UNIVERSITÀ PARTHENOPE
DiST
DIPARTIMENTO DI SCIENZE E TECNOLOGIE

Machine Learning (part II)

Foundations of Machine Learning

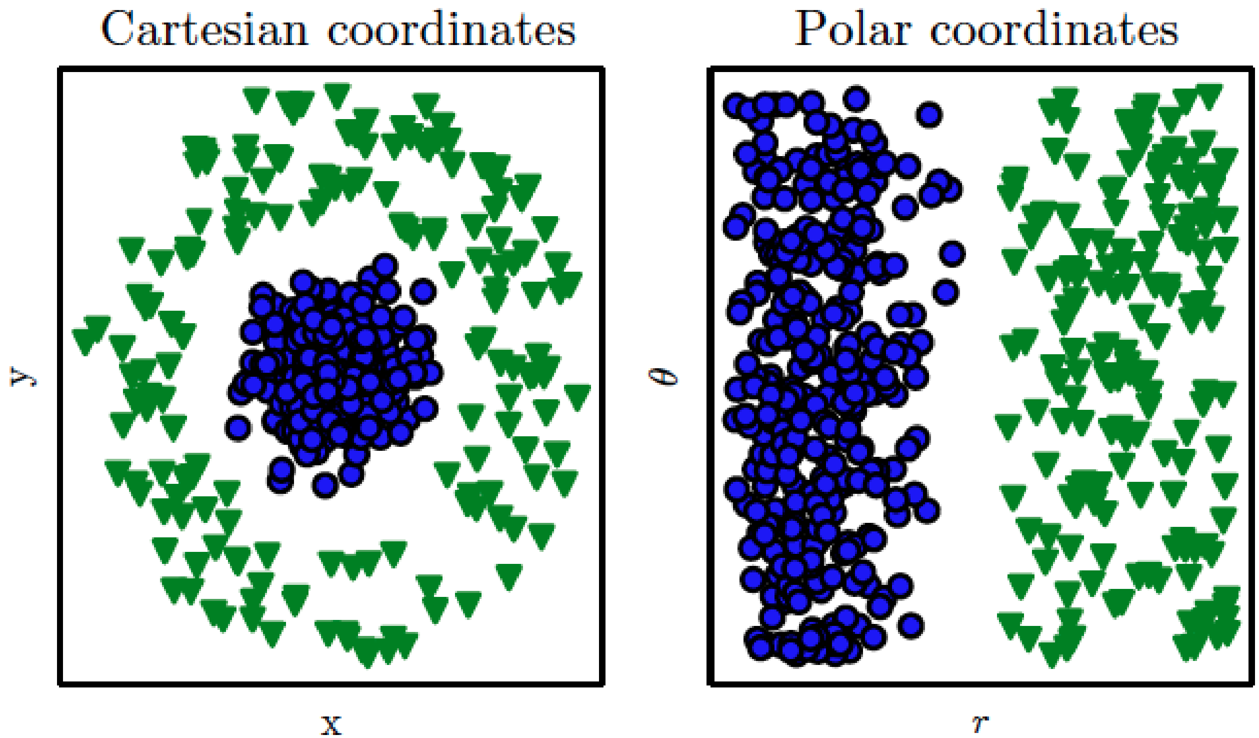
Angelo Ciaramella

Machine Learning

- Knowledge base
  - projects have sought to hard-code knowledge about the world in formal languages
  - logical inference rules
- Machine Learning (ML)
  - systems need the ability to acquire their own knowledge
  - extracting patterns from raw data
  - allowed computers to tackle problems involving knowledge of the real world
  - make decisions that appear subjective

Data representation

- Representation
  - performance of machine learning algorithms depends heavily on the representation of the data
  - information included in the representation is known as a feature



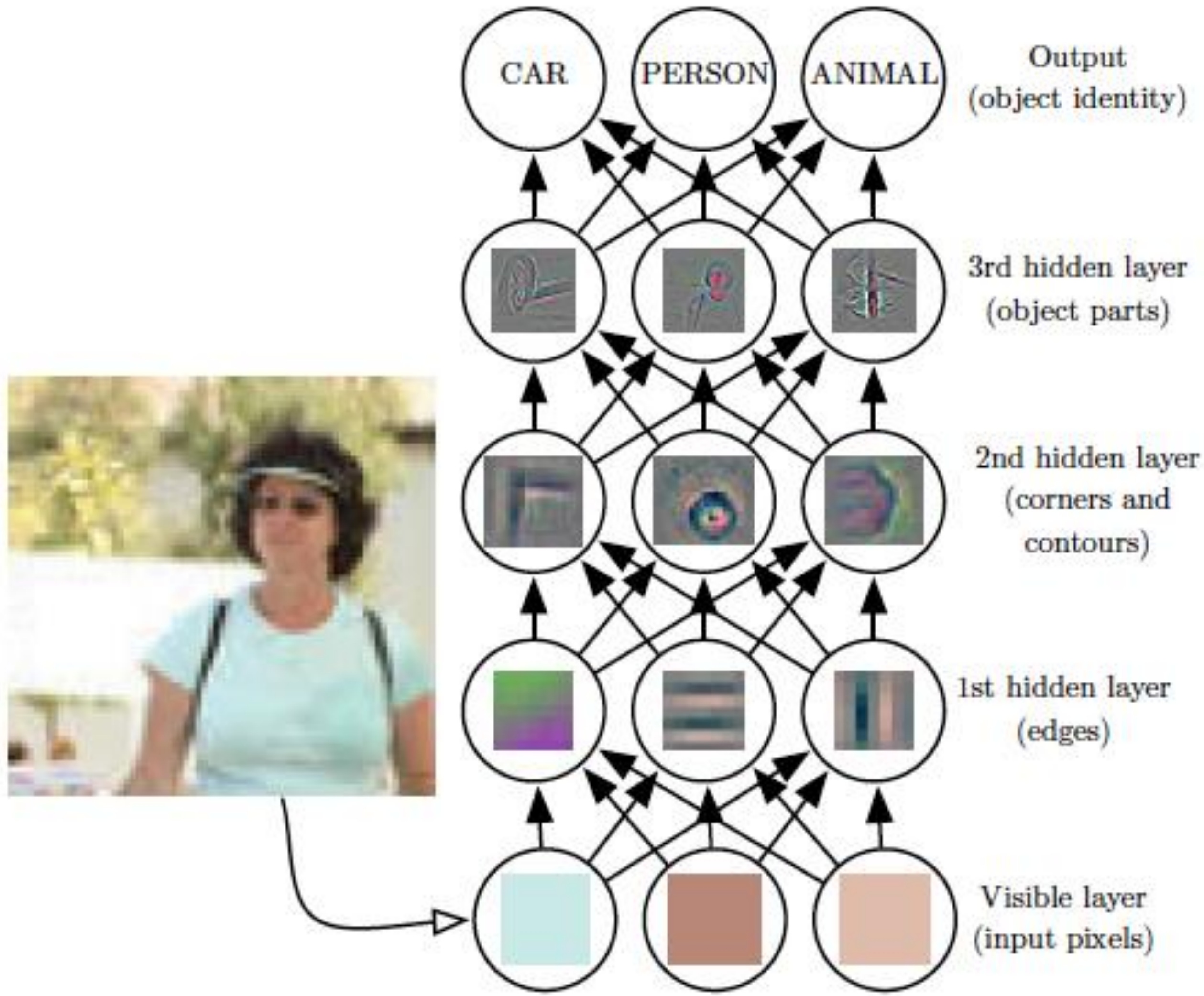
Representation learning

- Representation
  - for many tasks, it is difficult to know what features should be extracted
    - e.g., program to detect cars in photographs
  - solution
    - use machine learning to discover not only the mapping from representation to output but also the representation itself (representation learning)
    - the quintessential example of a representation learning algorithm is the autoencoder

Deep learning

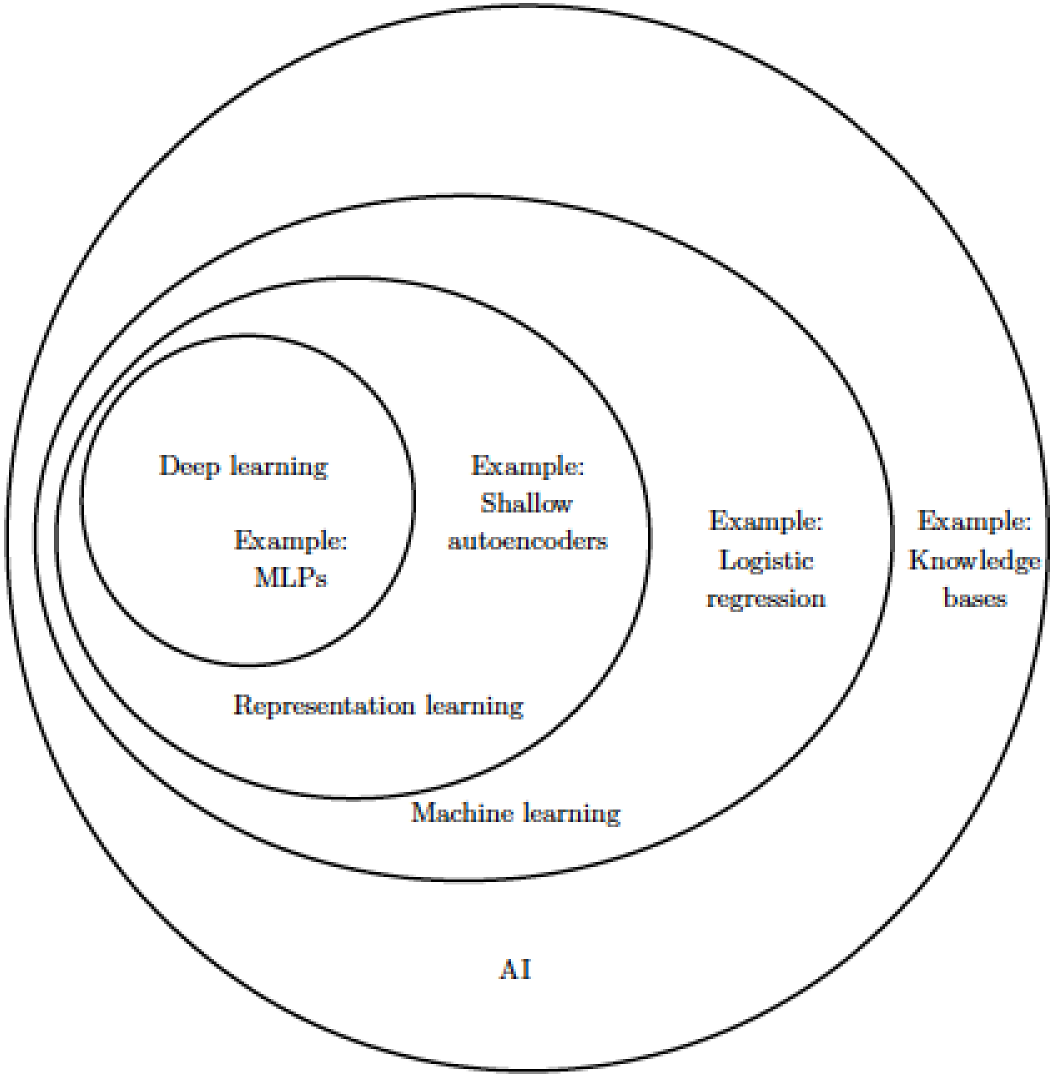
- Deep Learning (DL)
  - solves representation learning by introducing representations that are expressed in terms of other simpler representations
  - the quintessential example of a deep learning model is the feedforward deep network or MultiLayer Perceptron (MLP)
    - mathematical function mapping some set of input values to output values

