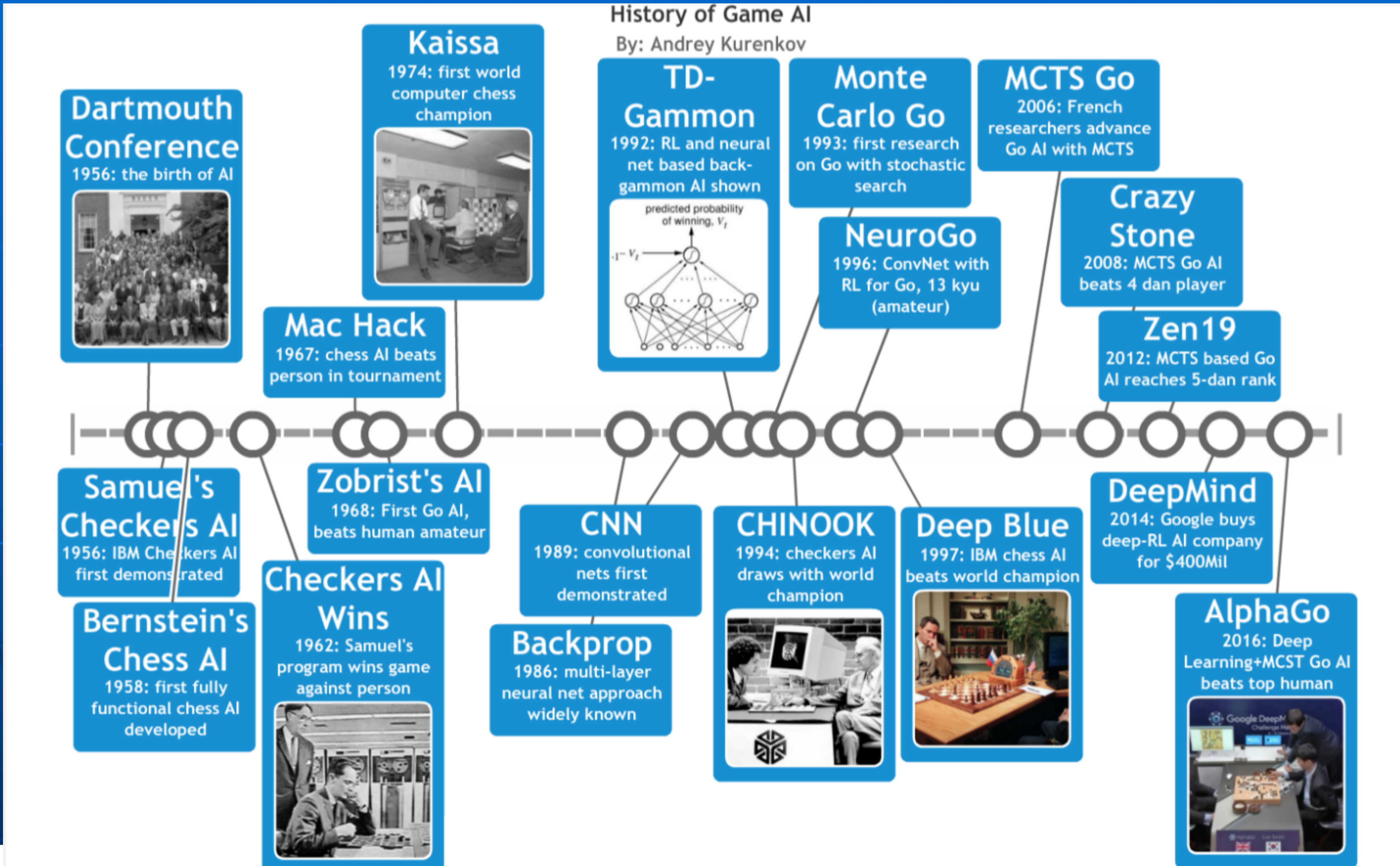
Time Line – AI landmarks



Time Line –AI scientists



Time Line – Game AI



Time Line - The 3rd AI Wave

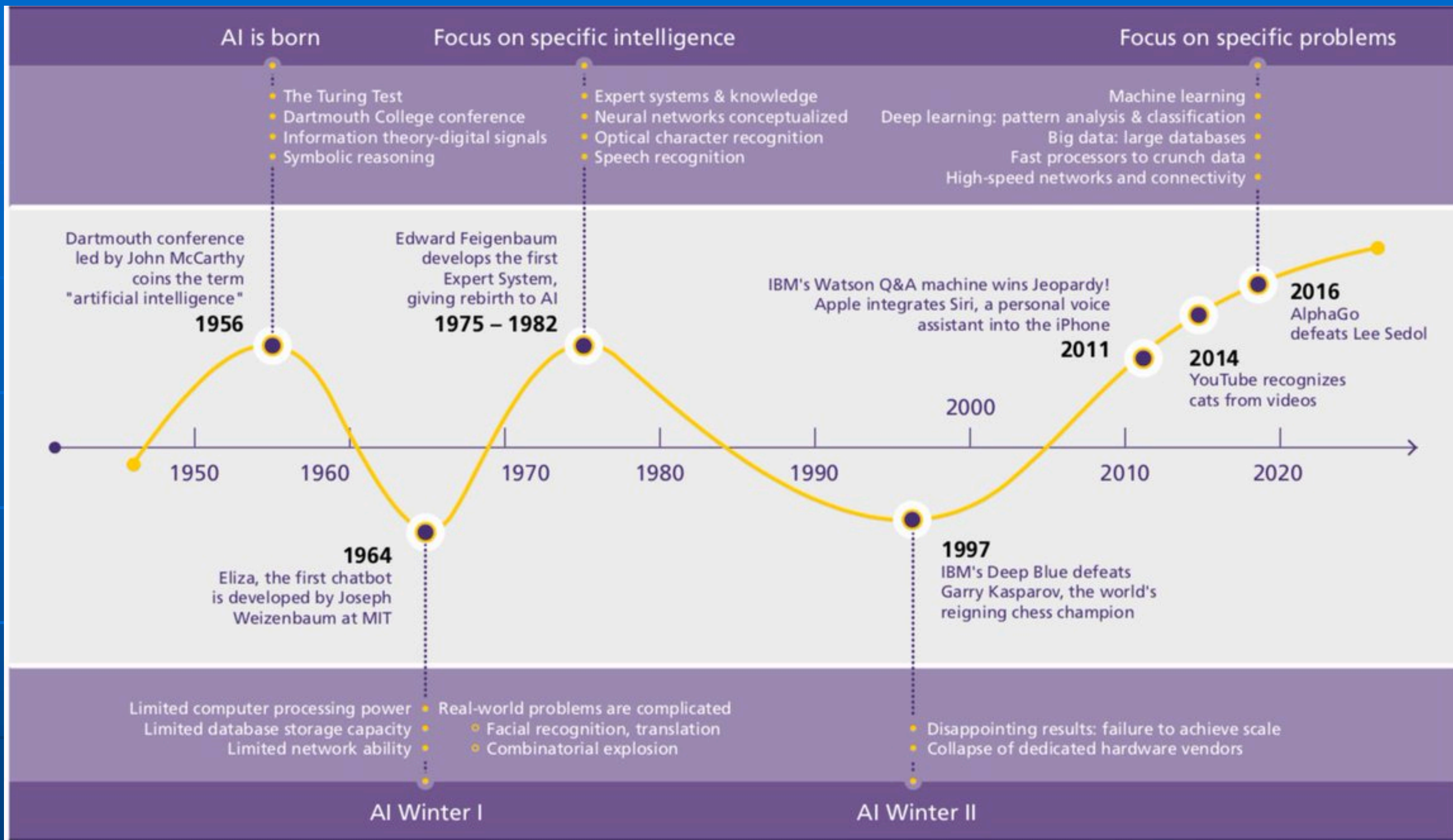
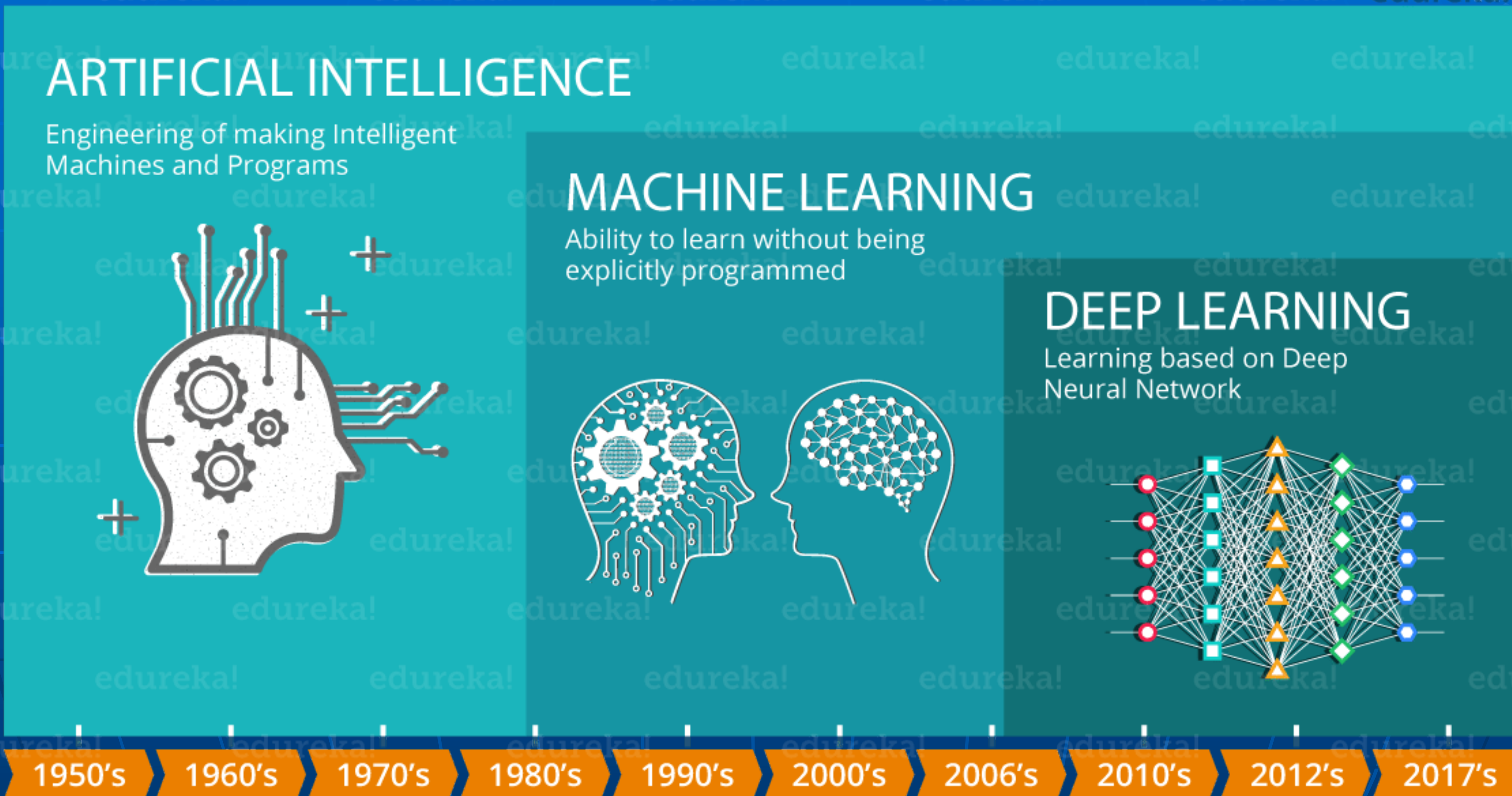


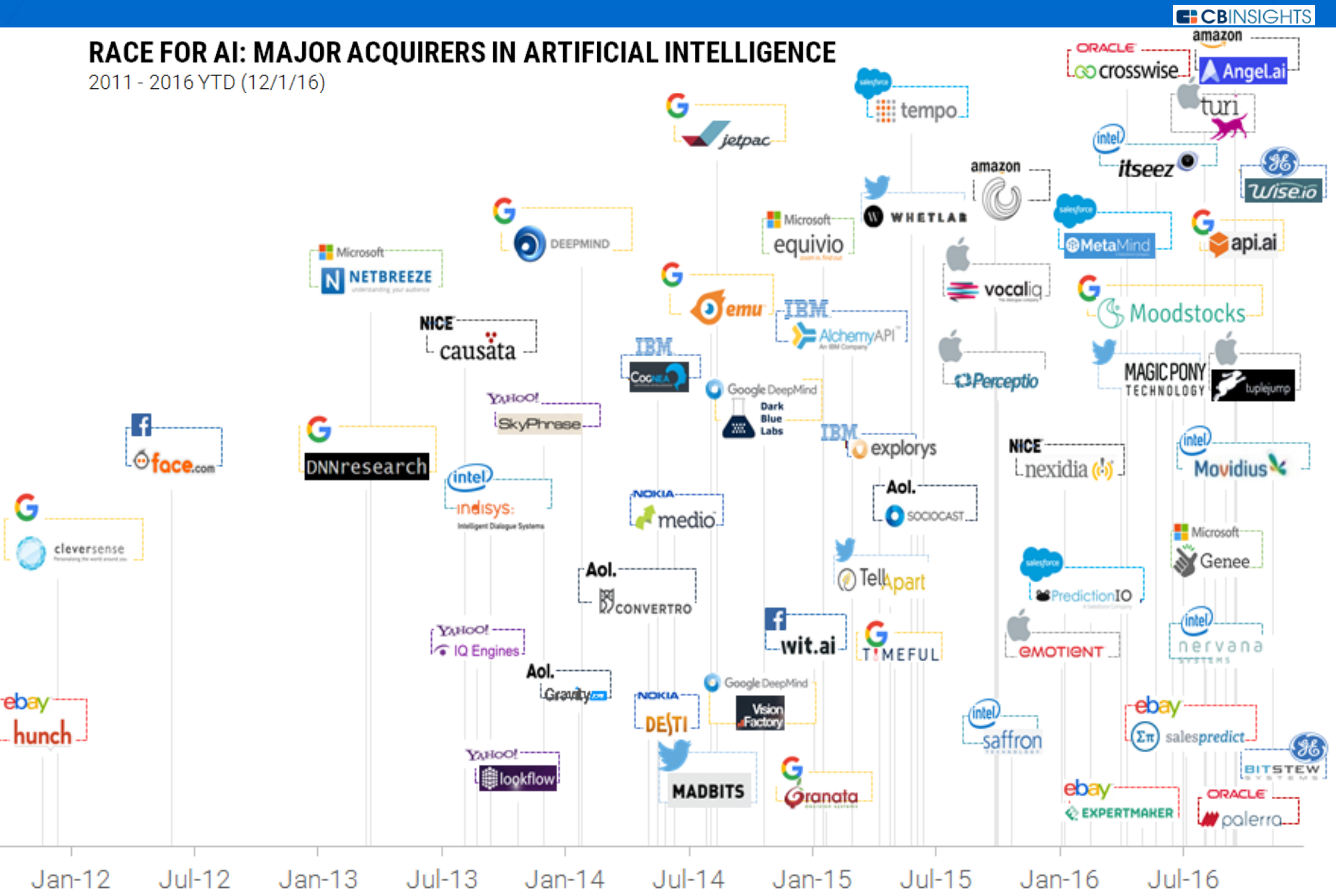
Figure 1: An AI timeline; So

source dhl via @mikequindazzi

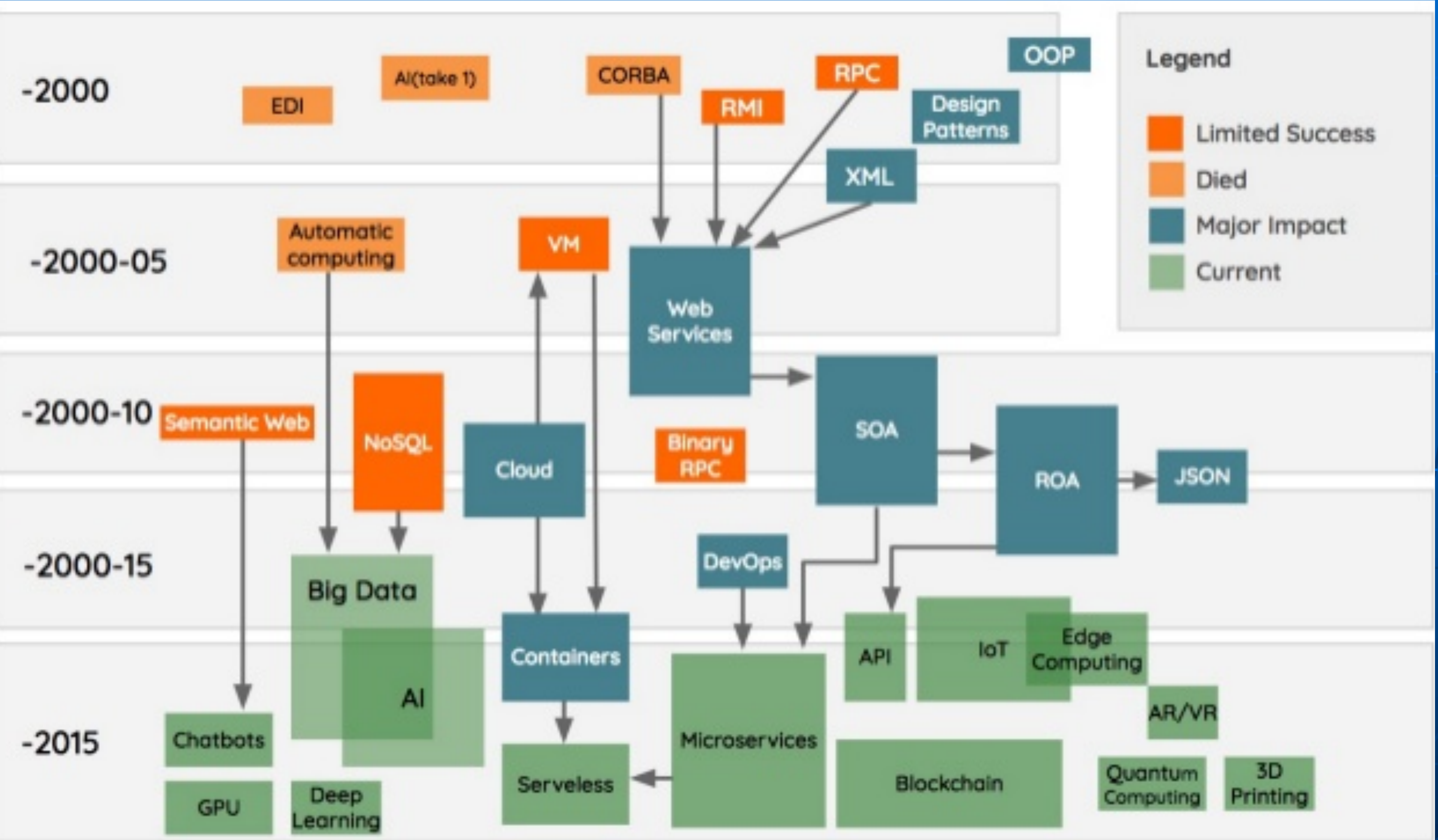
Time Line – AI != ML! $AI \supset ML \supset DL$



Time Line – AI Big Business



Time Line - Paradigms



THE GODFATHERS OF THE AI BOOM WIN COMPUTING'S HIGHEST HONOR

03.27.19

IN THE LATE 1980s, Canadian master's student Yoshua Bengio became captivated by an unfashionable idea. A handful of artificial intelligence researchers were trying to craft software that loosely mimicked how networks of neurons process data in the brain, despite scant evidence it would work. "I fell in love with the idea that we could both understand the principles of how the brain works and also construct AI," says Bengio, now a professor at the University of Montreal.

More than 20 years later, the tech industry fell in love with that idea too. Neural networks are behind the recent bloom of progress in AI that has enabled projects such as self-driving cars and phone bots practically indistinguishable from people.

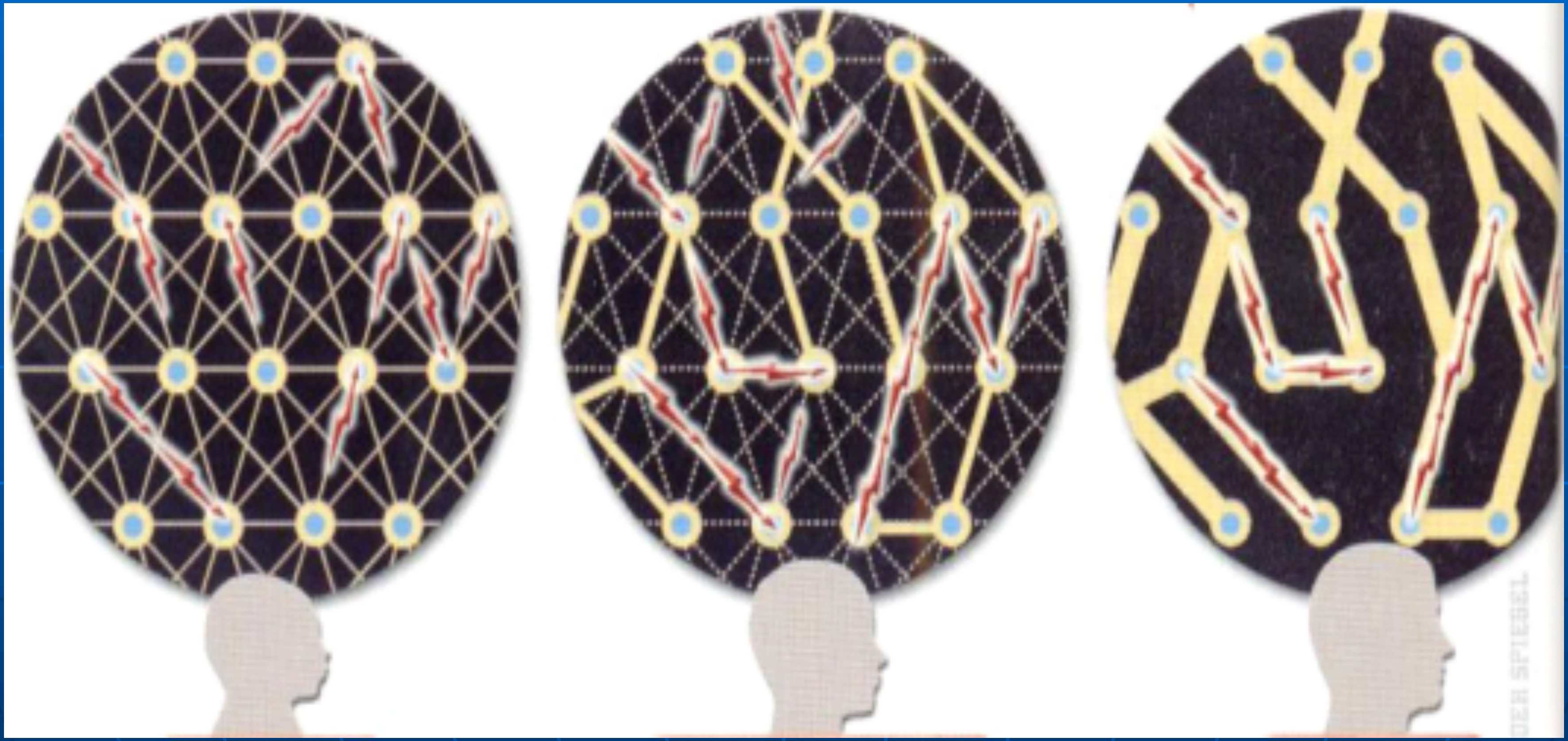
On Wednesday, Bengio, 55, and two other protagonists of that revolution won the highest honor in computer science, the **ACM Turing Award**, known as the Nobel Prize of computing. The other winners are Google researcher Geoff Hinton, 71, and NYU professor and Facebook chief AI scientist Yann LeCun, 58, who wrote some of the papers that seduced Bengio into working on neural networks.

