Introduction to AI

MACHINE LEARNING

Pedro Meseguer
Institut d'Investigació en Intel·ligència Artificial (IIIA)
Consejo Superior de Investigaciones Científicas (CSIC)

1

Contents

1. Introduction
2. Supervised learning: Decision trees
3. Unsupervised learning: Clustering
4. Reinforcement learning
5. Neural learning
   - Neural networks
   - Deep learning: GANs
6. ML names
7. Wrap-up

2

1.Machine Learning

- When a system improves performance after observations about the world
- Reasons for learning:
  - Unexpected future situations; capacity of generalization
  - Manual knowledge coding: hard and time consuming task
  - Perceived as more intelligent if shows adaptation
  - When there is no solving method

3

3

1.Machine Learning (cont.)

- Types of learning:
  - Supervised: with a teacher (examples are pairs input-output)
  - Unsupervised: without teacher (inputs only)
  - Reinforcement: agent acting, a reward/penalty aften many actions
- Existence of prior knowledge:
  - EBL: Explanation-based Learning
- Techniques of learning
  - Symbolic: Decision trees…
  - Neural: NNs // Deep learning

4

4

2.Supervised Learning

- It is learning with a teacher
- In the learning process, the system gets positive and negative examples
- Each example:
  - Input: vector of attributes
  - Correct output:
    - Discrete: classification task [yes/no (Boolean learning)]
    - Continuous: regression task (for instance, predicting temperature)
- From the set of examples
  - Take a subset, to perform training (training set)
  - Another subset to perform validation (validation set)
  - A third subset to assess system performance (test set)

5

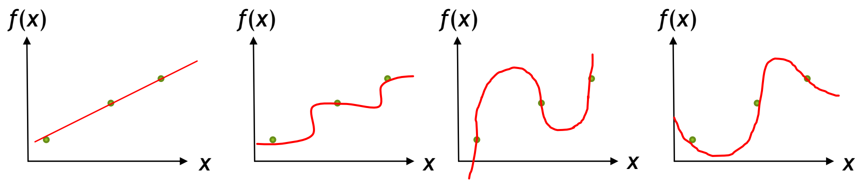
2.Supervised Learning: The examples

- Input-output pairs (each input with the correct output)
- The task of interest is to predict outpus of unseen inputs
- Input: factored representation – a vector of attribute values
- Style:
  - Symbolic
  - Neural

6

2.Sup. Learning: Hypothesis function

- Training set:
  - Pairs (x_1, y_1)… (x_n, y_n): x_i inputs, y_i outputs
  - Unknown funtion $f(x)=y$
  - Looking for function *h* (called hypothesis), such that *h* approximates *f*

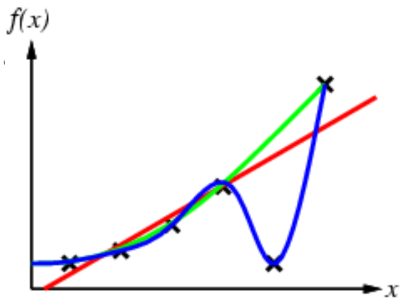


- Test set:
  - Sample of pairs (x_i, y_i), different from the training set
  - To assess if *h* valid

7

2.Inductive learning

- Construct/adjust *h* to agree with *f* on training set (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



- Ockham's razor: prefer the simplest hypothesis consistent with data. Can we find the blue curve easily? → decision trees

8

2.Example: Learning decision trees

Decide whether to wait for a table at a restaurant, based on the attributes:

- Alternate: is there an alternative restaurant nearby?
- Bar: is there a comfortable bar area to wait in?
- Fri/Sat: is today Friday or Saturday?
- Hungry: are we hungry?
- Patrons: number of people in the restaurant (None, Some, Full)
- Price: price range ($, $$, $$$)
- Raining: is it raining outside?
- Reservation: have we made a reservation?
- Type: kind of restaurant (French, Italian, Thai, Burger)
- WaitEstimate: estimated waiting time (0-10, 10-30, 30-60, >60)