DL and MLP



Information representation of a DL model

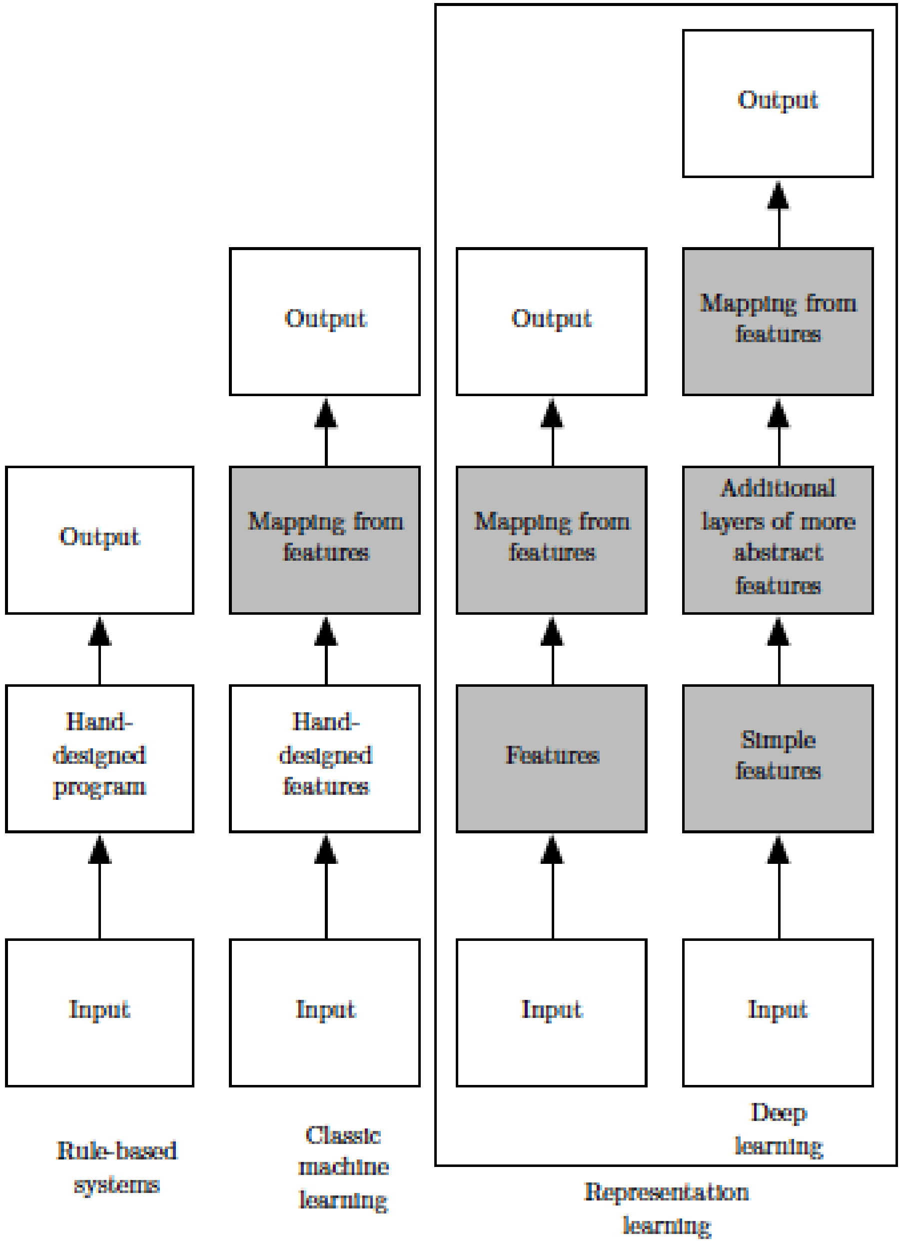
AI diagram



Venn diagram

AI systems

Flowcharts of AI systems



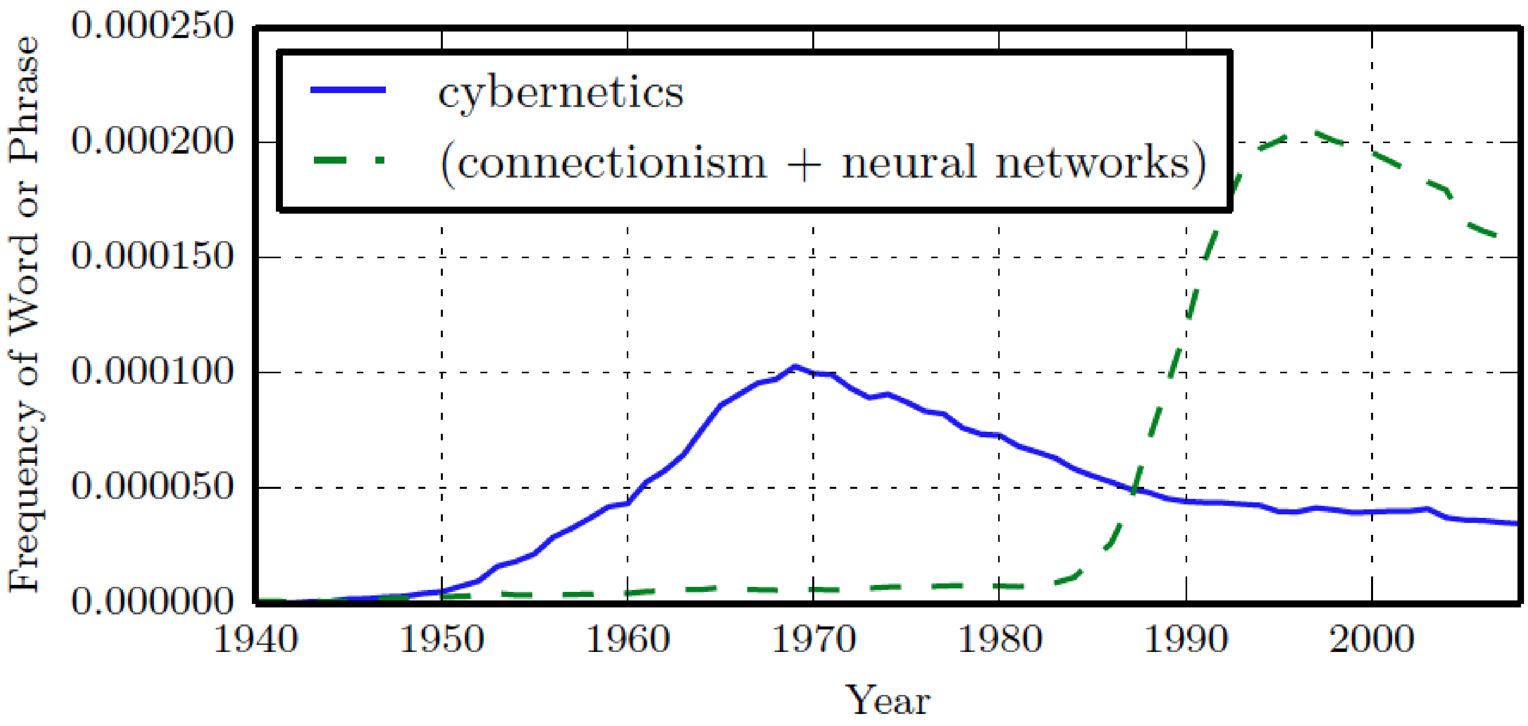
Historical trends

- Historical trends in DL
  - DL dates back to the 1940s

- Three waves of development
  - 1940s - 1960s cybernetics
  - 1980s - 1990s connectionism
  - trom 2006 deep learning

- Earliest learning algorithms
  - computational models of biological learning
  - Artificial Neural Networks
  - today learning frameworks are not necessarily neurally inspired

Historical trends



Historical waves of artificial neural nets research

First model

- McCulloch-Pitts Neuron
  - Warren Sturgis McCulloch and Walter Harry Pitts
  - 1943 – Article titled: *A logical calculus of the ideas immanent in nervous activity*
  - early model of brain function
  - the linear model could recognize two different categories of inputs
  - the weights needed to be set correctly by the human operator

McCulloch and Pitts research

Bulletin of Mathematical Biology Vol. 52, No. 1/2, pp. 99–115, 1990.
Printed in Great Britain.
0092-8240/90$3.00+0.00
Pergamon Press plc
Society for Mathematical Biology

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY*

■ WARREN S. MCCULLOCH AND WALTER PITTS
University of Illinois, College of Medicine,
Department of Psychiatry at the Illinois Neuropsychiatric Institute,
University of Chicago, Chicago, U.S.A.

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

1. Introduction. Theoretical neurophysiology rests on certain cardinal assumptions. The nervous system is a net of neurons, each having a soma and an axon. Their adjunctions, or synapses, are always between the axon of one neuron and the soma of another. At any instant a neuron has some threshold, which excitation must exceed to initiate an impulse. This, except for the fact and the time of its occurence, is determined by the neuron, not by the excitation. From the point of excitation the impulse is propagated to all parts of the neuron. The velocity along the axon varies directly with its diameter, from $<1\ \mathrm{ms}^{-1}$ in thin axons, which are usually short, to $>150\ \mathrm{ms}^{-1}$ in thick axons, which are usually long. The time for axonal conduction is consequently of little importance in determining the time of arrival of impulses at points unequally remote from the same source. Excitation across synapses occurs predominantly from axonal terminations to somata. It is still a moot point whether this depends upon irreciprocity of individual synapses or merely upon prevalent anatomical configurations. To suppose the latter requires no hypothesis *ad hoc* and explains known exceptions, but any assumption as to cause is compatible with the calculus to come. No case is known in which excitation through a single synapse has elicited a nervous impulse in any neuron, whereas any neuron may be excited by impulses arriving at a sufficient number of neighboring synapses within the period of latent addition, which lasts <0.25 ms. Observed temporal summation of impulses at greater intervals

* Reprinted from the *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115–133 (1943).

99

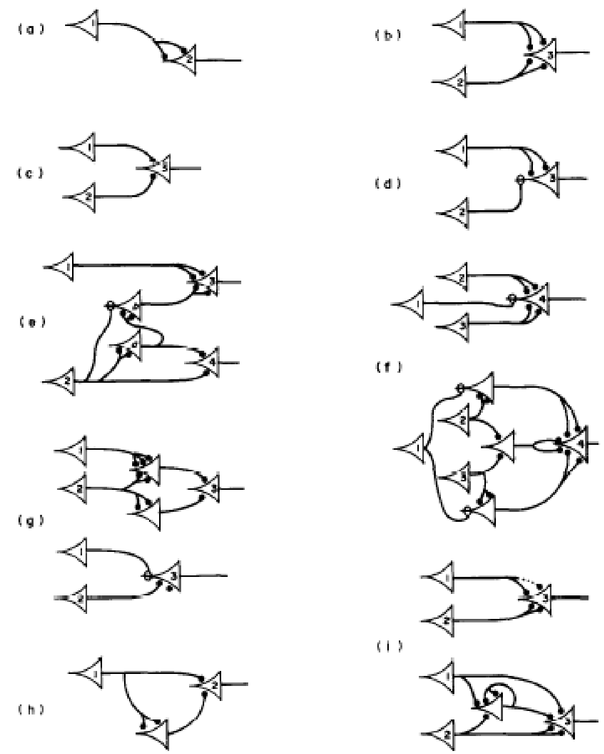
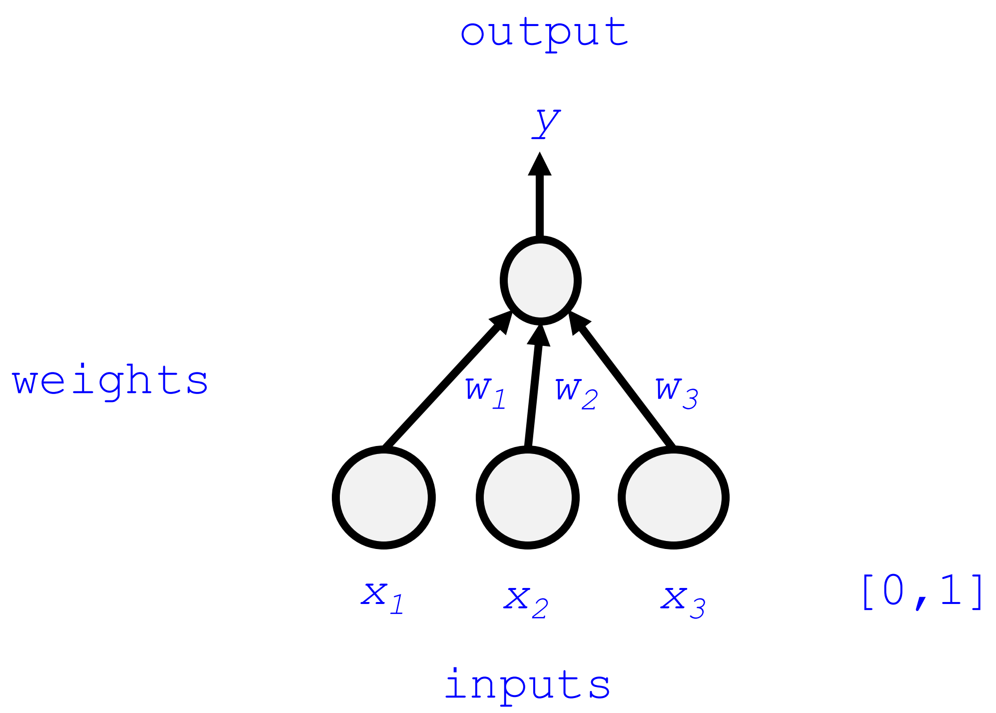


Figure 1. The neuron c_i is always marked with the numeral i upon the body of the cell, and the corresponding action is denoted by "N" with is subscript, as in the text:

(a) $N_2(t) . \equiv . N_1(t-1)$;
(b) $N_3(t) . \equiv . N_1(t-1) \vee N_2(t-1)$;
(c) $N_3(t) . \equiv . N_1(t-1) . N_2(t-1)$;
(d) $N_3(t) . \equiv N_1(t-1) . \sim N_2(t-1)$;
(e) $N_3(t) :\equiv: N_1(t-1) . \vee . N_2(t-3) . \sim N_2(t-2)$;
$N_4(t) . \equiv . N_2(t-2) . N_2(t-1)$;
(f) $N_4(t) :\equiv: \sim N_1(t-1) . N_2(t-1) \vee N_3(t-1) . \vee . N_1(t-1).$
$N_2(t-1) . N_3(t-1)$
$N_4(t) :\equiv: \sim N_1(t-2) . N_2(t-2) \vee N_3(t-2) . \vee . N_1(t-2).$
$N_2(t-2) . N_3(t-2)$;
(g) $N_3(t) . \equiv . N_2(t-2) . \sim N_1(t-3)$;
(h) $N_2(t) . \equiv . N_1(t-1) . N_1(t-2)$;
(i) $N_3(t) :\equiv: N_2(t-1) . \vee . N_1(t-1) . (Ex)t-1 . N_1(x) . N_2(x)$.

