Artificial Intelligence for Oil and Gas Without the Hype

Antônio de Pádua Pires

Cesar Calderon

Luiz Eduardo Queiroz

Pedro Pesce

Thiago Toribio

(Petrobras)

ice '25 RIO
AAPG International Conference & Exhibition

Co-hosted by
ABGP
ASSOCIAÇÃO BRASILEIRA DE GEÓLOGOS DO PETRÓLEO

The heart has its reasons which reason knows nothing of

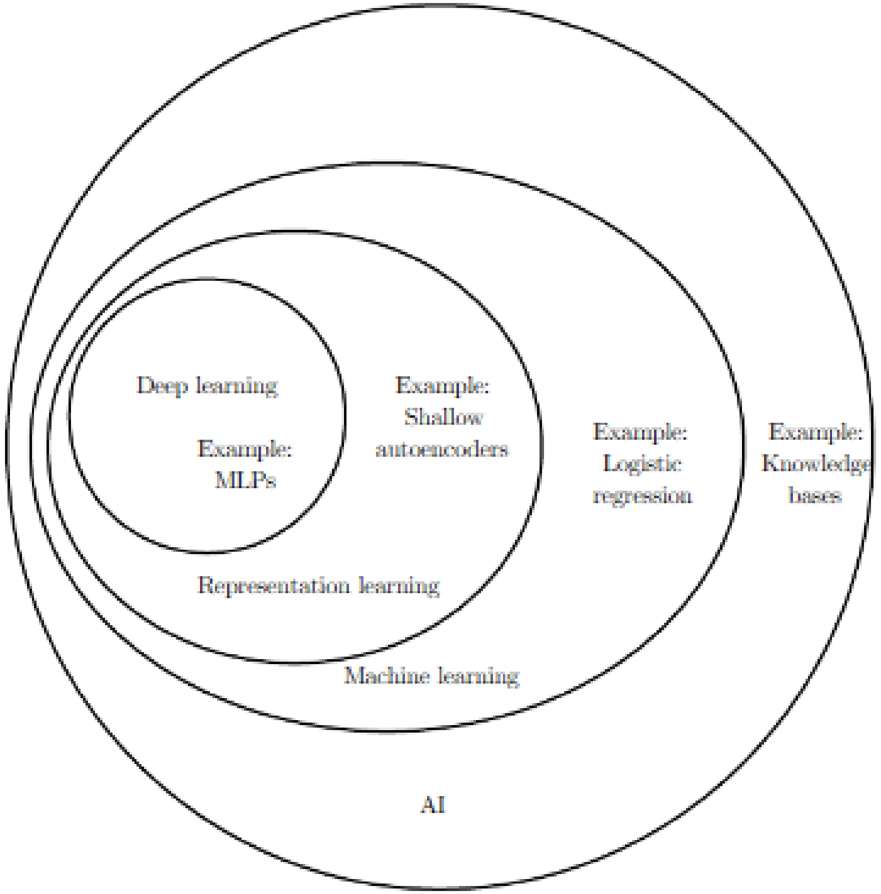
Blaise Pascal, *Pensées*, Section 4, 1669

What are we talking about? An incipient glossary...

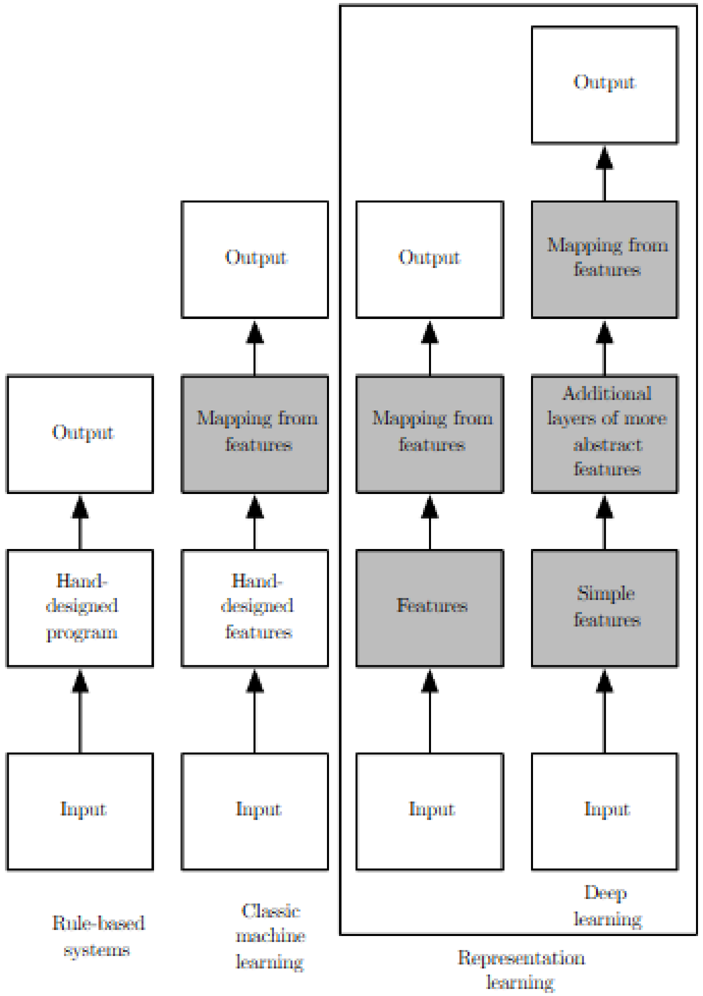
Intelligence
Artificial
Learning
Machine
Network
Neuron
Layer
Deep

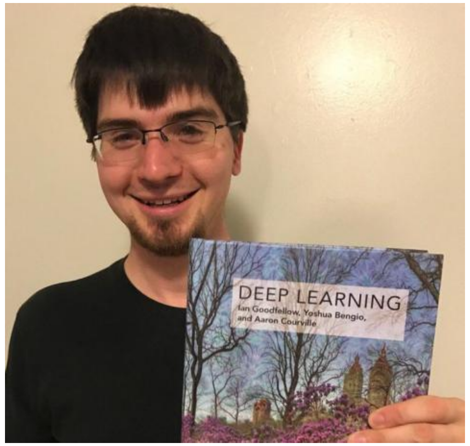
- Artificial intelligence (AI)
  - Hand designed algorithms
- Machine Learning (ML)
  - Linear regression
  - K-means
- Deep learning
  - Multilayer Perceptron

What are we talking about? A taxonomy and its elements...

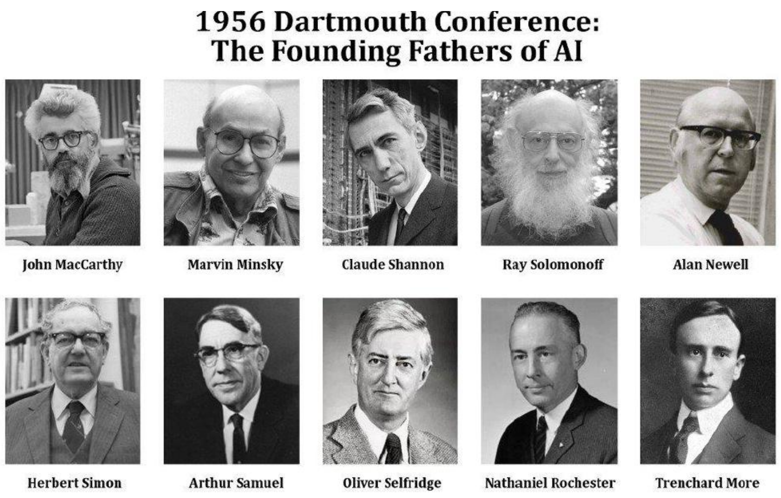


Deep Learning





Dartmouth Conference (1956)



A Proposal for the

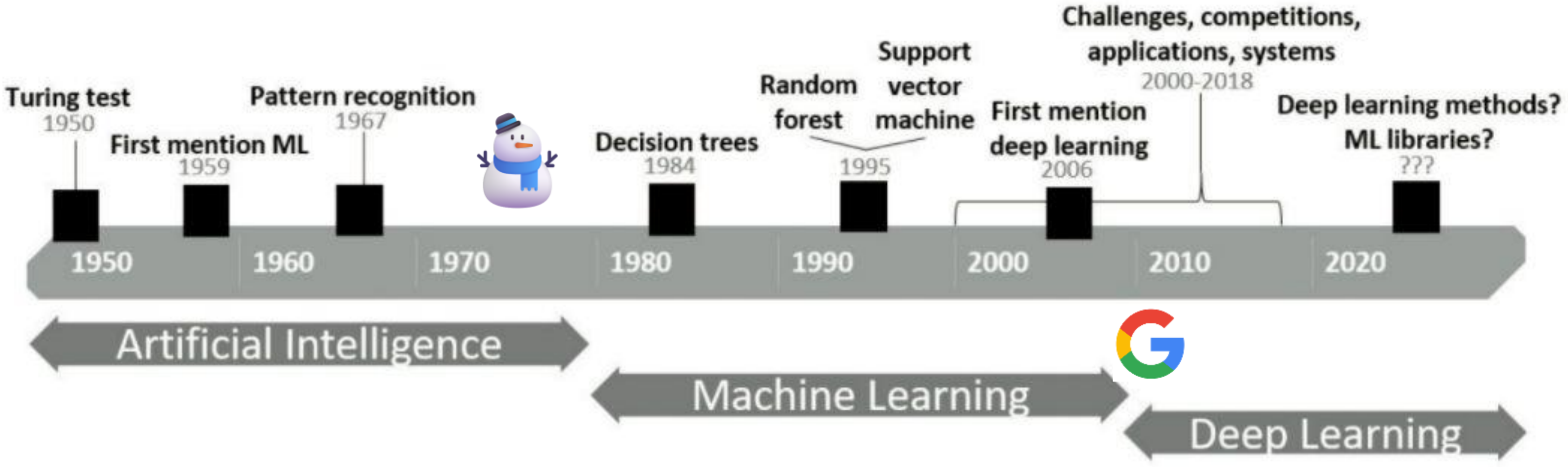
DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

June 17 - Aug. 16

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

Winters and springs...