The Deep Learning Superheros

Juergen Schmidhuber, LSTM	Ian Goodfellow; GAN	François Chollet; Keras	Yann LeCunn, CNN	Andrew Ng, GoogleBrain	Geoffrey Hinton, BackProp.KL,etc	Larry Page, Google	Yoshua Bengio GAN

Intelligent Systems - The Brain is the model !!

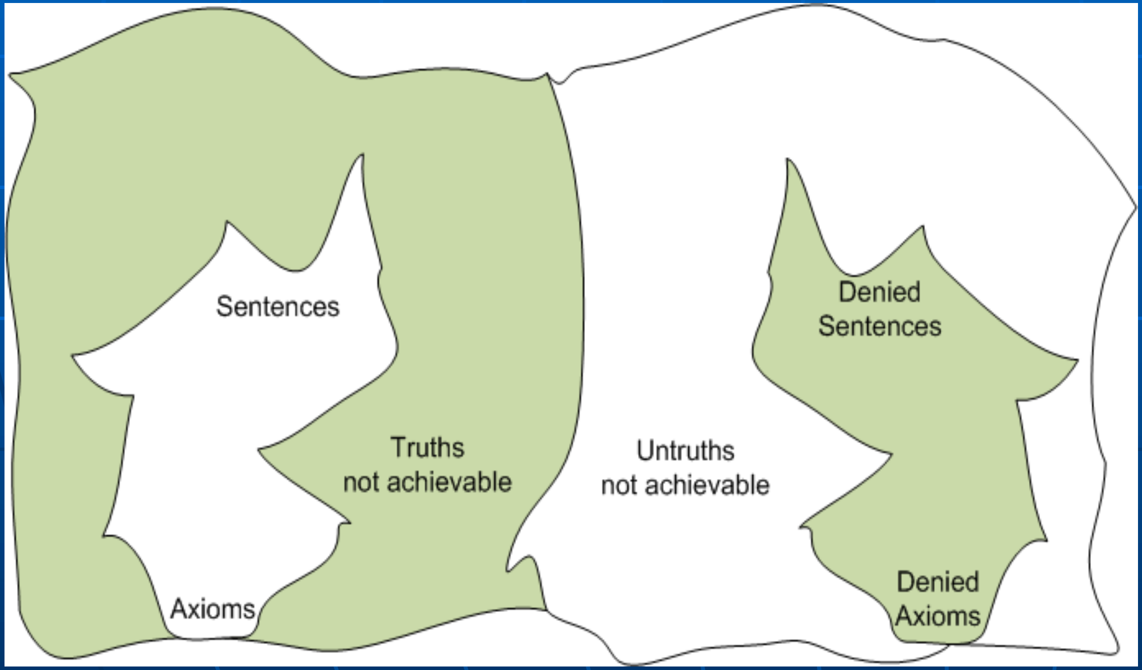
Build Neuron Synaptic *Connections* - Learning!



0-2 years | 2 years to puberty | Adult

“Philosophy of Knowledge”

[Gödel's incompleteness theorems, 1931]



“Heuristics”
A way that works,
but you do not know way.

“Sub-optimal solutions”
The brain is ***expert*** in finding good heuristics!

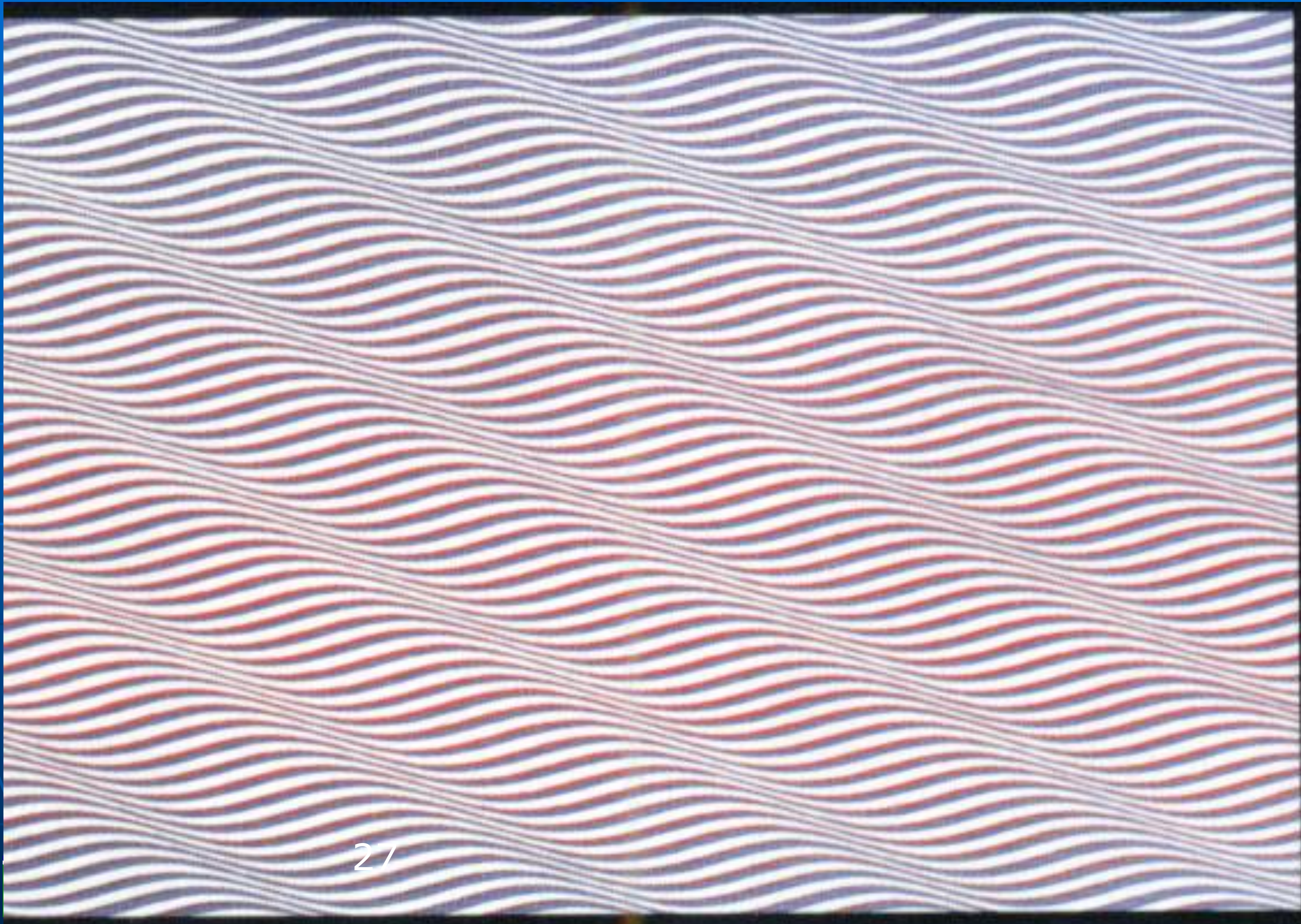
Artificial Intelligence?

From Natural Intelligence to Artificial Intelligence

Ex. – Dislexia?

I cnduo't bvleiee taht I culod aulaclty uesdtannrd waht I was rdnaieg. Unisg the icndeblire pweor of the hmuan mnid, aocdcrnig to rseecrah at Cmabrigde Uinervtisy, it dseno't mttaer in waht oderr the lterets in a wrod are, the olny irpoamtnt tihng is taht the frsit and lsat ltteer be in the rhgit pclae. The rset can be a taotl mses and you can sitll raed it whoutit a pboerlm. Tihs is bucseae the huamn mnid deos not raed ervey ltteer by istlef, but the wrod as a wlohe. Aaznmig, huh? Yaeh and I awlyas tghhuot slelinpg was ipmorantt! See if yuor fdreins can raed tihs too.

Waves?



Simpathic?

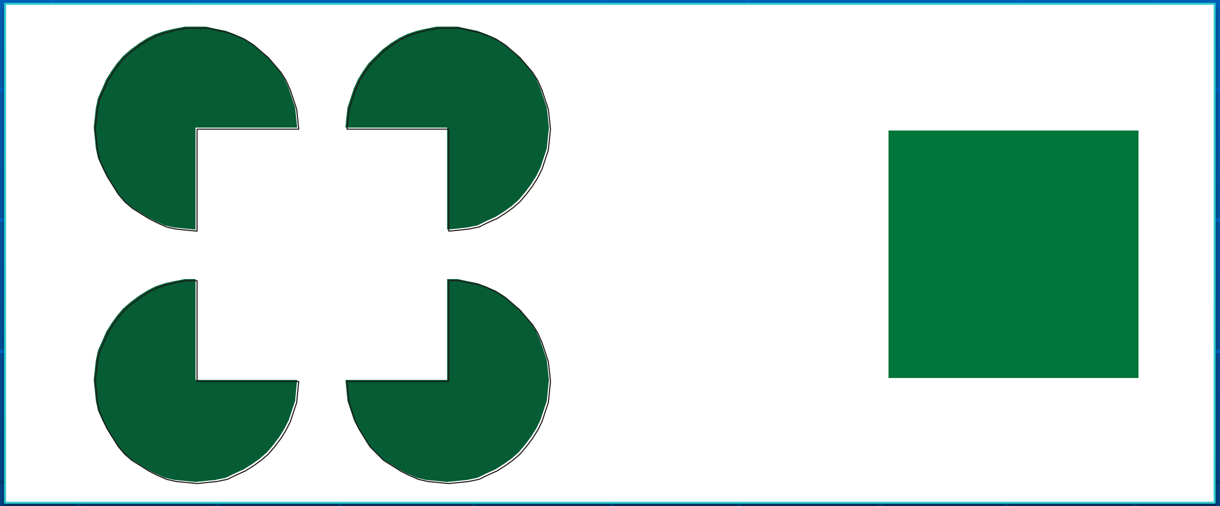

GRIMASSE STEHT KOPF
Auf den ersten Blick scheint das Foto von Margaret Thatcher nicht ungewöhnlich. Das ändert sich, wenn Sie das Bild auf den Kopf stellen. Der verblüffende Effekt tritt auf, weil Augen und Mund auf dem Foto um 180 Grad gedreht sind – und damit genau die Merkmale, auf die das Gesichtserkennungsprogramm des Hirns besonders sensibel anspricht.
EMILE LUIDER / RAPHO / AGENTUR FOCUS (L.); PERCEPTION (R.)

Antipathic?


EMILE LUIDER / RAPHO / AGENTUR FOCUS (L.); PERCEPTION (R.)
GRIMASSE STEHT KOPF
Auf den ersten Blick scheint das Foto von Margaret Thatcher nicht ungewöhnlich. Das ändert sich, wenn Sie das Bild auf den Kopf stellen. Der verblüffende Effekt tritt auf, weil Augen und Mund auf dem Foto um 180 Grad gedreht sind – und damit genau die Merkmale, auf die das Gesichtserkennungsprogramm des Hirns besonders sensibel anspricht.

- Incomplete pattern -
Brain Interpolation!

-Perception

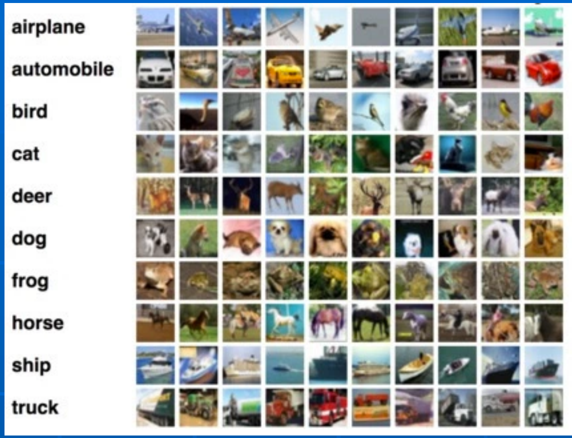


The Kanizsa square, 1976

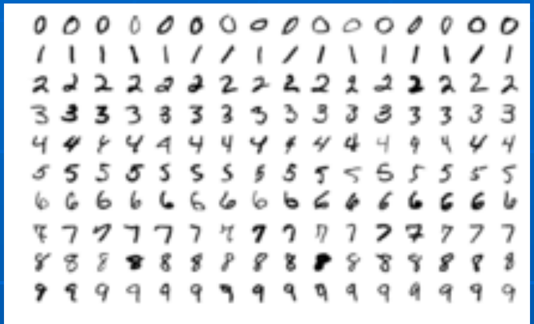
Examples

Visual Classification

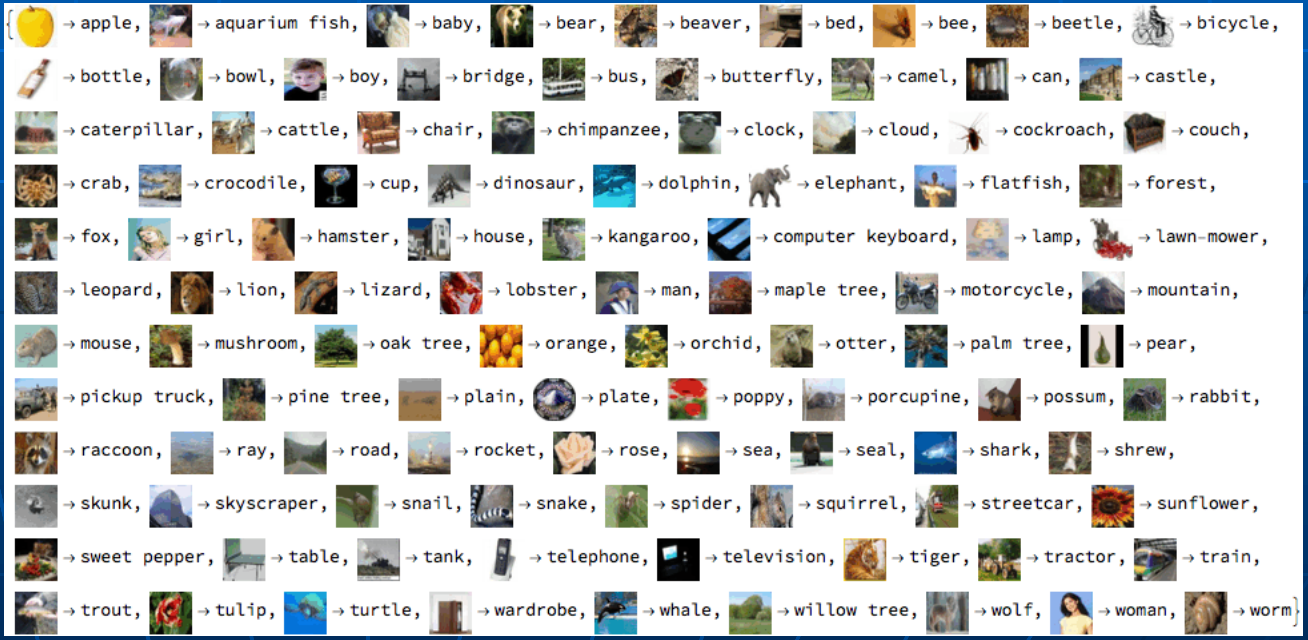
CIFAR 10



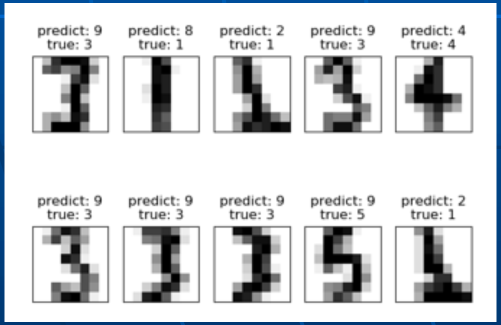
MNIST



CIFAR 100



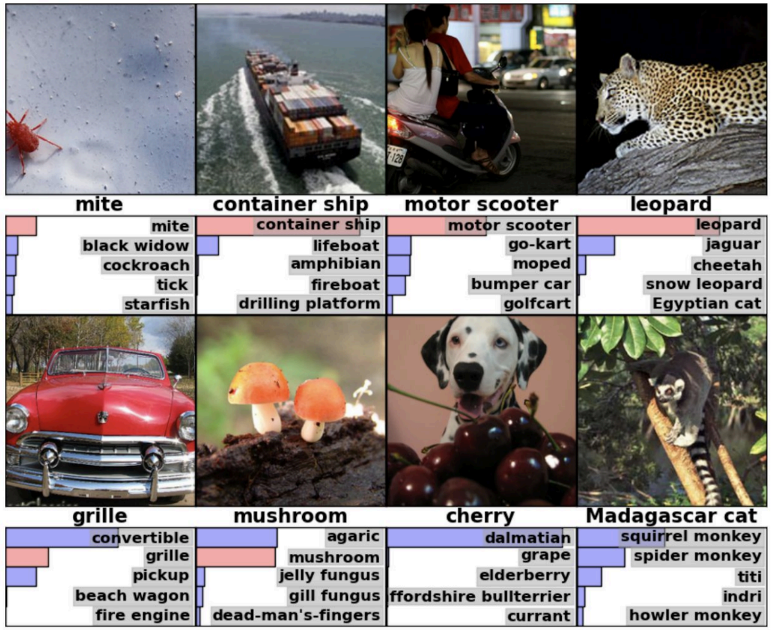
Sklearn.load_digits



ImageNet Challenge

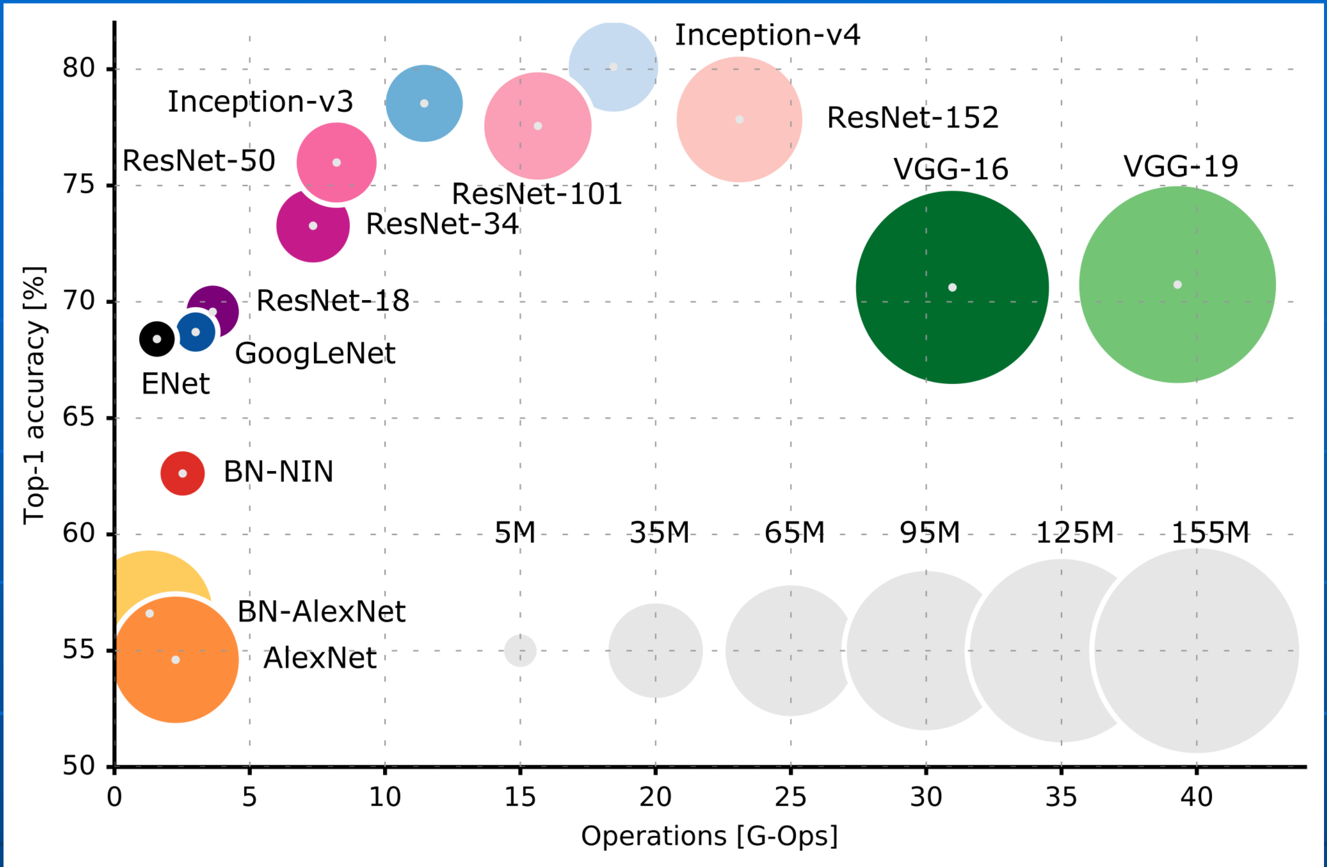
IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.



ILSVRC 2018 Winner (ImageNet Large Scale Visual Recognition Competition)

Inception-v4 (Evolved from GoogLeNet, Merged with ResNet Idea)



The **Top-1 error** is the percentage of the time that the classifier did not give the correct class the **highest** score. The **Top-5 error** is the percentage of the time that the classifier did not include the correct class among its **top 5** guesses.

