Analiza și procesarea datelor prin tehnici de Învățare Automată

2. Învățare Automată - concepte de bază

Șef Lucrări Dr. Ing. Horia Modran

Contact: horia.modran@unitbv.ro / modranhoria@gmail.com

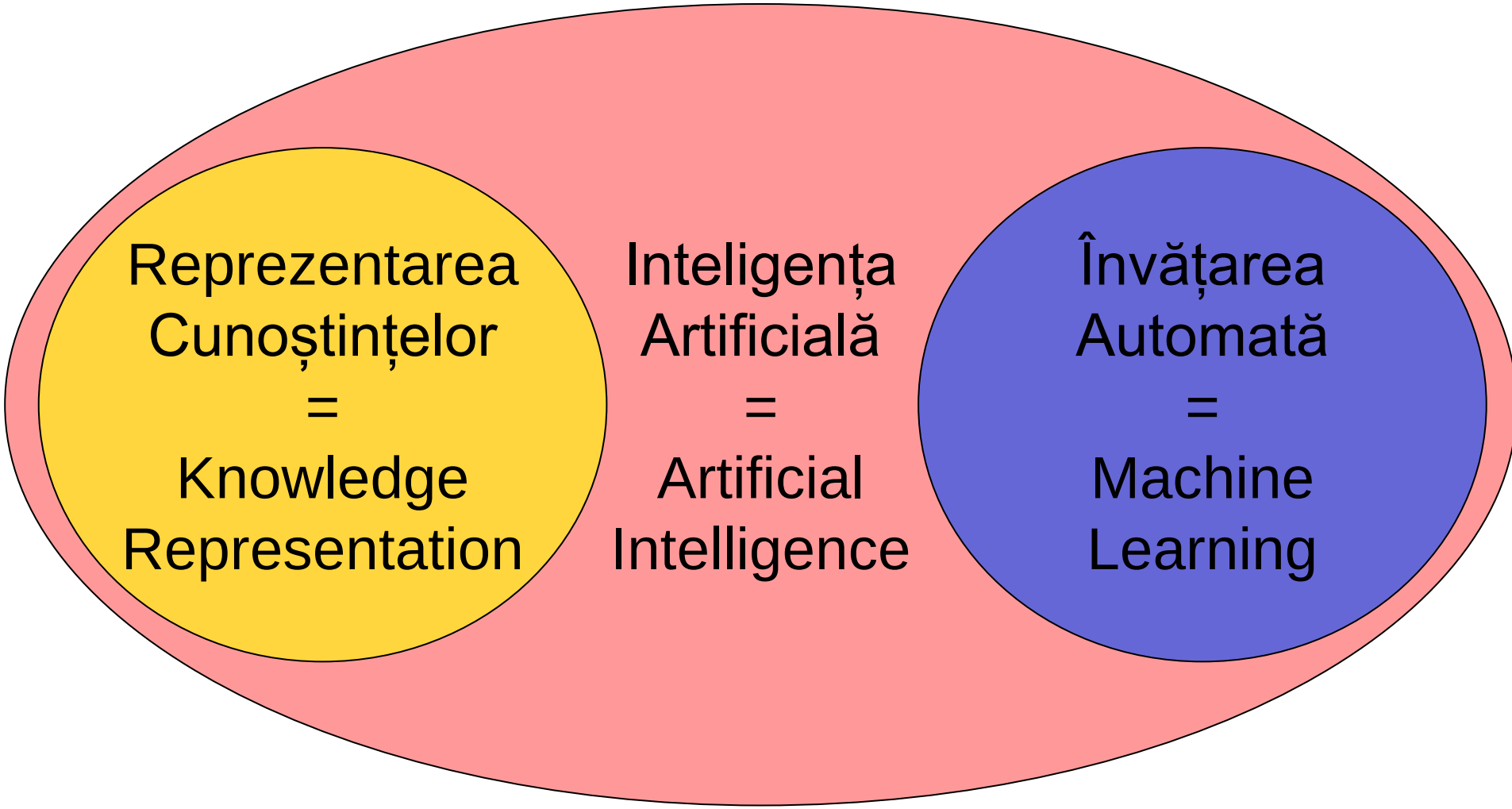
Tel: 0770171577

Universitatea Transilvania din Brașov

FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

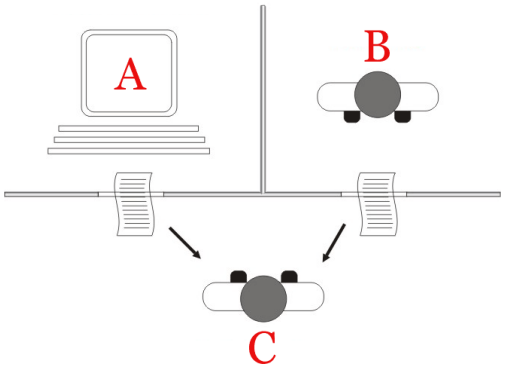
2024 - 2025

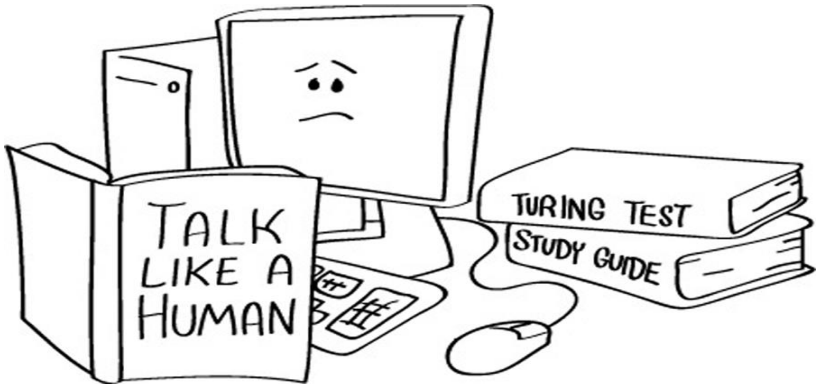
AI și ML


Reprezentarea Cunoștințelor
=
Knowledge Representation
Inteligența Artificială
=
Artificial Intelligence
Învățarea Automată
=
Machine Learning

Inteligența Artificială

- Scopul suprem al inteligenței artificiale este de a construi sisteme care să atingă nivelul de inteligență al omului
- Testul Turing: un computer prezintă un nivel de inteligență uman dacă un interlocutor uman nu reușește să distingă, în urma unei conversații în limbaj natural, că vorbește cu un om sau cu un calculator





Învățarea automată

- O mare parte din cercetători consideră că acest scop poate fi atins prin imitarea modului în care oamenii învață
- Învățarea automată – domeniu care studiază modul în care calculatoarele pot fi înzestrate cu abilitatea de a învăța, fără ca aceasta să fie programată în mod explicit
- În acest context, învățarea se referă la:
  - recunoașterea unor tipare / structuri (patterns) complexe
  - luarea deciziilor inteligente bazate pe observațiile din date

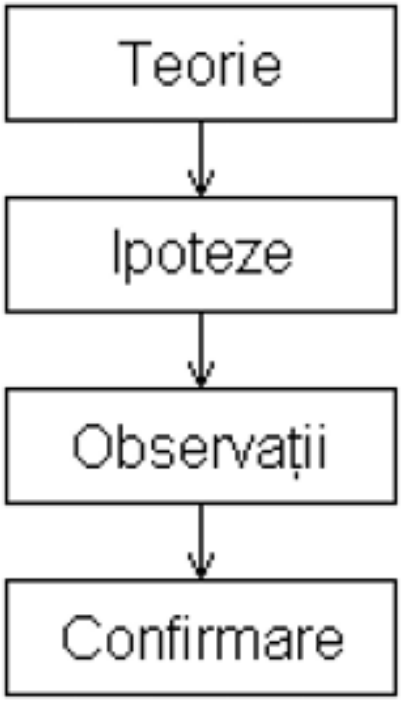
Ce este învățarea?