- Boom in the 60´s, big funding and hype (again in the 90´s, 2010´s, 2020´s...)
  - “Machines will be capable, within twenty years, of doing any work a man can do” – Herbert Simon (1965).
  - “Within a generation, (…) the problems of creating ‘artificial intelligence’ will be substantially solved”, Marvin Minsky (1967).
- AI Winter (funding cut in the 70´s):
  - General reasons and technical excuses
  - Change in focus: *weak AI* – specific tasks



Some “historical facts”

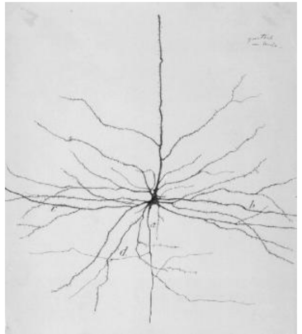
- 1958: Mark I Perceptron (IBM) - Frank Rosenblatt, psychologist at Cornell University
  - Biological/psychological (behaviorism) inspiration: the neuron
  - Capable of learning: “like humans, perceptron will make mistakes at first, but it will grow wiser as it gains experience”

Psychological Review
Vol. 65, No. 6, 1958

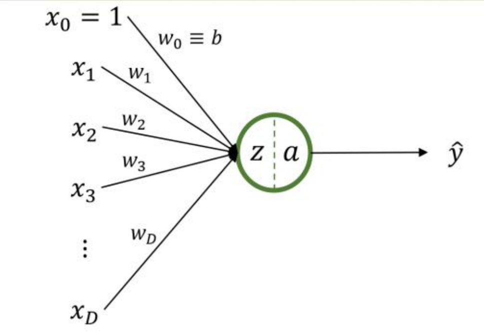
THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN[1]

F. ROSENBLATT

Cornell Aeronautical Laboratory



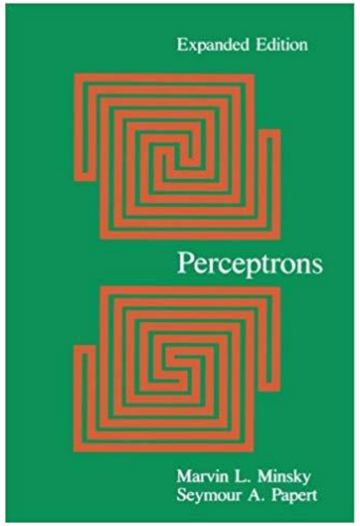
A neuron

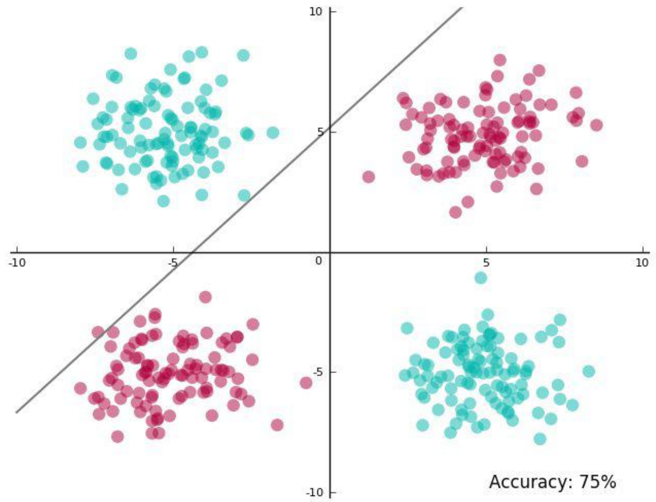


A perceptron

Some more “historical facts”

- Marvin Minsky and Seymour Papert, Perceptrons, 1969:
  - Mathematical foundations on the research about perceptron
  - Problems on XOR: perfectly separable but not linearly separable
  - Solved by the stacking of perceptrons – first glance at deep learning – Change in focus: *weak AI* – specific tasks





A little more “historical facts” - epilogue

- The rebirth of neural networks with *weak AI* approach in the 80´s
  - John Hopfield, *Neural networks and physical systems with emergent collective computational abilities*, PNAS 79 (8): 2554-2558, 1982.
  - Backpropagation training: David Rumelhart, Geoffrey Hinton e Ronald Williams, *Learning representations by back-propagating errors*, Nature 323, 533-536 (1986).

PRESS RELEASE

8 October 2024

The Nobel Prize in Physics 2024

The Royal Swedish Academy of Sciences has decided to award the Nobel Prize in Physics 2024 to

John J. Hopfield
Princeton University, NJ, USA

Geoffrey Hinton
University of Toronto, Canada

“for foundational discoveries and inventions that enable machine learning with artificial neural networks”

A theorem: the universal approximator

Neural Networks, Vol. 2, pp. 359–366, 1989
0893-6080/89 $3.00 + .00

ORIGINAL CONTRIBUTION

Multilayer Feedforward Networks are Universal Approximators

KURT HORNIK
Technische Universität Wien

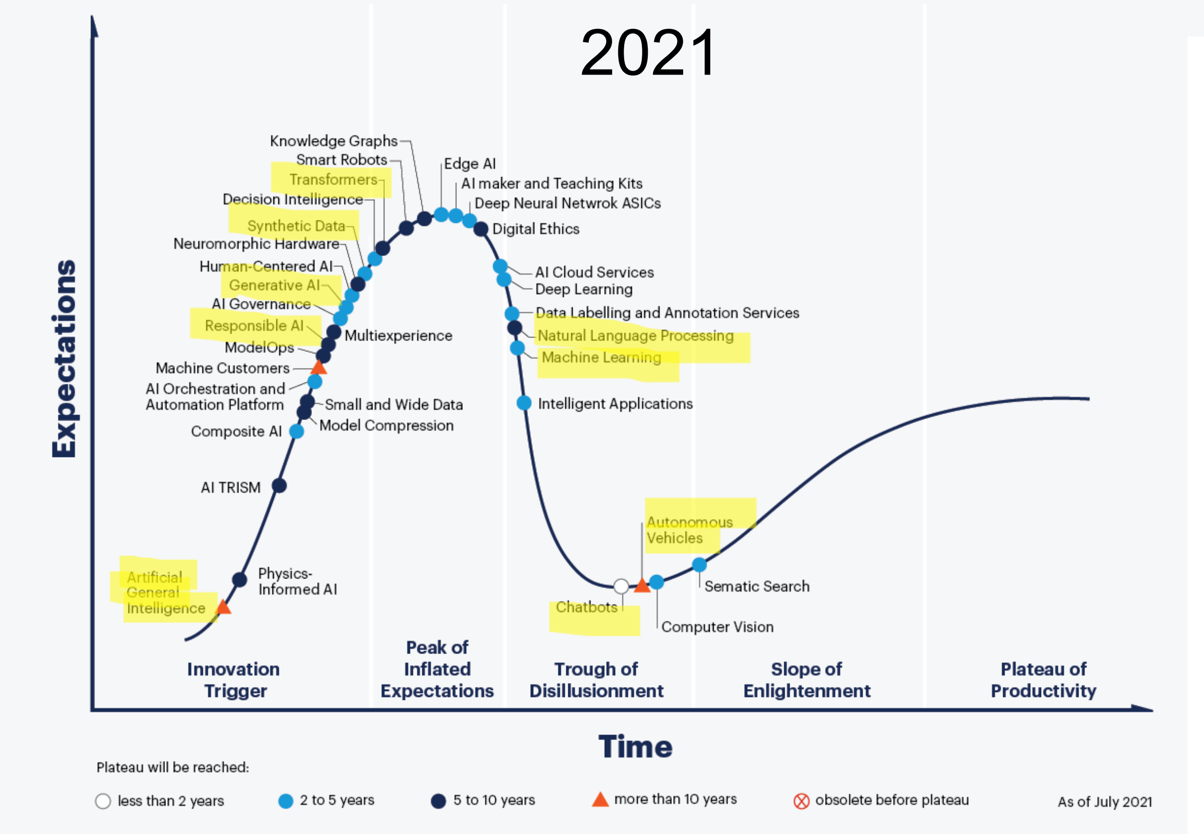
MAXWELL STINCHCOMBE AND HALBERT WHITE
University of California, San Diego