Network	Top-1 Error	Top-5 Error
BN-Inception (Ioffe and Szegedy 2015)	25.2%	7.8%
Inception-v3 (Szegedy et al. 2015b)	21.2%	5.6%
Inception-ResNet-v1	21.3%	5.5%
Inception-v4	20.0%	5.0%
Inception-ResNet-v2	19.9%	4.9%

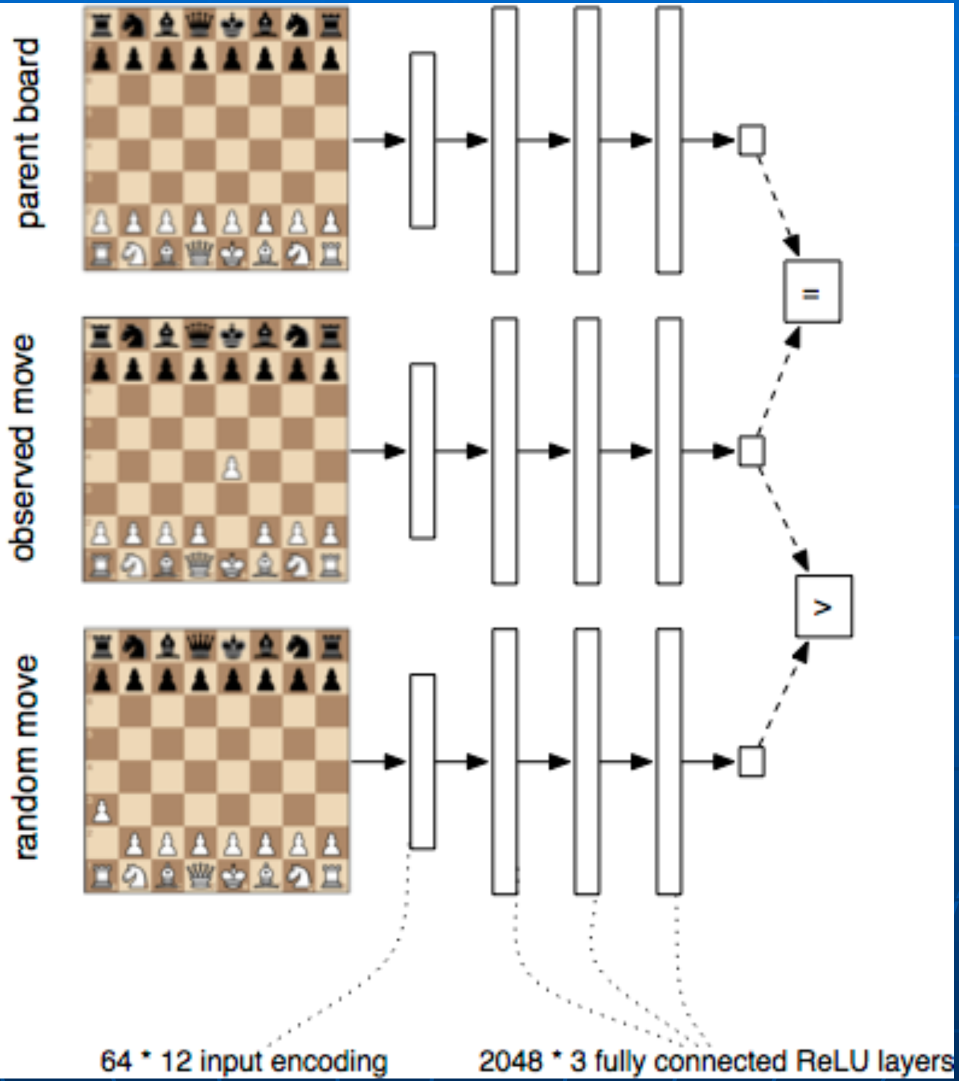
Chess

Chess: 10^{47}

Deep Blue, Feb 10, 1996

Go: 10^{170}

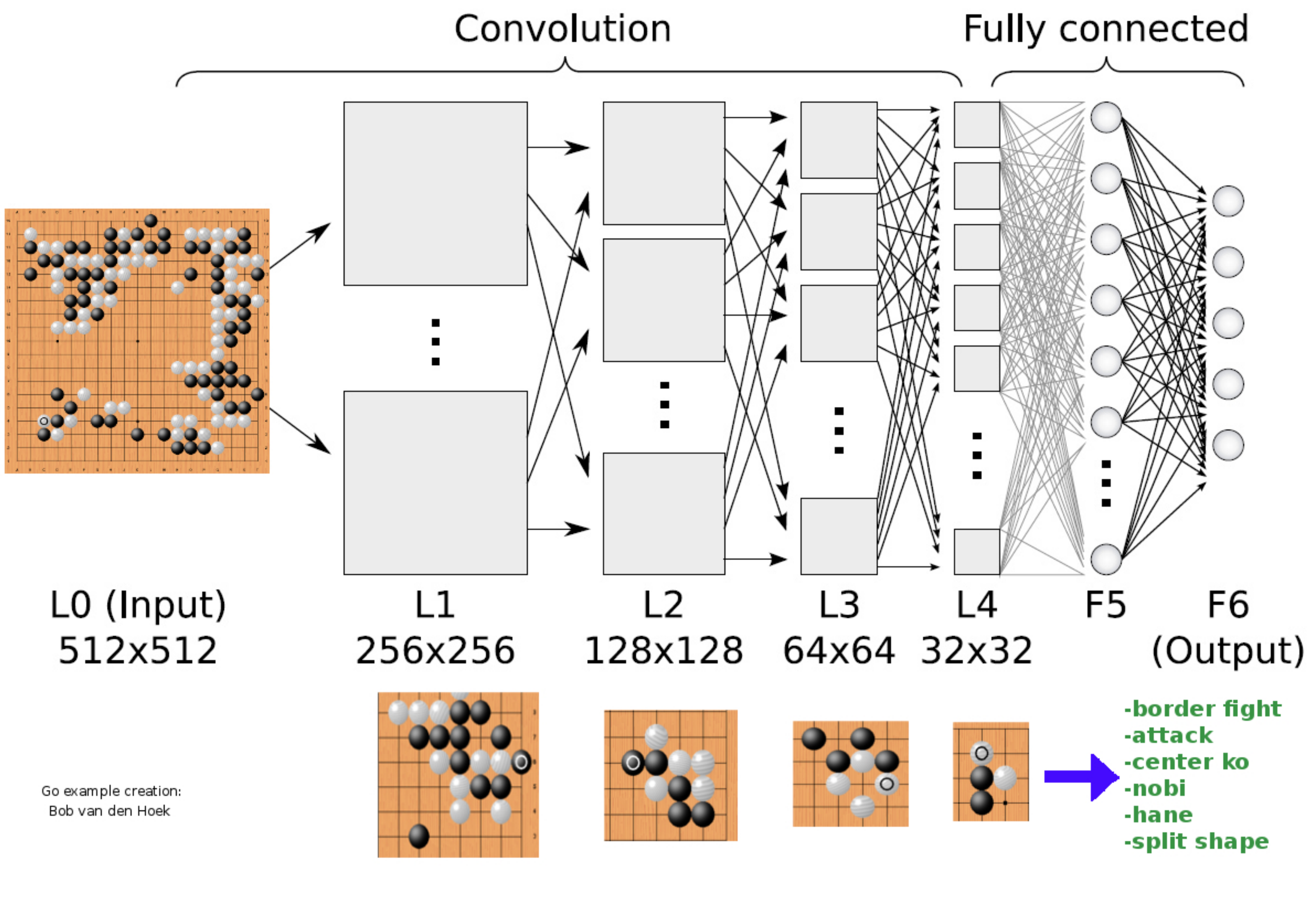
AlphaGo, March, 2016

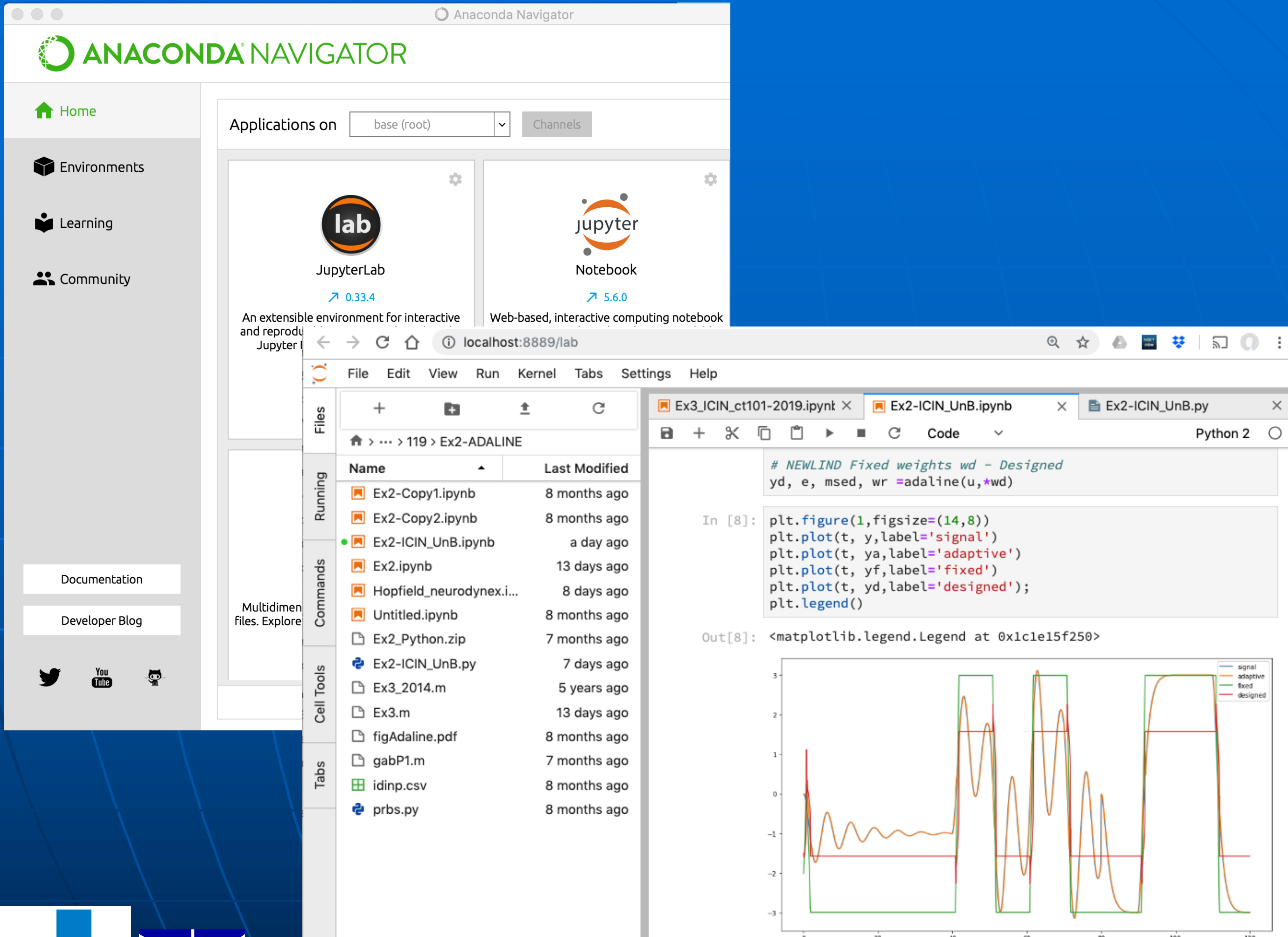


Go

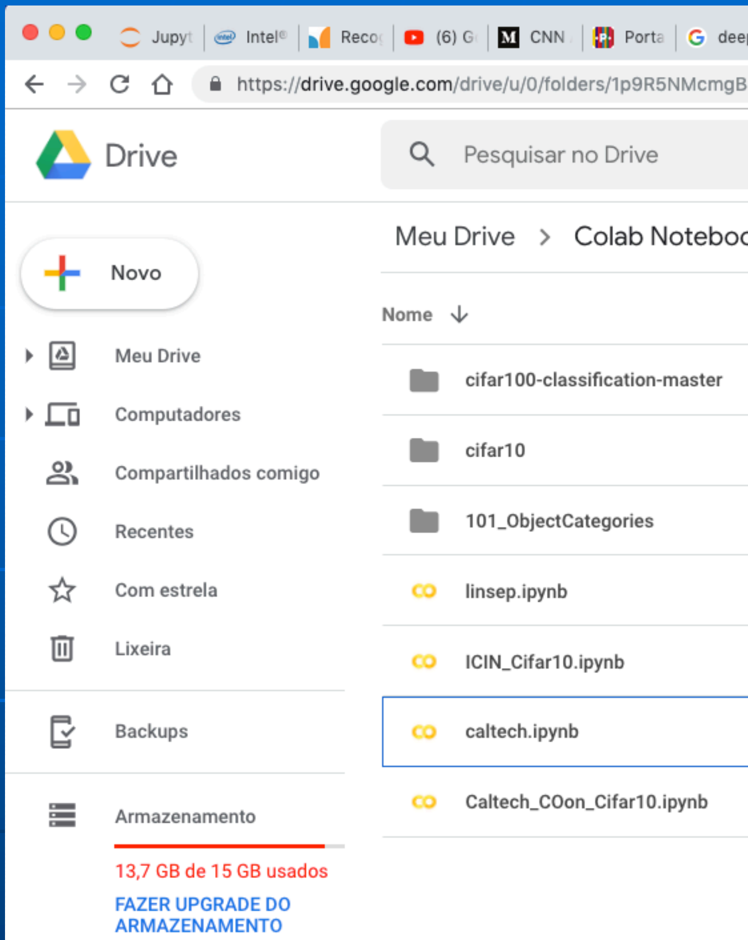
Chess: 10^{47}
Deep Blue, Feb 10, 1996

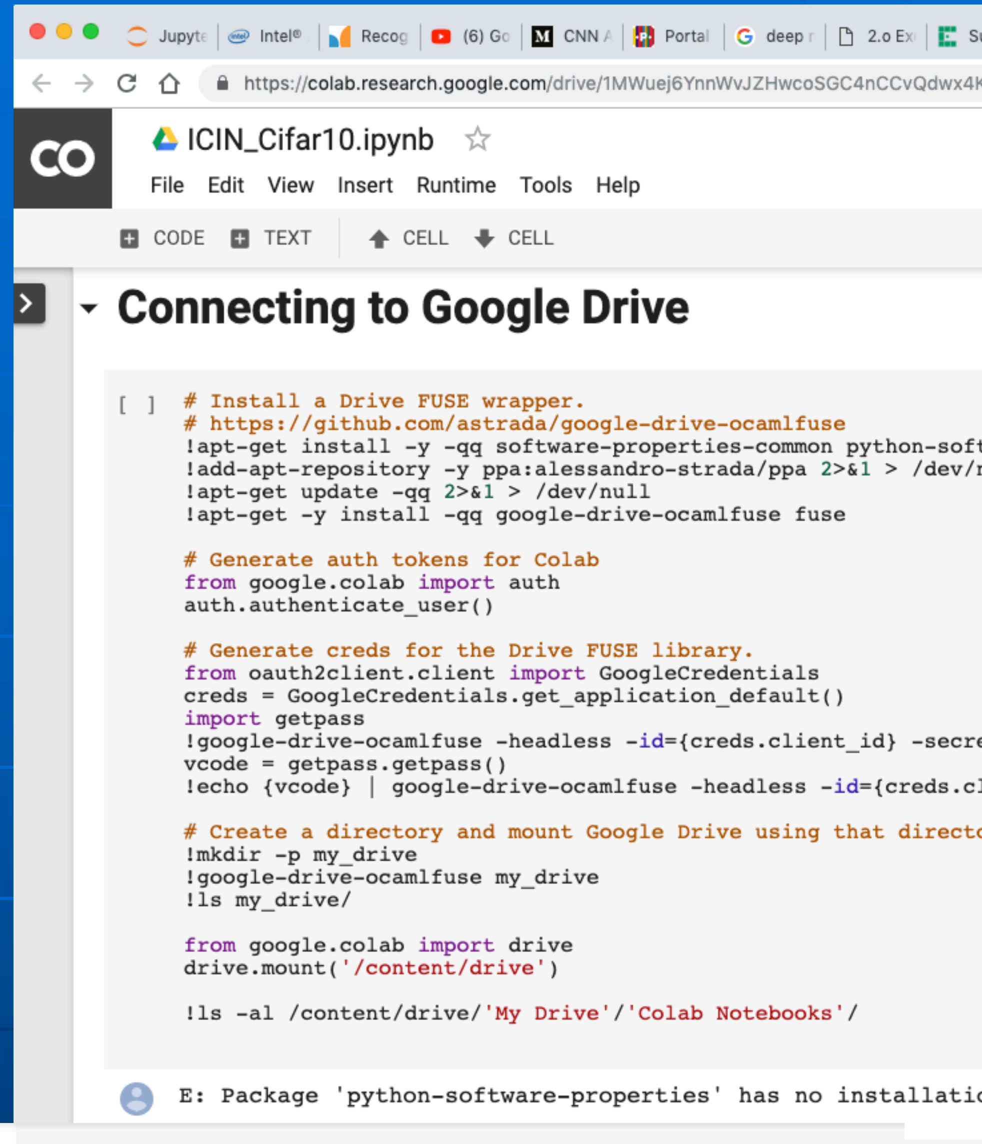
Go: 10^{170}
AlphaGo, March, 2016

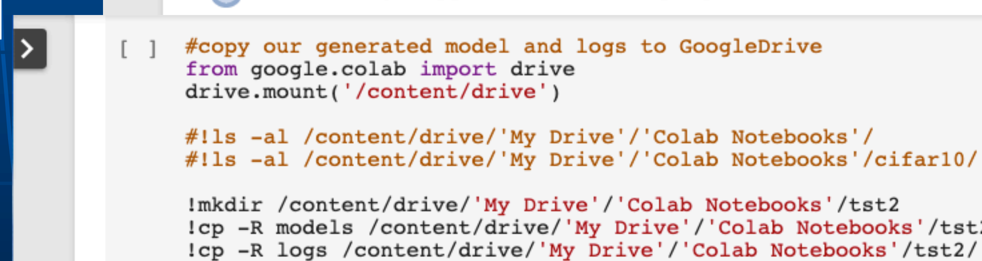



Anaconda Navigator
ANACONDA NAVIGATOR
Home
Environments
Learning
Community
Documentation
Developer Blog
Applications on
base (root)
Channels
JupyterLab
0.33.4
An extensible environment for interactive
Notebook
5.6.0
Web-based, interactive computing notebook
localhost:8889/lab
File
Edit
View
Run
Kernel
Tabs
Settings
Help
Files
Running
Commands
Cell Tools
Tabs
119 › Ex2-ADALINE
Name
Last Modified
Ex2-Copy1.ipynb 8 months ago
Ex2-Copy2.ipynb 8 months ago
Ex2-ICIN_UnB.ipynb a day ago
Ex2.ipynb 13 days ago
Hopfield_neurodynex.i... 8 days ago
Untitled.ipynb 8 months ago
Ex2_Python.zip 7 months ago
Ex2-ICIN_UnB.py 7 days ago
Ex3_2014.m 5 years ago
Ex3.m 13 days ago
figAdaline.pdf 8 months ago
gabP1.m 7 months ago
idinp.csv 8 months ago
prbs.py 8 months ago
Ex3_ICIN_ct101-2019.ipynb
Ex2-ICIN_UnB.ipynb
Ex2-ICIN_UnB.py
Code
Python 2
NEWLIND Fixed weights wd - Designed
yd, e, msed, wr =adaline(u,*wd)
In [8]:
plt.figure(1,figsize=(14,8))
plt.plot(t, y,label='signal')
plt.plot(t, ya,label='adaptive')
plt.plot(t, yf,label='fixed')
plt.plot(t, yd,label='designed');
plt.legend()
Out[8]:
<matplotlib.legend.Legend at 0x1c1e15f250>
signal
adaptive
fixed
designed

Google Colab







Intel DevCloud (Colfax)

adolfobs — ssh ◂ ssh colfax — 78×36

```
ABMac:~ adolfobs$ ssh colfax
/etc/profile.d/lang.sh: line 19: warning: setlocale: LC_CTYPE: cannot change
ocale (UTF-8): No such file or directory
########################################################################
# Welcome to Intel AI DevCloud!
#
# 1) See README.txt for information about usage policies and tips,
#    including the location of machine learning frameworks and datasets
#
# 2) See access portal https://access.colfaxresearch.com/
#    for additional information. Your invitation email contains
#    the authentication URL.
#
# 3) If you have any questions regarding the cloud usage, post them at
#    https://forums.intel.com/s/topic/0TO0P00000018NNWAY/intel-ai-academy
#
# Intel AI DevCloud Team
#
########################################################################
#
# Note: Cryptocurrency mining on the Intel AI DevCloud is forbidden.
#       Mining will lead to immediate termination of your account.
#       For complete terms of servicee rules, see
# https://access.colfaxresearch.com/doc/Colfax_Cluster_Service_Terms.pdf
#
########################################################################
Last login: Sat Apr 20 14:08:02 2019 from 10.9.0.249
u25367@login-1:~$ cd linsep
u25367@login-1:~/linsep$ qsub launch
22703.v-qsvr-1.aidevcloud
u25367@login-1:~/linsep$ qstat
Job ID                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
22703.v-qsvr-1             linsep_tf        u25367                 0 Q batch

u25367@login-1:~/linsep$
```

Jupyter | Brows | Unive | ENE/U | how d | Train_ | Tra

https://hub.colfaxresearch.com/user...

jupyter Train_CIFAR10

Logout | Control Panel | Time left: 3:58:31

Trusted | Python 2 (Intel, 2019 update 2)