- Herbert Simon: „Învățarea este orice proces prin care un sistem îmbunătățește performanța prin experiență.”
- “Se spune că un program de calculator învață din experiența E cu privire la o anumită clasă de sarcini T și măsura performanței P, dacă performanța sa la sarcinile din T, măsurată de P, se îmbunătățește cu experiența E.

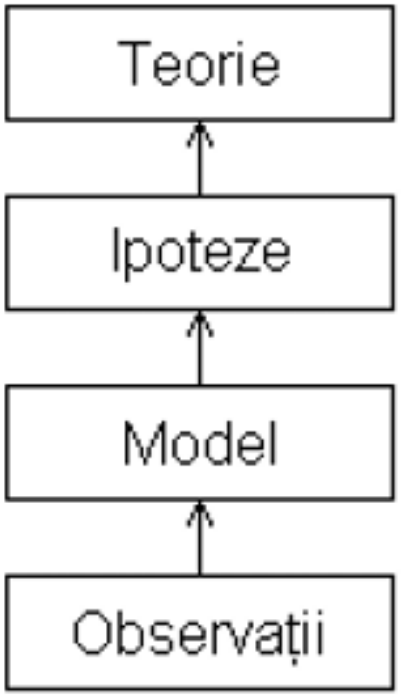
– Tom Mitchell

Tipuri de raționament

Raționament deductiv (general -> particular)



Raționament inductiv (particular -> general)



Învățare inductivă

- Cea mai simplă formă: învață o funcție din exemple
  - f este funcția țintă -> un exemplu este o pereche (x, f(x))
- sarcină de inducție pură:
  - având în vedere o colecție de exemple de f, returnează o funcție h care aproximează f.
  - se caută o ipoteză h, astfel încât h ≈ f, având în vedere un set de exemple de antrenament
- acesta este un model foarte simplificat de învățare reală:
  - ignoră cunoștințele anterioare
  - presupune că sunt furnizate suficiente exemple

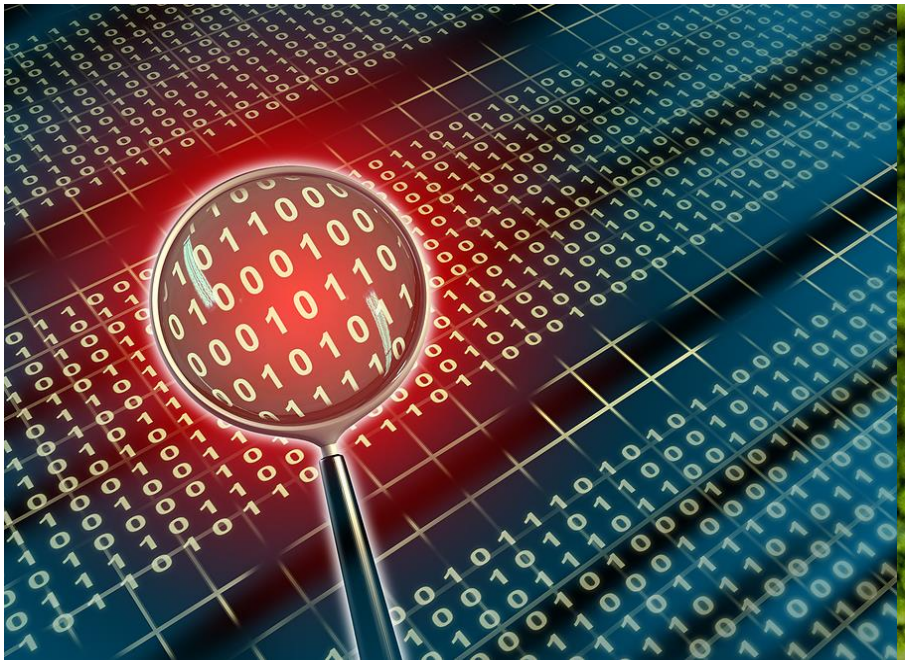
Aplicare

- Ce probleme pot fi rezolvate* folosind învățarea automată?
- Se aplică în situații în care este foarte greu (imposibil) să definim un set de reguli de mână / să scriem un program
- Exemple de probleme unde putem aplica învățarea automată:
  - Detectarea facială
  - Înțelegerea vorbirii
  - Prezicerea prețului acțiunilor
  - Recunoașterea obiectelor

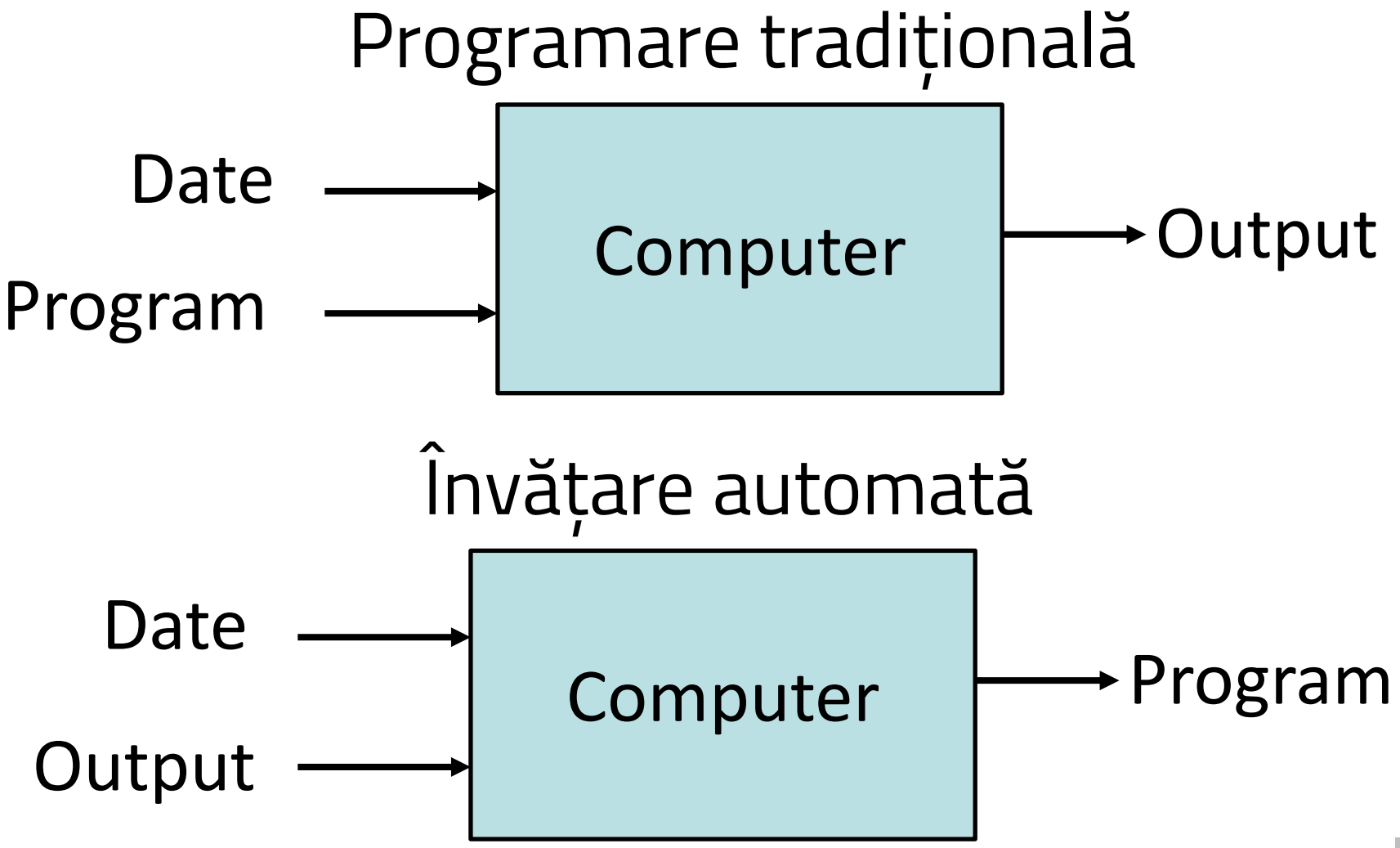
(*) rezolvate cu un anumit grad de acuratețe/încredere