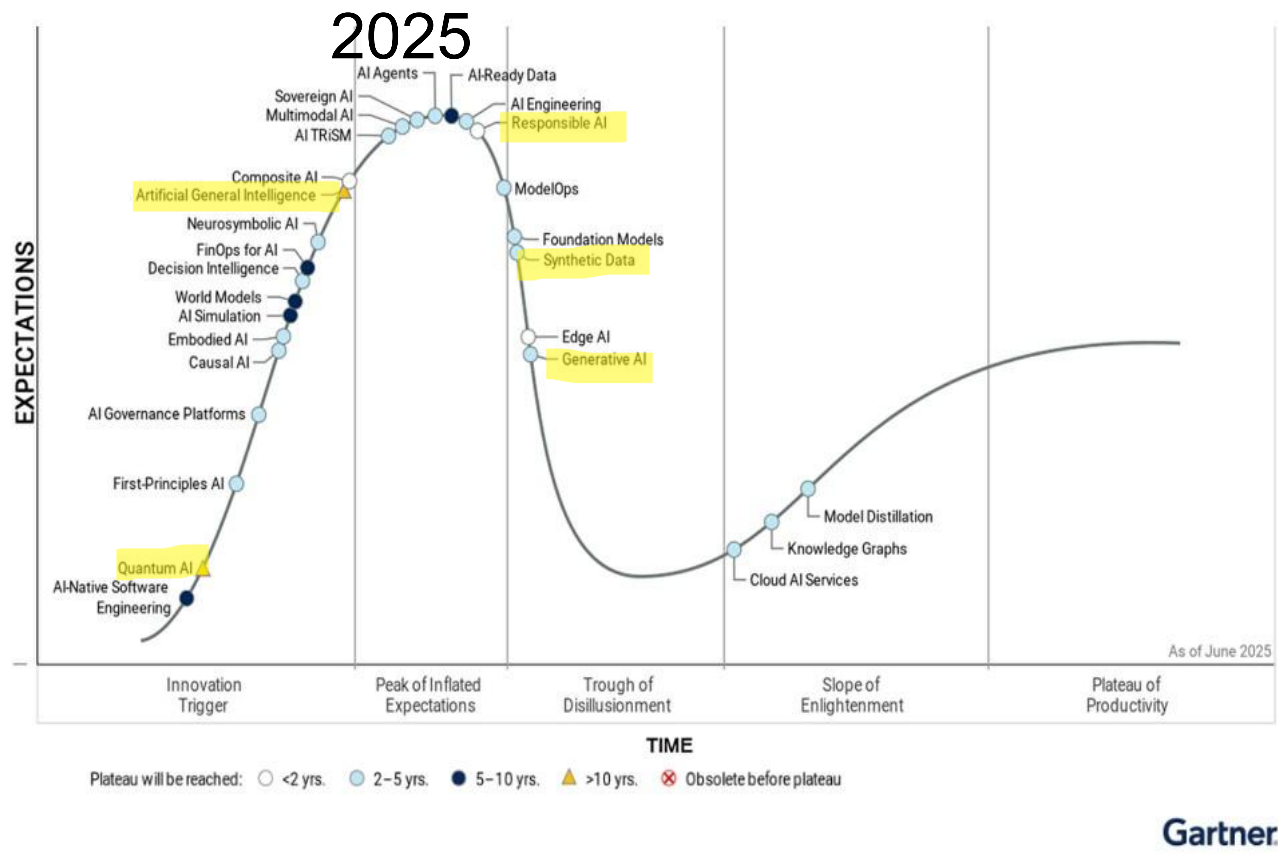
(Received 16 September 1988; revised and accepted 9 March 1989)

Abstract:

This paper rigorously establishes that standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available. In this sense, multilayer feedforward networks are a class of universal approximators.

Without the hype?





https://www.gartner.com/en/articles/the-4-trends-that-prevail-on-the-gartner-hype-cycle-for-ai-2021

Gartner Hype Cycle Identifies Top AI Innovations in 2025

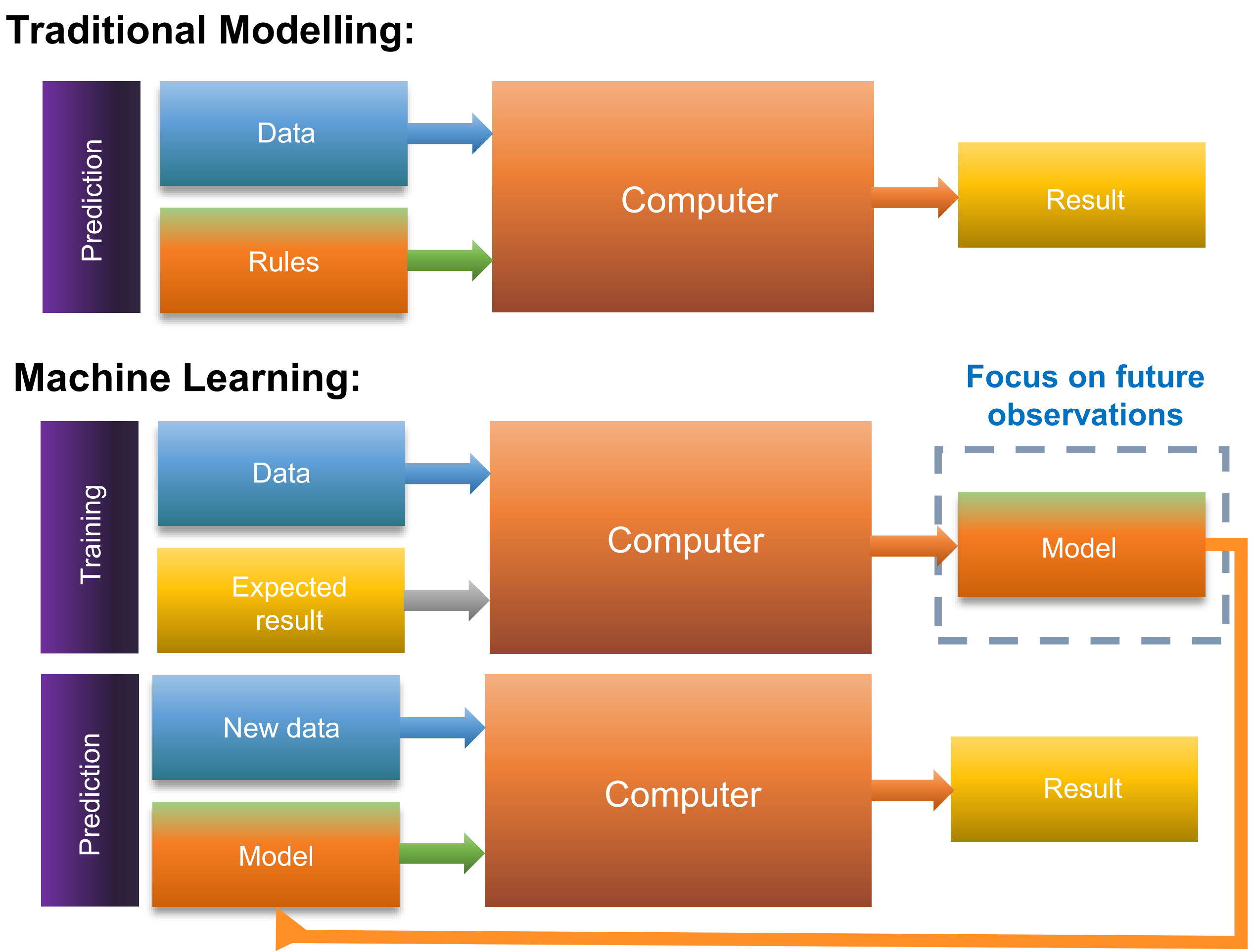
Why “now”? World and geosciences

- “Increasing datasize”
  - Digital surveillance, tracking and data collection, including unstructured data (natural text, images, videos). Huge amounts of publicly available data and data obtained on the internet (GAFAM and other “big tech” companies)
  - Seismic, well log, image and unstructured text data
- “Increasing modelsize”
  - Gigantic data centers with massive use of parallel processing (GPU), in O&G and seismic processing companies as well as in “big tech” companies
- “Increasing Accuracy, Complexity and **Real-World Impact**”
  - Big profits in “real world” data and model transactions, through accuracy, logistic and distribution optimization and living labor savings
  - Big profits for O&G companies?