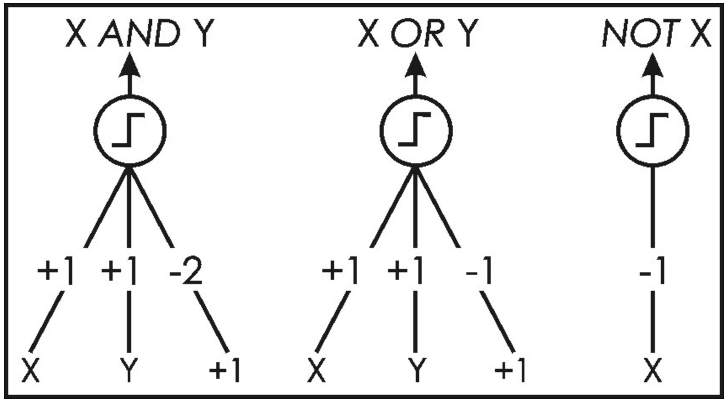
ML – Foundations of ML

McCulloch and Pitts linear model



$$y = \sum_{i=1}^{3} w_i x_i$$

McCulloch and Pitts model



Output response (threshold)

$$T \geq 0 \rightarrow 1$$
$$T < 0 \rightarrow 0$$

Learning

- Hebbian theory
  - Donald O. Hebb was a Canadian psychologist
  - First learning hypotheses
  - 1949 – Book titled: *The organization of behavior*
  - *links to complex brain models have been proposed*
    - *Hebbian learning - Hebb's rule*

The perceptron

- Connessionism and learning
  - Frank Rosenblatt introduced the perceptron
  - 1957 – Article titled: *The Perceptron - a perceiving and recognizing automaton*
  - system consists of binary activations
  - a variable threshold value is used
  - perceptron learn the weights defining the categories given examples of inputs from each category

The perceptron

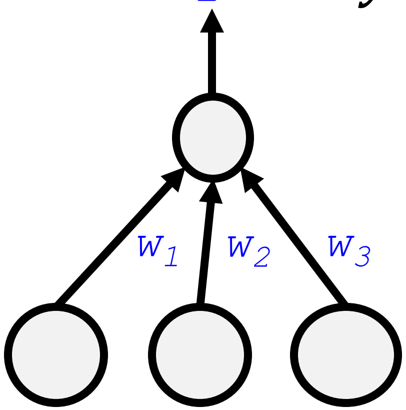
$$\theta = \begin{cases} 1 & if \ \ f(\boldsymbol{w}, \boldsymbol{x}) > 0 \\ 0 & \qquad otherwise \end{cases}$$

output

y

$$y = f(\boldsymbol{w}, \boldsymbol{x}) = \theta\left(\sum_{i=1}^{3} w_i \, x_i \right)$$

weights



$$\boldsymbol{w} = [w_1, w_2, w_3]$$

x_1 x_2 x_3

$$\boldsymbol{x} = [x_1, x_2, x_3]$$

[0,1]

inputs

Delta rule

- Learning approach
  - Bernard Widrow and Ted Hoff
  - 1960 – Article titled: *Adaptive Switching Circuits*
  - Delta rule
    - gradient descent learning rule for updating the weights of the inputs to artificial neurons in a single-layer neural network
  - Adaptive filters
    - Adaline - Adaptive Linear Neuron

Learning and generalization

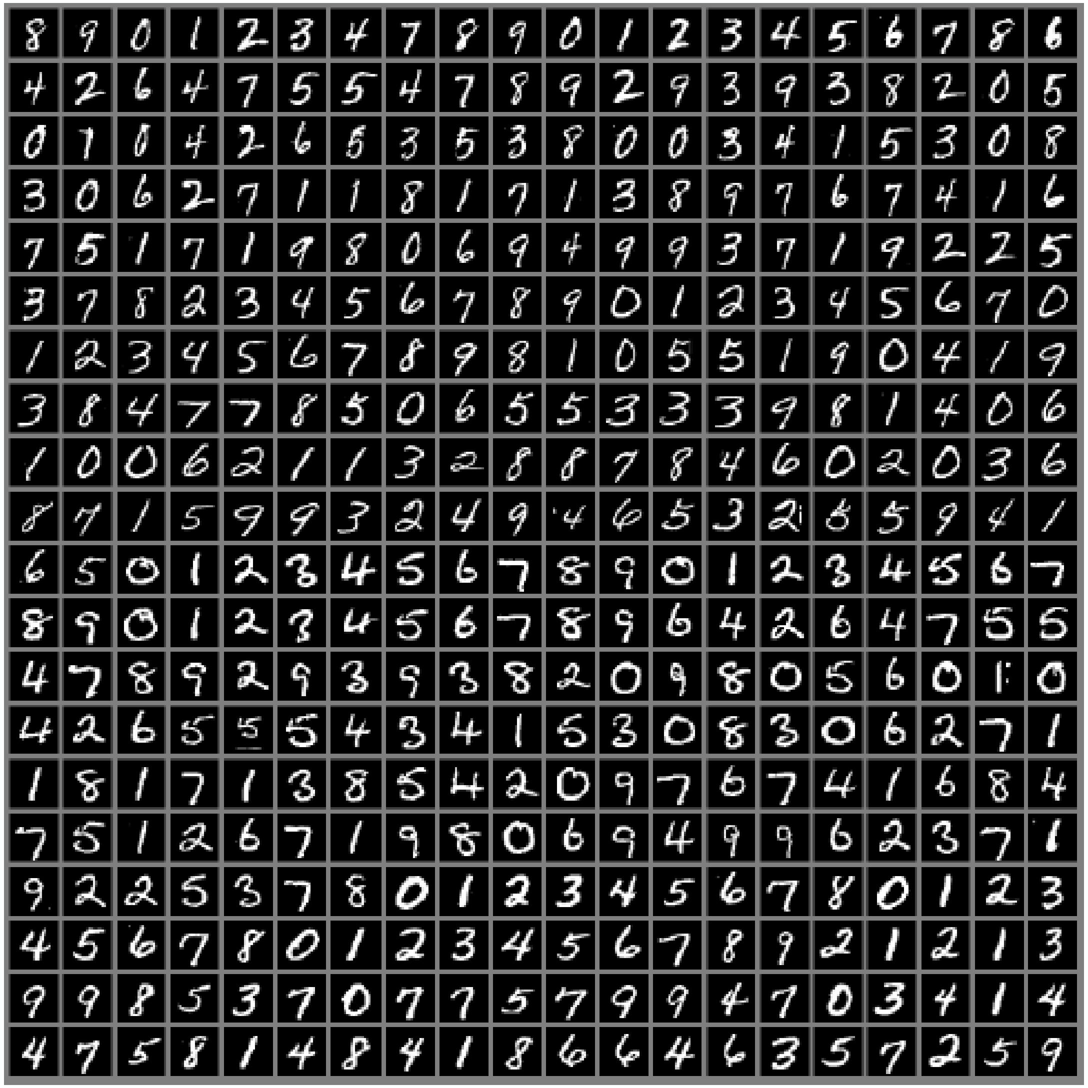
- MultiLayer Perceptron (MLP) and learning
  - Paul Werbos
  - 1974 - generalization of delta rule could be used for MLP
    - doctoral dissertation

- Backpropagation and recognition
  - David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams
    - 1986 – Article titled: Learning representations by back-propagating errors
  - James McClelland
  - Connectionism - large number of simple computational units can achieve intelligent behavior when networked together
    - distributed representation

Deep learning

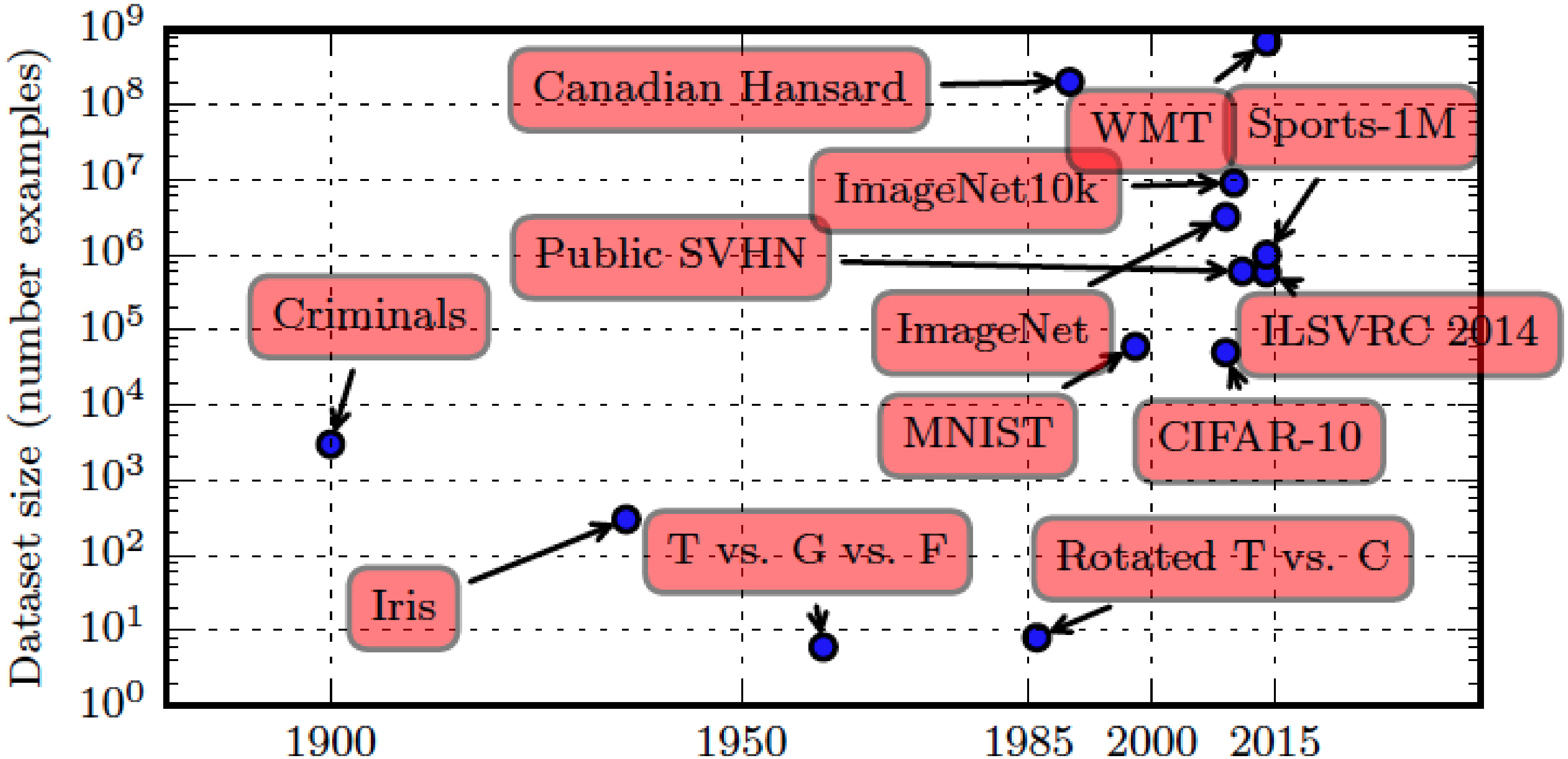
- Deep Neural Netorks
  - Geoffrey Hinton
  - 2006 - efficiently trained a deep belief network using a strategy called greedy layer-wise pretraining
  - train deeper neural networks focusing attention on the *theoretical importance of depth*