9

2.Attribute-based representations

- 9216 possible combinations; 12 examples class T /F
- Examples described by attribute values (Boolean, discrete)

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

10

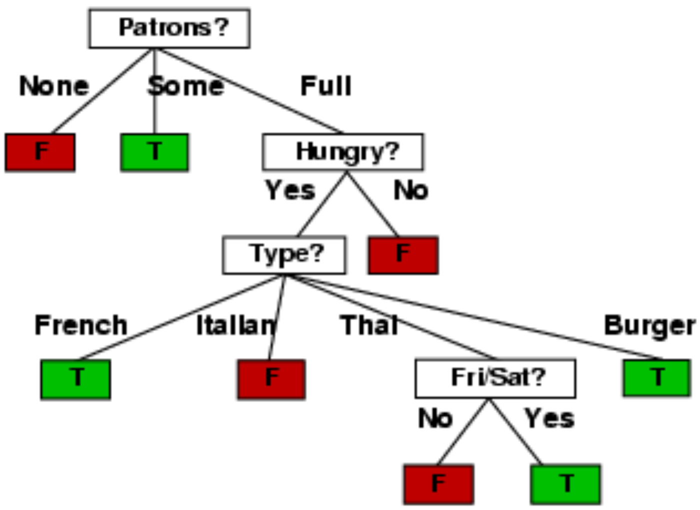
2.What is a decisión tree?

- It is a classifier with a tree structure:
  - Internal nodes: check for an attribute
    - As many successors as values for the attribute
    - At each successor you locate the example(s) with that value
  - Leaves:
    - Contain positive examples only (green)
    - Contain negative examples only (red)
- Selection of attributes in the tree: essential to obtain small trees

11

11

2.Decision tree learned



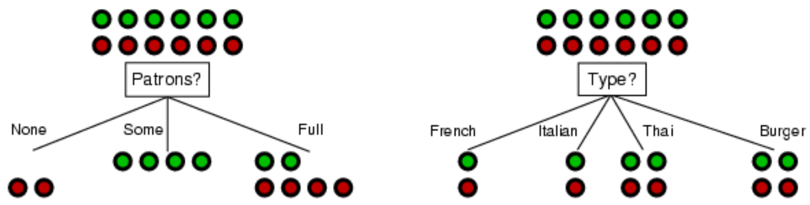
implements function h

12

2.Decision tree building

Choosing an attribute:

- Idea: a good attribute splits the examples into subsets that are (ideally) "all positive" or "all negative”



- *Patrons?* is a better choice

13

2.Decision tree building (cont.)

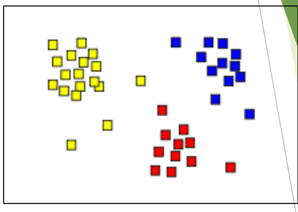
- A chosen attribute *A* divides the training set *E* into subsets E_1, … , E_v according to their values for *A*, where *A* has v distinct values.
- Information Gain (IG) or reduction in entropy from the attribute test
- Choose the attribute with the largest IG