Google Amazon Facebook Apple Microsoft

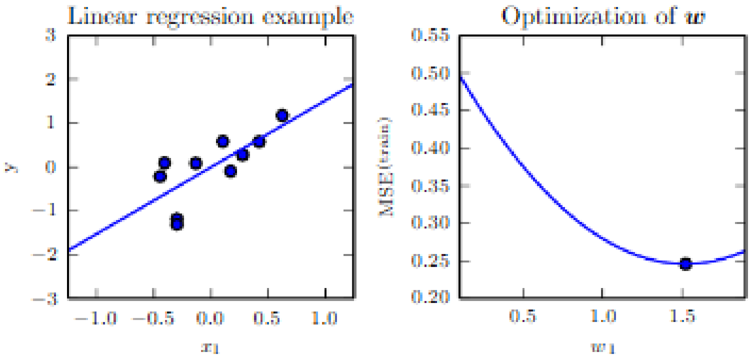
NVIDIA

Machine learning strategy



Machine learning algorithms: *E, T, P*

- Learn from data – *data-driven*.
- Mitchell, 1997 (Apud Goodfellow, 2016):
  - “A computer program is said to learn from **experience *E***
  - with respect to some **class of tasks *T***
  - and **performance measure *P***,
  - if its performance at tasks in ***T***, as measured by ***P***, improves with experience ***E***.”
- Example: linear regression
  - **Experience *E***: dataset y_measured (x_measured) – supervised learning
  - **Task *T***: predict y from x (regression)
  - **Performance measure *P***: squared error

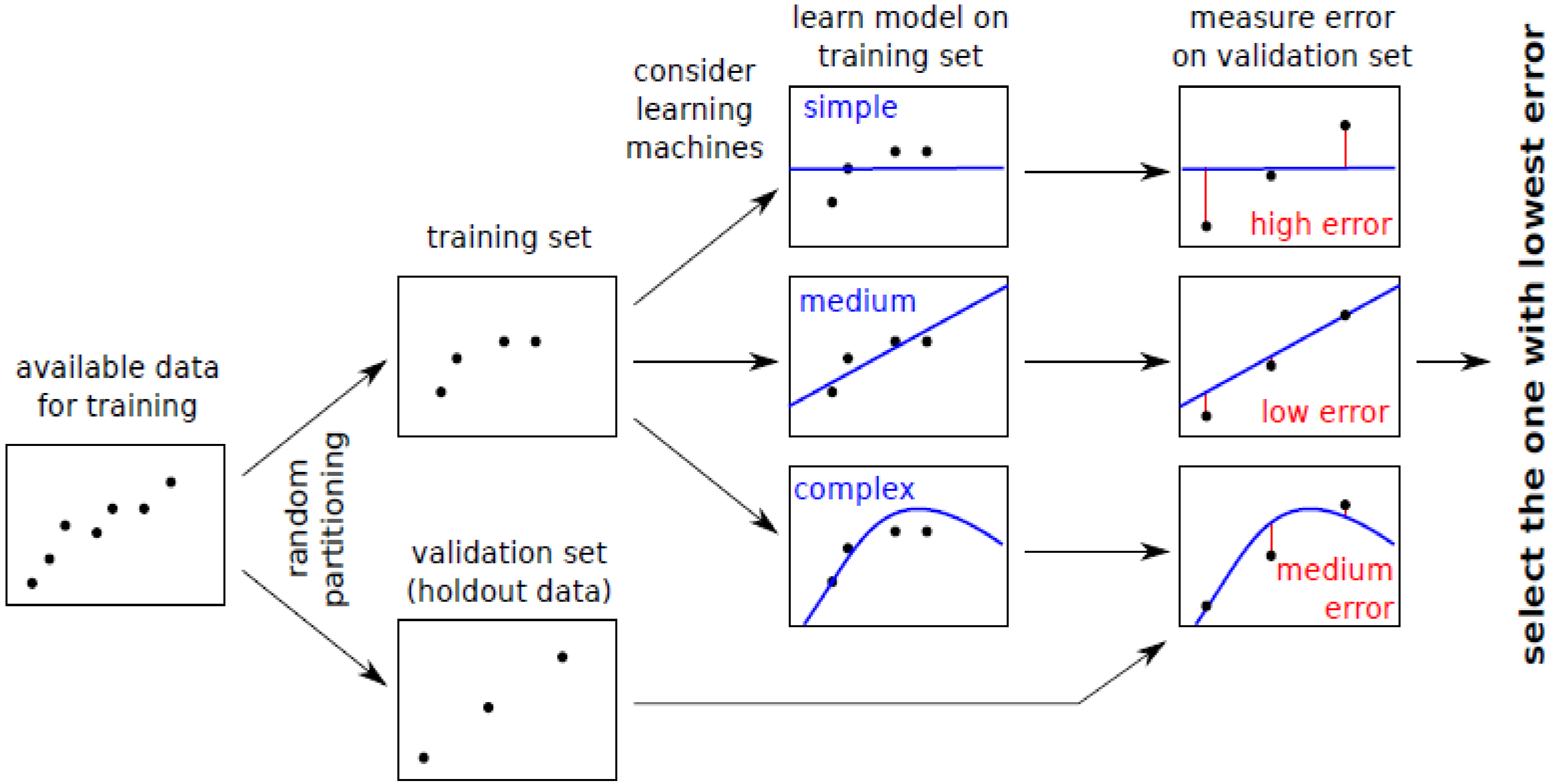


[Deep Learning](#)

The experience *E* and the task *T*

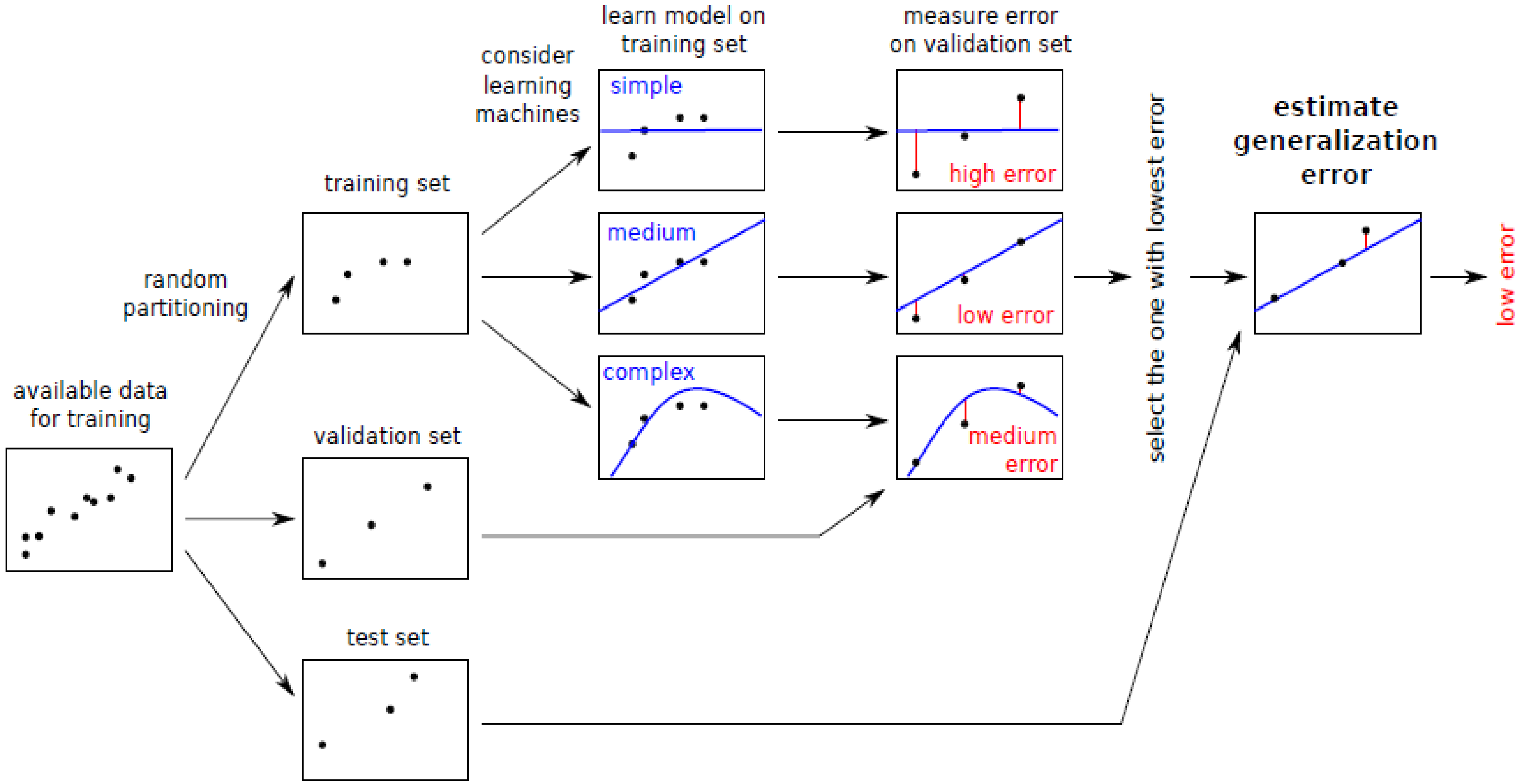
- Supervised learning (**Experience *E***)
  - Classification (**Task *T***)
  - Regression (**Task *T***)
- Unsupervised learning (**Experience *E***)
  - Clustering (**Task *T***)
  - Dimensionality reduction (**Task *T***)
- Reinforcement learning (**Experience *E*** and **Task *T***?)
- Semi-supervised learning (**Experience *E*** and **Task *T***?)