Esența Învățării Automate

- Există un tipar/șablon
- Dar nu îl putem exprima programatic / matematic
- Avem date / exemple în care regăsim acest tipar



Paradigma


Programare tradițională
Date
Program
Computer
Output
Învățare automată
Date
Output
Computer
Program

Ce este Învățarea Automată

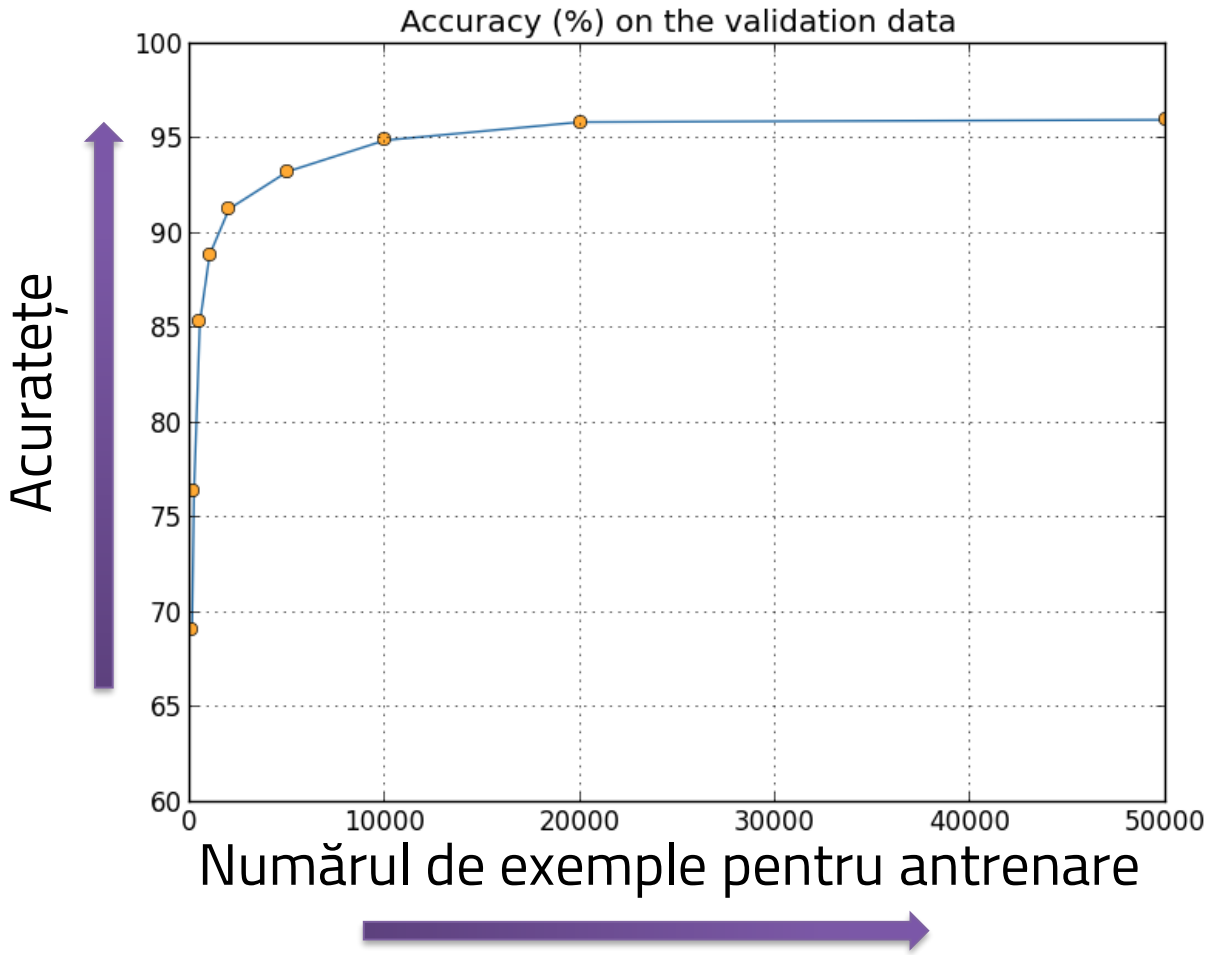
- **Arthur Samuel** (1959) - domeniu care oferă computerelor capacitatea de a învăța fără a fi explicit programate
- **Kevin Murphy** – algoritmi care:
  - detectează automat modele/șabloane în date
  - utilizează tiparele descoperite pentru a prezice date viitoare sau alte rezultate de interes
- **Tom Mitchell** – algoritmi care își îmbunătățesc performanța (P) la o anumită sarcină (T) cu experiență (E)

Scurt istoric al AI

- Anii 1950: Perceptronul lui Rosenblatt (1957)
- Anii 1960-1980: "AI Winter"
- Anii 1990: Rețelele neuronale domină, în principal datorită descoperirii algoritmului de propagare a erorii înapoi (*backpropagation*) pentru rețele cu mai multe straturi
- Anii 2000: Metodele kernel domină, în principal din cauza instabilității rețelelor neuronale
- Anii 2010: Revenirea la rețele neuronale, în principal datorită conceptului de învățare profundă (deep learning)

Prezent

- De ce funcționează în prezent?
  - Mai multă putere de calcul
  - Mai multe date
  - Modele mai bune



Esența ML

- Mii de algoritmi de învățare automata existenți
  - Cercetătorii publică sute de noi algoritmi în fiecare an
- Simplificând decenii de cercetare în domeniu, putem reduce învățarea automată la:
  - **Învățarea unei funcții f care să mapeze un input X către un output Y, anume $f: X \rightarrow Y$**
    - Exemplu:

      X: email-uri

      Y: {spam, non-spam}

Esența ML

- Input: X (imagini, texte, email-uri…)
- Output: Y (spam sau non-spam…)
- Funcție Target (necunoscută)
  - $f: X \rightarrow Y$ (realitatea / "adevărata" mapare)
- Date

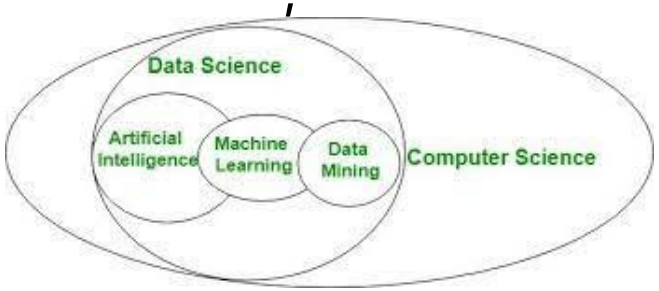
  $(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)$
- Model
  - $g: X \rightarrow Y$
  - $y = g(x) = sign(w^T x)$

Machine Learning

- Machine learning: studiul algoritmilor care se pot îmbunătăți automat prin experiență și prin utilizarea datelor
  - Cum să dezvolți un model pe baza datelor/experienței
    - Învățând parametri (e. g. Probabilități)
    - Învățând structuri (e. g. Rețele Bayesiene)
    - Învățând concepte ascunse (e. g. clustering)

Machine Learning și Data Mining

- Învățarea automată și Data Mining folosesc adesea aceleași metode și se suprapun în mod semnificativ
- Învățarea automată se concentrează pe predicție, pe baza proprietăților cunoscute învățate din datele de antrenament
- Data Mining se concentrează pe descoperirea de proprietăți (anterior) necunoscute în date (etapa de analiză a descoperirii cunoștințelor în baze de date)
- Data Mining folosește multe metode de învățare automată, dar cu scopuri diferite



Paradigme ale învățării

- Învățare supervizată: sunt furnizate date și etichete
- Învățare nesupervizată: sunt furnizate numai datele
- Învățare semi- supervizată: unele (dacă nu toate) etichete sunt prezente
- Învățare prin consolidare: un agent care interacționează cu lumea exterioară face observații, ia decizii și este recompensat sau pedepsit; ar trebui să învețe să aleagă acțiunile în așa fel încât să obțină o mulțime de recompense

Învățare supervizată

- Învățarea supervizată este sarcina de a învăța o funcție care mapează o intrare la o ieșire pe baza exemplelor de perechi intrare-ieșire
- deduce o funcție din datele de antrenament etichetate
- fiecare exemplu constă dintr-un obiect de intrare și o ieșire
- un algoritm analizează datele de antrenament și produce o funcție dedusă, utilizată pentru maparea de noi exemple
- scenariu optim: modelul ML va determina corect etichetele de clasă pentru instanțe nevăzute
- calitatea statistică - măsurată prin eroarea de generalizare

Terminologie

- Input: X – vector de p componente (intrări, regresor, caracteristici, variabile independente)
- Output: Y – datele de ieșire (variabila dependentă, țintă)
- Variabilă cantitativă (ex, vârsta, înălțimea, preț, venit, etc.) vs. variabilă calitativă (genul unei persoane, diagnotic medical, etc.)
- În problemele de regresie Y este o variabile cantitativă (ex: preț, tensiunea arterială, etc.)
- În problemele de clasificare Y ia valori într-o mulțime finită Datele de antrenare sunt de forma (x_1, t_1), (x_2, t_2), ,...,(x_n, t_n).