File Edit View Insert Cell Kernel Help

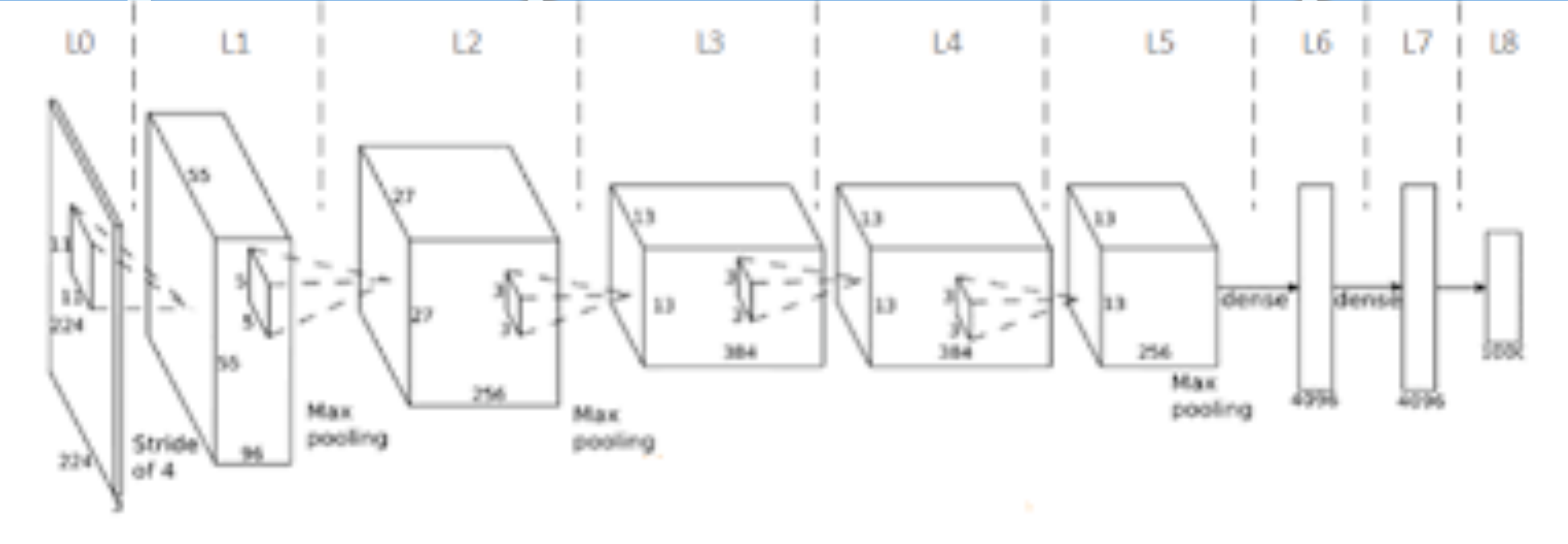
Run | Code

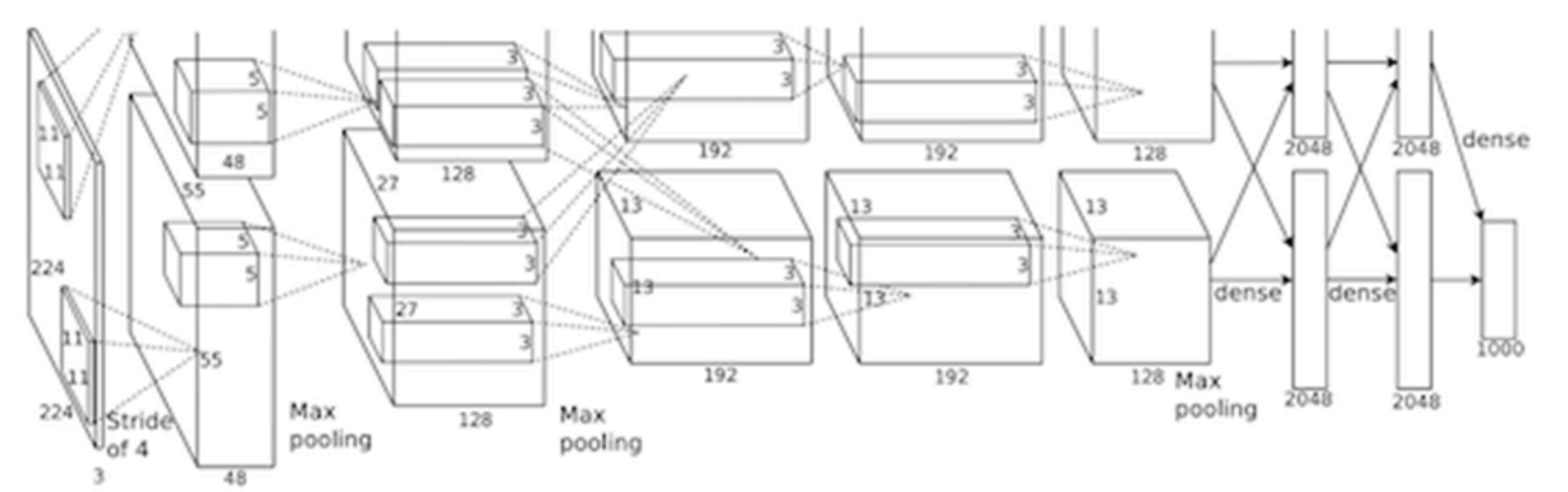
In [1]:

```python
# Simple CNN model for CIFAR-10
import numpy
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.constraints import maxnorm
from keras.optimizers import SGD, Adam
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.utils import np_utils
from keras import backend as K
K.set_image_dim_ordering('th')

seed = 7
numpy.random.seed(seed)
# load data
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train = X_train / 255.0
X_test = X_test / 255.0
# one hot encode outputs
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]

# Create the model Large scales
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(3, 32, 32), activation='
model.add(Dropout(0.2))
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

https://hub.colfaxresearch.com/hub/user/u25367t/

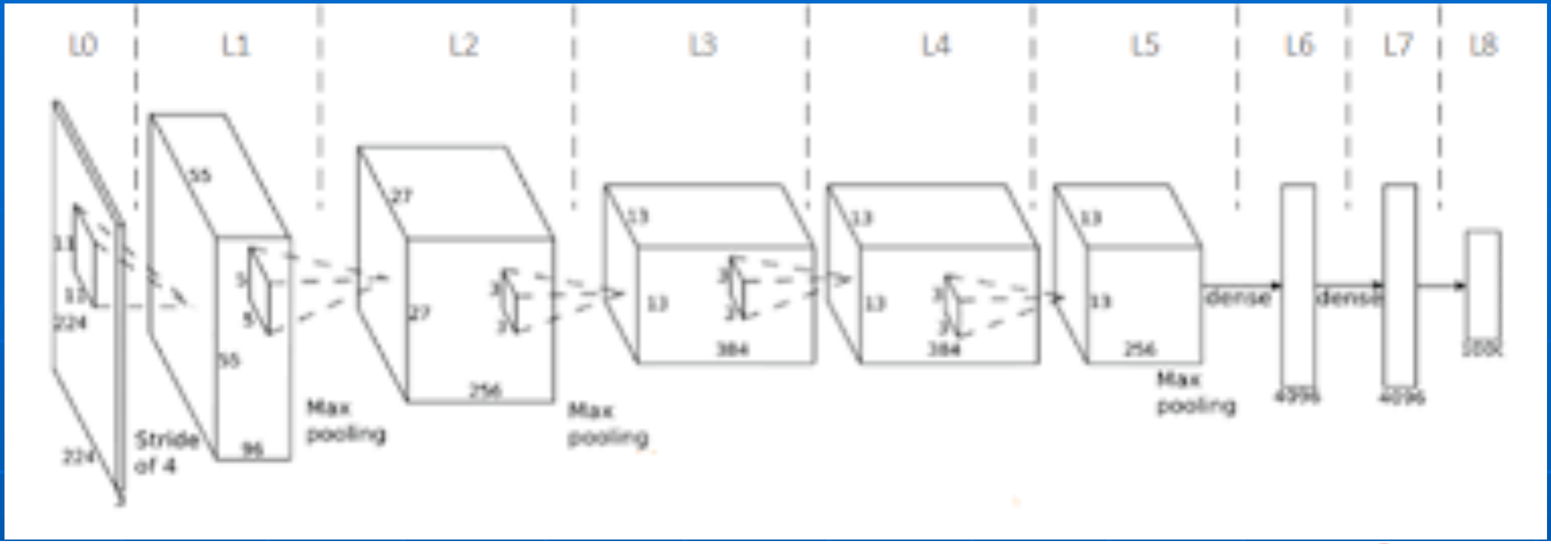
Deep Learning – CIFAR10 - ImageNet

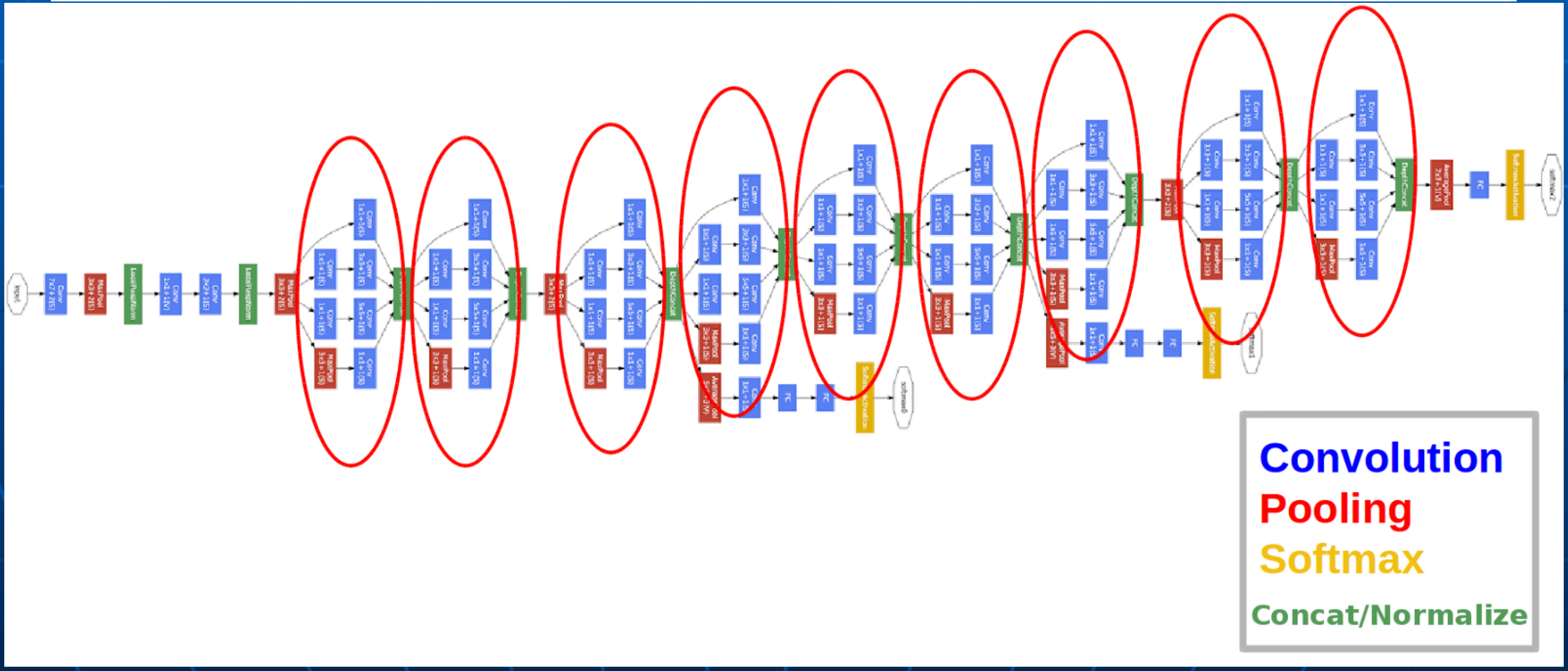




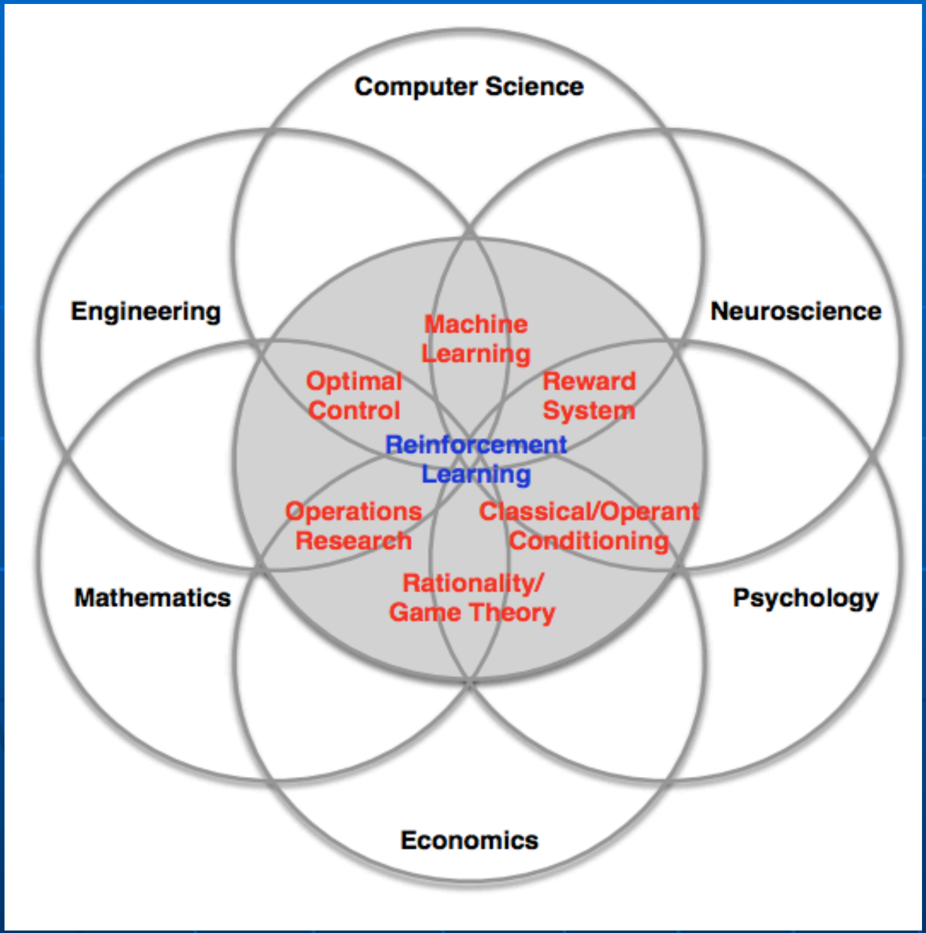
AlexNet architecture (May look weird because there are two different "streams". This is because the training process was so computationally expensive that they had to split the training onto 2 GPUs)

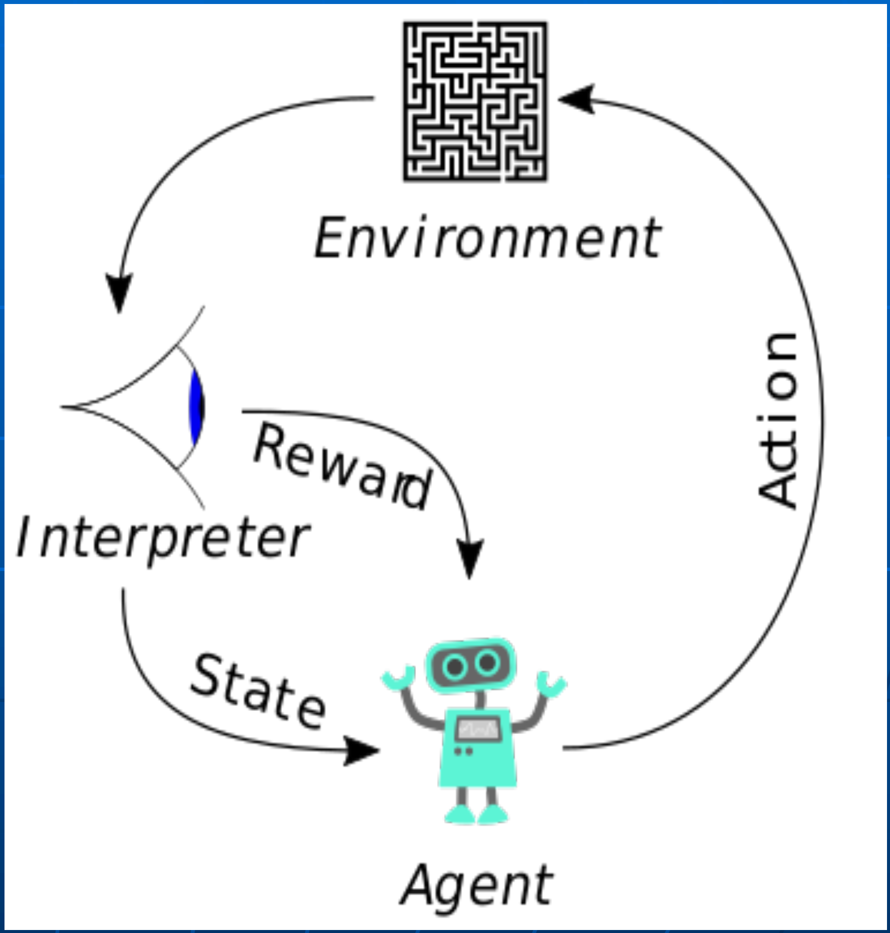
DL on CIFAR10 – CNN x GoogLeNet



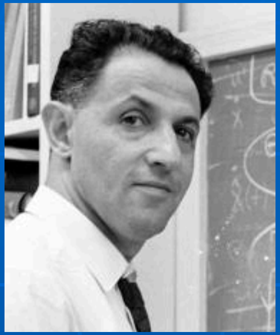


Reinforcement Learning





RL
– Optimal Control – Trial & Error – Temporal Difference



Bellman
Silver

Sutton-Barto

Dynamic
Programming
1953

Princeton

Reinforcement
Learning
1984

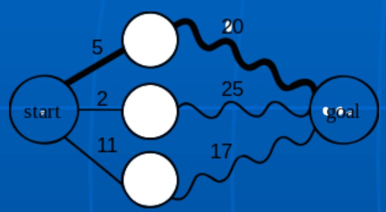
Massaschussets-Amhearst

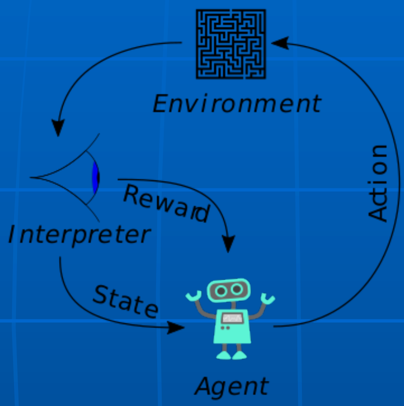
Deep RL
OpenMind Google
2010

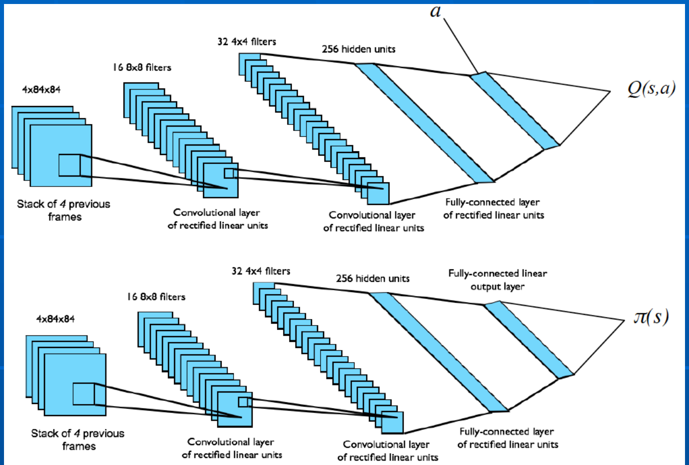
Alberta/London

RL

– Optimal Control – Trial & Error – Temporal Difference







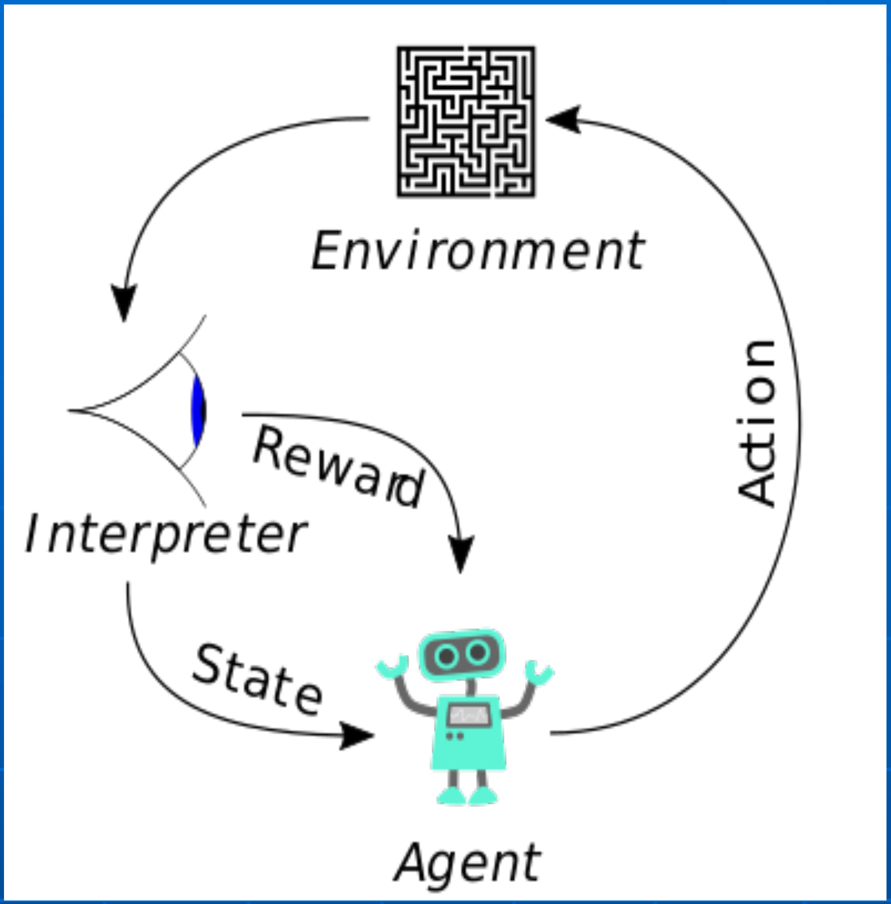
Dynamic Programming 1953

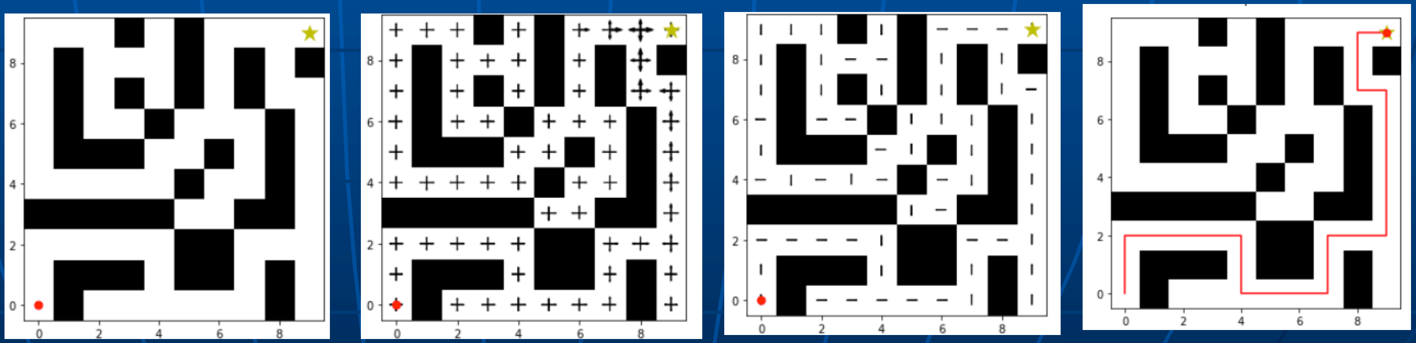
Reinforcement Learning 1984

Deep RL DeepMind Google 2010

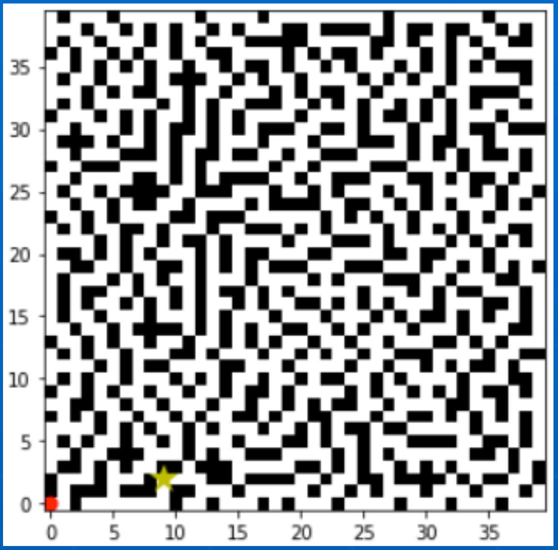
Reinforcement Learning

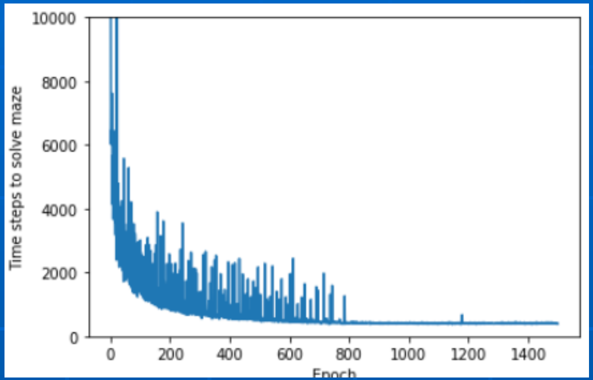
Learn from experience

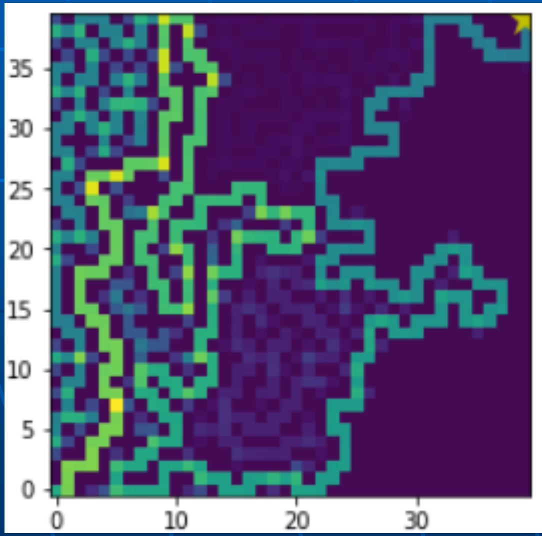


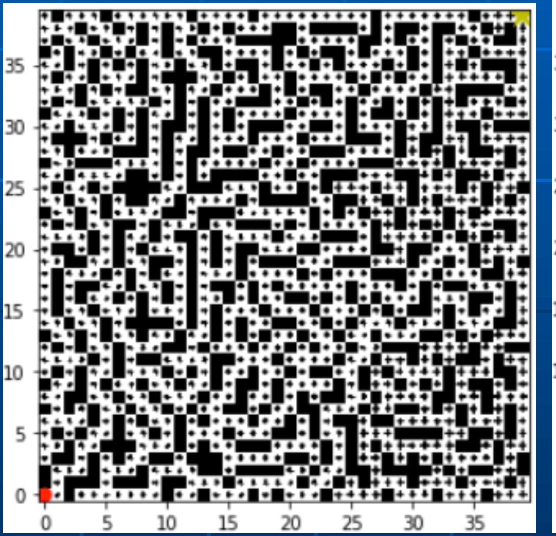


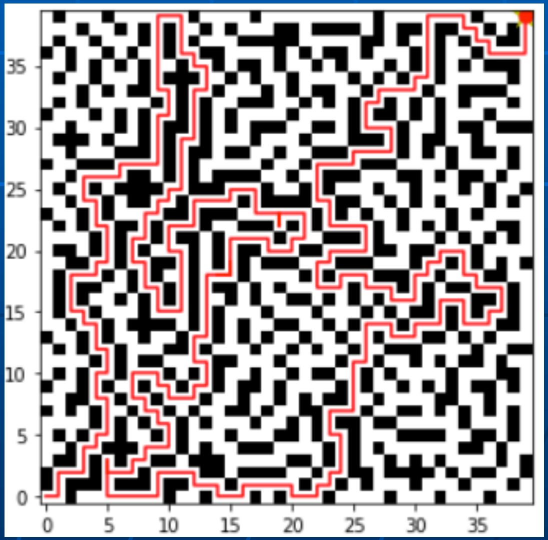
Maze RL





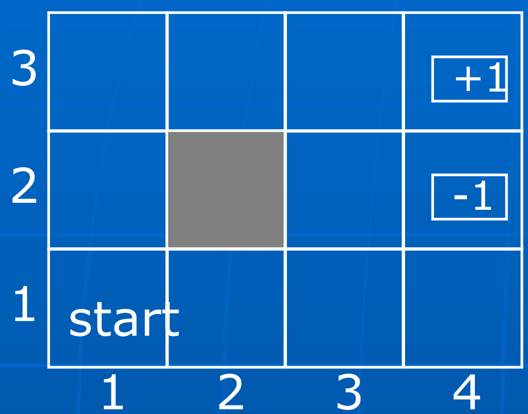






env.treasure = (mx-1,my-1)

Robot navigation

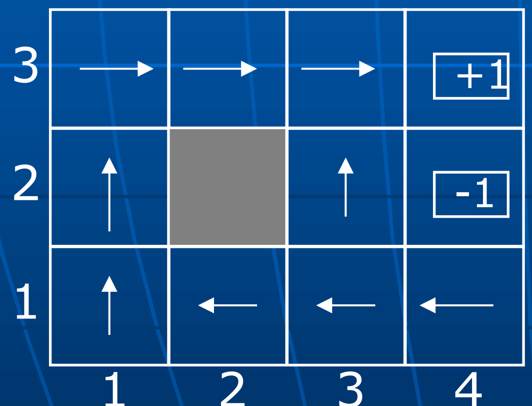


Epochs = training sequences:

(1,1)→(1,2)→(1,3)→(1,2)→(1,3)→(1,2)→(1,1)→(1,2)→(2,2)→(3,2) <u>-1</u>
(1,1)→(1,2)→(1,3)→(2,3)→(2,2)→(2,3)→(3,3) <u>+1</u>
(1,1)→(1,2)→(1,1)→(1,2)→(1,1)→(2,1)→(2,2)→(2,3)→(3,3) <u>+1</u>
(1,1)→(1,2)→(2,2)→(1,2)→(1,3)→(2,3)→(1,3)→(2,3)→(3,3) <u>+1</u>
(1,1)→(2,1)→(2,2)→(2,1)→(1,1)→(1,2)→(1,3)→(2,3)→(2,2)→(3,2) <u>-1</u>
(1,1)→(2,1)→(1,1)→(1,2)→(2,2)→(3,2) <u>-1</u>

Policy: mapping from states to actions

An optimal policy for the stochastic environment:



utilities of states:

	1	2	3	4