Performance measure *P*: Train, test, validation



Bishop, *Machine Learning and Pattern Recognition*

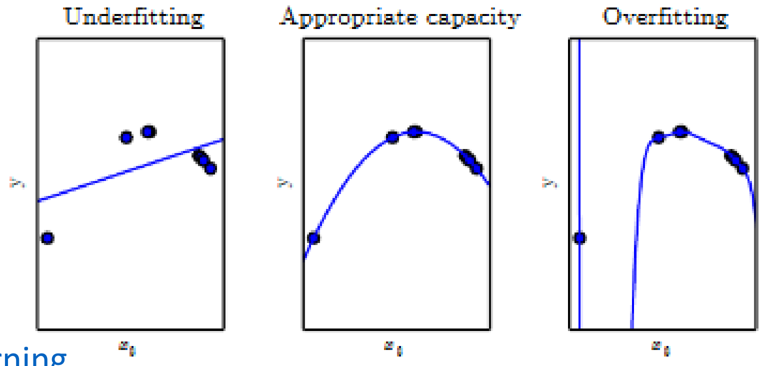
Performance measure *P*: Train, test, validation

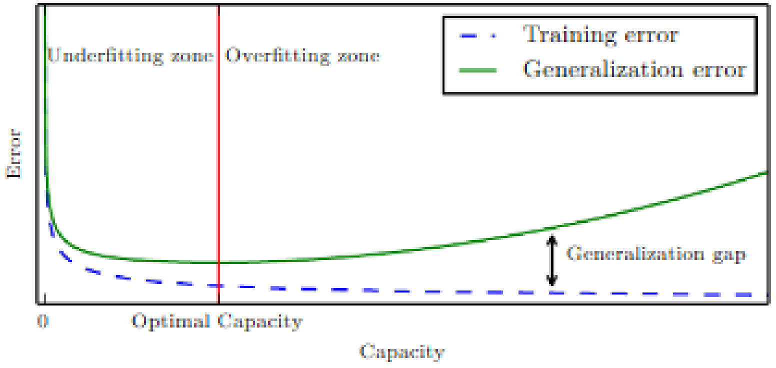


Bishop, *Machine Learning and Pattern Recognition*

Performance measure *P*: *capacity, overfitting, underfitting*

- Warning – Goodhart´s law: “When a measure becomes a target, it ceases to be a good measure”. The right task might not be (simply) to improve the performance.
- Train/test, statistics care (representativity), *overfitting, underfitting*.
- Goodfellow: “The central challenge in machine learning is that our algorithm must perform well on [...] previously **unseen inputs** [...]. The ability to perform well on previously **unobserved inputs** is called **generalization**. [...] What separates machine learning from optimization is that we want the **generalization error**, also called the **test error**, to be low as well”.

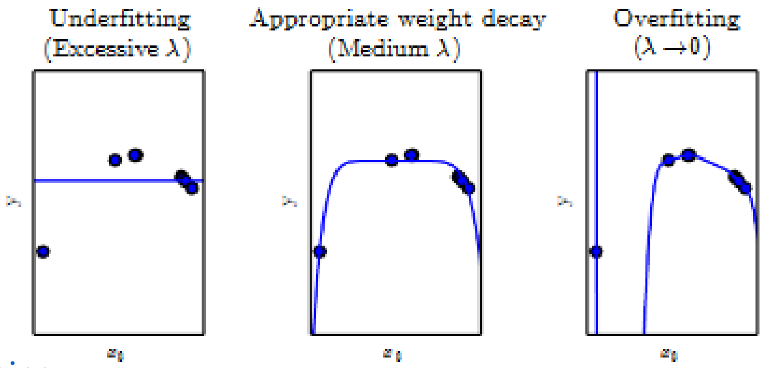


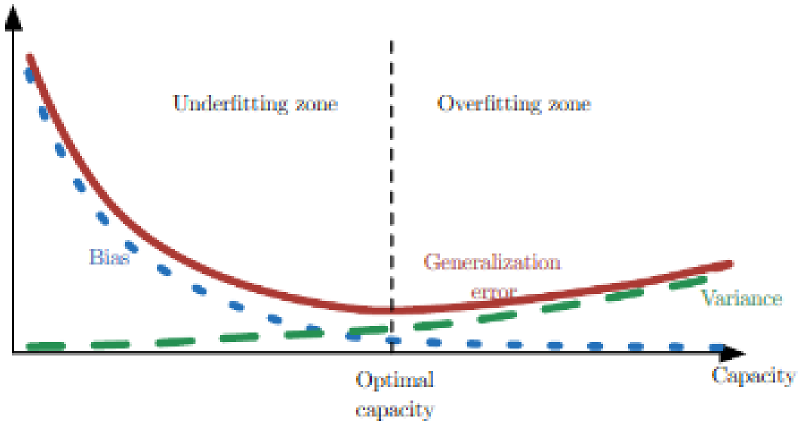


Deep Learning

Regularization, validation, *bias-variance trade-off*

- Regularization: reducing the possibilities of weight choices to reduce the generalization error.
- Weighting the regularization – choice, and not optimization, of hyperparameter.
- Introduce *bias*, reduce *variance*.
- *Bias-variance trade-off* (“*big boy math (statistics)*” – C.H.).
- Cross-validation strategy