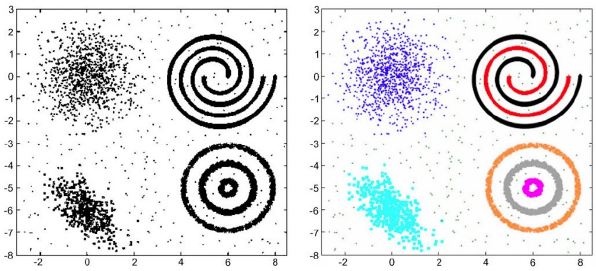
14

14

3.Unsupervised Learning

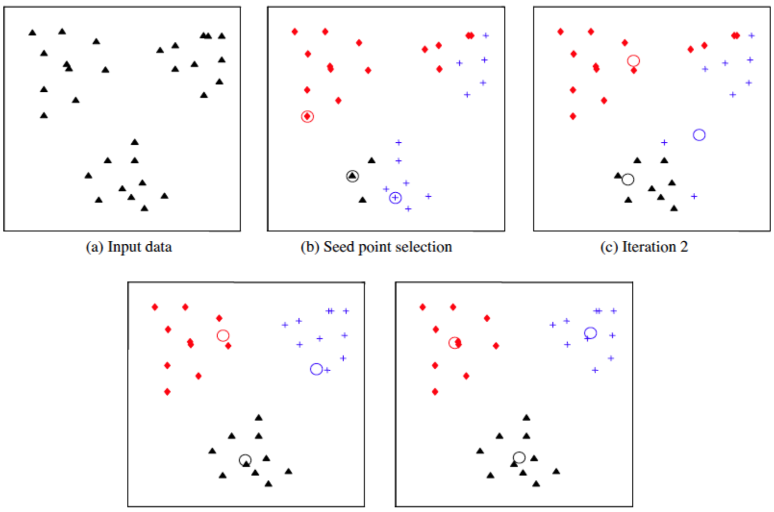
- Without teacher
- Inputs, but not the correct outputs
- Clustering: grouping inputs according to their characteristics
- To find structure in the data





15

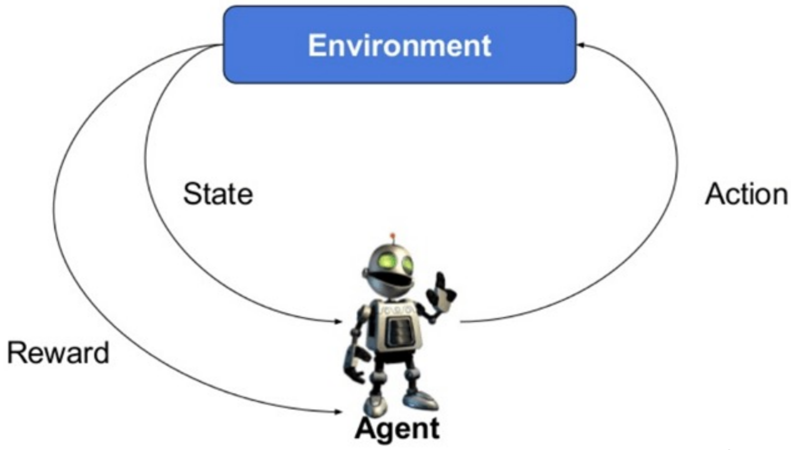
3.Unsupervised Learning: K-means



(a) Input data
(b) Seed point selection
(c) Iteration 2
(d) Iteration 3
(e) Final clustering