3	0.812	0.868	0.912	+1
2	0.762		0.660	-1
1	0.805	0.655	0.611	0.388

Deep Reinforcement Learning

Approaches To Reinforcement Learning

Value-based RL

- Estimate the optimal value function $Q^*(s, a)$
- This is the maximum value achievable under any policy

Policy-based RL

- Search directly for the optimal policy π^*
- This is the policy achieving maximum future reward

Model-based RL

- Build a model of the environment
- Plan (e.g. by lookahead) using model

Adapted from: Deep Reinforcement learning – D. Silver, Google DeepMind, 2016

Generalization

With table lookup representation (of U,M,R,Q) up to 10,000 states or more

Chess ~ 10^{47} Backgammon ~ 10^{50}

Hard to represent & visit all states!

Implicit representation,
e.g. $U(i) = w_1 f_1(i) + w_2 f_2(i) + \ldots + w_n f_n(i)$

Chess 10^{47} states → n weights
This compression does generalization

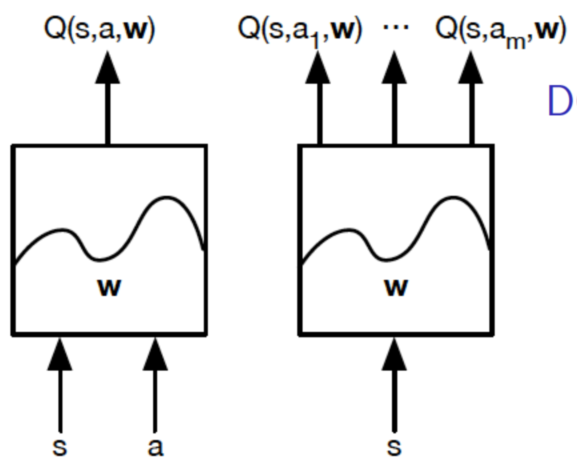
E.g. Backgammon:

Observe $1/10^{44}$ state space and beat any human.

Deep Reinforcement Learning

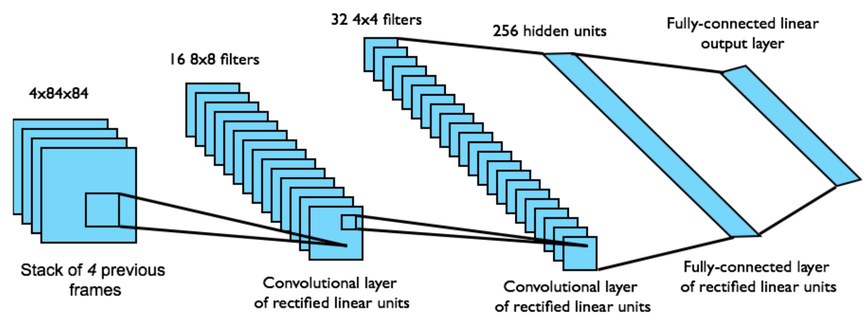
Represent value function by Q-network with weights **w**

$$Q(s, a, \mathbf{w}) \approx Q^*(s, a)$$



DQN in Atari

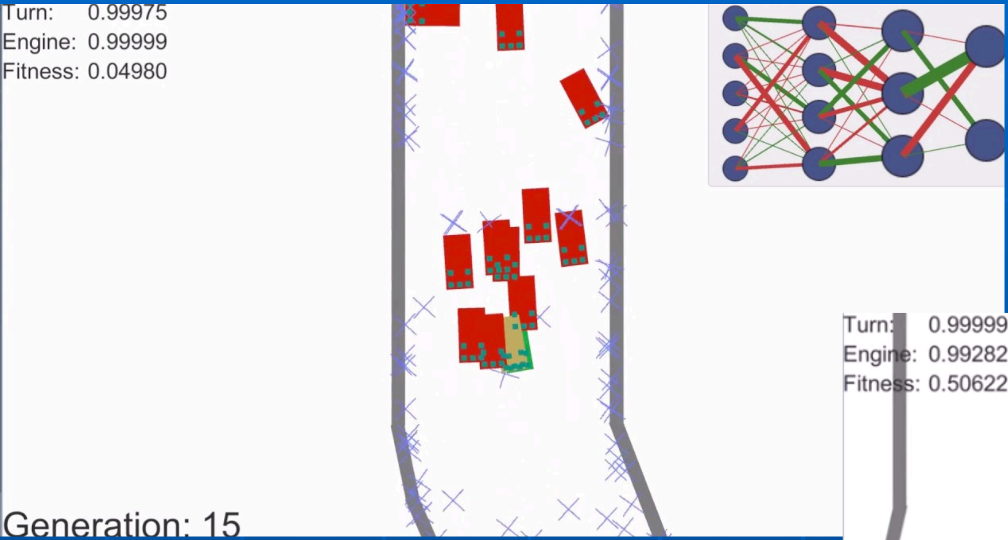
- End-to-end learning of values $Q(s, a)$ from pixels s
- Input state s is stack of raw pixels from last 4 frames
- Output is $Q(s, a)$ for 18 joystick/button positions
- Reward is change in score for that step

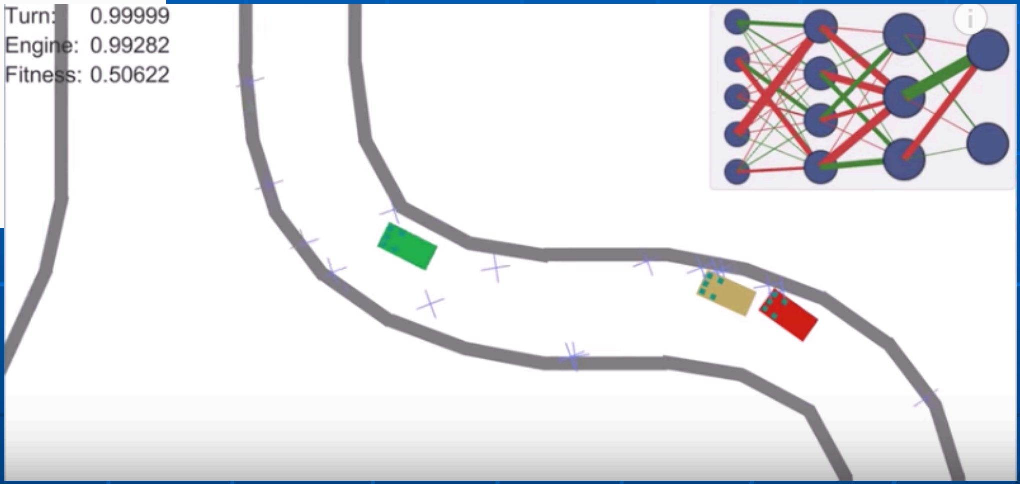


Network architecture and hyperparameters fixed across all games

Adapted from: Deep Reinforcement learning – D. Silver, Google DeepMind, 2016

RL – Examples: Self Driving Cars

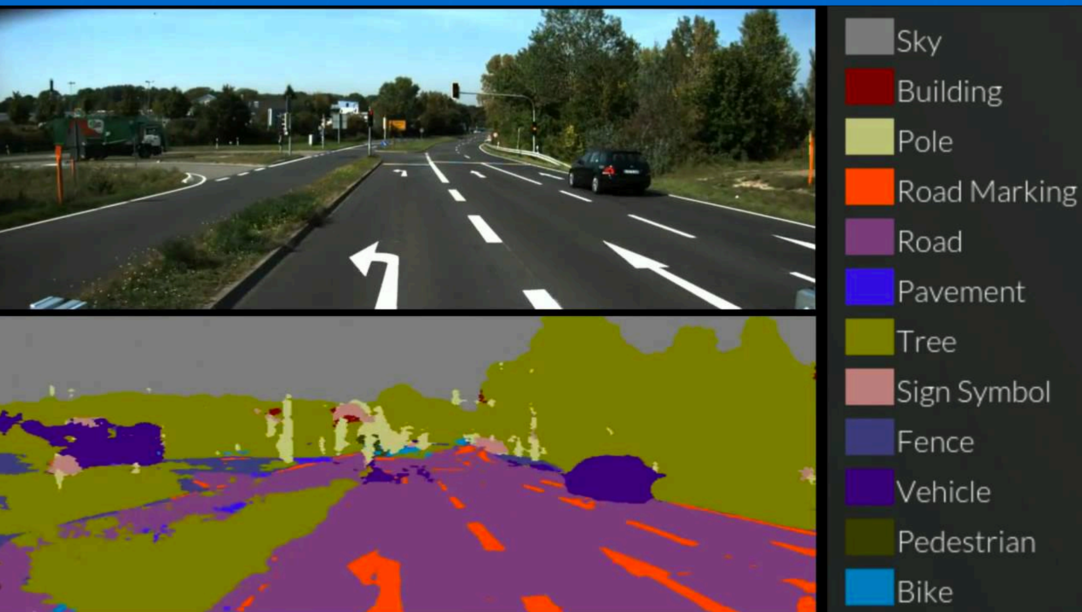




Deep Learning Cars
1.286.087 visualizations
Samuel Arzt
Pub. 23/oct/2016
Date: 1/Nov/2018

https://www.youtube.com/watch?v=Aut32pR5PQA

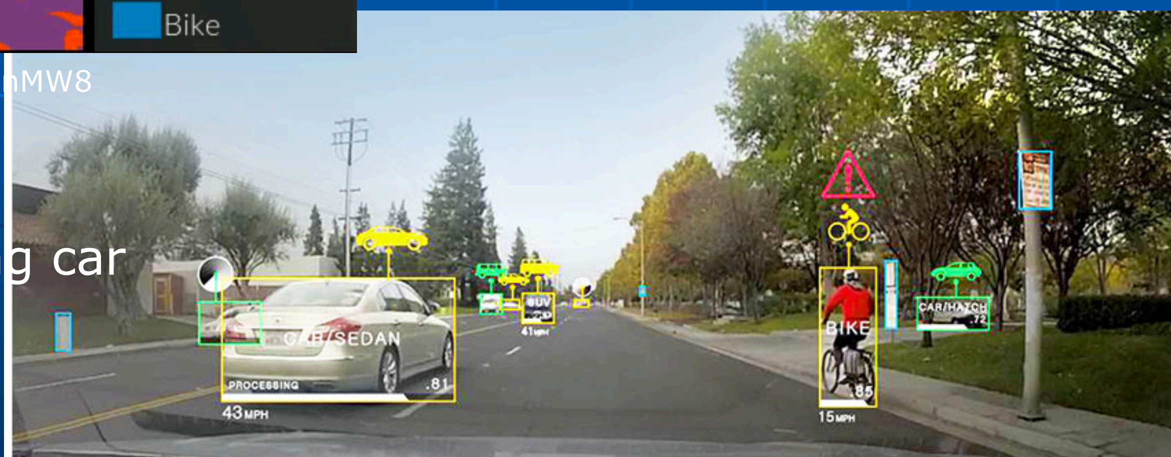
RL – Examples: Self Driving Cars



https://www.youtube.com/watch?v=kMMbW96nMW8

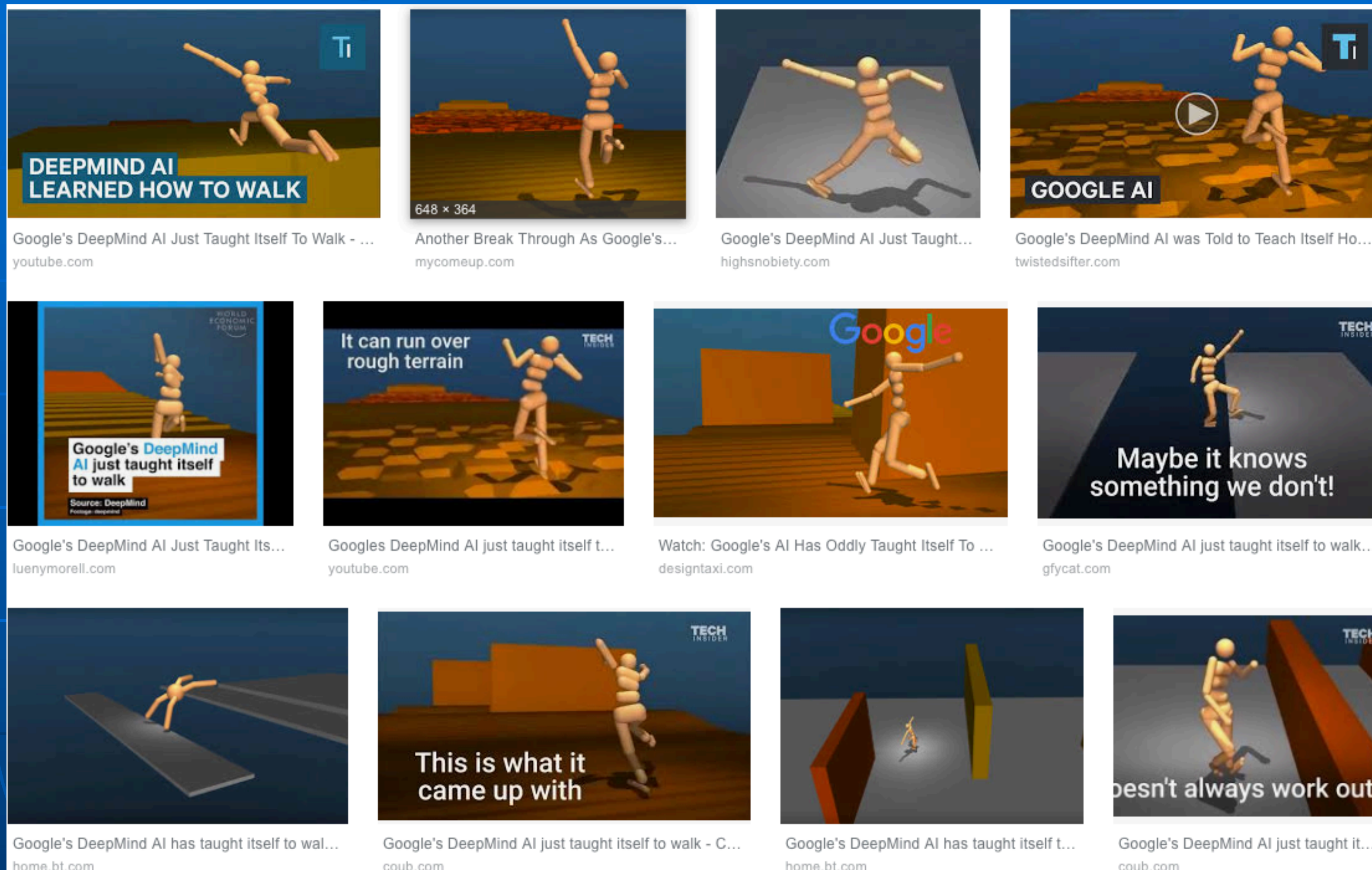
https://www.linkedin.com/pulse/machine-lea
fundamentals-self-driving-cars-david-silver/

Deep Learning:
Technology behind self-driving car
6.194 visualizations
Pub. 25/dec/2016



http://www.alphr.com/cars/1001713/practice-makes-perfect-driverless-cars-will-learn-from

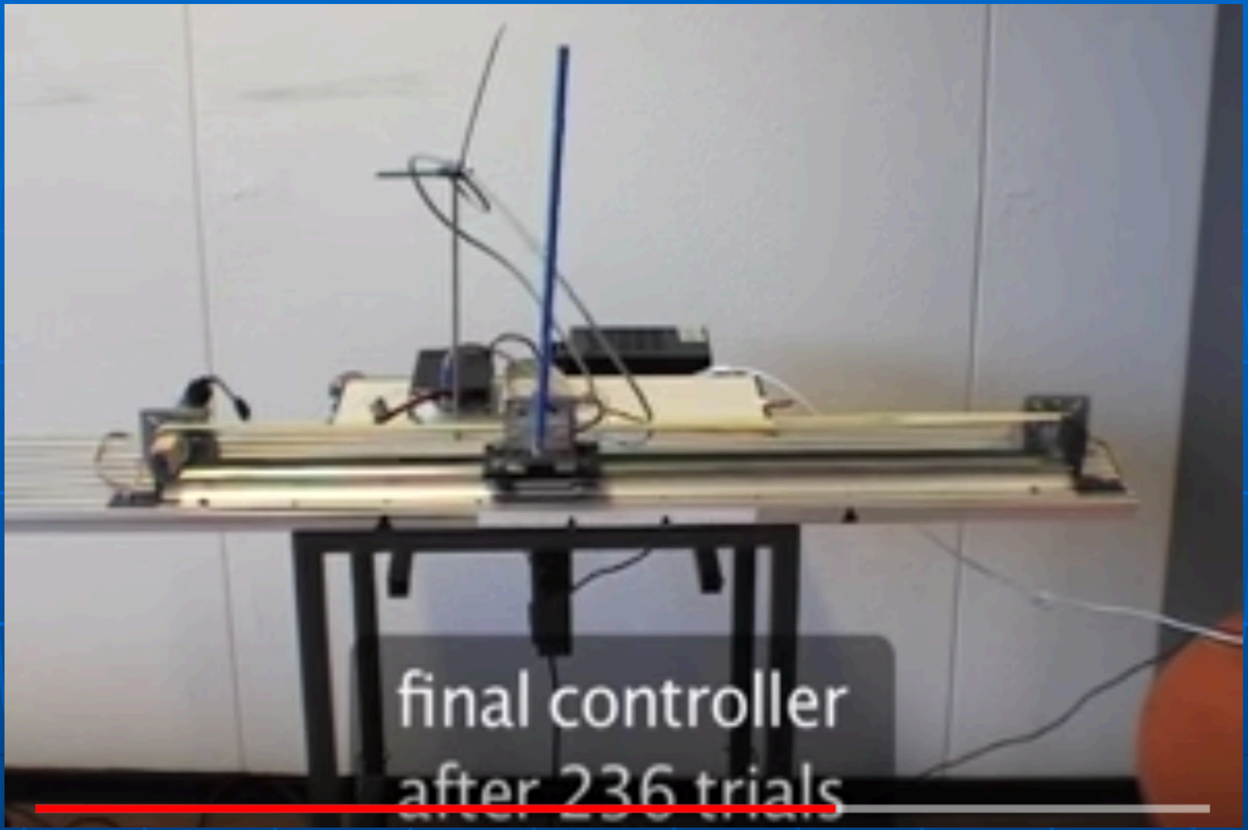
RL – Examples: Learn To Walk



Google's DeepMind AI
Just Taught Itself to Walk
5.985.455 vis. 12/jul/2017

https://www.youtube.com/watch?v=gn4nRCC9TwQ

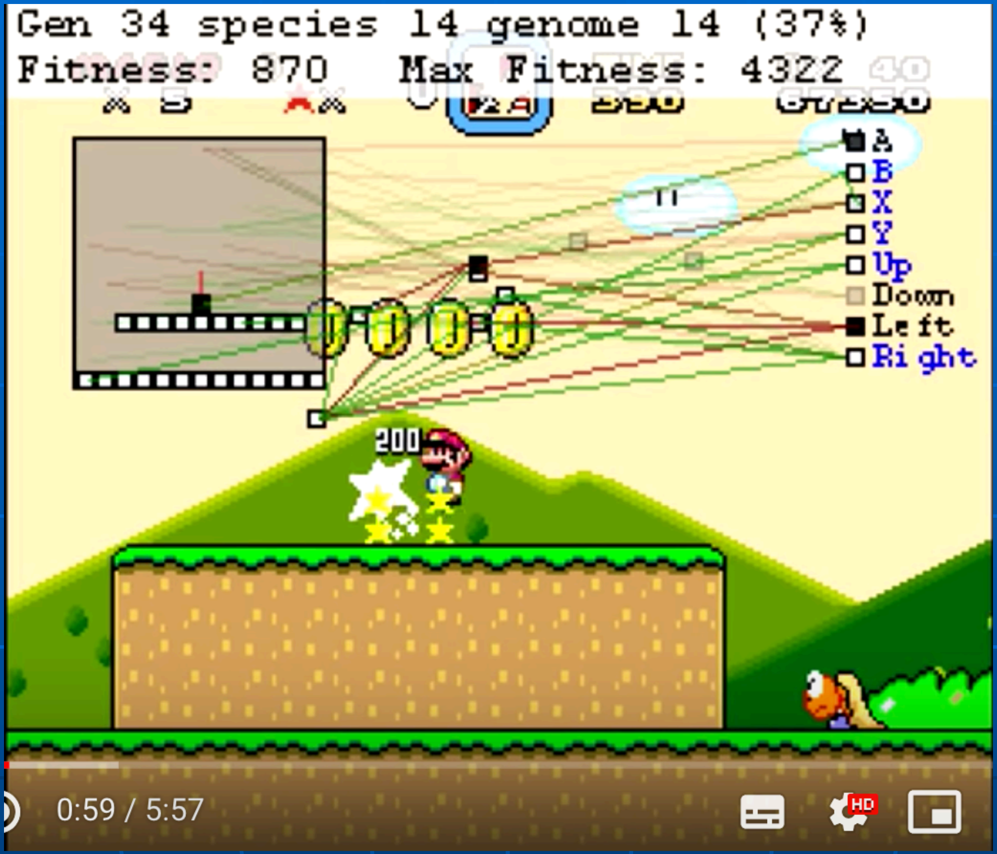
RL – Examples: Swing-Up & Balance



Learning to Swing-Up and Balance from Scratch
48.860 visualizations
Pub. 16/mar/2010

https://www.youtube.com/watch?v=Lt-KLtk

Reinforcement Learning - Examples



MarI/O – Machine Learning for Video Games
7.291.721 visualizations
Pub. 13/jun/2015 https://www.youtube.com/watch?v=qv6UVOQ0F44&t=129s

Deep Reinforcement Learning

Q-Learning

- Optimal Q-values should obey Bellman equation