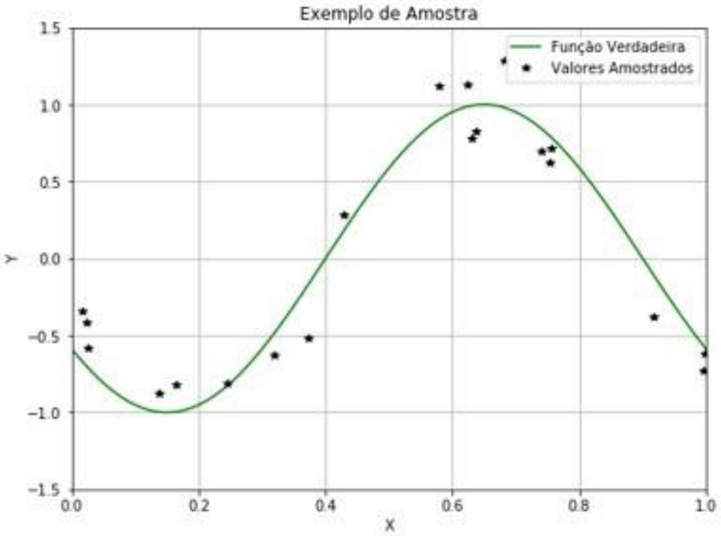


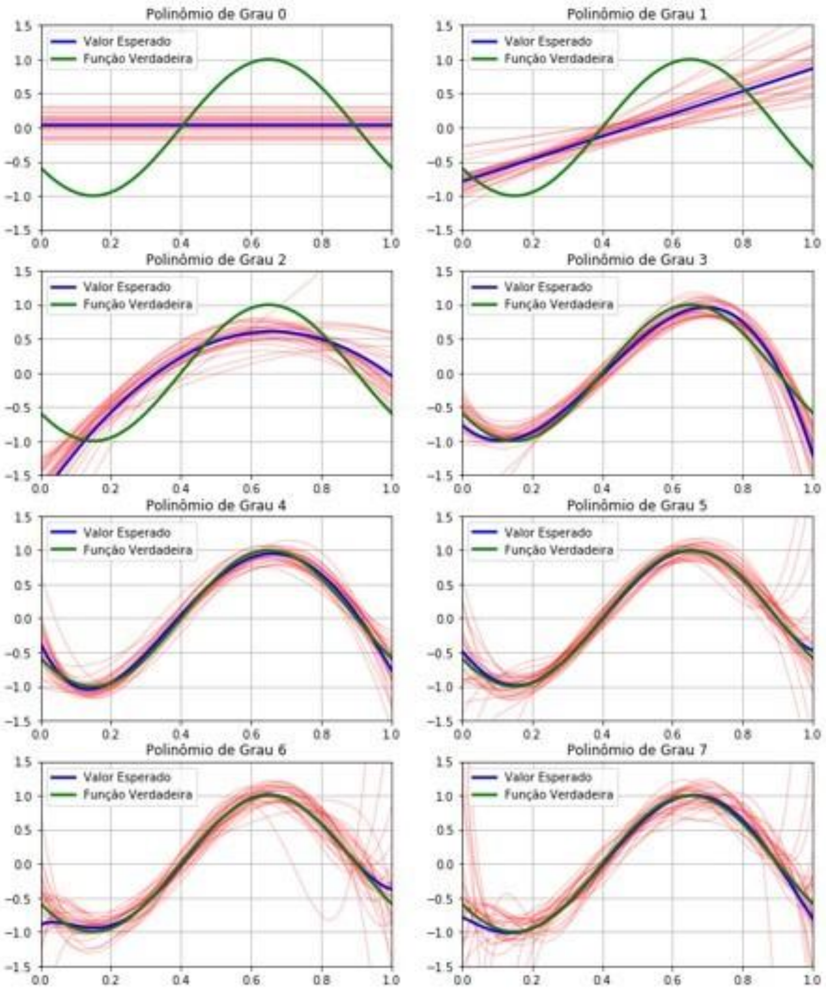


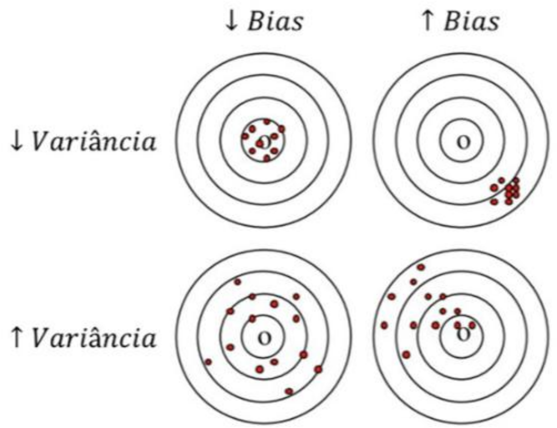
[Deep Learning]

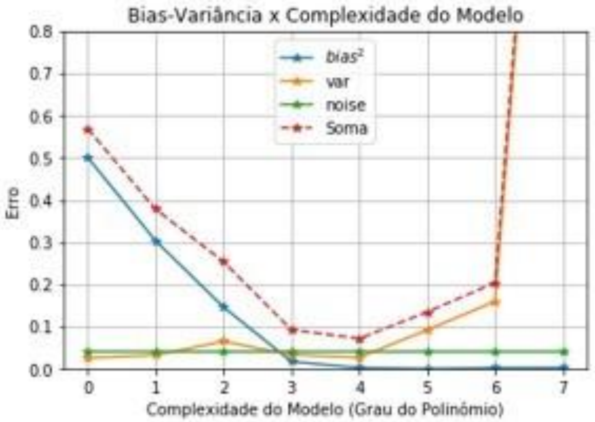
Bias-variance trade-off

Example of over and underfitting with polynomial regression. What is the best degree of the polynomial regression in this case?



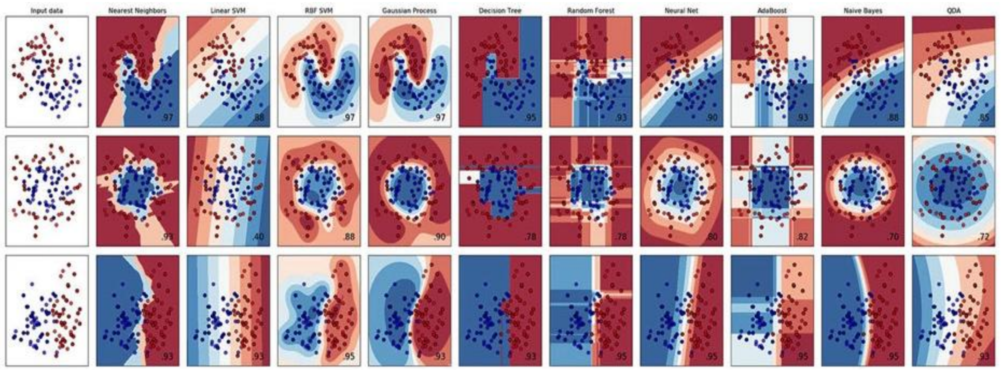




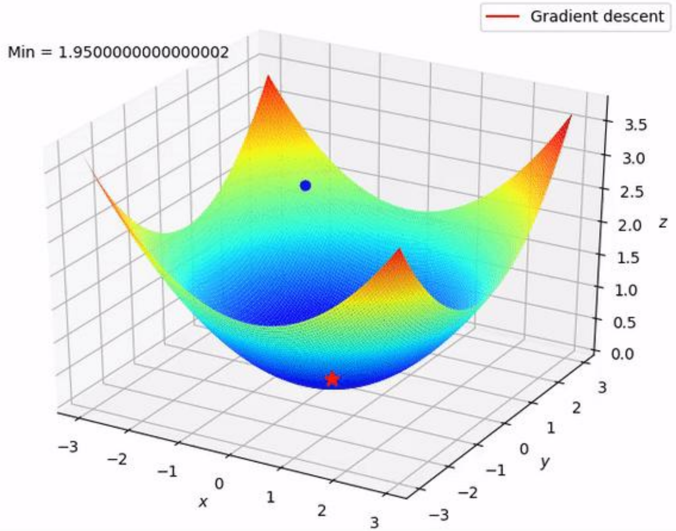


Machine learning – some further concepts

- **Representation**: re-organizing knowledge
  - Imitation (Aquino);
  - Manifestation of knowledge (Kant);
  - A way of placing knowledge in the model



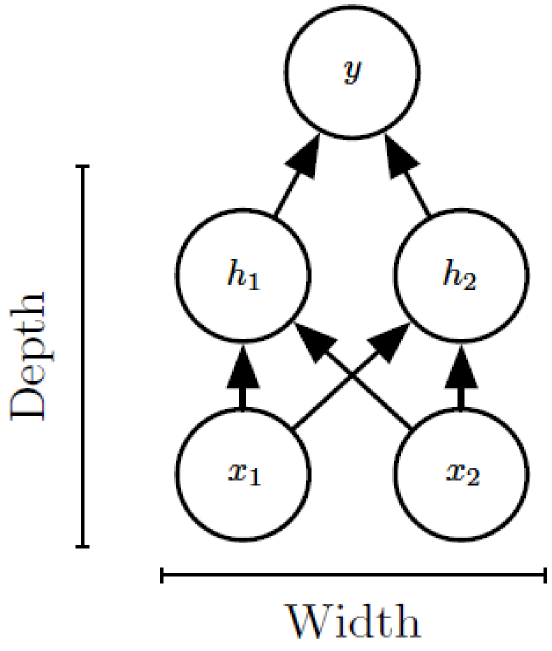
- **Optimization**: method to obtain the optimal of a function (in the case, the *loss function*)
  - In the optimization stage the learning *effectively* happens
  - Example: Using the gradient descent method to optimize a non-linear least squares loss function through back-propagation calculations

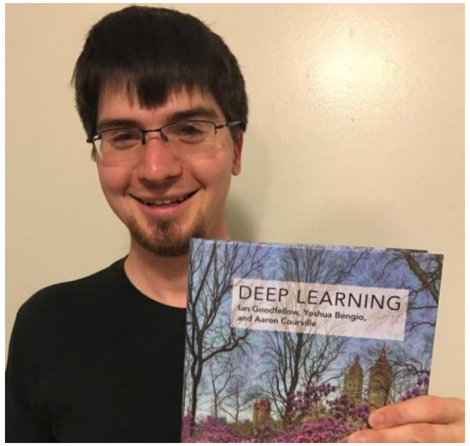


Back to the universal approximator theorem

- One hidden layer is enough to *represent* (not *learn*) an approximation of any function to an arbitrary degree of accuracy

- So why deeper?
  - Shallow net may need (exponentially) more width
  - Shallow net may overfit more