MNIST dataset



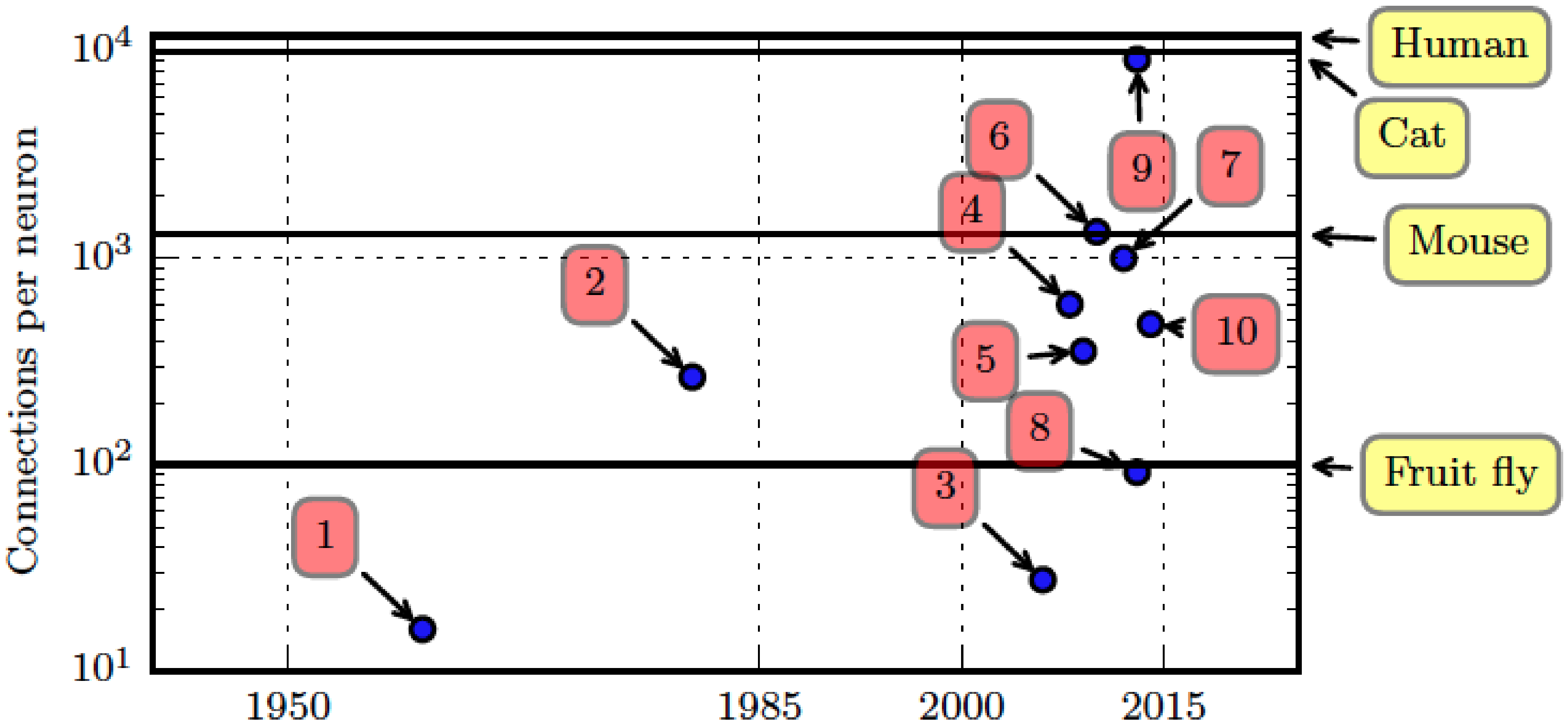
MNIST dataset of scans of handwritten numbers

Growing datasets



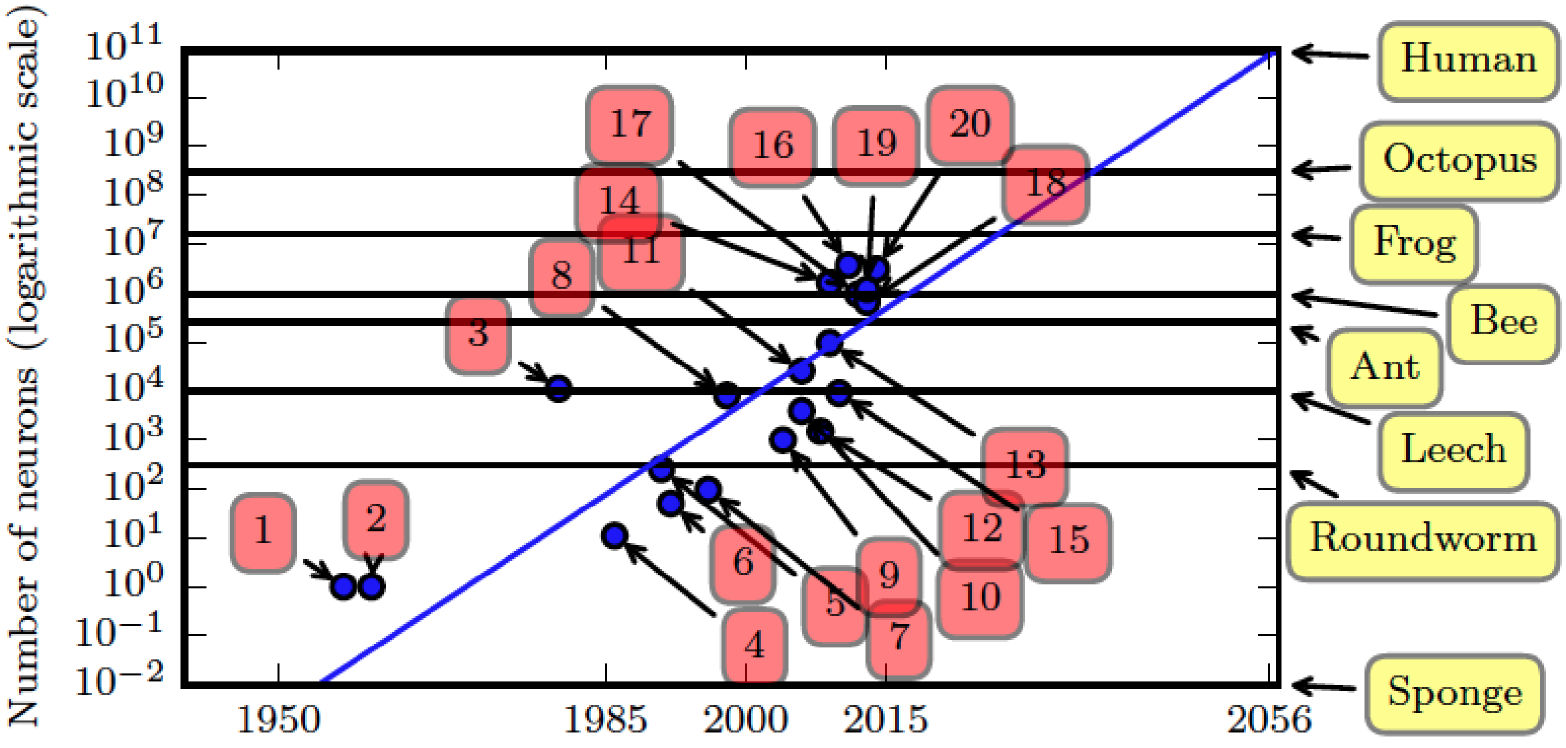
Increasing dataset over time

Growing connections



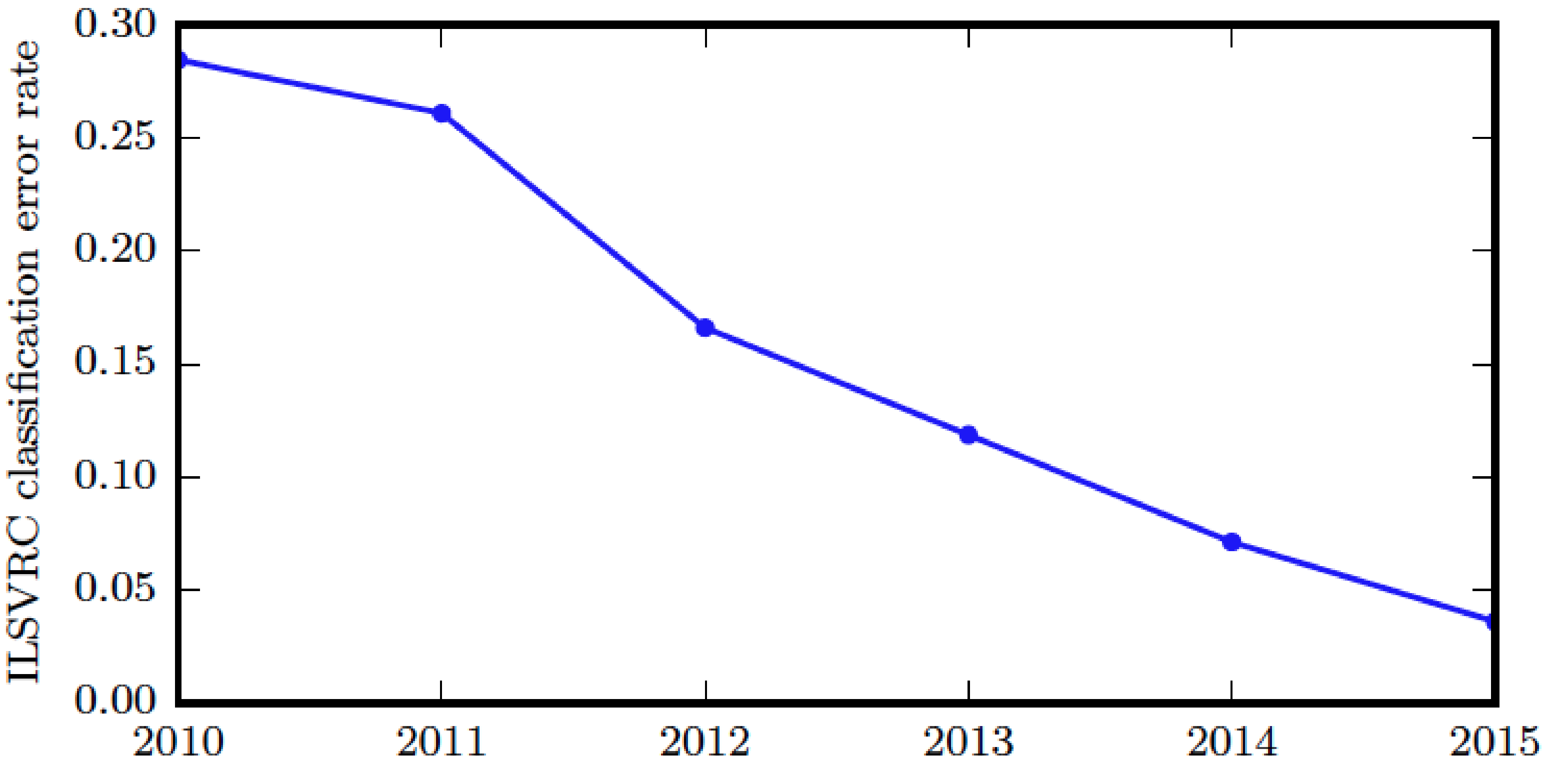
Number of connections per neuron over time

Growing neurons



Number of neurons over time

Deep Learning and challenge



ImageNet Large Scale Visual Recognition Challenge

Deep learning

- Companies using DL
  - Google, Microsoft, Facebook, IBM, Baidu, Apple, Adobe, Netflix, NVIDIA and NEC

- Software libraries
  - Scikit-learn (Pedregosa et al., 2011)
  - Theano (Bergstra et al., 2010; Bastien et al., 2012)
  - PyLearn2 (Goodfellow et al., 2013)
  - Torch (Collobert et al., 2011),
  - DistBelief (Dean et al., 2012)
  - Caffe (Jia, 2013)
  - MXNet (Chen et al., 2015)
  - Keras (Chollet et al., 2015)
  - TensorFlow (Abadi et al., 2015)