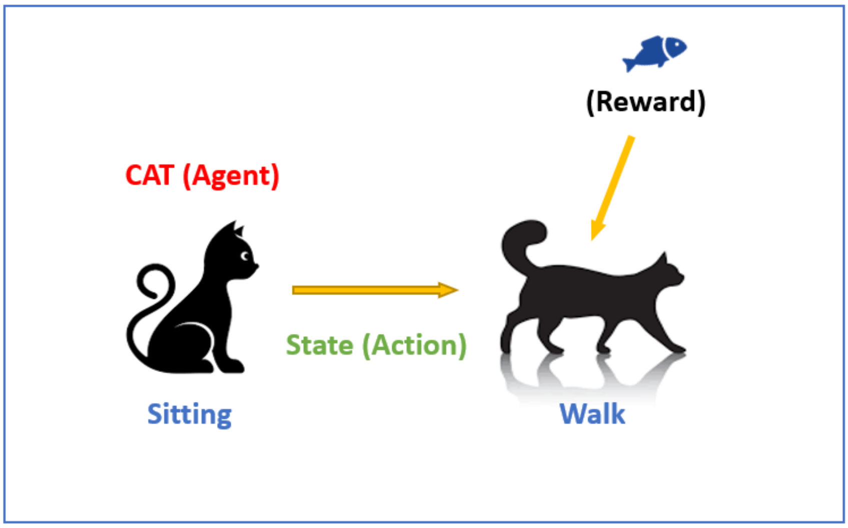
16

4.Reinforcement Learning



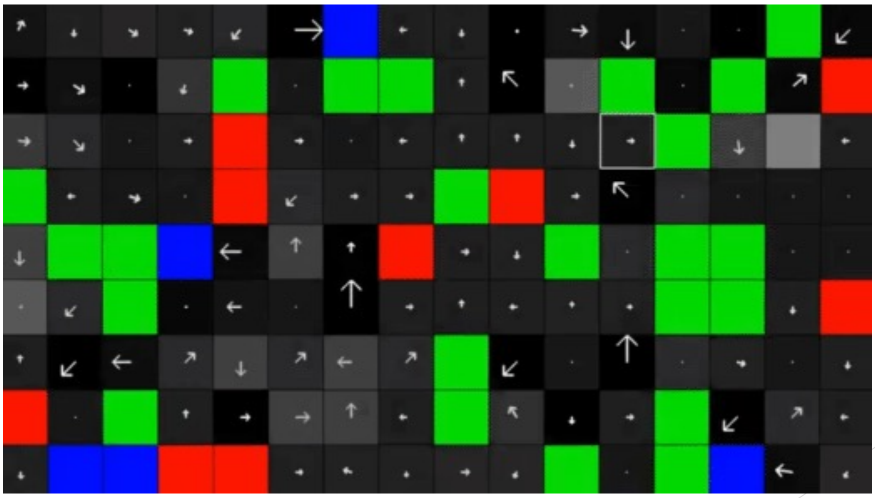
17

4.Reinforcement Learning (II)



18

4.Escaping from mazes



https://towardsdatascience.com/maze-rl-d035f9ccdc63

19

19

4.RL: summary

- Goal: maximize cumulative rewards over a time horizon
- Trial and error
- There is no supervisor, only a real number or reward signal
- The agent learns a “good” sequential decision making
- Time plays a crucial role in Reinforcement problems
- Feedback is always delayed, not instantaneous
- Agent’s actions determine the subsequent data it receives
- RL treatment is rather technical

20

20

5.Neural Learning

- Artificial neural networks:
  - Layers: input, hidden, output
  - Artificial Neurons or units
  - Training: backpropagation
- Deep neural networks (deep learning):
  - Autoencoders
  - Architectures (convolutional)
  - Generative Adversarial Networks

21

21

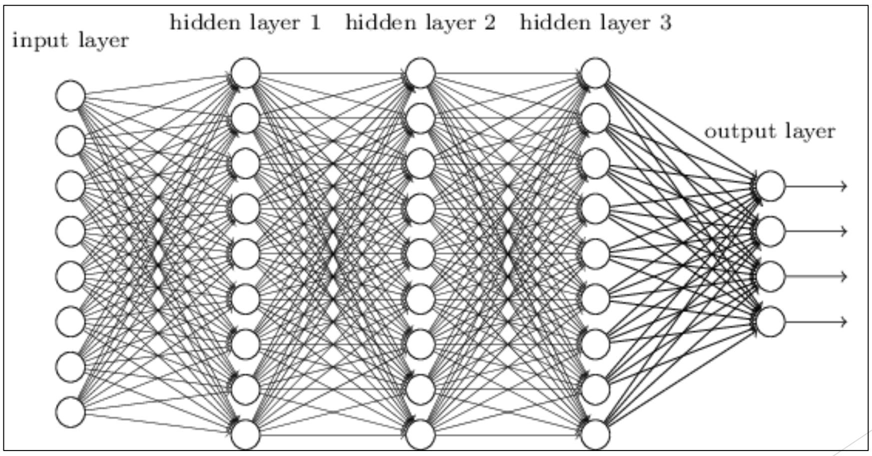
5.Artificial Neural Networks

- Inspired by mathemathical models of the biological brain
- Artificial Neurons — Connections — Signals
- Artificial Neuron Model [McCulloch & Pitts 1943]
- Perceptron: input-output layers [Minsky & Papert 1969]
- Hidden layers (1980s):
  - Feed-forward NN
  - Recurrent network