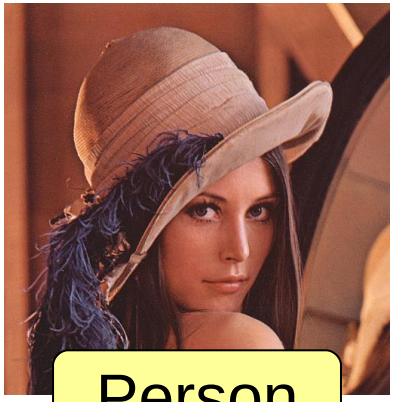
Exemplu 1: Clasificare imagini

- Avem la dispoziție exemple de obiecte etichetate
- Exemplu 1: recunoașterea obiectelor din imagini cu eticheta obiectelor conținute

Car

Person

Car



Person

Dog

Exemplu 2: Filtrare spam-uri

Input: email

Output: spam/ham

Setup:

- Get a large collection of example emails, each labeled "spam" or "ham"
- Note: someone has to hand label all this data!
- Want to learn to predict labels of new, future emails

Features: The attributes used to make the ham / spam decision

- Words: FREE!
- Text Patterns: $dd, CAPS
- Non-text: SenderInContacts
- …

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

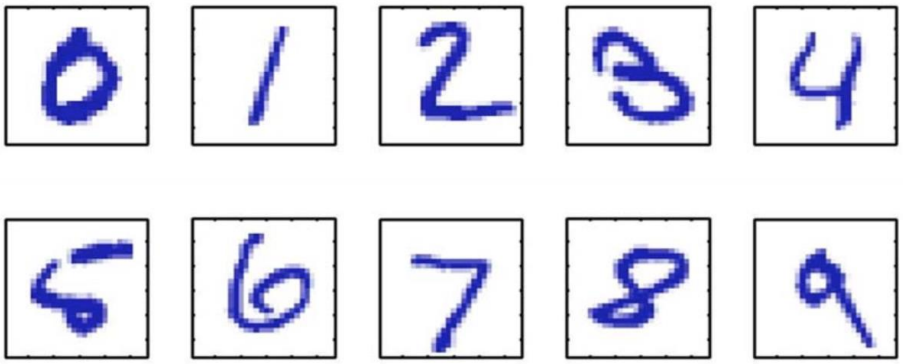
TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

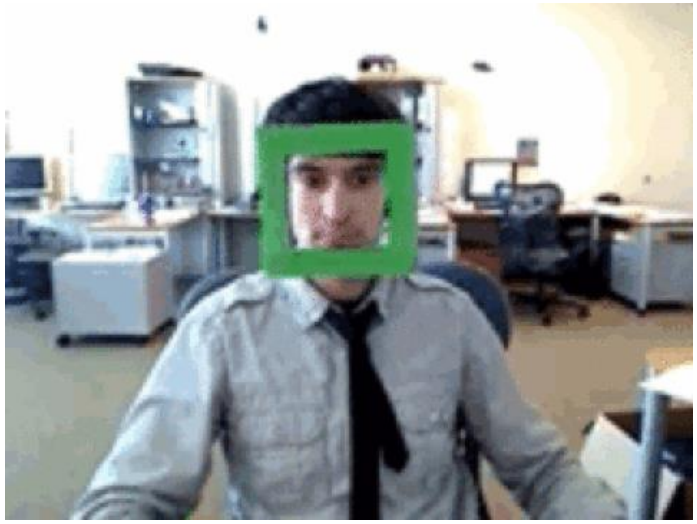
Examplu 3: Recunoașterea cifrelor

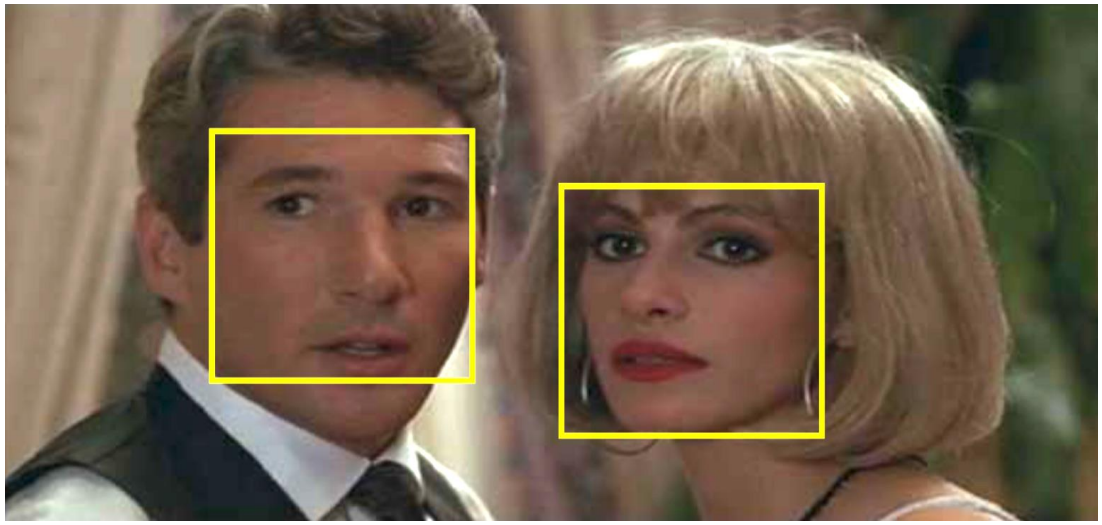
- Recunoașterea caracterelor scrise de mână (setul de date MNIST)
- Imagini de 28 x 28 de pixeli
- Reprezentăm o imagine ca un vector x cu 784 de componente
- Antrenăm un clasificator f(x) astfel încât:
  - $f : x \rightarrow \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$



Examplu 4: Recunoașterea facială

- O abordare constă în plimbarea unei ferestre peste imagine
- Scopul este să clasificăm fereastra într-una din cele două clase posibile: față sau non-față (transformarea problemei într-una de clasificare)