Machine learning

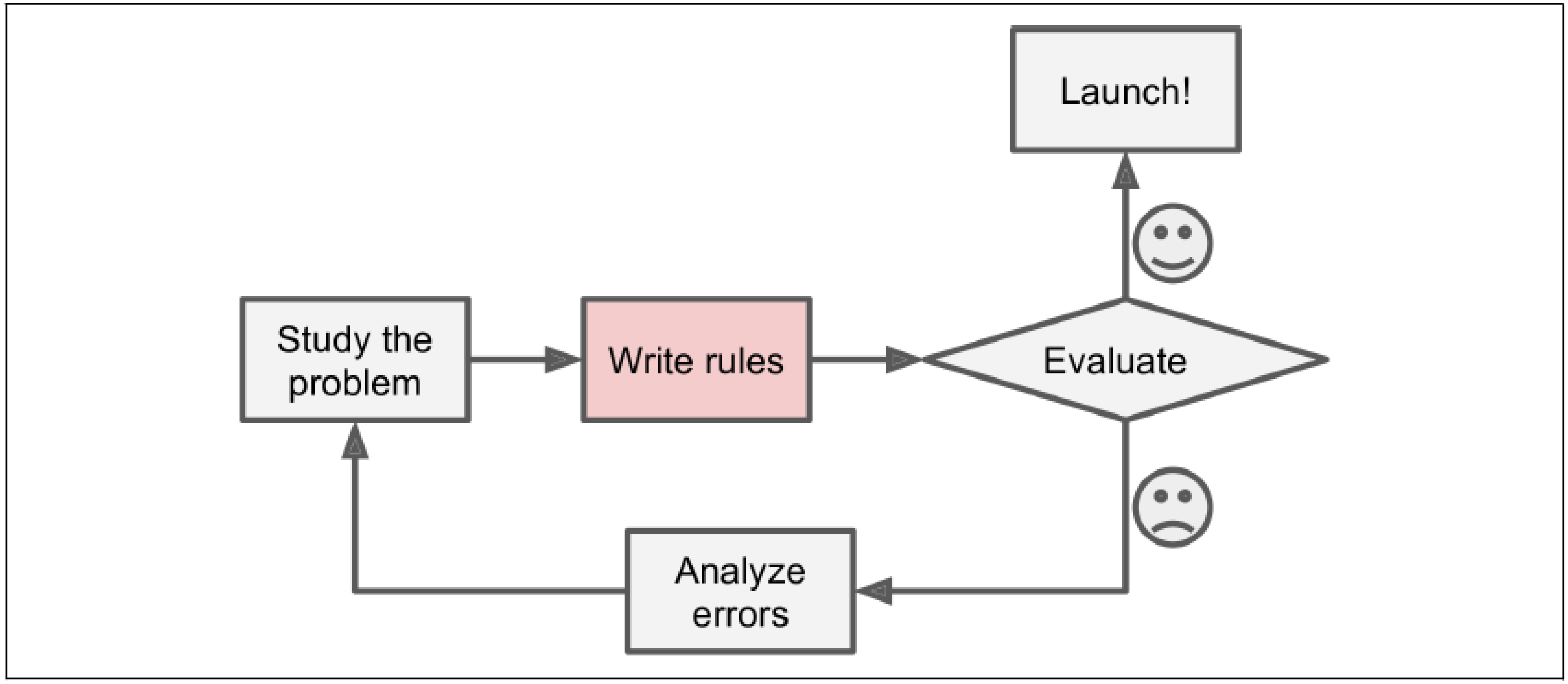
- Machine Learning (ML) is the science (and art) of programming computers so they can learn from data

- The examples that the system uses to learn are called the training set (experience)

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

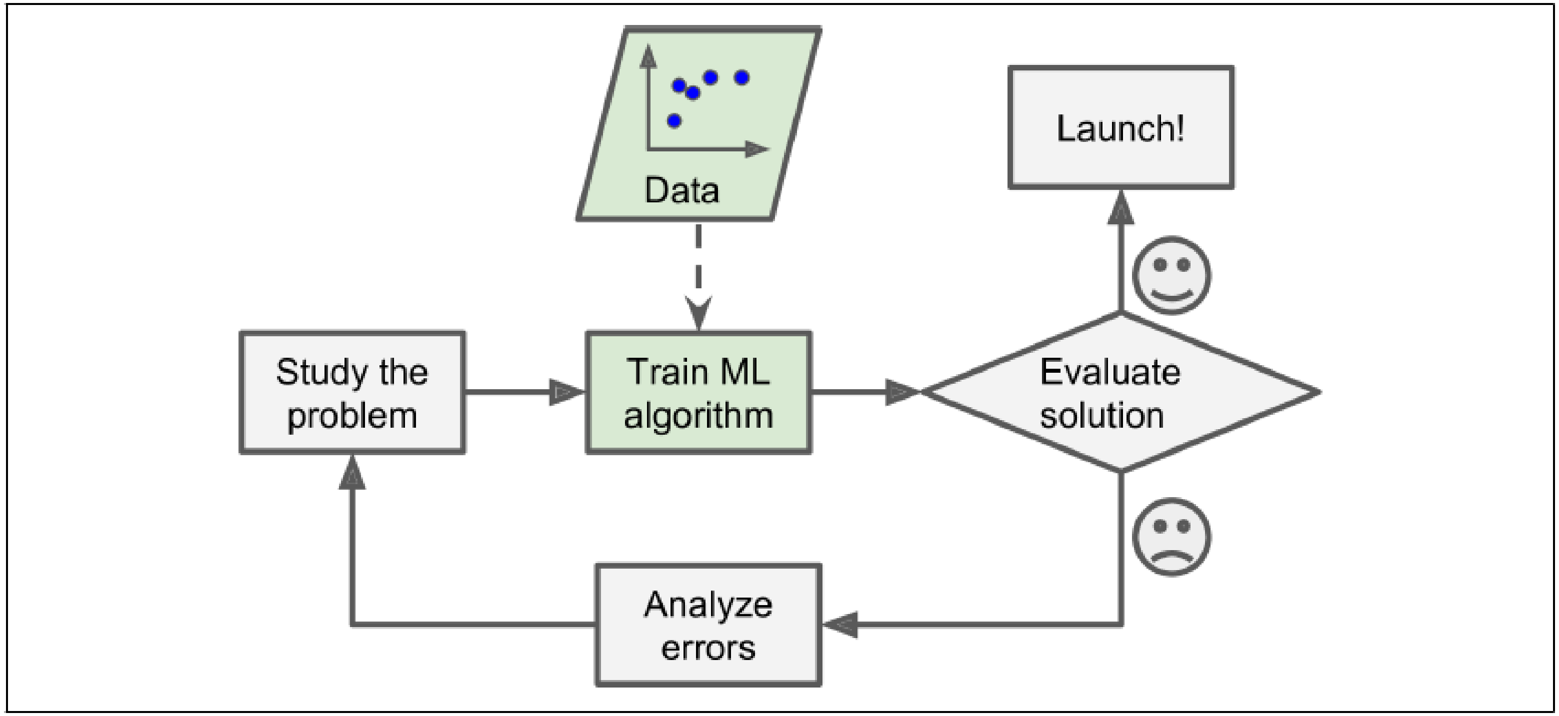
—Tom Mitchell, 1997

Machine learning



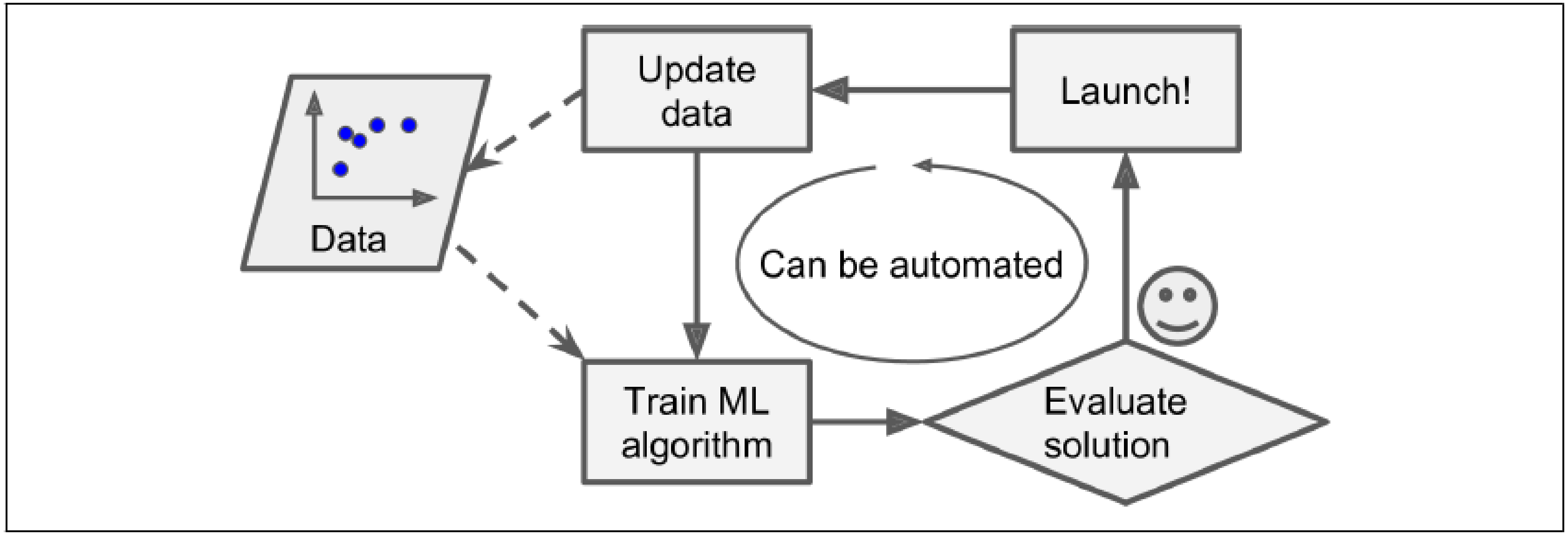
Traditional approach

Machine learning



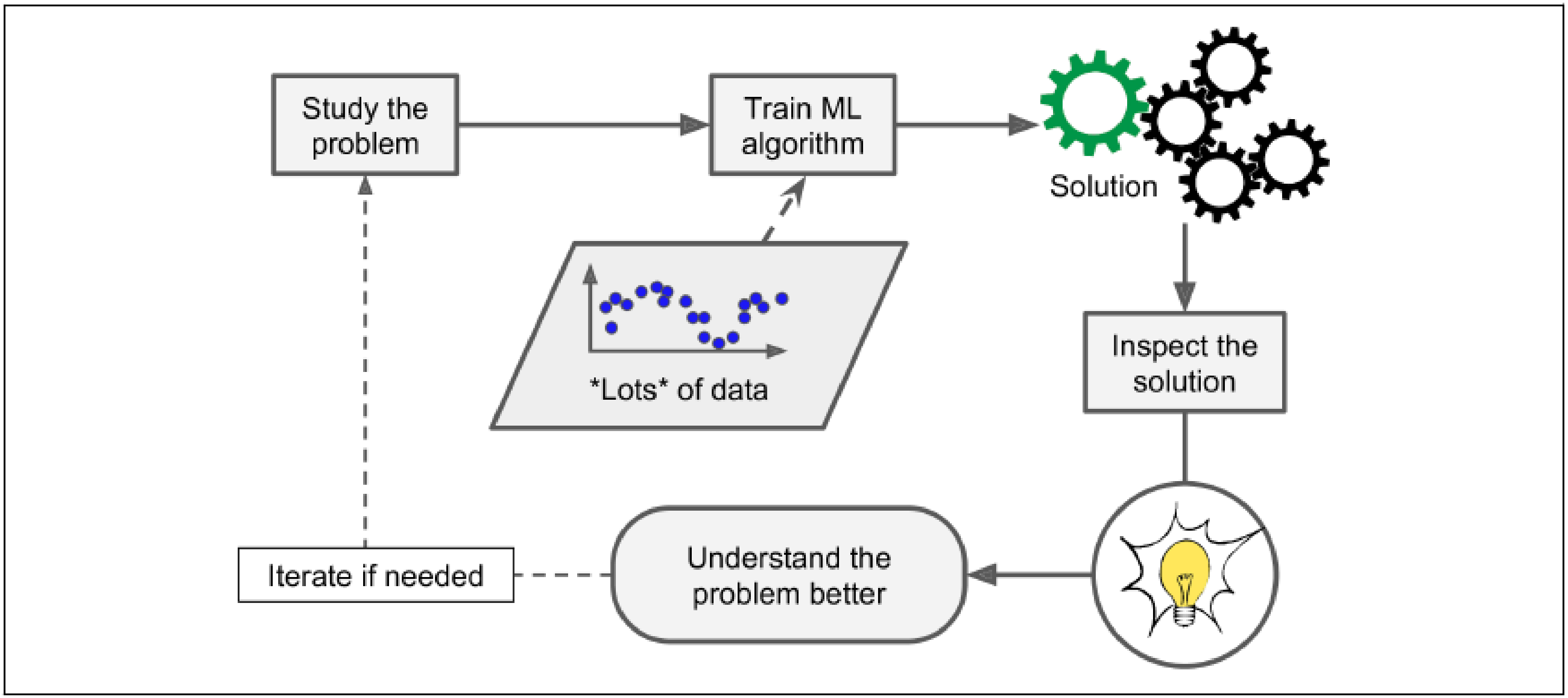
Machine Learning approach

Machine learning



Automatically adapting to change

Machine learning



ML can help humans learn (data mining)