22

22

5.Feed-forward NNs: Layers

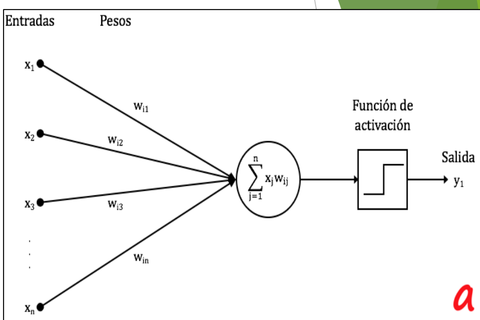


23

23

5.Artificial Neuron

- Simple computation at each neuron
  - Connected with all neurons of previous level
  - Each connection has a weight *Wij* (between neuron *i* and neuron *j*) in [-1,1]
  - Output 0 / 1
- Computes this function $\sum_{j=1}^{n} x_j w_{ij}$
  - If the sum of inputs times their weights is greater than the theshold: output 1
  - otherwise: output 0



24

24

5.ANNs: Backpropagation

- Supervised learning schema
- Error: correct output - real output (vector)
- Culprits?
  - Weights distributed across the network
  - Backpropagation: gradiend descend in parameter space
- In practice: training from examples is an iterative procedure
  - Take an example
  - Execute: input → output
  - Determine error and change weights by backpropagation
  - Until minimum error or exhausting resources

25

25

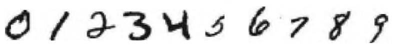
5.ANNs: Design

- Usually input and output layers can be easily decided
  - #hidden layers
  - #neurons at each hidden layer
  - *trial and error*
  - Weights: automatically adjusted by backpropagation
- Caution: backpropagation changes in weights
  tend to vanish if many layers

26

26

5.ANNs: Example



- Handwritten digit recognition
  - Useful in the automation of many tasks
    - Post offices
    - Banks
  - NN
    - Input layer: 400 units (20 x 20 image, one unit per pixel)
    - One hidden layer: 300 units (experimentally determined)
    - Output layer: ten units (one per class)

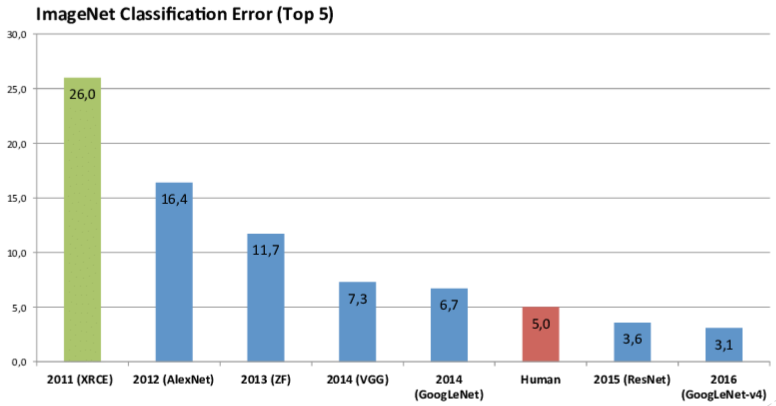
27

5.Deep Learning

- Recent approach
- Outstanding results
- It needs huge amounts of data for training
- Vision and natural language:
  - Currently are based on DL models
  - The whole thing started in 2012, with a contribution to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC)
  - AlexNet (a deep learning network) won, 10% better than its predecesor (error 26% → 16%)
  - Since then, “deep” is pervasive in AI

28

5.Deep Learning, ImageNet

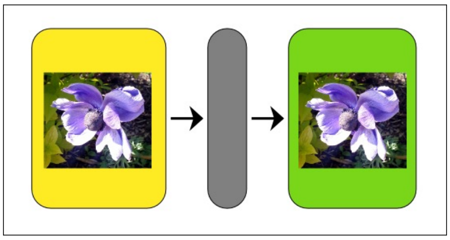


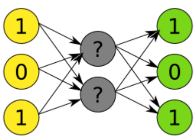
Human performance: 5% error

29

5.Deep Learning: Autoencoders

- Imagine a NN with 3 layers, which produces an output exactly equal to the input: encoder/decoder
- NN of one hidden layer, smaller (or larger) than input/output layer