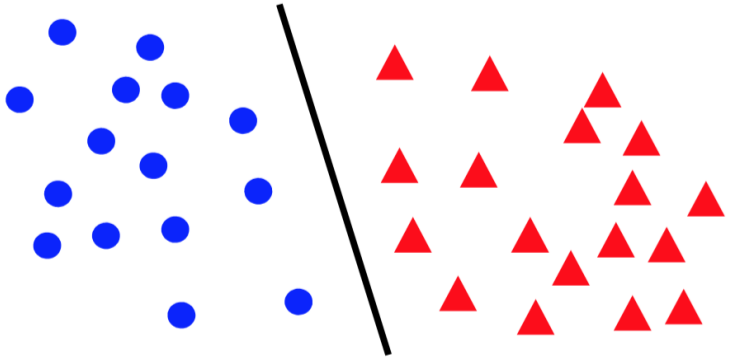


Examplu 5: Predicție preț

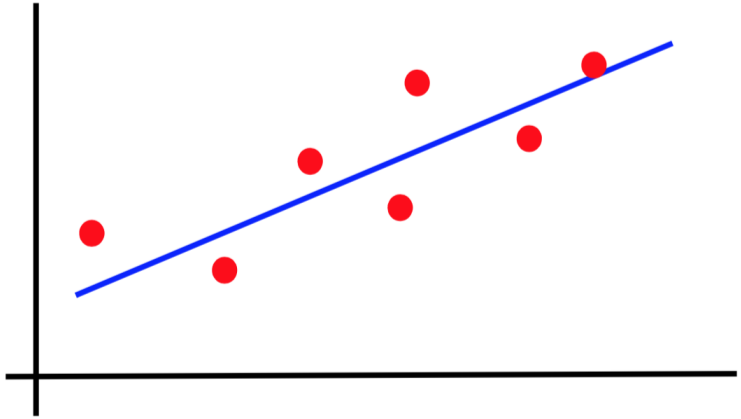
- Scopul este de a prezice prețul la o dată din viitor, de exemplu peste câteva zile
- Acesta este un task de regresie, deorece output-ul este unul continuu

Forme canonice

- Clasificare



- Regresie



Exemple de clasificare

- În clasificare, prezicem etichetele y (clasele) pentru intrările x
- Exemple:
  - OCR (intrare: imagini, clase: caractere)
  - Diagnostic medical (input: simptome, clase: boli)
  - Evaluator automat de eseuri (input: document, clase: note)
  - Detectarea fraudei (input: activitate contului, clase: fraudă / fără fraudă)
  - direcționarea e-mailului către serviciului clienți
  - articole recomandate într-un ziar, cărți
  - identificarea secvenței de ADN și proteine
  - investitii financiare

Paradigma de învățare supervizată

Functions $\mathcal{F}$

$f : \mathcal{X} \to \mathcal{Y}$

Training data

$\{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}$

LEARNING

$\text{find } \hat{f} \in \mathcal{F}$
$\text{s.t. } y_i \approx \hat{f}(x_i)$

Learning machine

PREDICTION

$y = \hat{f}(x)$

New data

x

Algoritmi de învățare supervizată

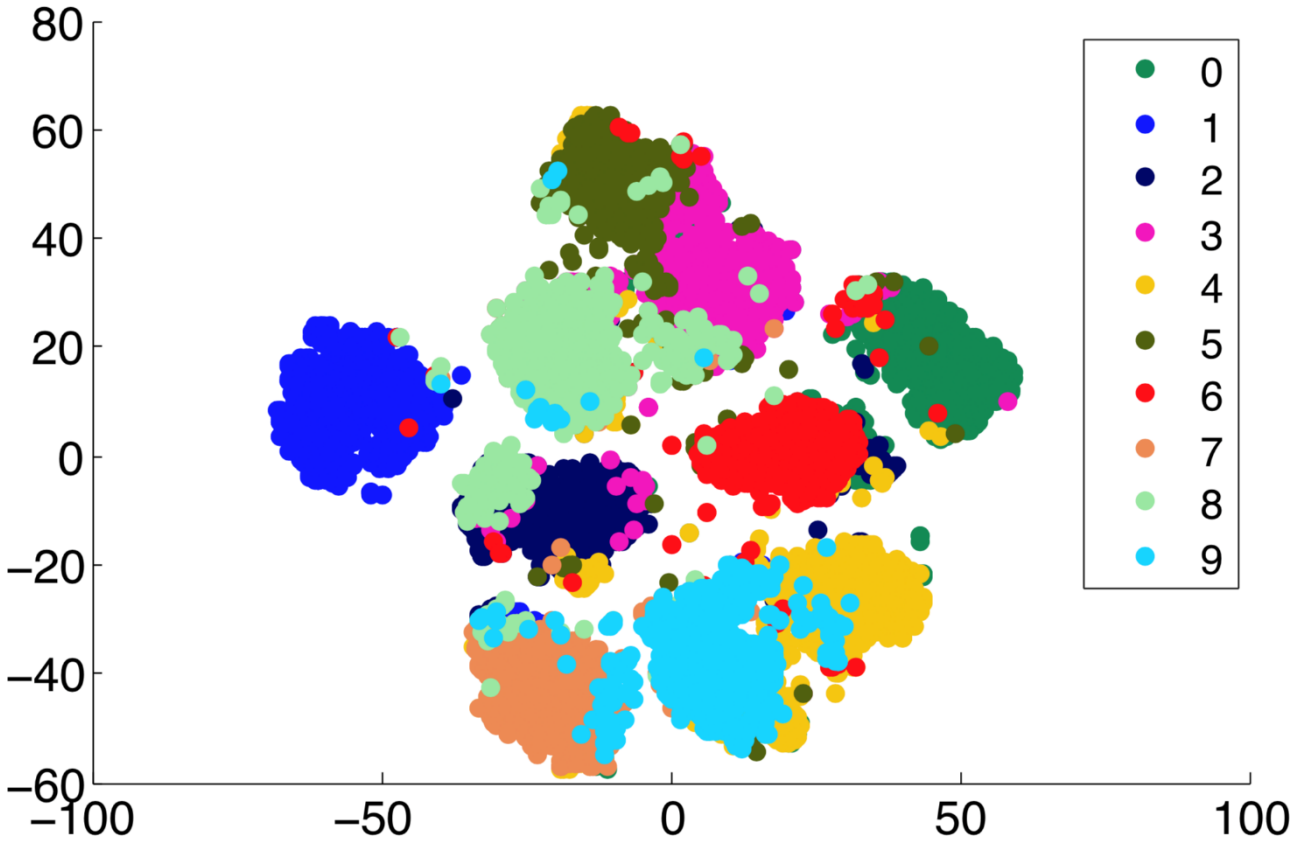
- Clasificatorul Bayes naiv
- Metoda celor mai apropiați vecini
- Clasificatorul cu vectori suport (SVM)
- Metode kernel
- Rețele neuronale și învățare “deep”
- Arbori de decizie și random forests
- ... și altele

Învățare nesupervizată

- Avem la dispoziție exemple de obiecte fără etichete
- Exemplu 1: gruparea imaginilor după similaritate

Exemplu 2: Clustering pe MNIST

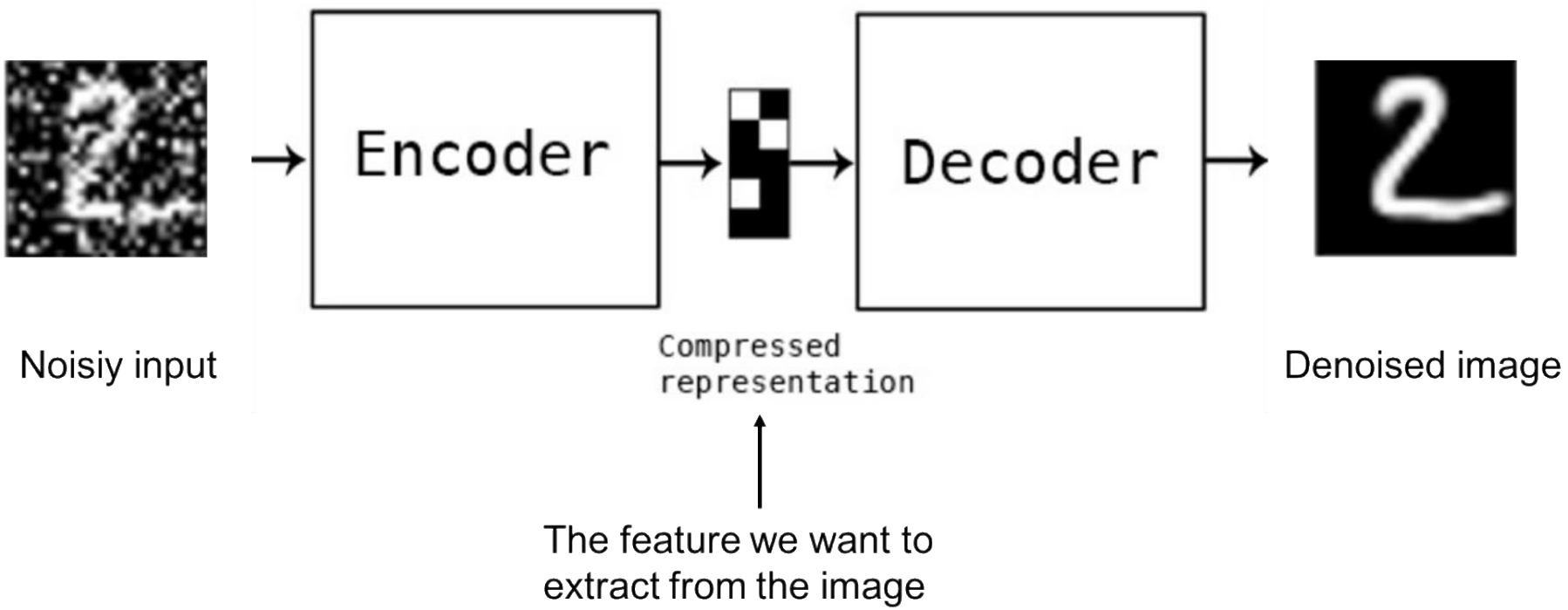
- Clusterizarea aglomerativă a imaginilor MNIST