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q(s',a')^* \mid s,a\right]$$

- Treat right-hand side $r + \gamma \max_{a'} Q(s',a',\mathbf{w})$ as a target
- Minimise MSE loss by stochastic gradient descent

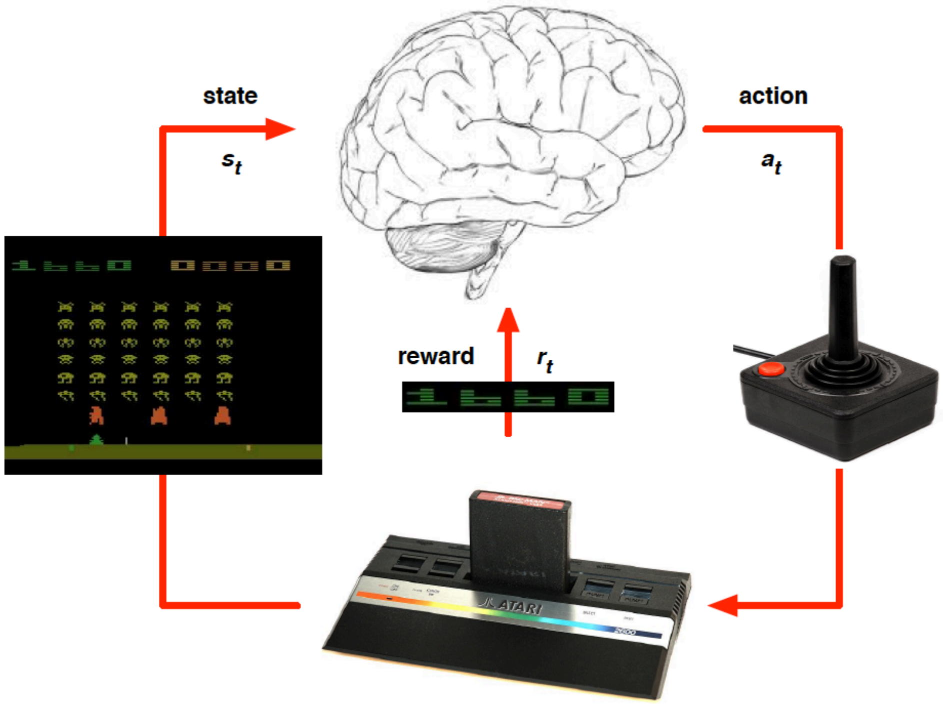
$$l = \left(r + \gamma \max_{a} Q(s',a',\mathbf{w}) - Q(s,a,\mathbf{w})\right)^2$$

- Converges to Q^* using table lookup representation
- But diverges using neural networks due to:
  - Correlations between samples
  - Non-stationary targets

Adapted from: Deep Reinforcement learning – D. Silver, Google DeepMind, 2016

Deep Reinforcement Learning

Deep Reinforcement Learning in Atari



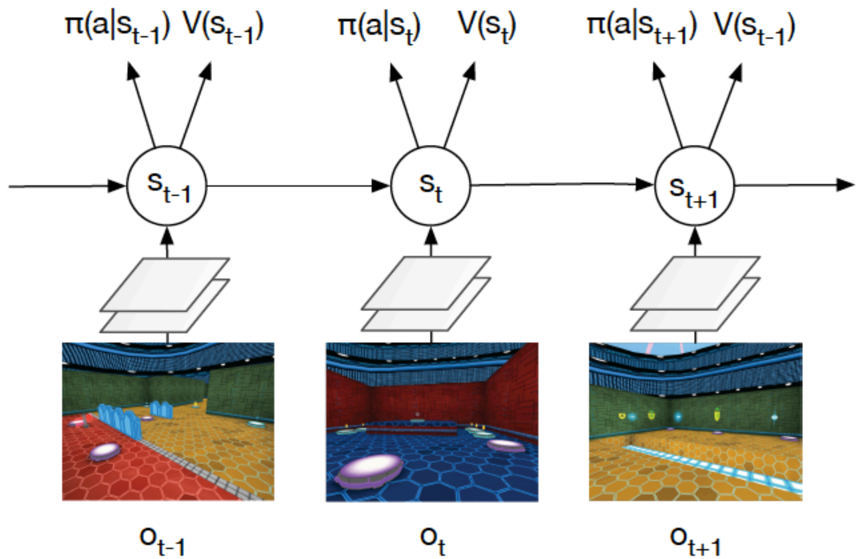
27 Fev 2015

28 jan 2016

Adapted from: Deep Reinforcement learning – D. Silver, Google DeepMind, 2016

Deep Reinforcement Learning

A3C in Labyrinth



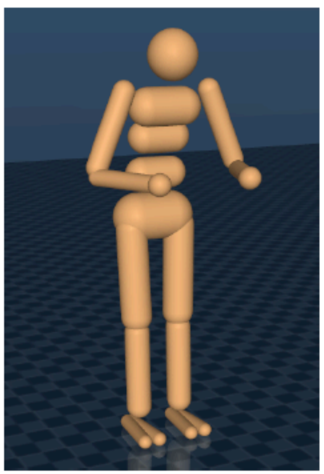
- End-to-end learning of softmax policy $\pi(a|s_t)$ from pixels
- Observations o_t are raw pixels from current frame
- State $s_t = f(o_1, ..., o_t)$ is a recurrent neural network (LSTM)
- Outputs both value $V(s)$ and softmax over actions $\pi(a|s)$
- Task is to collect apples (+1 reward) and escape (+10 reward)

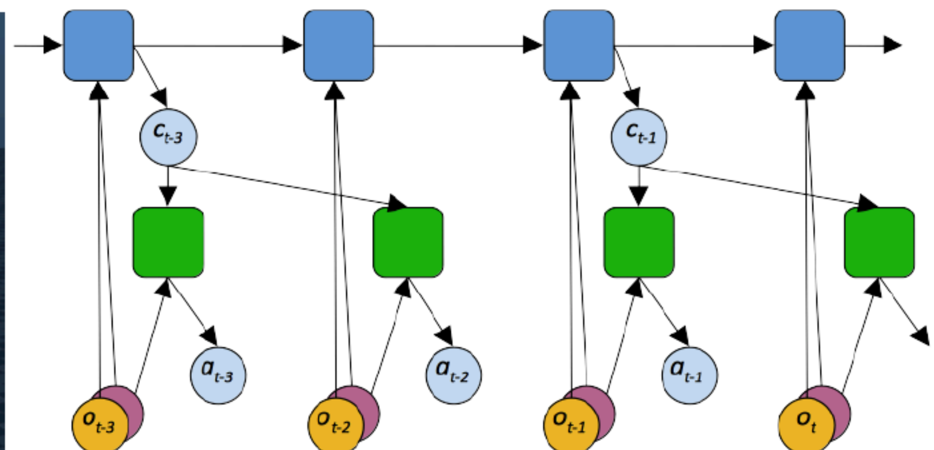
Adapted from: Deep Reinforcement learning – D. Silver, Google DeepMind, 2016

Deep Reinforcement Learning

A3C in Simulated Physics Demo

- Asynchronous RL is viable alternative to experience replay
- Train a hierarchical, recurrent locomotion controller
- Retrain controller on more challenging tasks

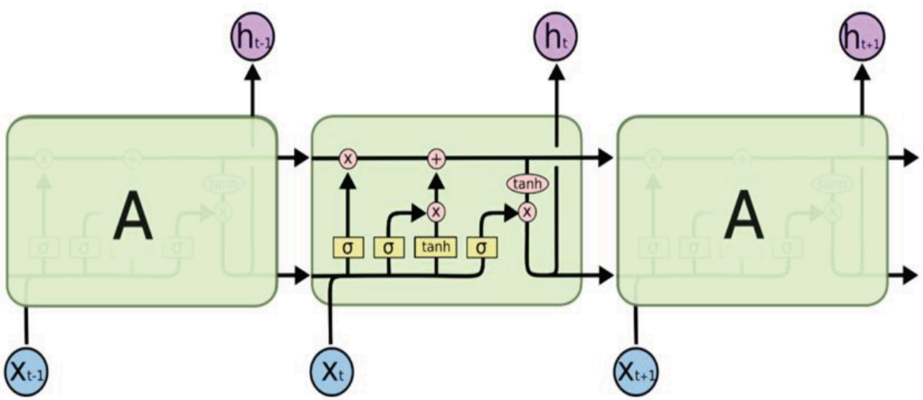




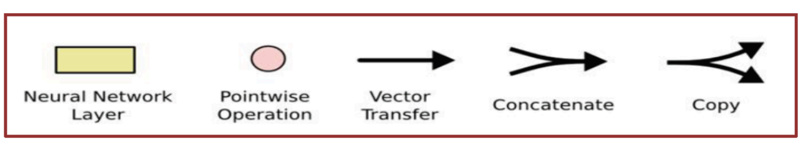
Adapted from: Deep Reinforcement learning – D. Silver, Google DeepMind, 2016

Long Short-Term Memory

(LSTM)[Hochreiter & Schmidhuber (1997)]



The repeating module in an LSTM contains four interacting layers.

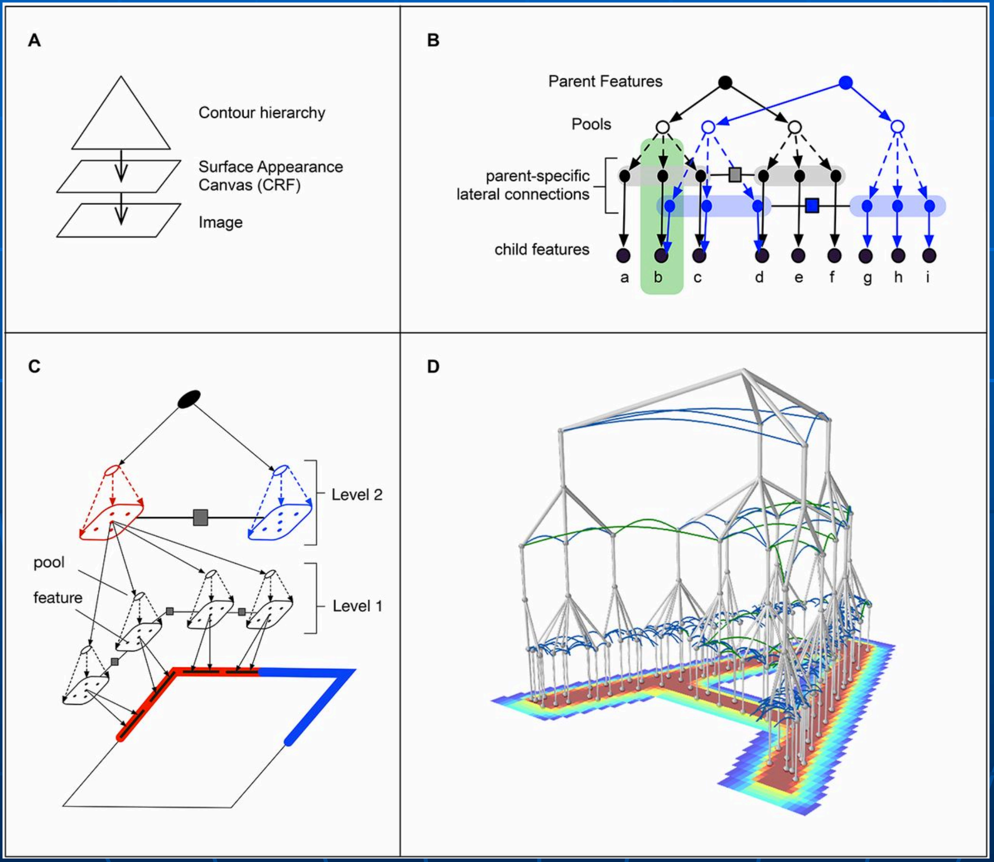


LSTM Segmentation - Zheng et al ICCV 2015

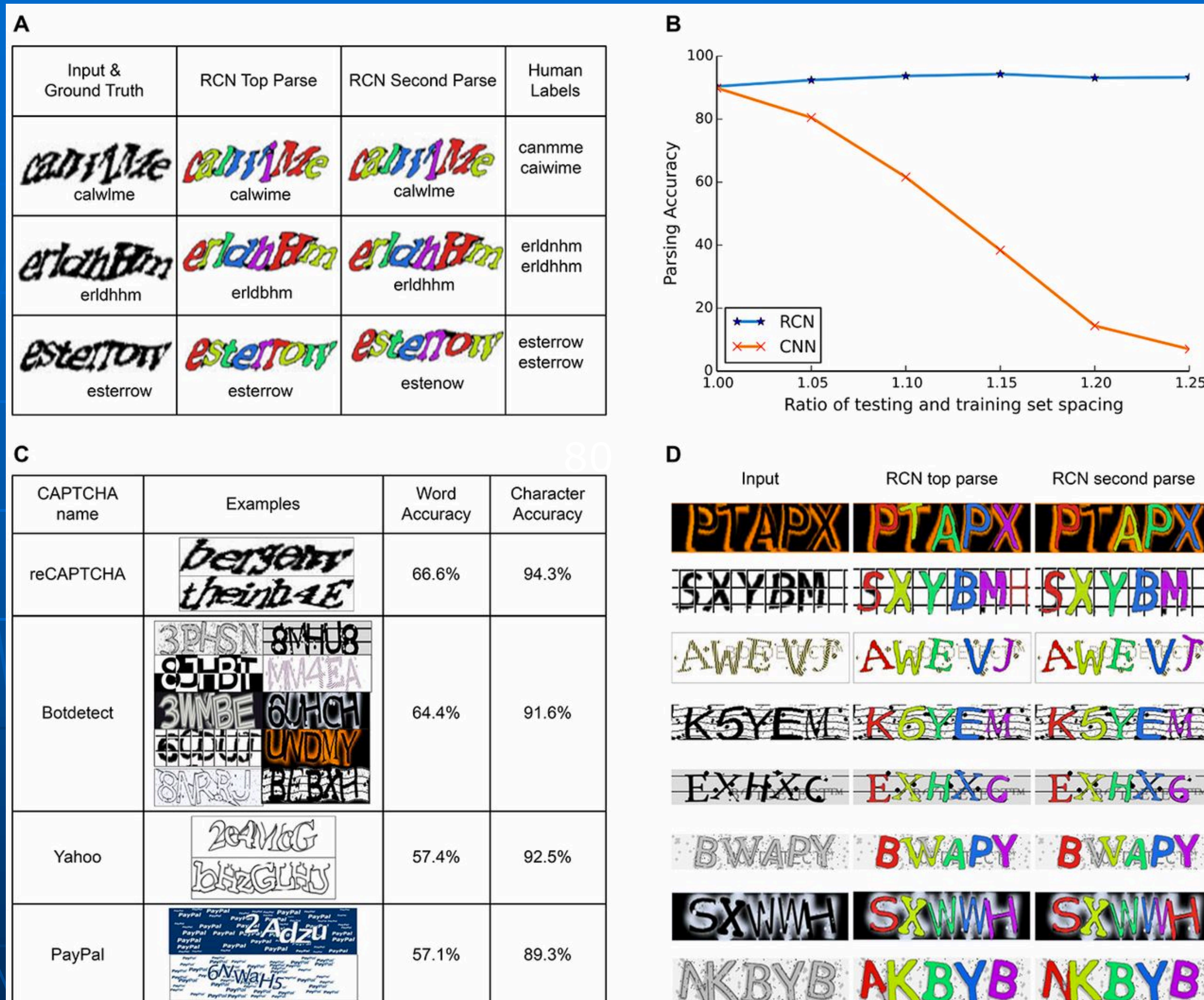
Cortical Recurrent Networks

A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs

D. George, W. Lehrach, K. Kansky, M. Lázaro-Gredilla, C. Laan, B. Marthi, X. Lou, Z. Meng, Y. Liu, H. Wang, A. Lavin, D. S. Phoenix
Science Vol.: eaag2612 DOI: 10.1126/science.aag2612, Dec 2017

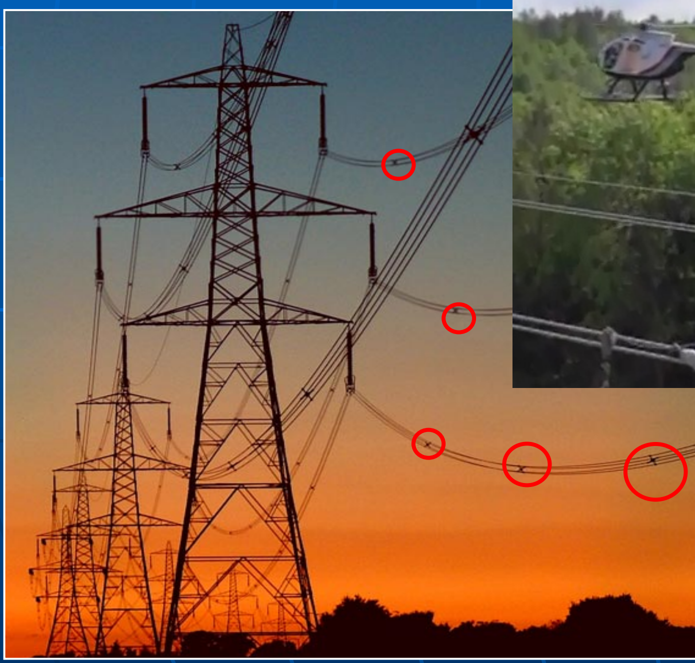


CAPTCHA - CRN


A
Input & Ground Truth
RCN Top Parse
RCN Second Parse
Human Labels
calwlme
calwime
calwlme
canmme
caiwime
erldhhm
erldbhm
erldhhm
erldnhm
erldhhm
esterrow
esterrow
estenow
esterrow
esterrow
B
Parsing Accuracy
Ratio of testing and training set spacing
RCN
CNN
100
80
60
40
20
0
1.00
1.05
1.10
1.15
1.20
1.25
C
CAPTCHA name
Examples
Word Accuracy
Character Accuracy
reCAPTCHA
66.6%
94.3%
Botdetect
64.4%
91.6%
Yahoo
57.4%
92.5%
PayPal
57.1%
89.3%
D
Input
RCN top parse
RCN second parse

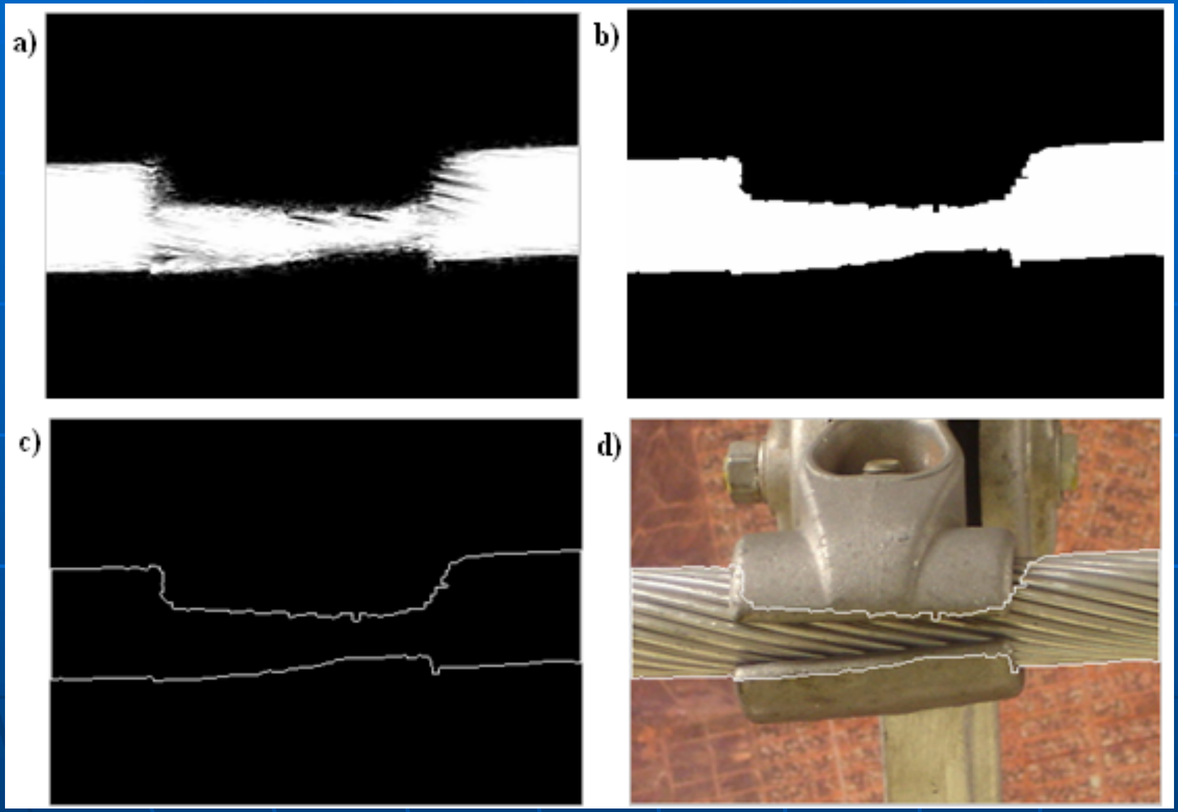
Inspection of Transmission Lines

- Autonomous system - visual inspection of electricity transmission lines
- Detection of flaws in the gripper of the line spacers