Types of ML systems

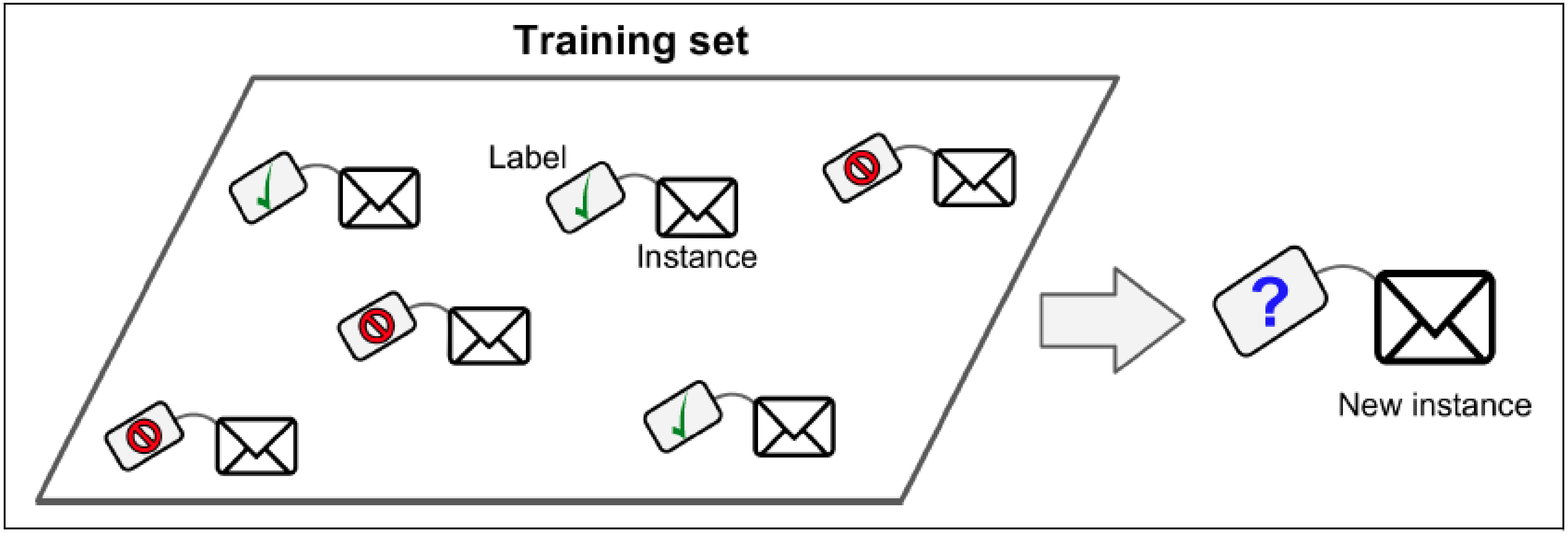
- Learning systems
  - *Supervised*
    - The training set you feed to the algorithm includes desidered soultions called label (e.g., target)
  - *Unsupervised*
    - The training set you feed to the algorithm is unlabeled

Types of ML systems

- target
  - *class*
    - e.g., spam or ham
    - classification
  - *numeric value*
    - e.g., price of a car
    - predictiors and the task is regression

- features
  - attribute with a value
  - Age (attribute) – 20 (value)

Supervised learning



A labelled training set for supervised learning (e.g., spam classification)

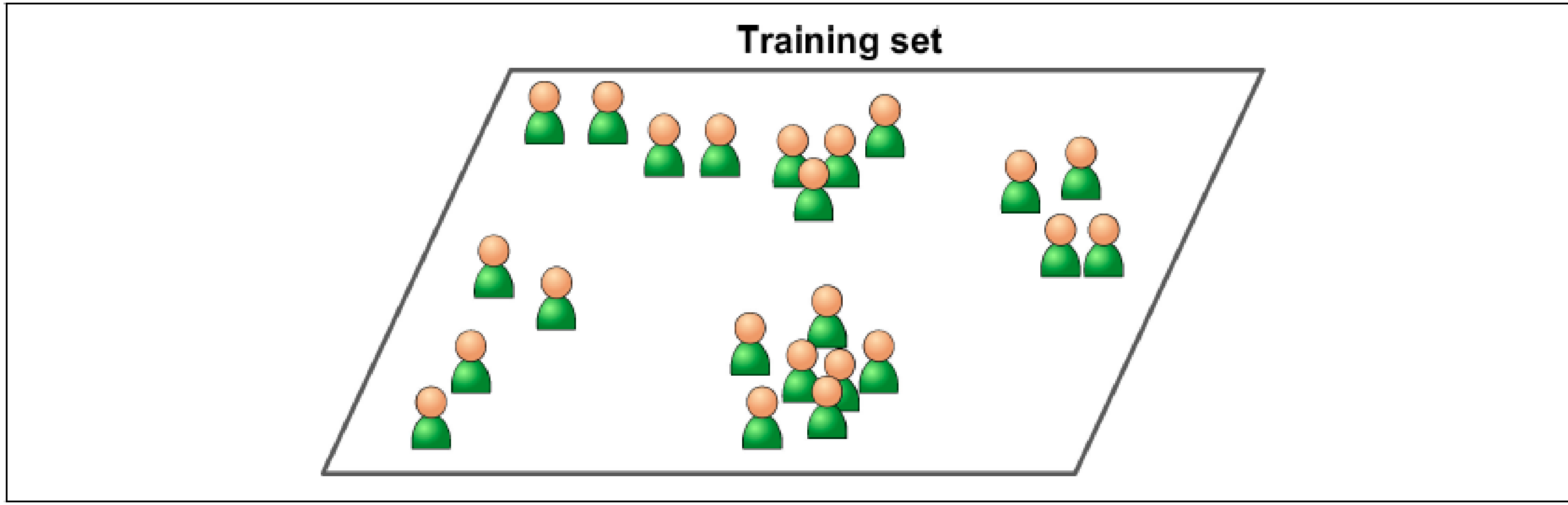
Supervised learning

Regression

Supervised algorithms

- Supervised learning algorithms
  - *K-Nearest Neighbors*
  - *Linear regression*
  - *Logistic regression*
  - *Support Vector Machines*
  - *Decision Trees and Random Forests*
  - *Neural Networks*

Unsupervised learning

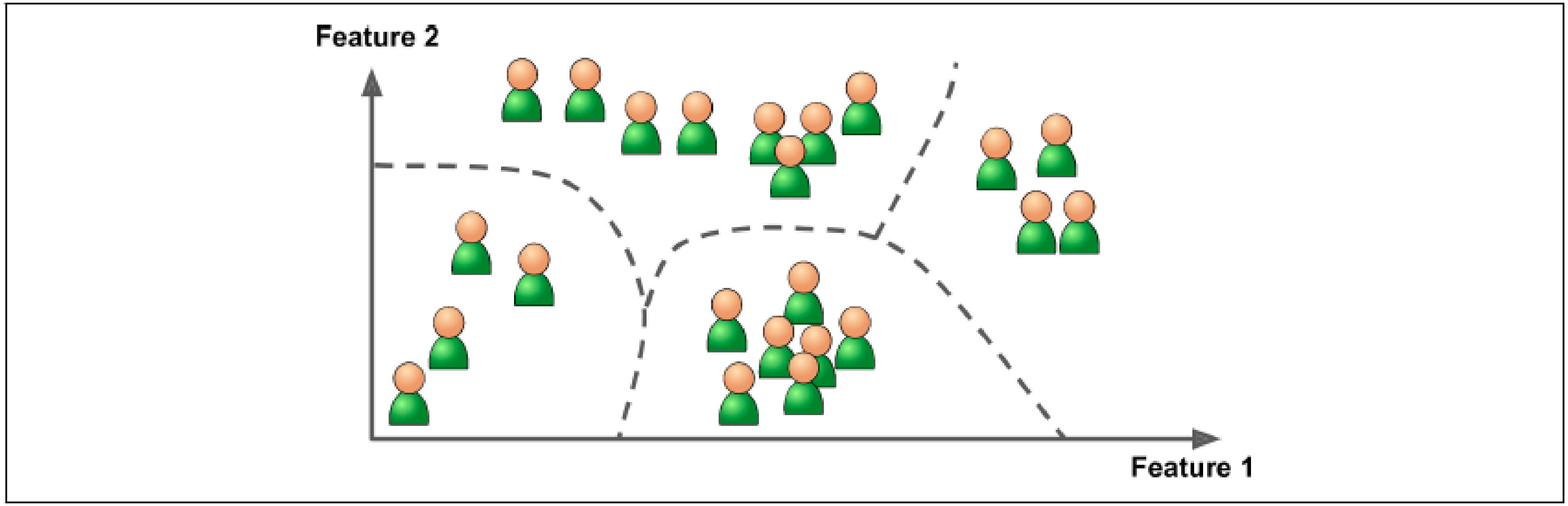


Unlabeled training set for unsupervised learning

Unsupervised algorithms

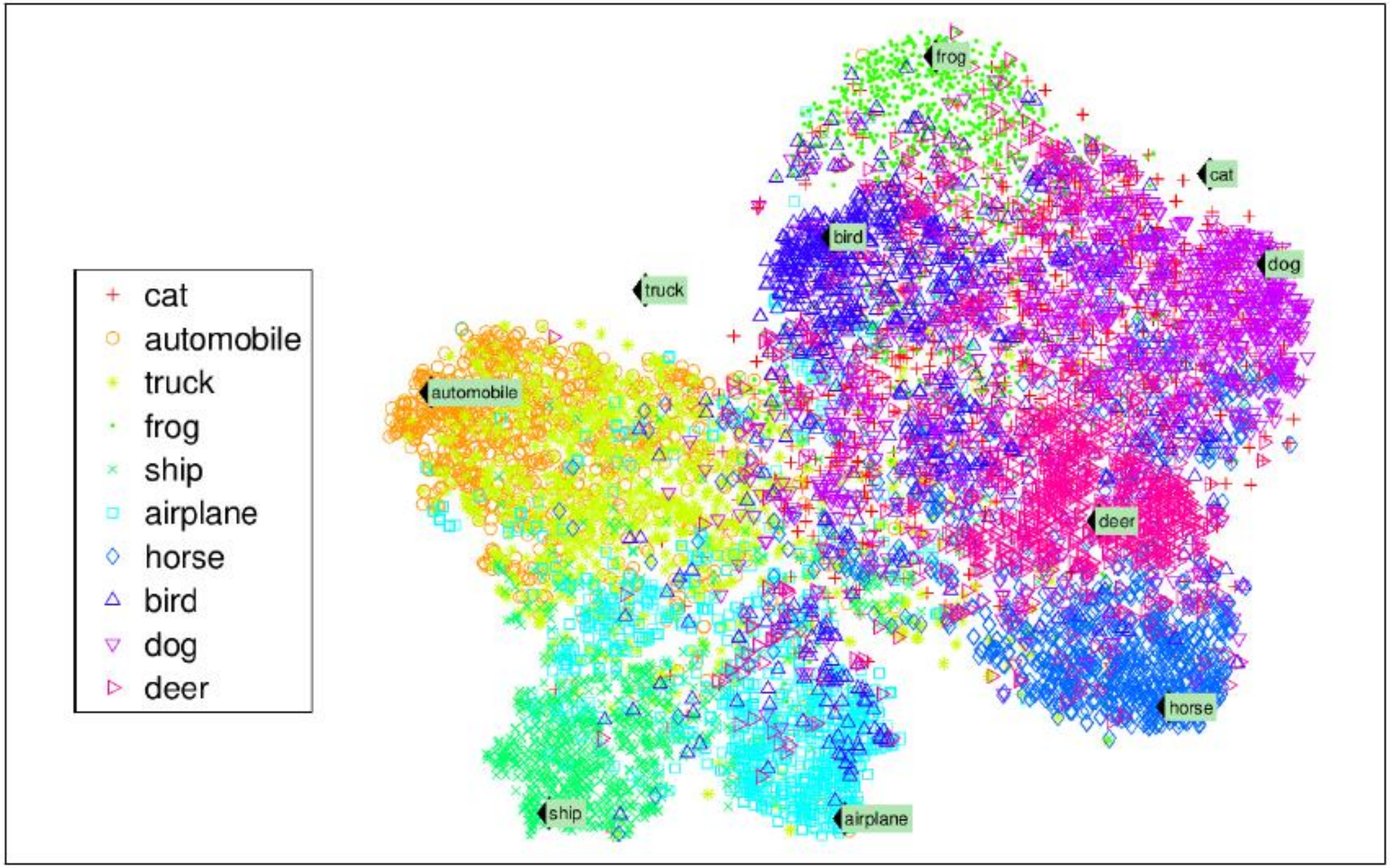
- Clustering
  - *K-Means*
  - *Fuzzy C-Means*
  - *DBSCAN*
  - *Hierarchical Cluster Analysis*
- Anomaly detection and novelty detection
  - *One-class SVM*
  - *Isolation Forest*
- *Visualization and dimensionality reduction*
  - *Self Organizing Maps*
  - *Isomap*
  - *Principal Component Analysis (PCA)*
  - *Kernel PCA*
  - *Locally-Linear Embedding (LLE)*