[Georgescu et al. ICIP2019]

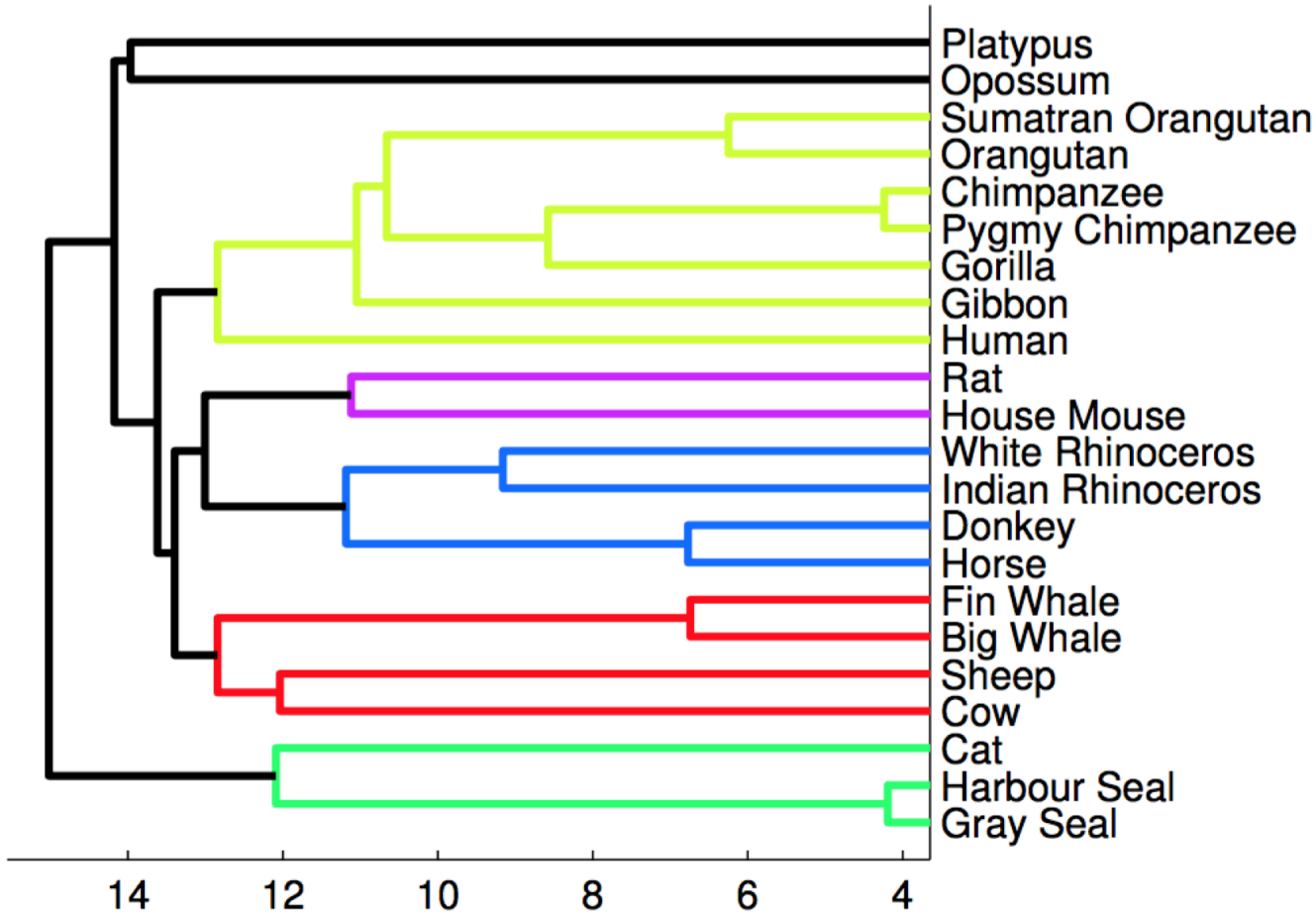
Exemplu 3: Autoencoder

- Învățarea de trăsături folosind principiul “bottleneck”



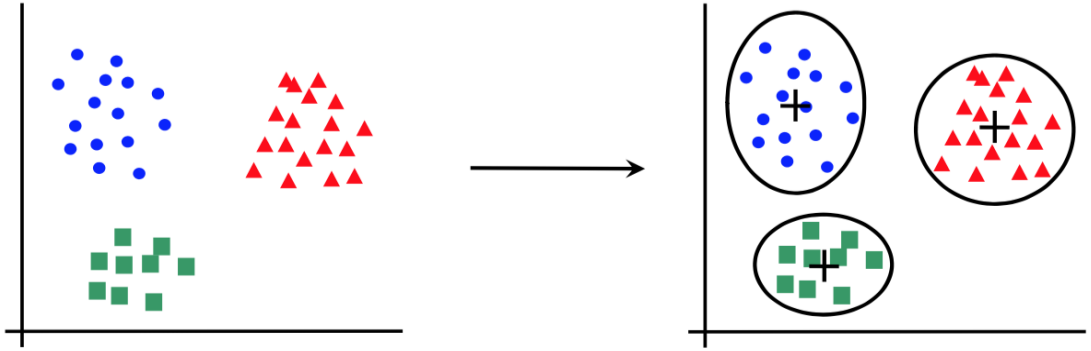
Exemplu 4: Hierachical Clustering

- Gruparea mamiferelor pe familii/specii/etc.

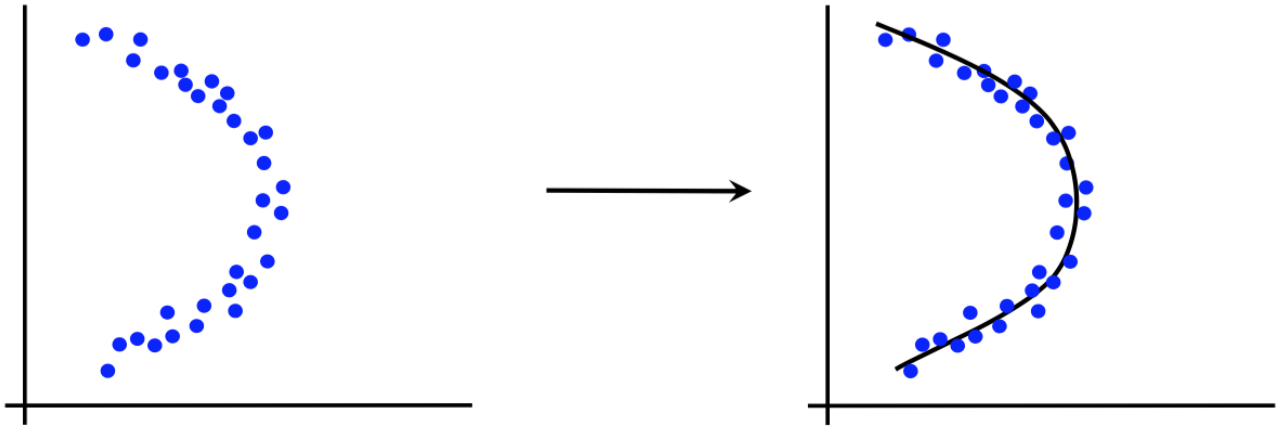


Forme canonice

- Grupare (*Clustering*)