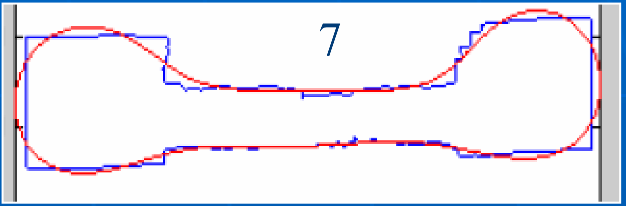


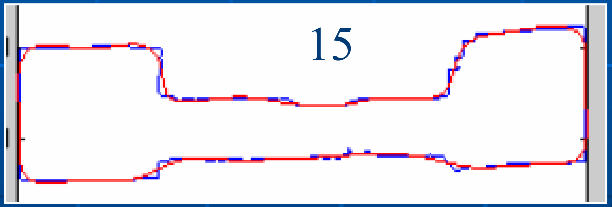
Need Maintenance!!

Gripped cable contour: FFT coefficients of directional chains



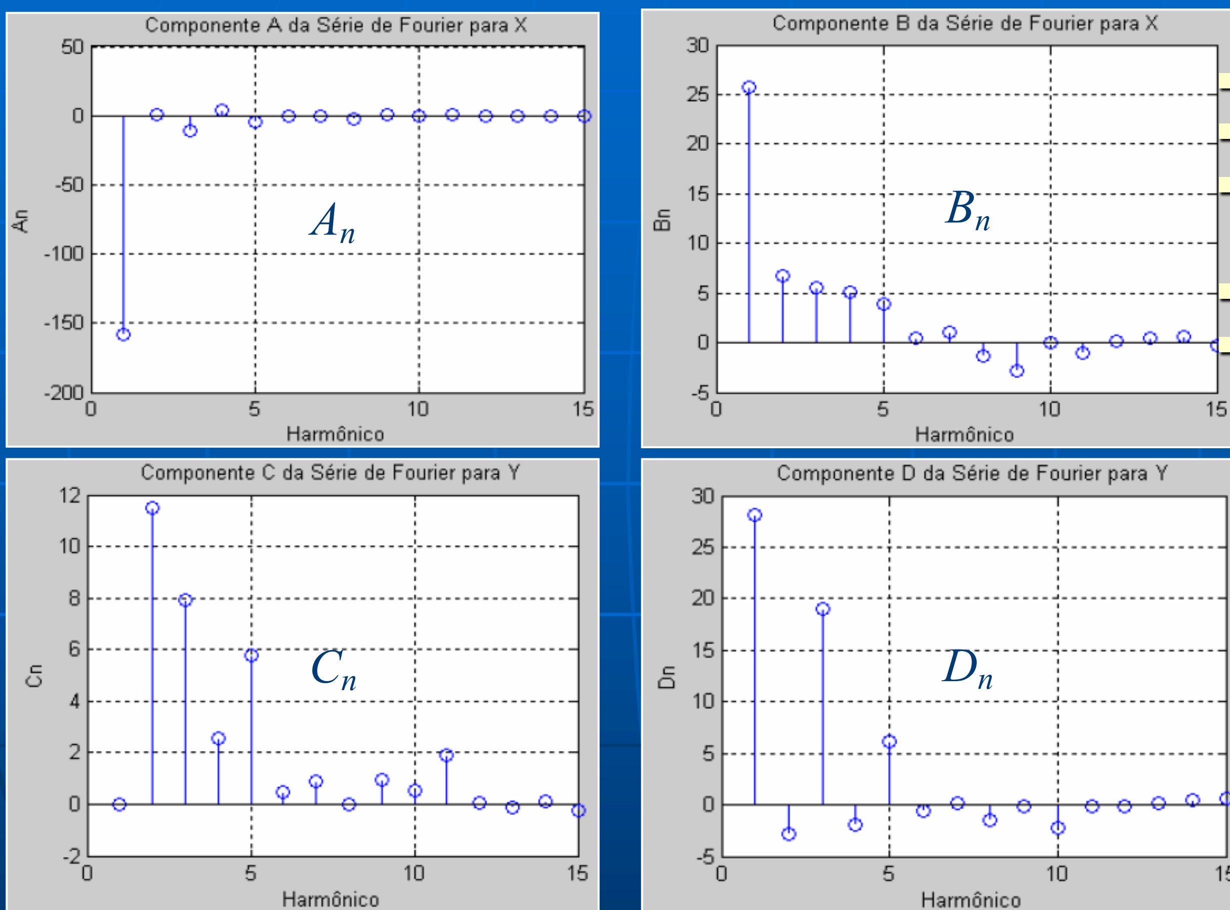
a) Gabor – b) Closing – c) Border – d)Image



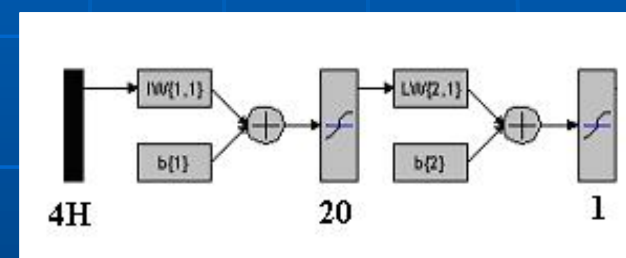


Reconstruction -
7 and 15 Harmonics

Gripped cable contour:
FFT coefficients of directional chains



- ANN – "Need Maintenance" classification
- Training, Test, Validation 80, 25, 25 images



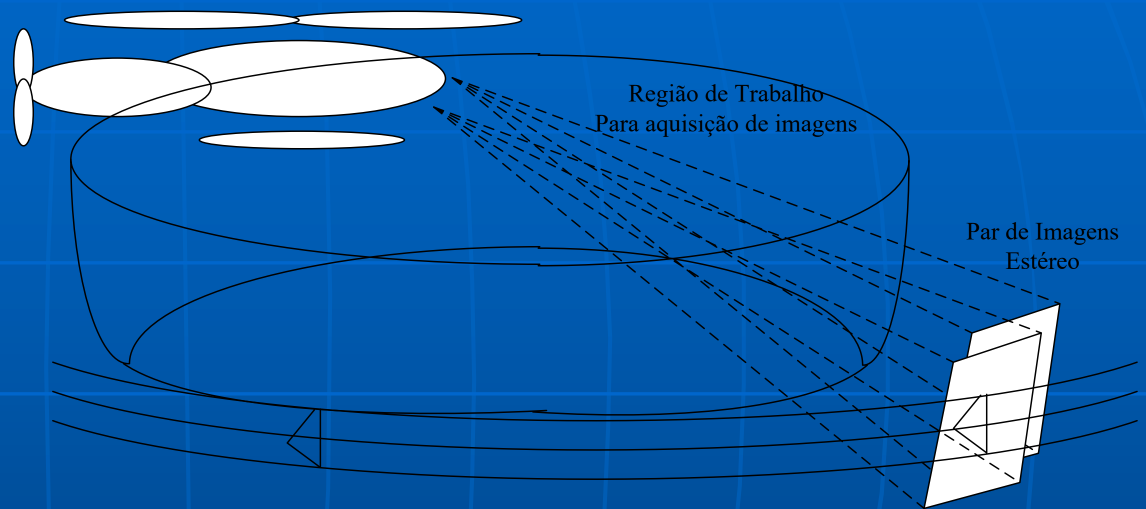
H=10 ➛
2 Misclassified images

H=12 ➛
1 Misclassified image

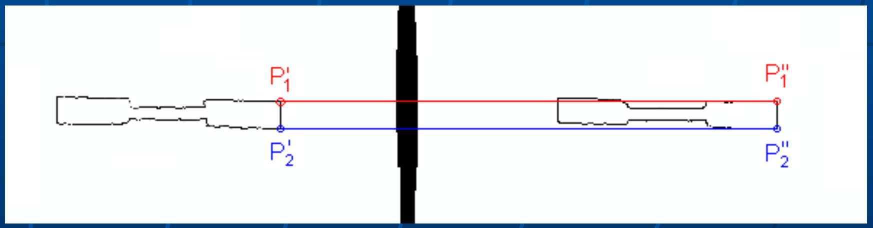
Flávio Oliveira & Alexandre Dias, 2007

Gripper inspection with 3D reconstr.

- It is not possible to train an ANN for every position/orientation in the visual field of the VANT.
- ANN trained for a fixed point of view.
- Build 3D contour model
- Reproject 3D contour to ANN point of view
- Classify with ANN

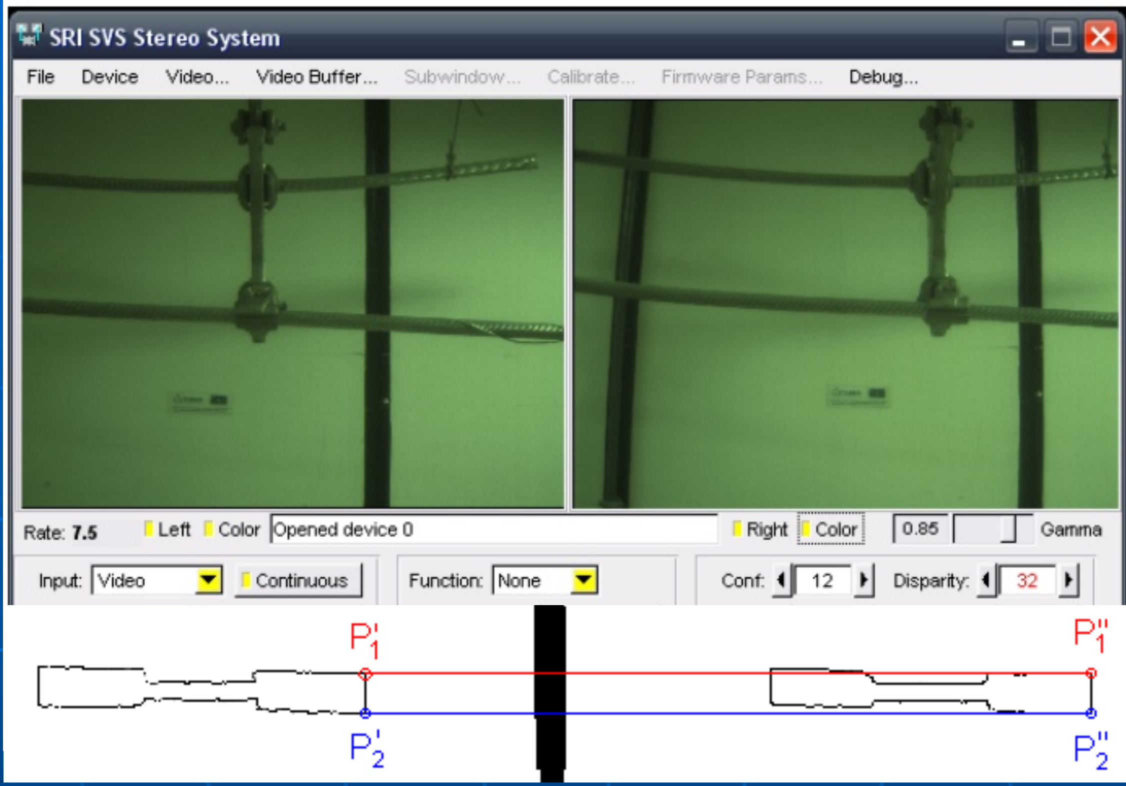


Different ROI's

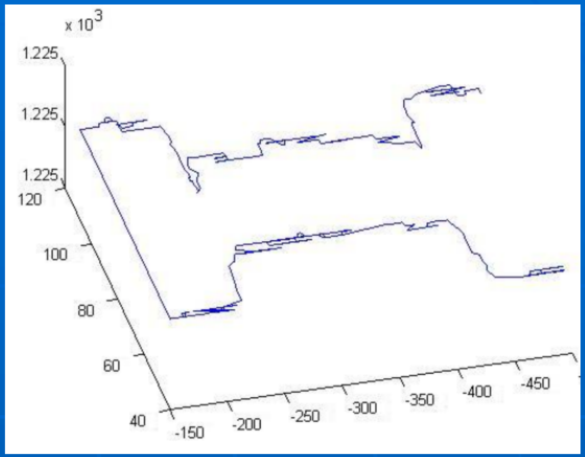


Correspondence –ROI contour in stereo pair

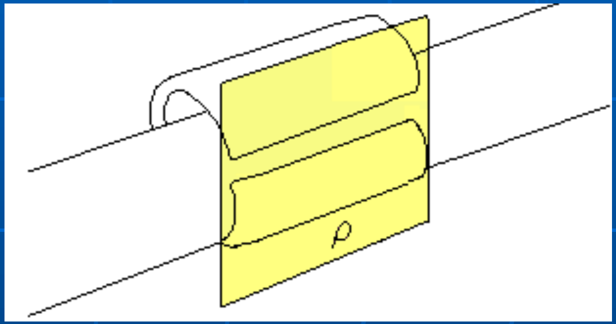
3D gripped cable



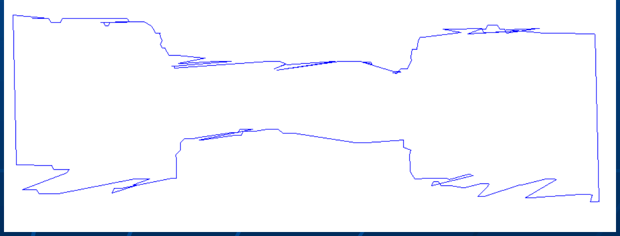
3D Reconstruction



ANN data bank Image plane



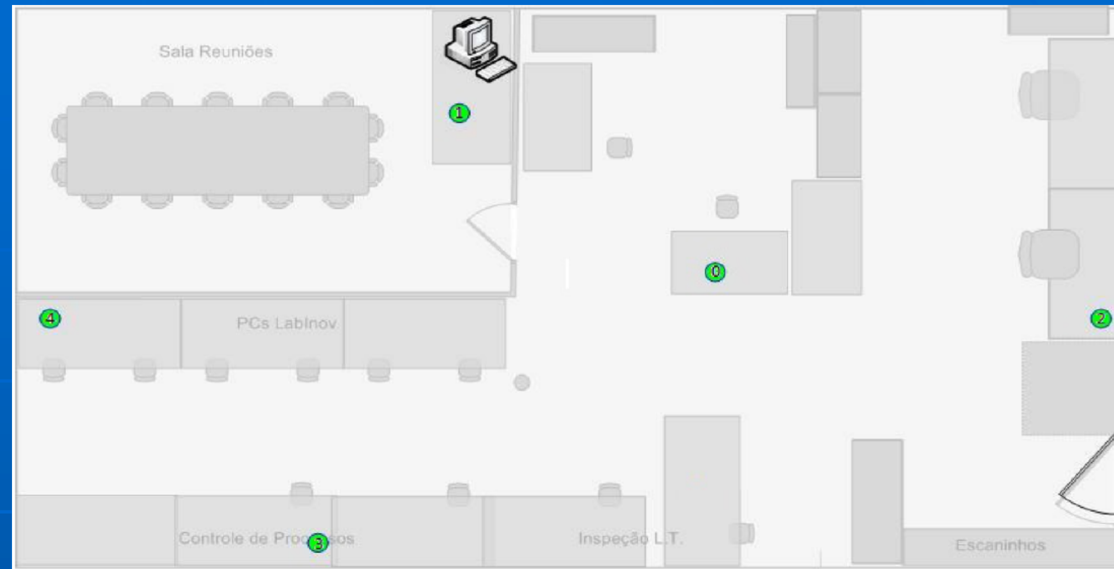
Reprojected contour for ANN

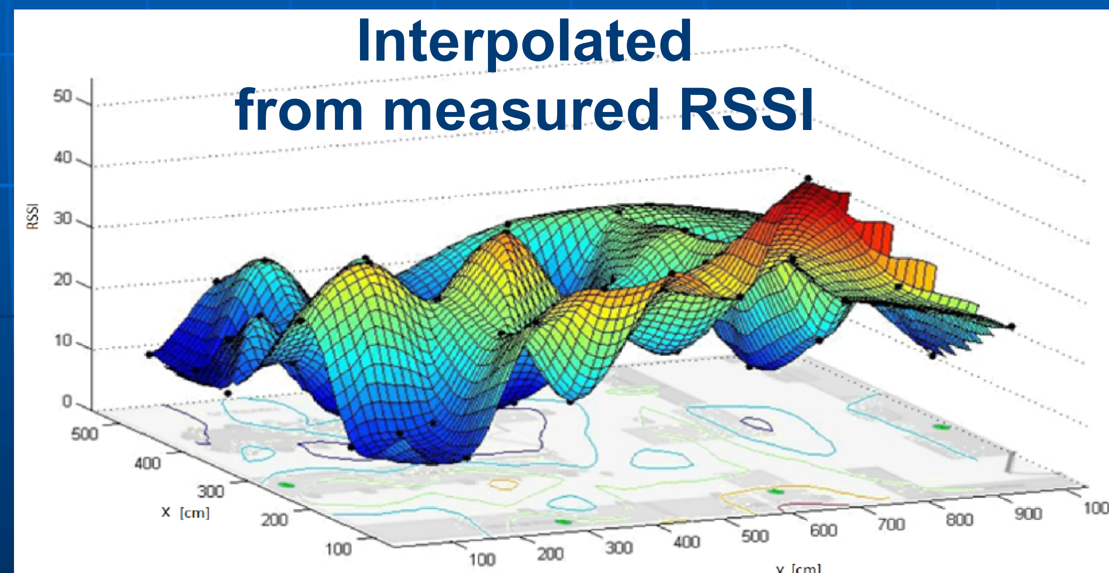


Results:
20 stereo pairs – 1 false pos., 1 false neg.
Elder Oroski, 2011

RFID occupancy identification

for thermal load estimation

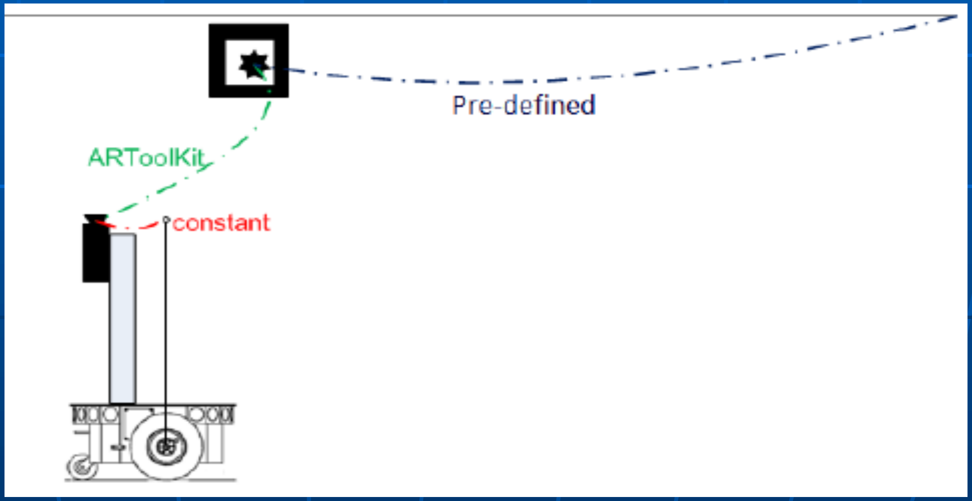




Indoor RFID Localization

in the Context of Mobile Robotics with Application in Ambient Intelligence

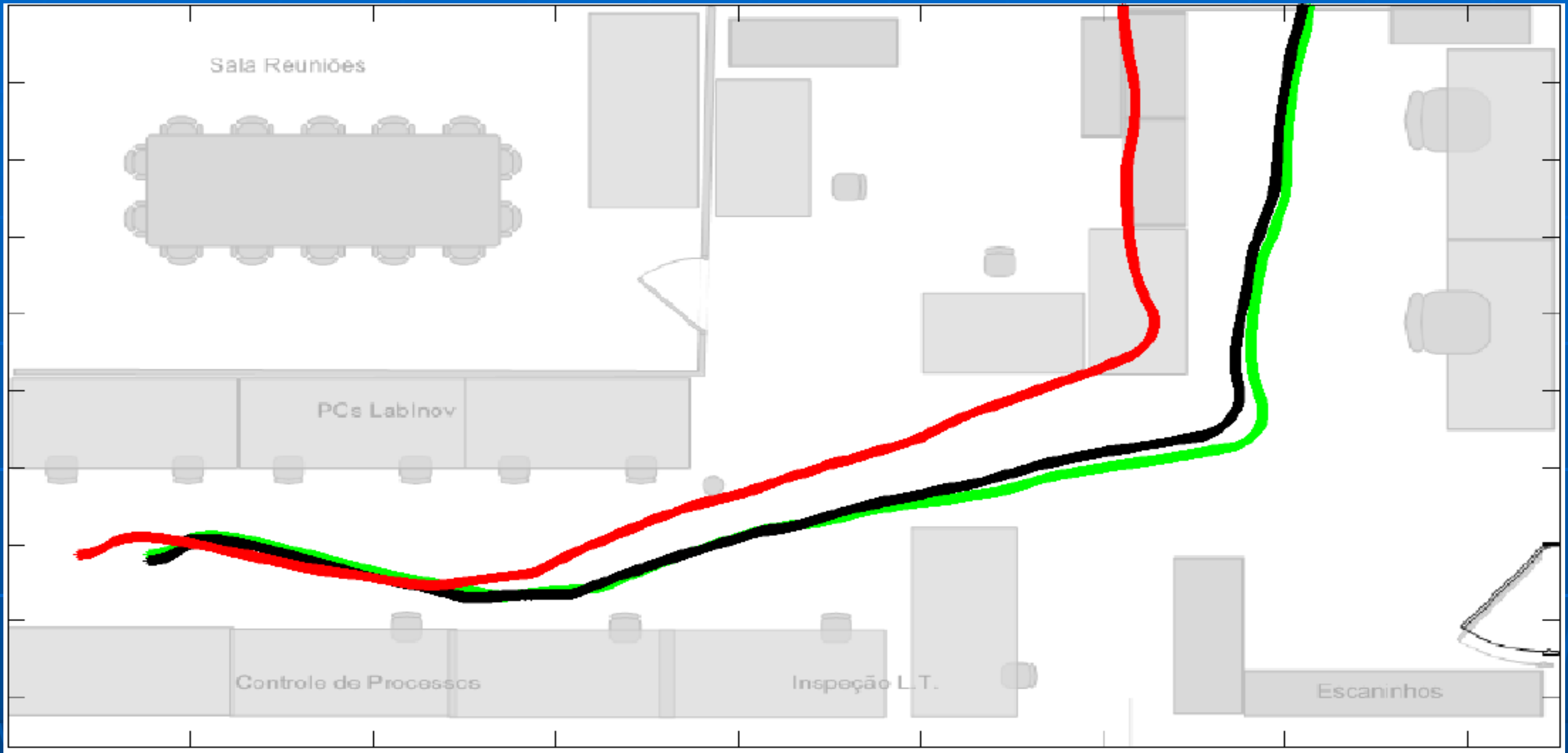
Augmented Reality Localization



(Gabriel Figueiró e André Luiz Gama, 2011)

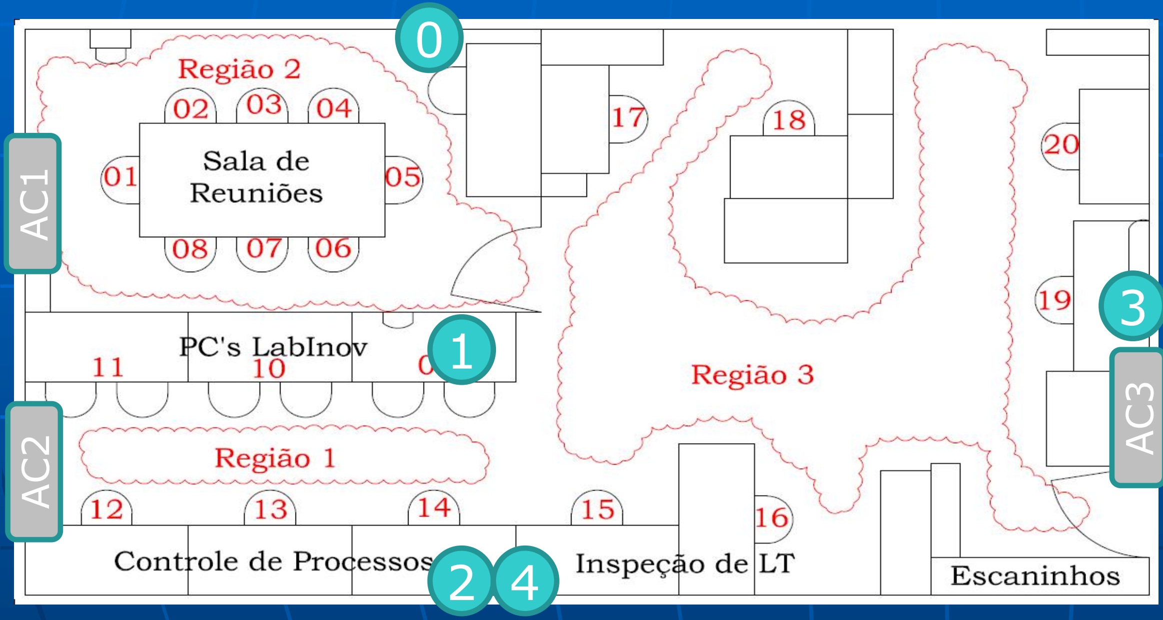
Indoor Localization

(Gabriel Figueiró e André Luiz Gama, 2011)



Red – odometry
Black – odometry + vision (augmented reality)
Green – odometry + vision + ANN RFID RSSI

RFID occupation estimation – AC zones



	a)	b)	c)
MLP10	72,2%	76,9%	**99,3%**
MLP50	74,5%	78,9%	**99,3%**
MLP100	74,8%	78,3%	**99,2%**
MLP150	74,0%	77,9%	**99,3%**
MLP200	73,8%	76,9%	**99,1%**
LVQ10	65,5%	73,5%	**97,3%**
LVQ100	65,4%	74,8%	**96,3%**
LVQ200	65,4%	74,6%	**96,1%**
SVM - RBF	76,3%	82,1%	**97,6%**
SVM – Lin.	**44,6%**	**73,1%**	**92,5%**

a) Raw RSSI (missing RSSI ->0)
b) RSSI hold last valid RSSI
c) EKF

(Cristovam Silva Jr., 2012)

LARA/SG11/UnB

01-20 Data collecting points

RFID Reader

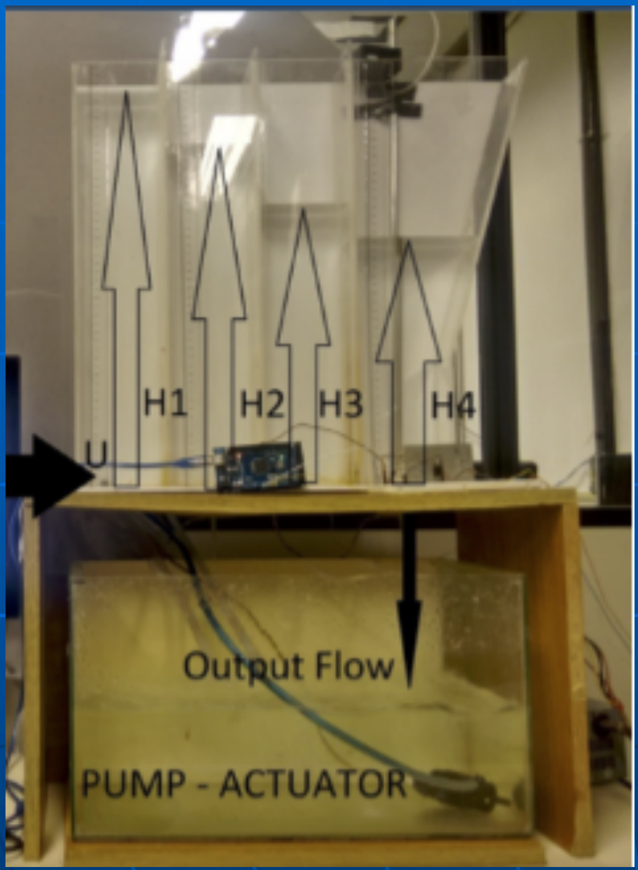
AC Air Conditioner

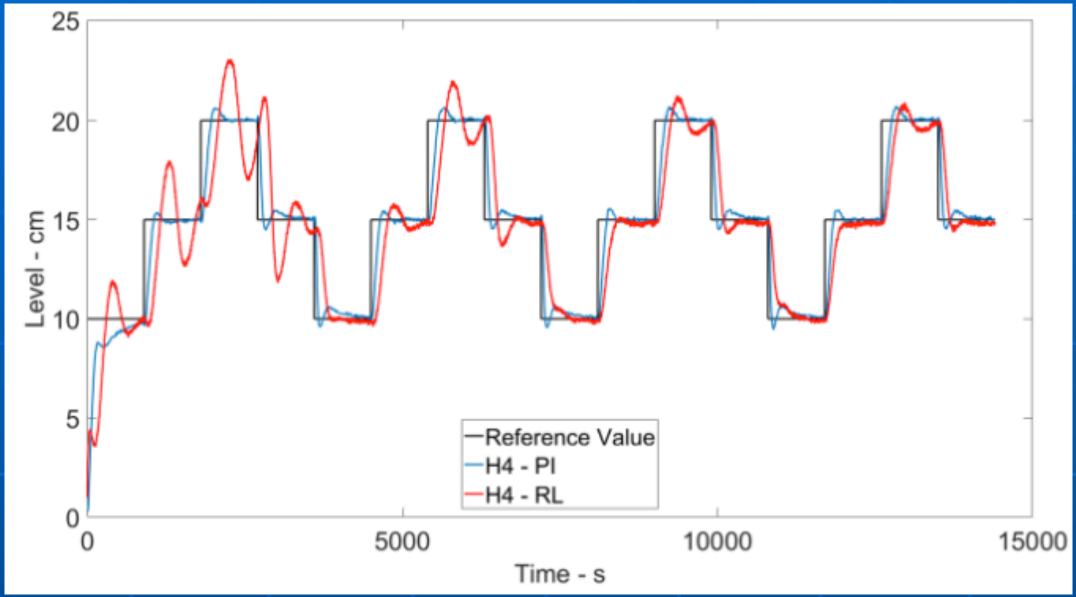
RFID occupation Id. – HVAC areas

	RSSI (some missing)	RSSI (if miss, use last)	RSSI (if miss EKF)
MLP10	72,2%	76,9%	**99,3 %**
MLP50	74,5%	78,9%	**99,3 %**
MLP100	74,8%	78,3%	**99,2 %**
MLP150	74,0%	77,9%	**99,3 %**
MLP200	73,8%	76,9%	**99,1 %**
LVQ10	65,5%	73,5%	**97,3 %**
LVQ100	65,4%	74,8%	**96,3 %**
LVQ200	65,4%	74,6%	**96,1 %**