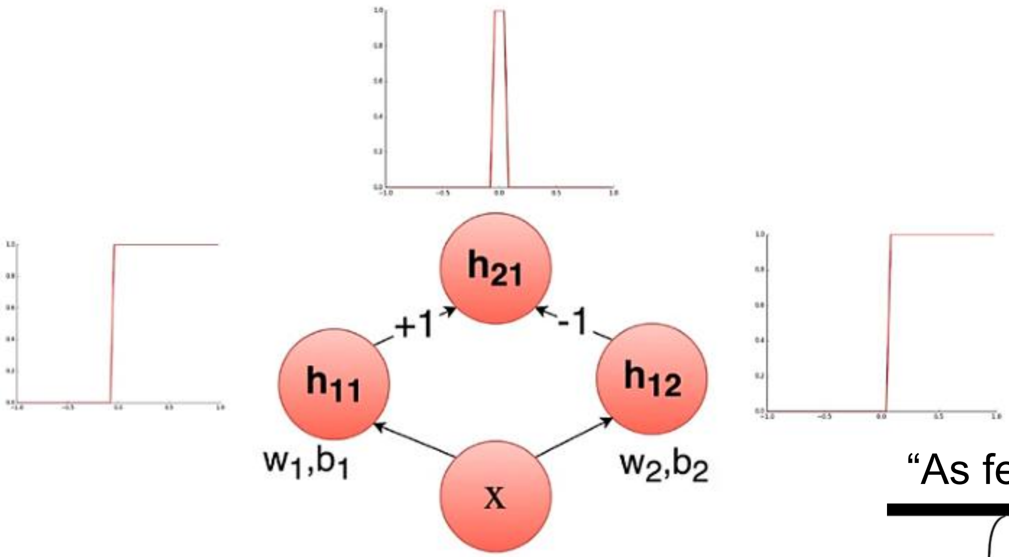




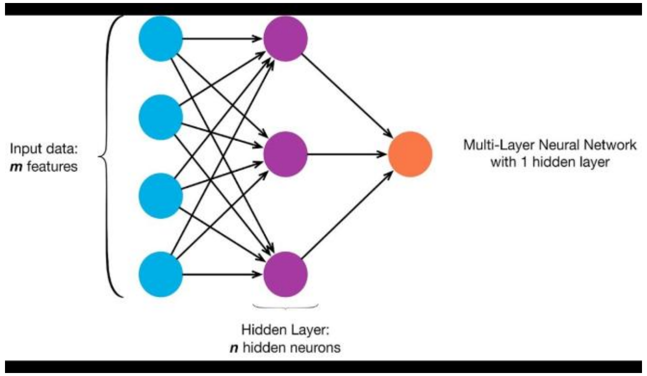
Deep Learning

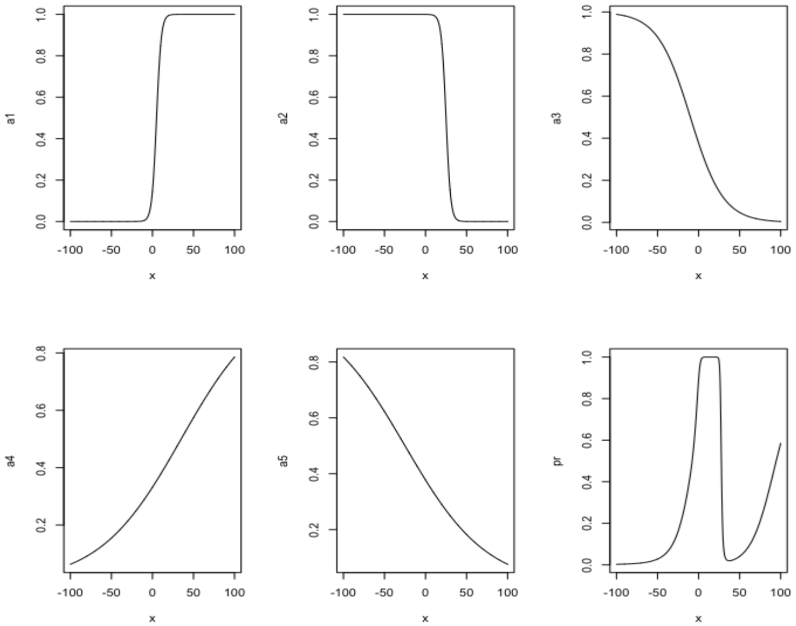
The universal approximator

$$a_1(x) = \frac{1}{1+e^{-(-1.75+0.35x)}}$$
$$a_2(x) = \frac{1}{1+e^{-(8.75-0.35x)}}$$
$$a_3(x) = \frac{1}{1+e^{-(-0.5-0.05x)}}$$
$$a_4(x) = \frac{1}{1+e^{-(-0.7+0.02x)}}$$
$$a_5(x) = \frac{1}{1+e^{-(-0.5-0.02x)}}$$

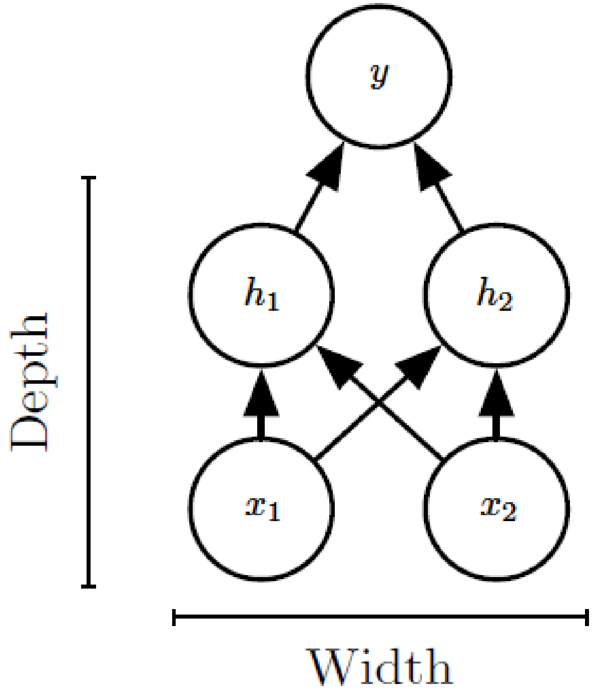


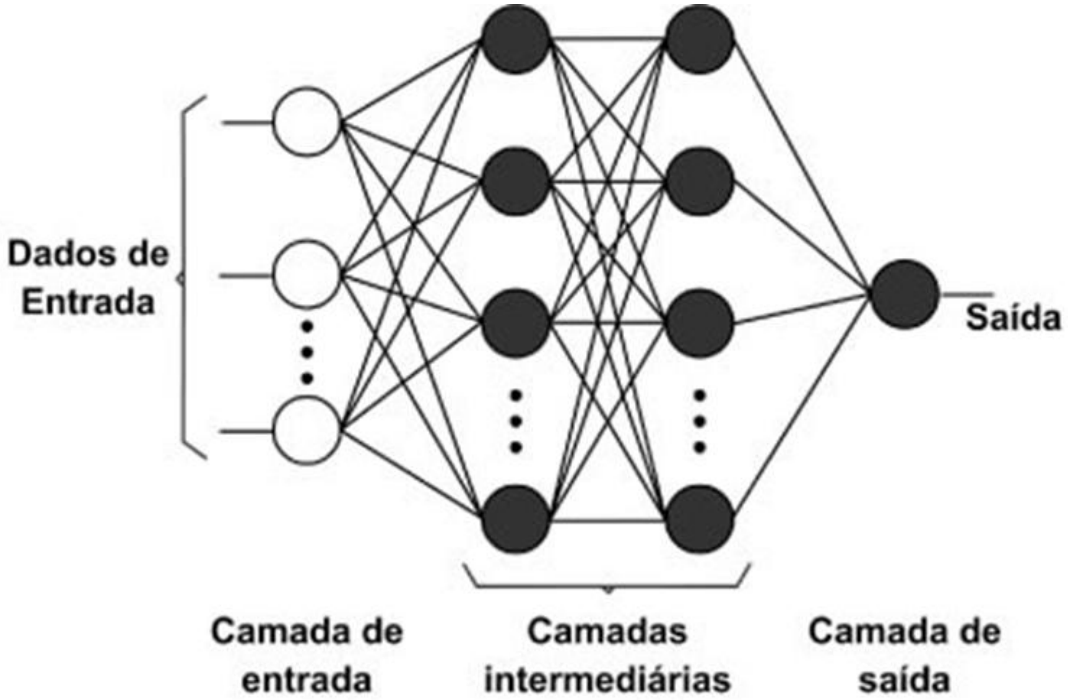
"As few as ***one hidden layer***"





Feedforward neural network, multilayer perceptron (MLP) or fully connected network





[Deep Learning]