- Reducerea dimensiunii

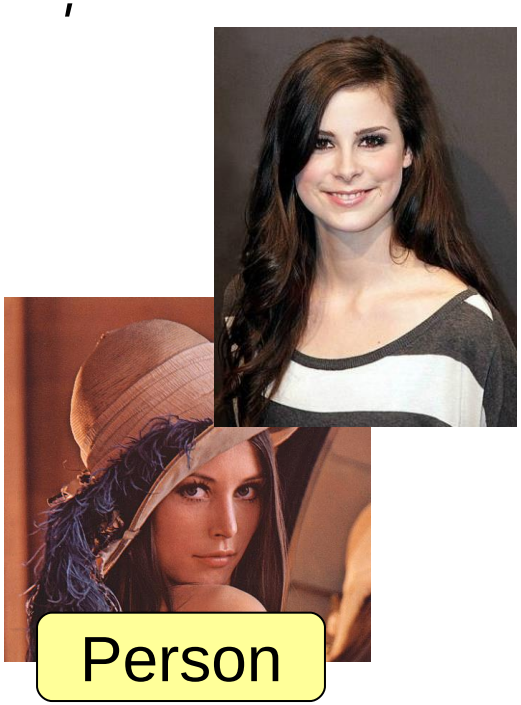


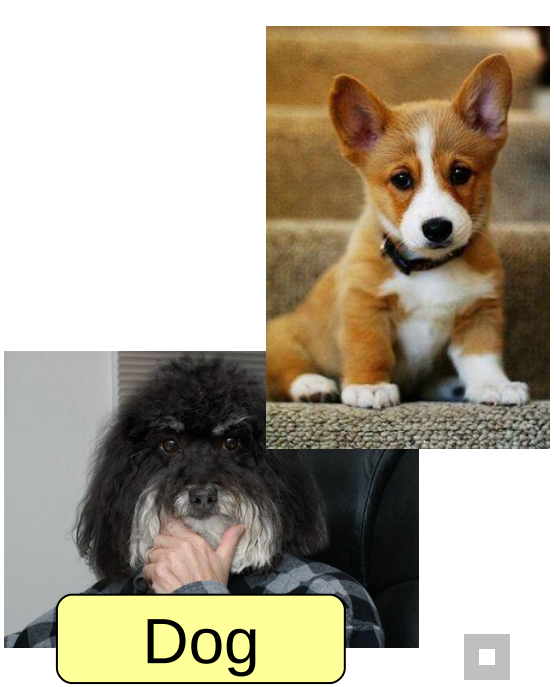
Algoritmi de învățare nesupervizată

- K-means clustering
- Clustering ierarhic (Hierachical Clustering)
- Analiza în componente principale (PCA)
- Modele de tip auto-encoder
- ... și altele

Învățare semi-supervizată

- Avem la dispoziție exemple de obiecte etichetate și exemple de obiecte netichetate
- Exemplu 1: recunoașterea obiectelor din imagini, unele cu eticheta obiectelor conținute



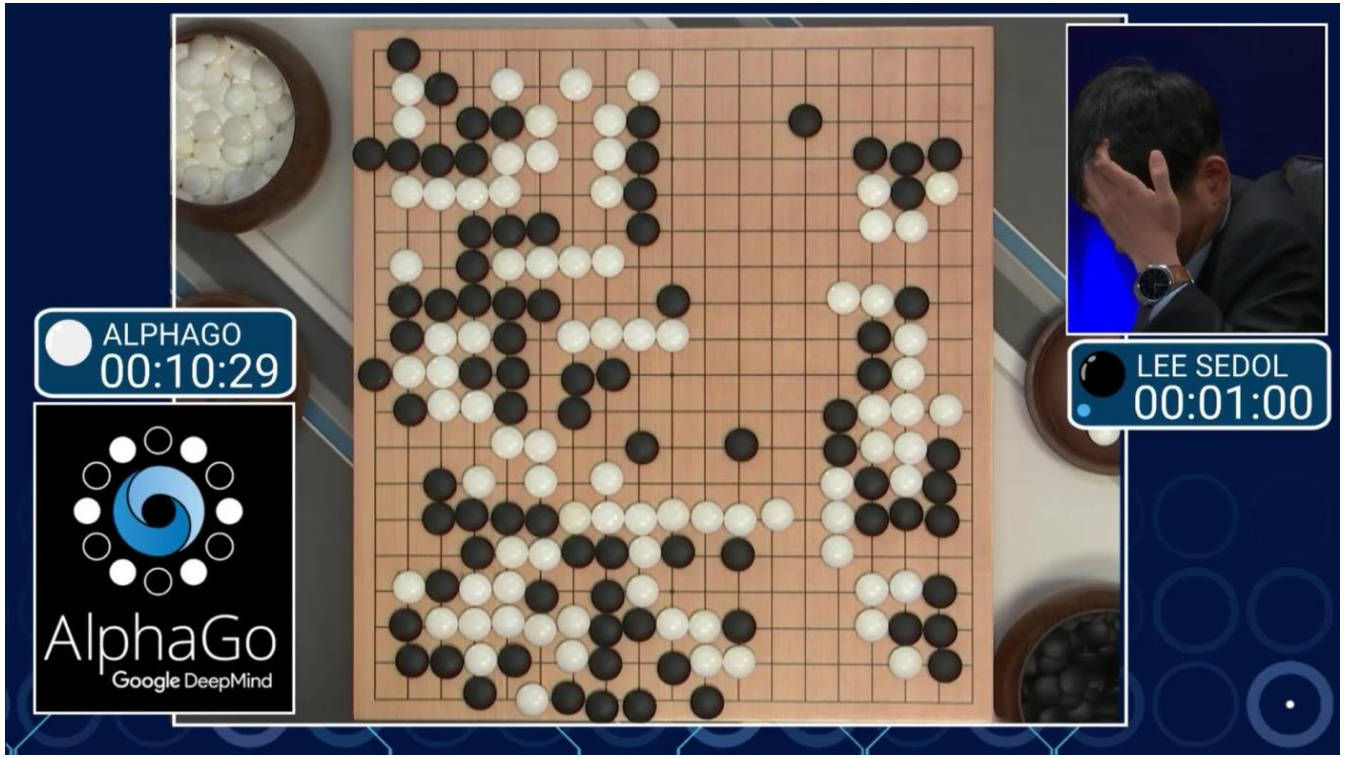


Învățare prin consolidare

- Cu ce diferă această paradigmă de învățare?
  - Sistemul învață comportamentul inteligent pe baza unei recompense (reinforcement signal)
  - Recompensa este primită după mai multe acțiuni (nu vine instant)
  - Timpul contează (datele sunt secvențiale)
  - Acțiunea sistemului influențează datele