30

5.Deep Learning: Autoencoders (II)

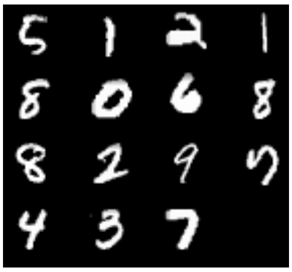
- Backpropagation computes weights involving hidden layer
- It forces the hidden layer to produce a "new" representation = it discovers an alternative representation
- Interesting:
  - this representation identifies problema characteristics
  - it is meaninful to our eyes: it captures meaning
- Once the autoencoder is trained, the decoder is no longer useful

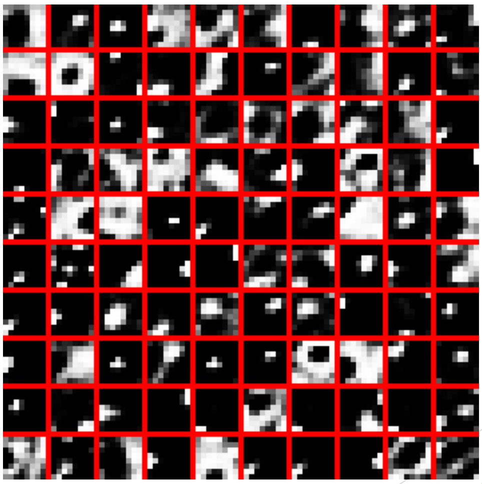
31

31

5.Deep Learning: Example

Handwritten digits





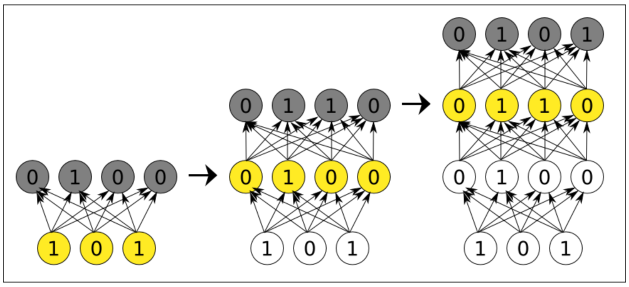
Patterns in the hidden layer

32

32

5.Deep Learning: Nested Autoencoders

- The hidden layer of an autoencoder → input layer for another autoencoder



- The 2nd hidden layer discovers a more sophisticated representation
- Nest autoencoders, until finding the representation that is meaninful for your problem