Back-propagation

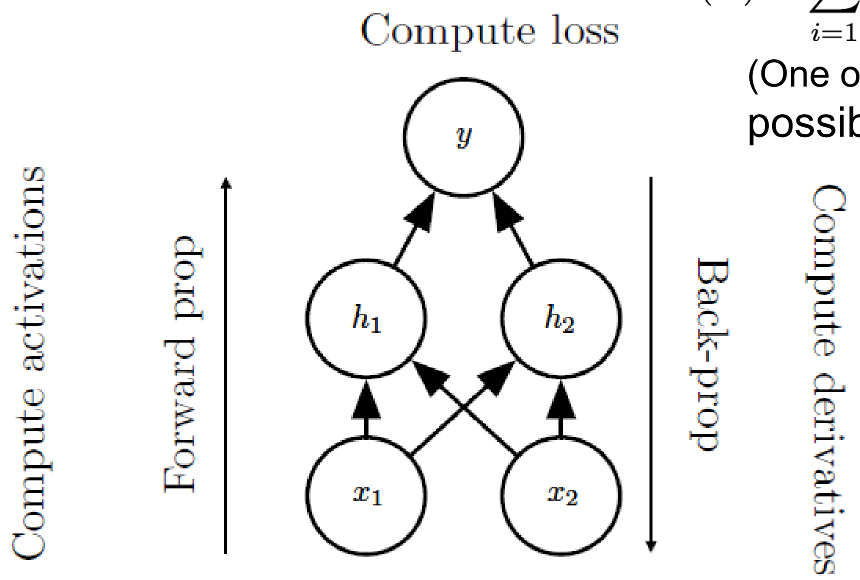
Back-Propagation

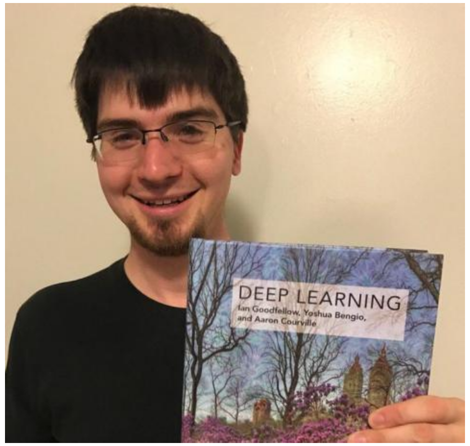
- Back-propagation is "just the chain rule" of calculus
- But it's a particular implementation of the chain rule
  - Uses dynamic programming (table filling)
  - Avoids recomputing repeated subexpressions
  - Speed vs memory tradeoff

$$\frac{\partial error}{\partial w1} = \frac{\partial error}{\partial output} * \frac{\partial output}{\partial hidden2} * \frac{\partial hidden2}{\partial hidden1} * \frac{\partial hidden1}{\partial w1}$$

$$loss(\Theta) = \sum_{i=1}^{n} (y_i - h(x_i, \Theta))^2$$

(One out of many possibilities...)

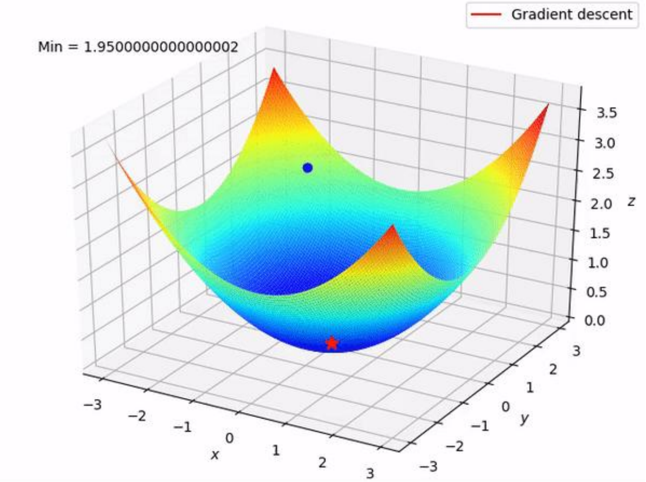


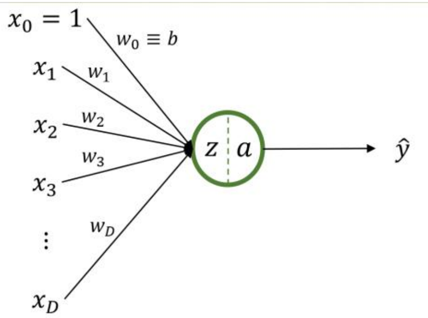


Deep Learning

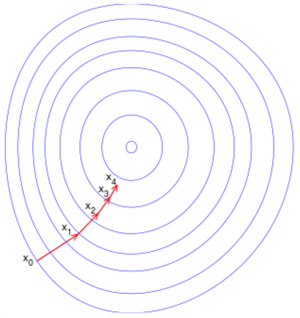
Determining the machine parameters

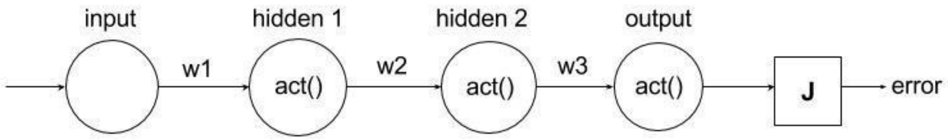
- How to know the correct weights and bias?
- The machine *learns* from data!
- Search for values that result the lowest classification error
- Optimization: gradient descent method
- Back-propagation used to calculate the derivatives with relation to each weight and bias – successive (and tedious) chain rule applications!





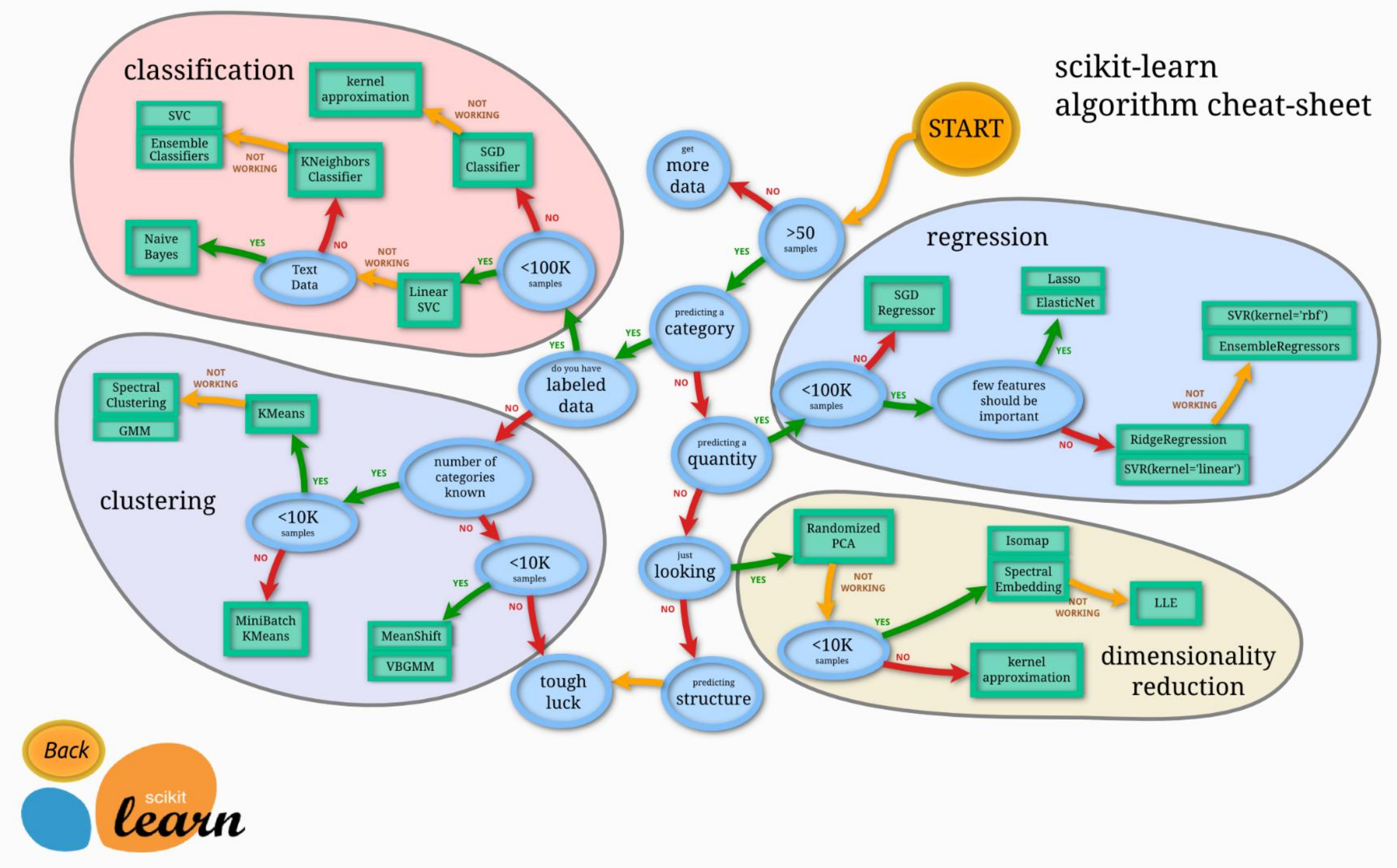
$$Grad\ f = \nabla f = \left(\frac{\partial f}{\partial w_0}, \frac{\partial f}{\partial w_1}, \dots \frac{\partial f}{\partial w_n}\right)$$





$$\frac{\partial error}{\partial w1} = \frac{\partial error}{\partial output} * \frac{\partial output}{\partial hidden2} * \frac{\partial hidden2}{\partial hidden1} * \frac{\partial hidden1}{\partial w1}$$

A toolkit for *classic* machine learning: Scikit-Learn



 http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

More tools for the kit: deep learning *frameworks*

torch

Caffe

Keras

PyTorch

TensorFlow
Google

theano

University of Montreal

CNTK
Cognitive Toolkit
Microsoft

Chainer

Obrigado!