SVM - RBF	76,3%	82,1%	**97,6 %**
SVM - Linear	**44,6%**	**73,1%**	**92,5 %**

(Cristovam Silva Jr., 2012)

RL: 4th order liquid leve Process





Actor-Critic Q-Learning

LARA/UnB, Lucas Matos, 2018

Universal Approximation Theorem

Considerando o Teorema da Aproximação Universal, abaixo, proposto inicialmente por Cybenko em 1989, para funções sigmoidais e expandido em 2017 por Lu et al. Estes mostraram que todas as funções integráveis no sentido de Lebesgue podem ser aproximadas por redes ReLU (Rectified Linear Unit).

Qual a necessidade de se utilizar Redes Neurais Profundas?

Teorema da Aproximação Universal

Seja $\varphi(.)$ uma função não constante, com limites, e numa função monotônica contínua positiva. Seja I_m uma unidade com dimensões $m[0,1]^m$. O espaço de funções contínuas de I_m escreve-se $C(I_m)$. Assim, para cada $\varepsilon > 0$ e cada função $f \in C(I_m)$, há uma integral N, com constantes reais $v_i, b_i \in \Re$ e vetores $w_i \in \Re^m$, onde $i = 1, ..., N$, tal que se possa definir:

$$F(x) = \sum_{i=1}^{N} v_i \varphi(w_i^T x + b_i), \tag{1}$$

como uma realização aproximada da função f onde f independe de φ; ou seja,

$$|F(x) - f(x)| < \varepsilon, \tag{2}$$

para todo $x \in I_m$. Em suma, funçõesdo tipo $F(x)$ são densas em $C(I_m)$.

Classic Dynamic Programming (DP)

Start from last step & move backward

Complexity of

Naïve search $O(|A|^n)$

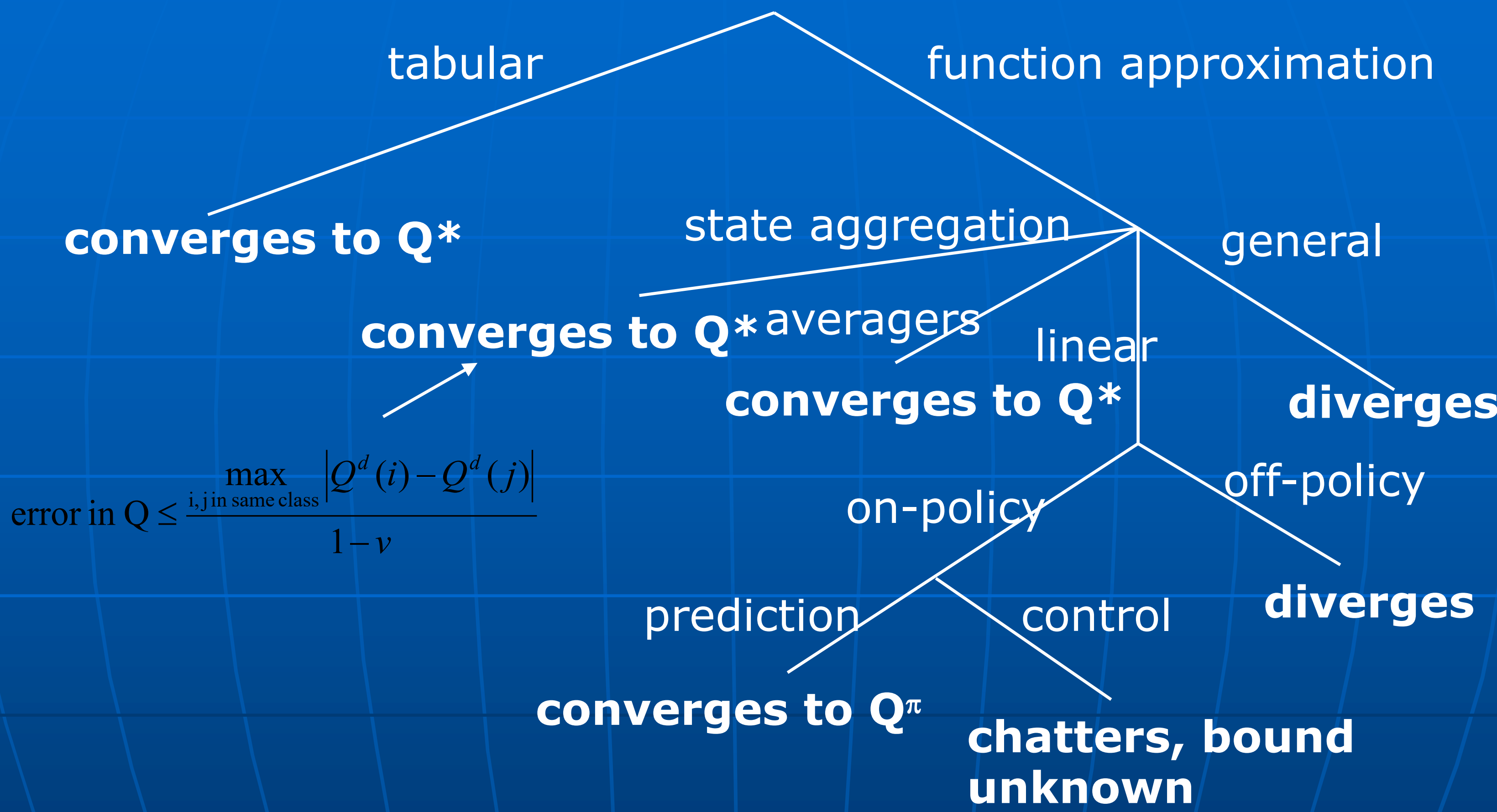
Actions per step

DP $O(n|A||S|)$

possible states

Problem: $n=\infty$ if loops or otherwise infinite horizon

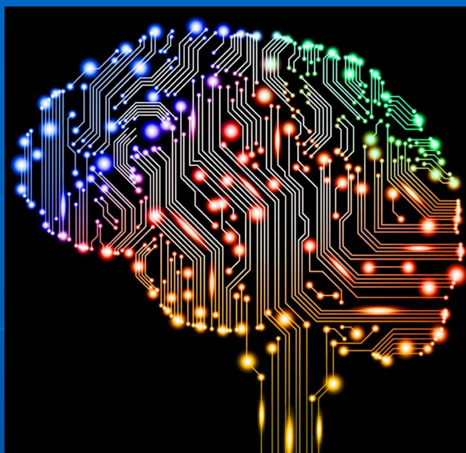
Convergence results of Q-learning



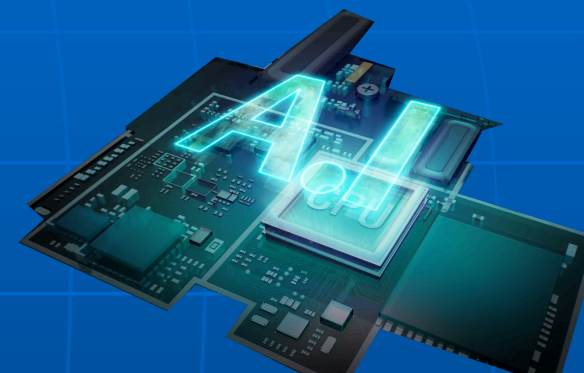
$$\text{error in Q} \leq \frac{\max\limits_{i,j \text{ in same class}} \left| Q^d(i) - Q^d(j) \right|}{1 - v}$$

Perspectives

Top 5 Trends of AI 2019



1. Deep learning
2. Facial Recognition
3. Privacy and Policy
4. AI-Enabled Chips
5. Role of Cloud in Artificial Intelligence Applications



Obs: Anyone can give a guess, make another list!!

Adapted from: https://hackernoon.com/top-5-trends-of-artificial-intelligence-ai-2019-693f7a5a0f7b

paperswithcode/sota

https://paperswithcode.com/sota

Search for papers, code and tasks

Browse state-of-the-art | Follow | Discuss | About | Log In/Register

Browse state-of-the-art

921 leaderboards • 1145 tasks • 968 datasets • 12701 papers with code

Follow on Twitter for updates

Computer Vision

Semantic Segmentation
17 leaderboards
434 papers with code

Image Classification
47 leaderboards
384 papers with code

Object Detection
36 leaderboards
312 papers with code

Image Generation
31 leaderboards
165 papers with code

Denoising
13 leaderboards
153 papers with code

See all 614 tasks

Natural Language Processing

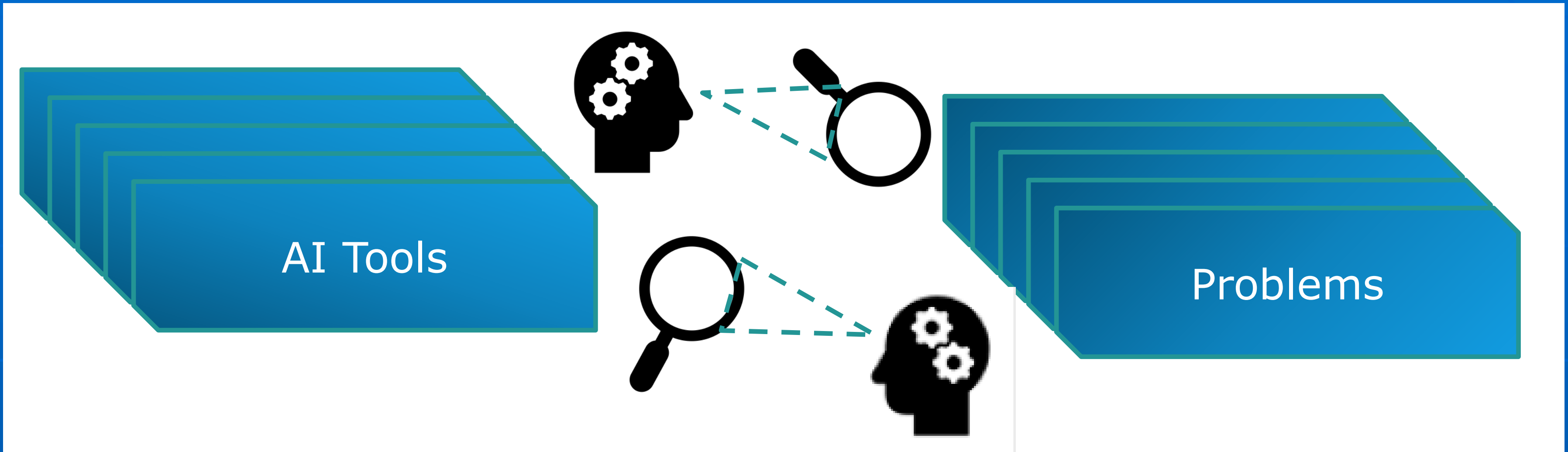
Machine Translation

Language Modelling

Question Answering

Sentiment Analysis

Information Retrieval



Thank You!

Adolfo Bauchspiess

www.ene.unb.br/adolfo

adolfobs@ene.unb.br