33

33

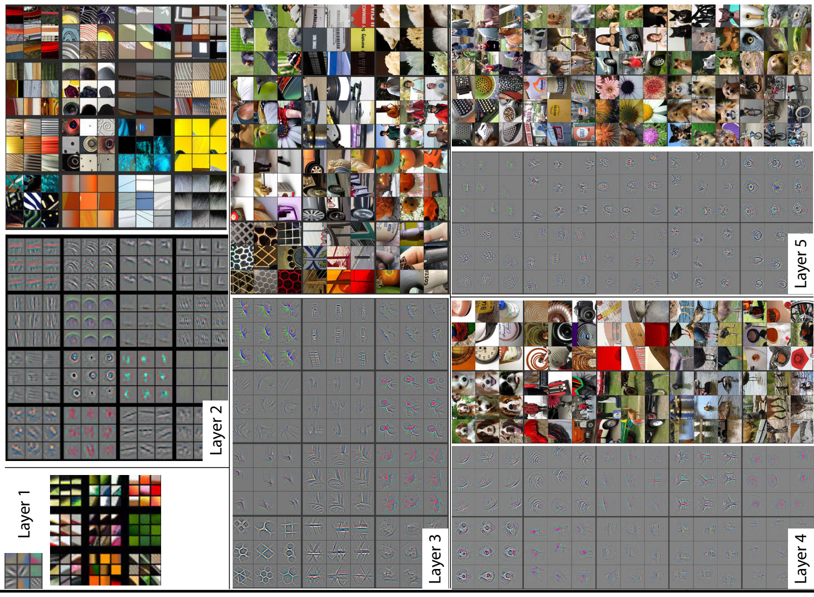
5.Deep Learning

- Unsupervised learning: features are automatically discovered
- With nested autoencoders, we can train networks as deep as we want
  - Train one autoencoder at time
  - Backpropagation problems do not happen

34

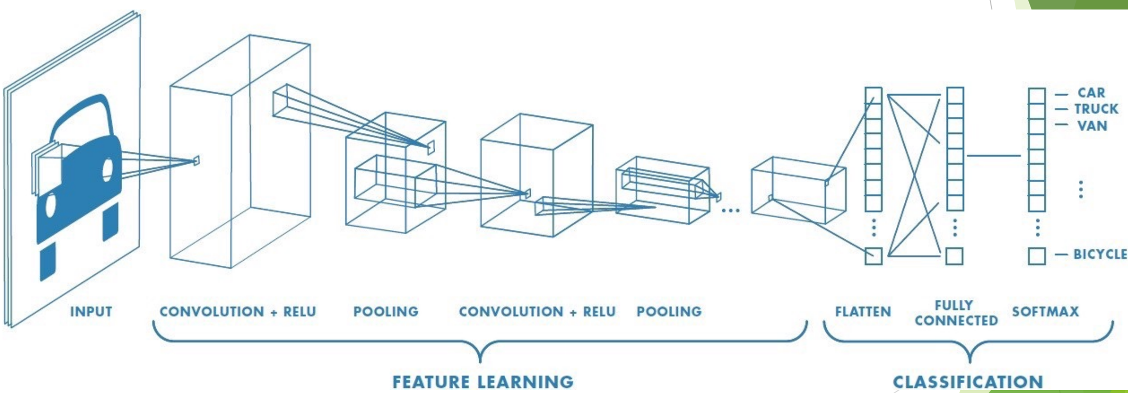
34

5.Deep Neural Networks: Example



35

5.Deep Learning: Convolutional Network



36

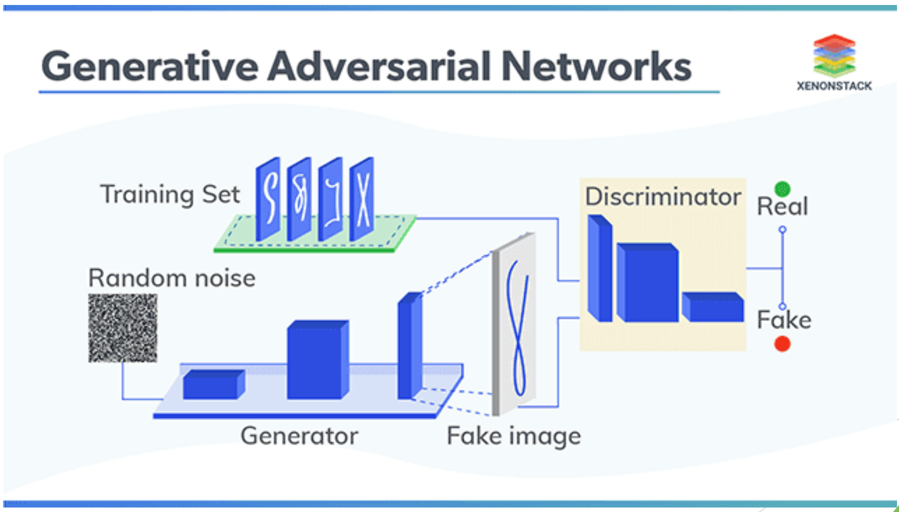
5.Deep Learning: GANs (I)