Unsupervised learning



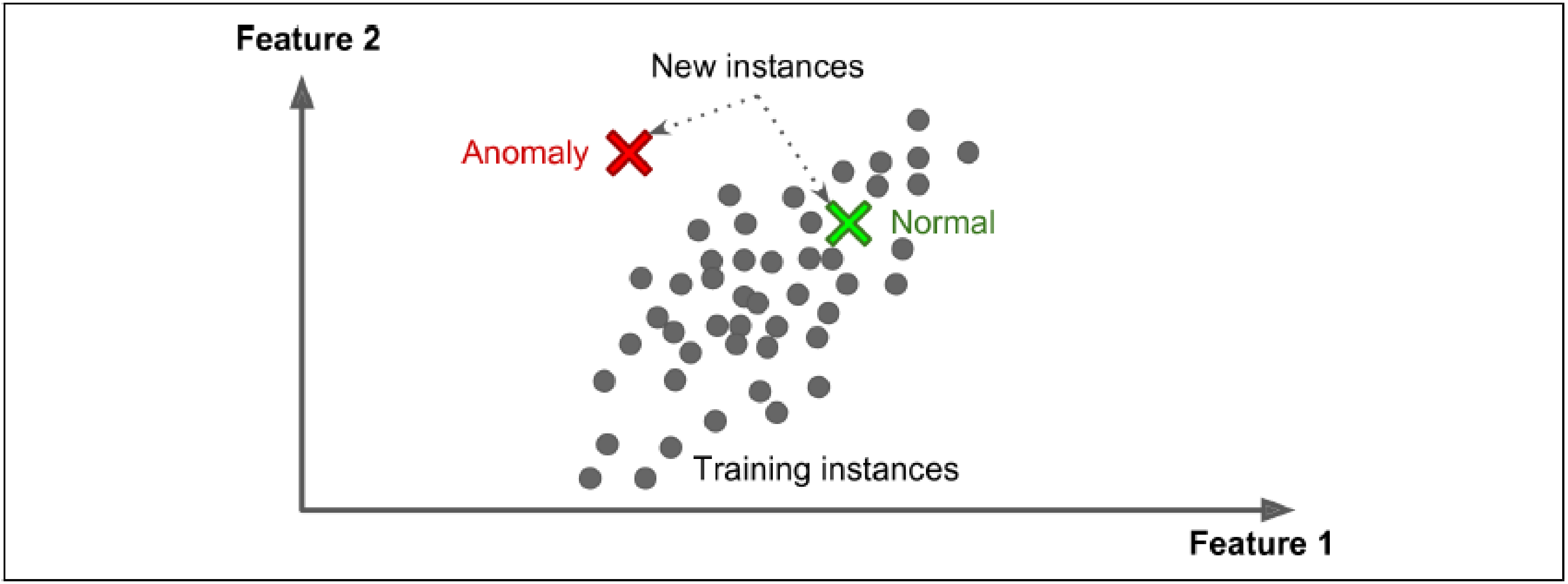
Clustering

Unsupervised learning



Clustering and visualization

Unsupervised learning



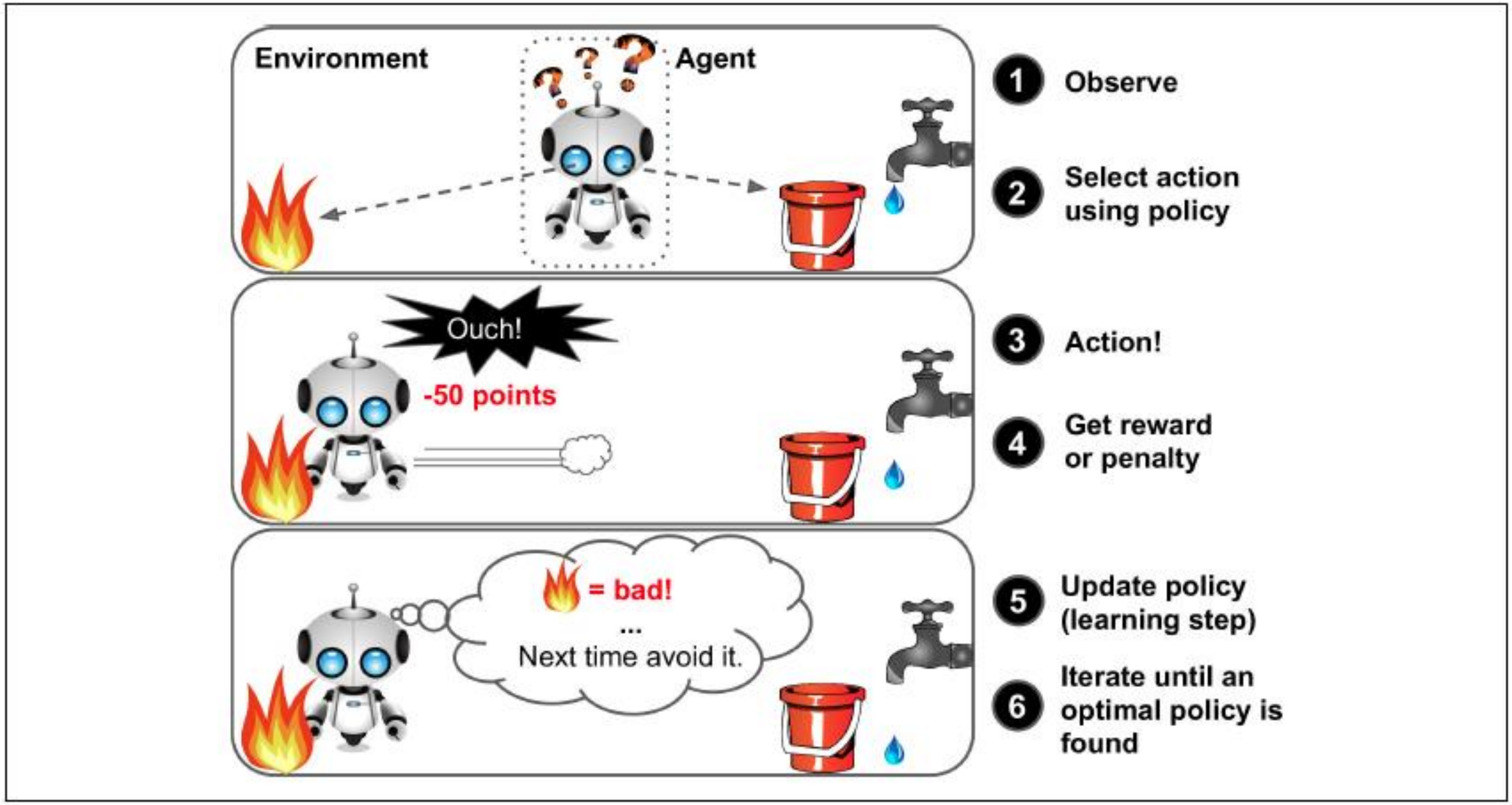
Anomaly detection

Reinforcement learning

- Agent
  - can observe the environment
  - select and perform actions
  - get rewards in return
    - penalties in the form of negative rewards

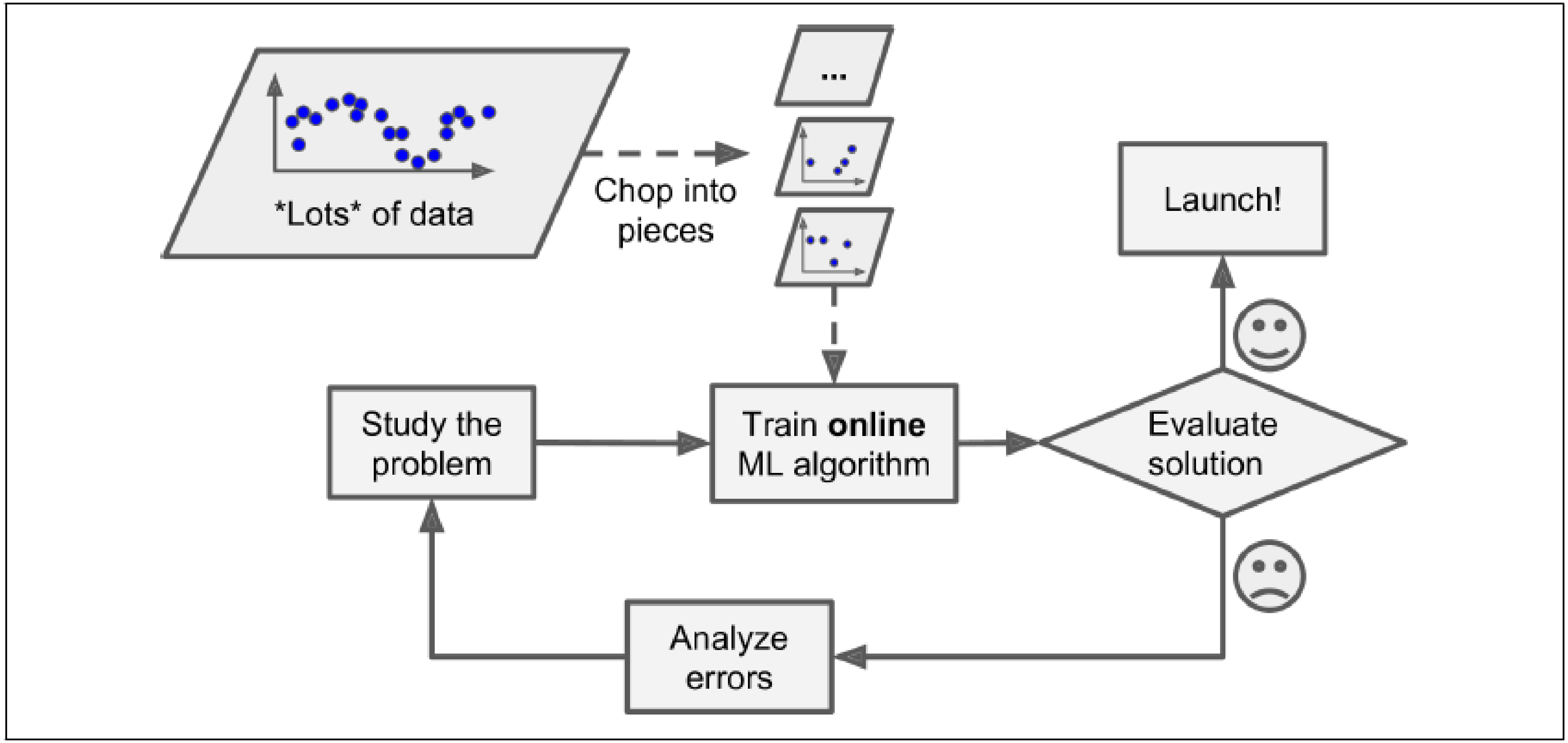
- Examples
  - DeepMind's AlphaGo program