- Two Deep networks:
  - DISCRIMINATOR
    - Trained with real images
    - Connected to the GENERATOR
    - Detects if an image is real or synthetic
  - GENERATOR
    - Computes synthetic images
    - It tries to fool the DISCRIMINATOR
- On an image: if DISCRIMINATOR succeeds → backpropagation on GENERATOR
- After training, keep GENERATOR, discard DISCRIMINATOR

37

37

5.Deep Learning: GANs (II)



38

5.Deep fake

- Deep fakes: images of people that do not exist; also videos
- Generated by GANs
- From internet images

39

5. Deep fake video

40

6.ML names: Tom Mitchell

- (1951-)
- Full profesor Carnegie-Mellon
- Creator of EBL
- Author of the book Machine Learning
- Many contributions

41

41

6.ML names: Geoffrey Hinton

- (1947-)
- Full profesor univ Toronto, Google
- Co-creator of Deep Learning
- Many contributions to neural learning
- Turing Prize

42

42

6.ML names: Tom Ditterich

- (–)
- Emeritus full professor Univ. Oregon
- Many contributions to Machine Learning

43

43

6.ML names: Pedro Domingos

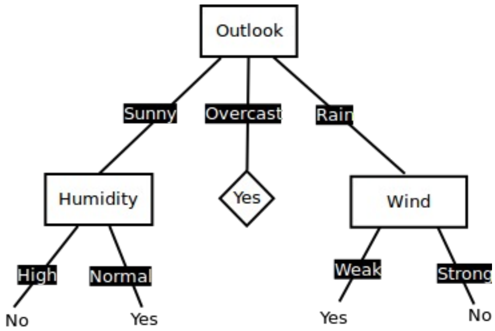
- (194? –), portuguese
- Emeritus full profesor Univ. Washington
- Many contributions to Machine Learning
- In particular, Markov logic network

44

44

4.ML names: ID3

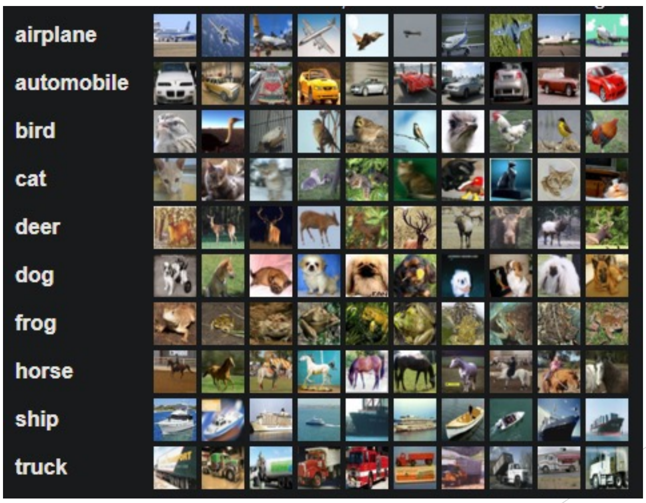
- Decision trees building algorithm
- Proposed by Quinlan in 1986



45

6.ML names: AlexNet

- 2012, the network that triggered deep learning



46

6. Wrap-up

1. Introduction
2. Supervised learning: with a teacher
   - Positive and negative examples: (input, output) pairs
   - Decision trees
3. Unsupervised learning: without teacher (inputs only)
   - Clustering: data points forming clusters
4. Reinforcement learning: reward/penalty after moves
5. Neural learning
   - Neural networks: design, backpropagation
   - Deep learning: new designs, GANs
6. ML names

47

47

Further reading

- Russel & Norvig 3rd ed:
  18.1, 18.2, 18.3, 18.7.1, 18.7.2, 18.7.3, 18.11.1, 20.3.1, 22.1 plus
  Bibliographical Notes chapter 18

- For the deep learning part:
  Deep Learning, Goodfellow, Bengio, Courville, The MIT Press
  accesible at https://deeplearningbook.org
  chapter 1, section 14.1

48

48