Reinforcement learning



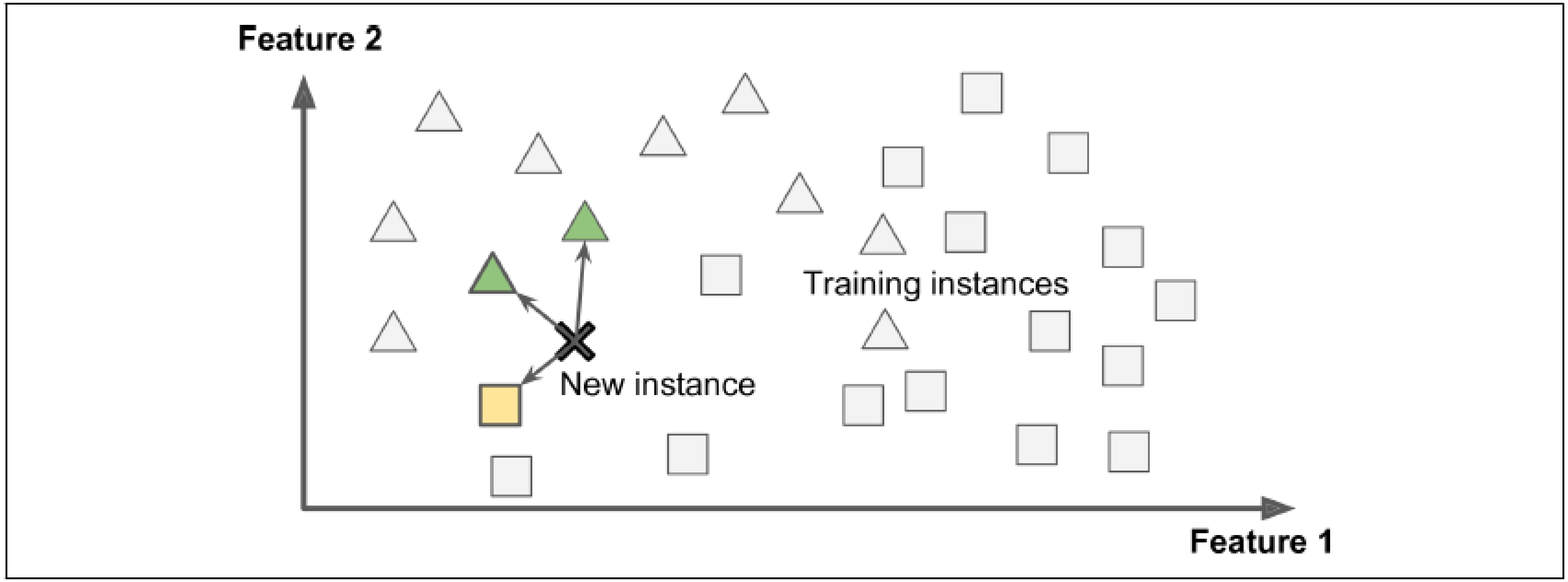
Reinforcement learning strategy

On-line learning



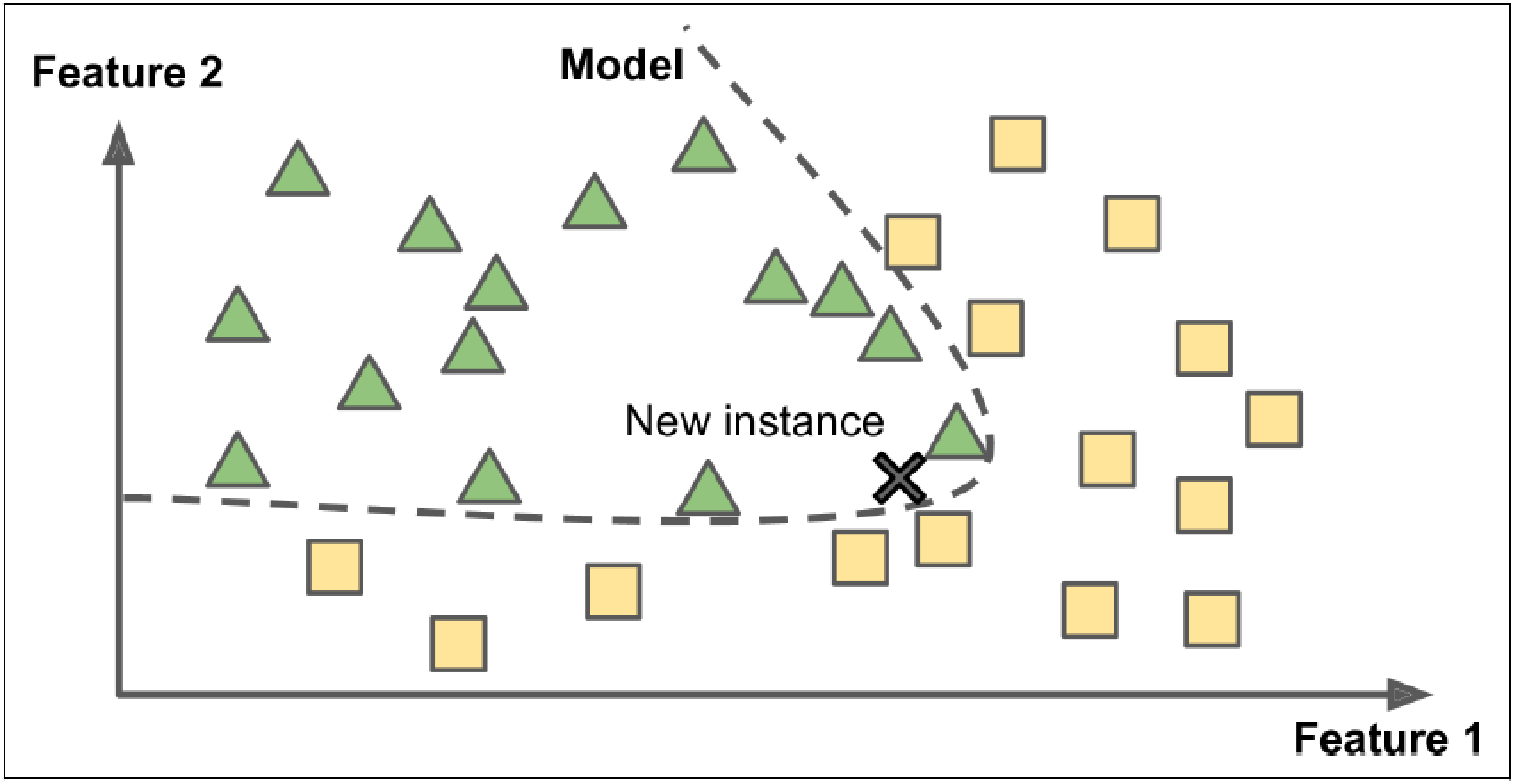
On-line learning to handle huge datasets

Istance-based learning



Instance-based learning

Model-based learning

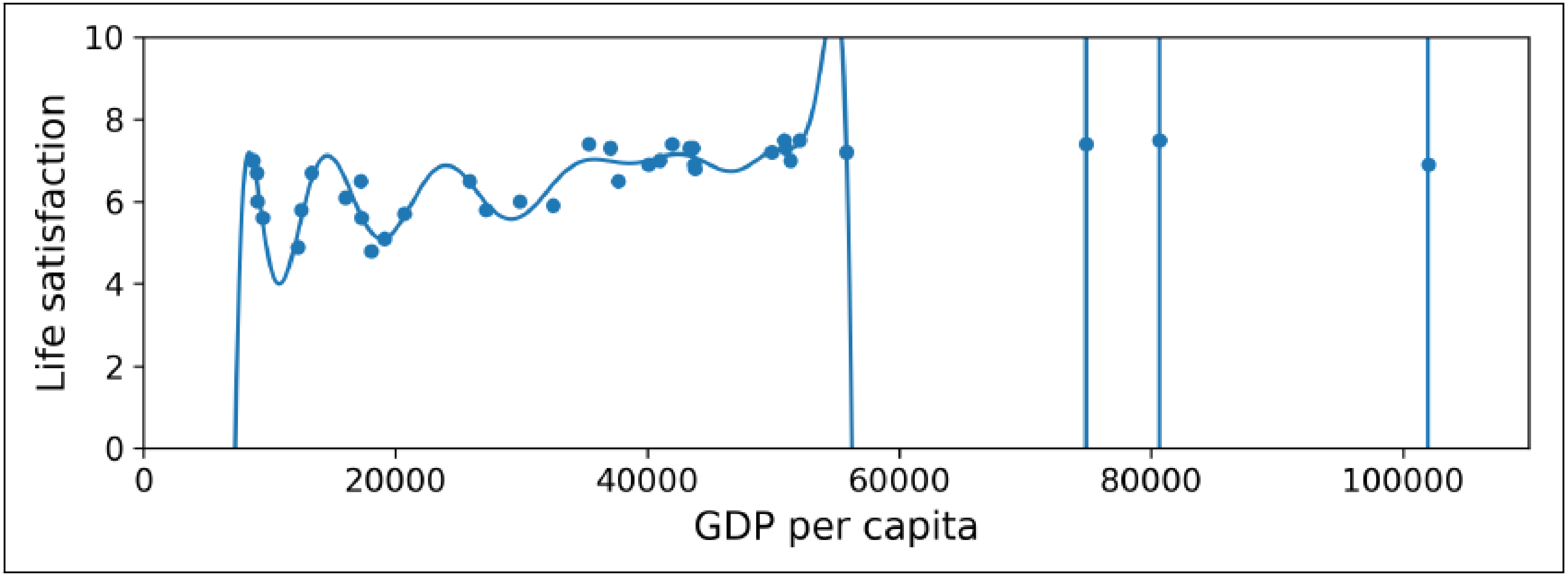


Model-based learning

Data

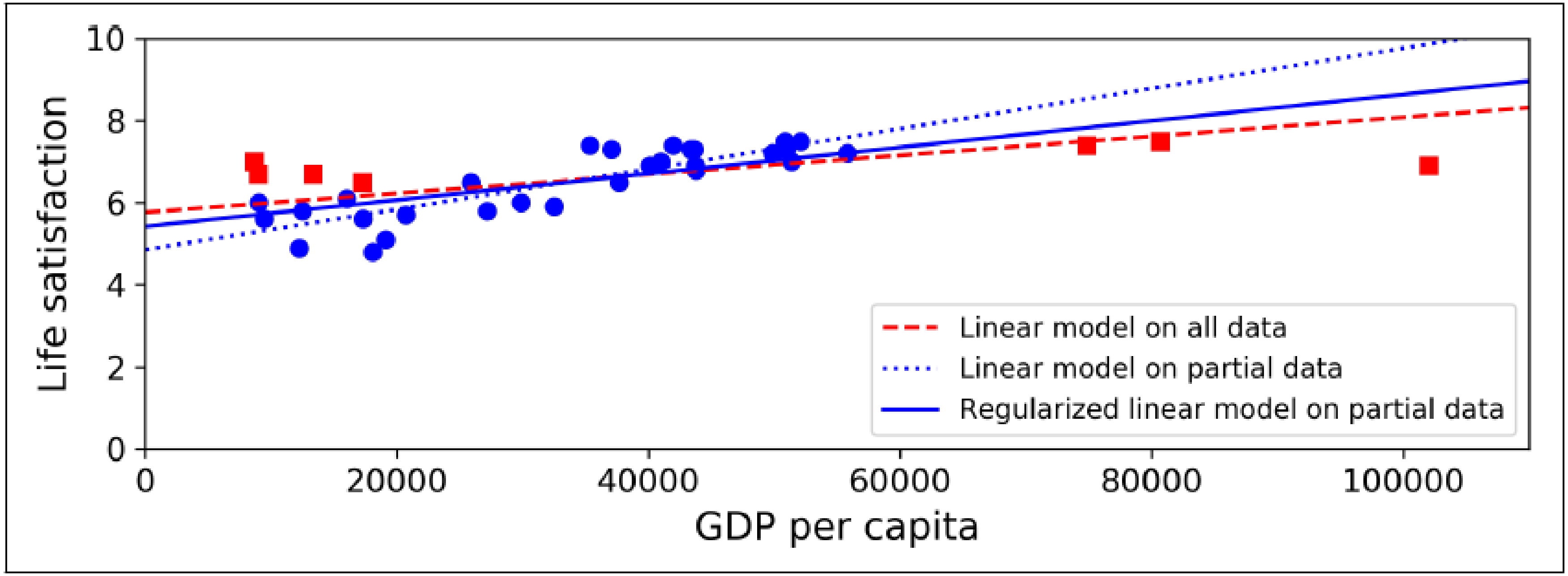
- Feature engineering
  - Feature selection
  - Feature extraction
- Data generalizzation
  - Overfitting
  - Underfitting

Overfitting



Overfitting of the data (regularization should be used)

Overfitting



Using regularization for avoiding overfitting of the data

Data mismatch

- Data
  - It is easy to get a large amount of data for training bit it is not perfectly representative of the data that will be used in production

- No Free Lunch Theorem
  - David Wolpert
  - 1996 – Article titled: The lack of *a priori* distinctions between learning algorithms
  - *If you make absolutely no assumption about the data, then there is no reason to prefer one model over any other*