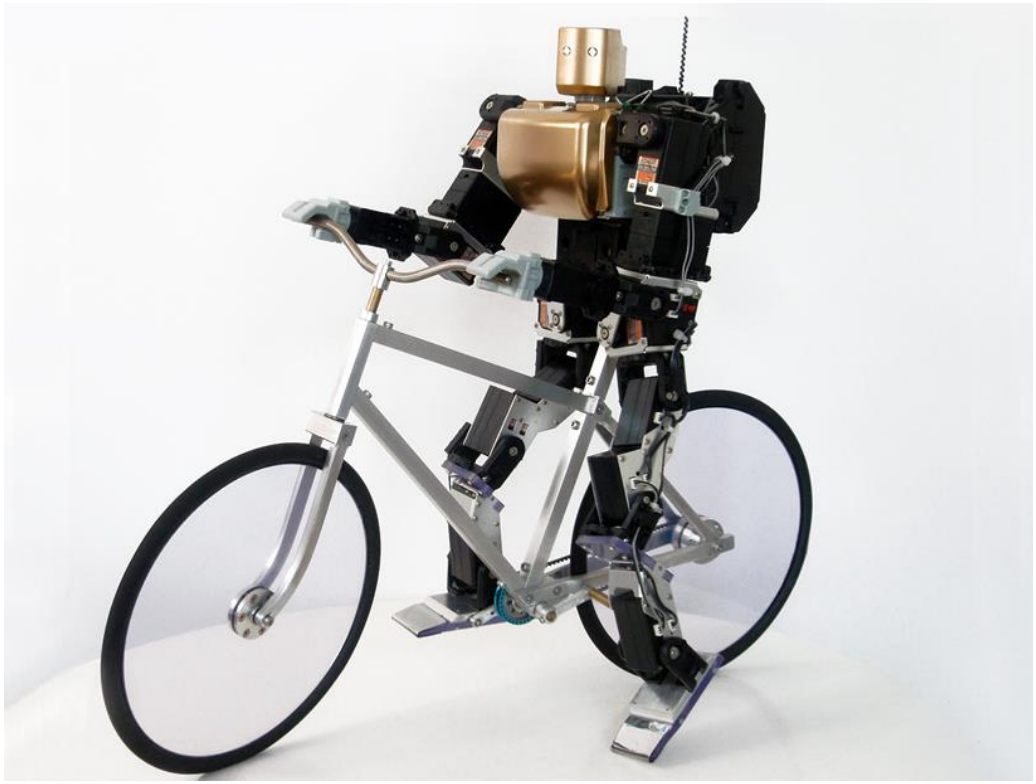
Exemplu 1

- Exemplu 1: învățarea jocului Go
  - recompensă +/- pentru câștigarea/pierderea unui joc



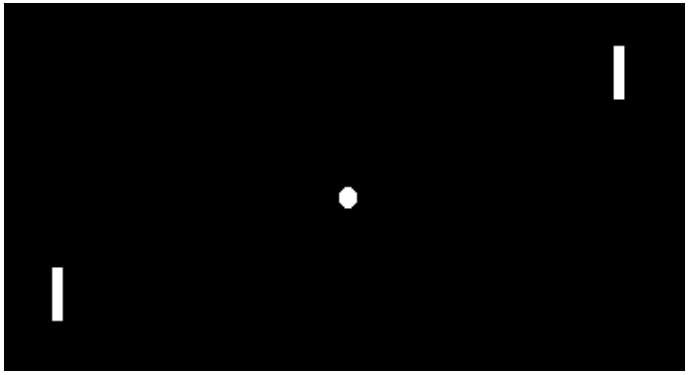
Exemplu 2

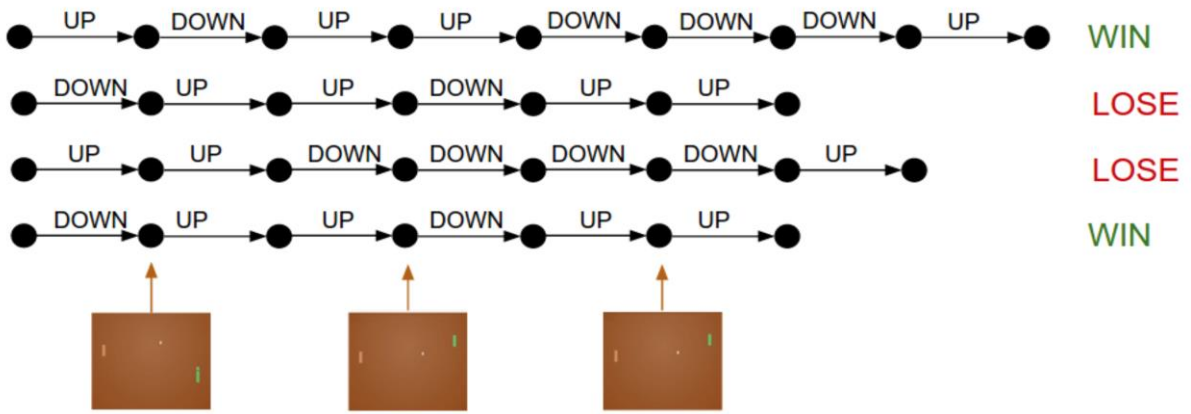
- Exemplu 2: învățarea unui robot să meargă pe bicicletă
  - recompensă +/- pentru mișcare înainte/cădere



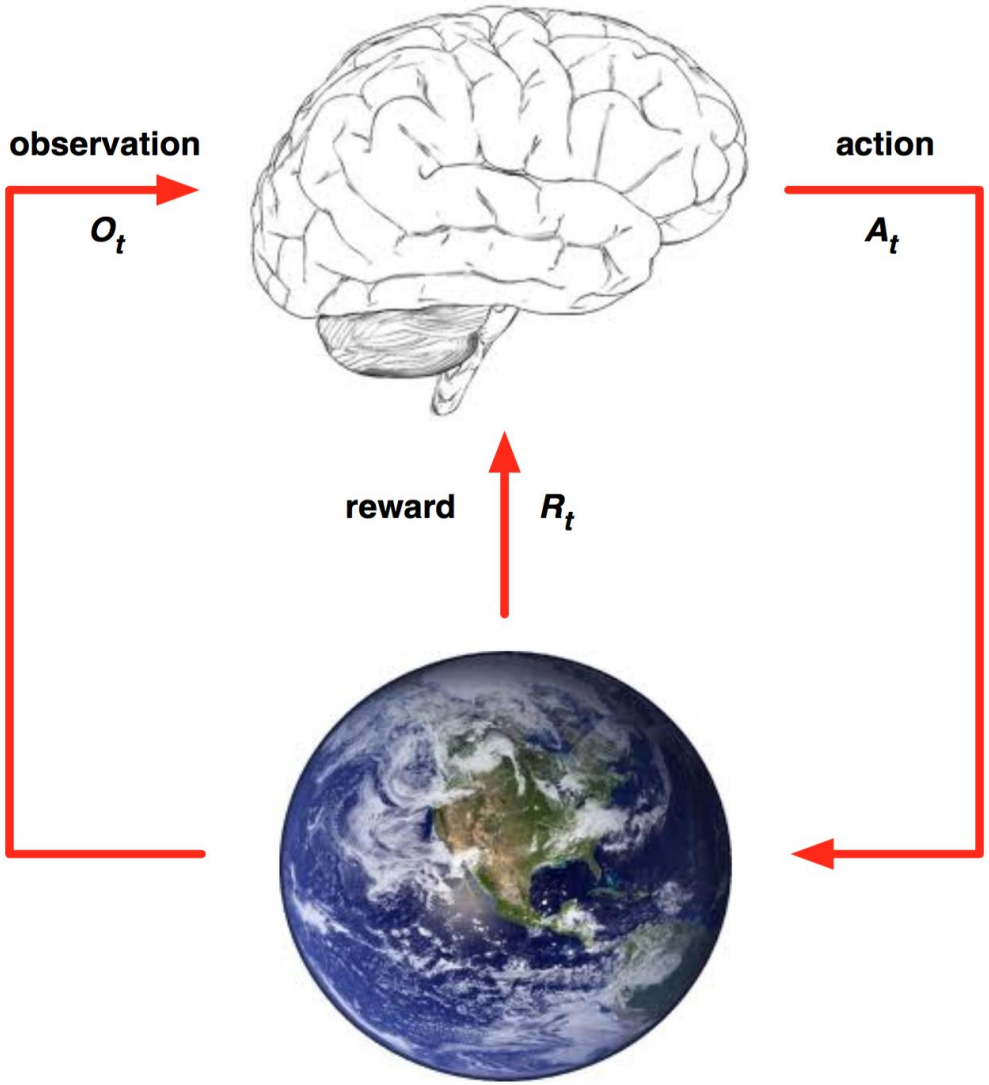
Exemplu 3

- Exemplu 3: învățarea jocului Pong din pixeli
  - recompensă +/- pentru creșterea scorului personal/al adversarului



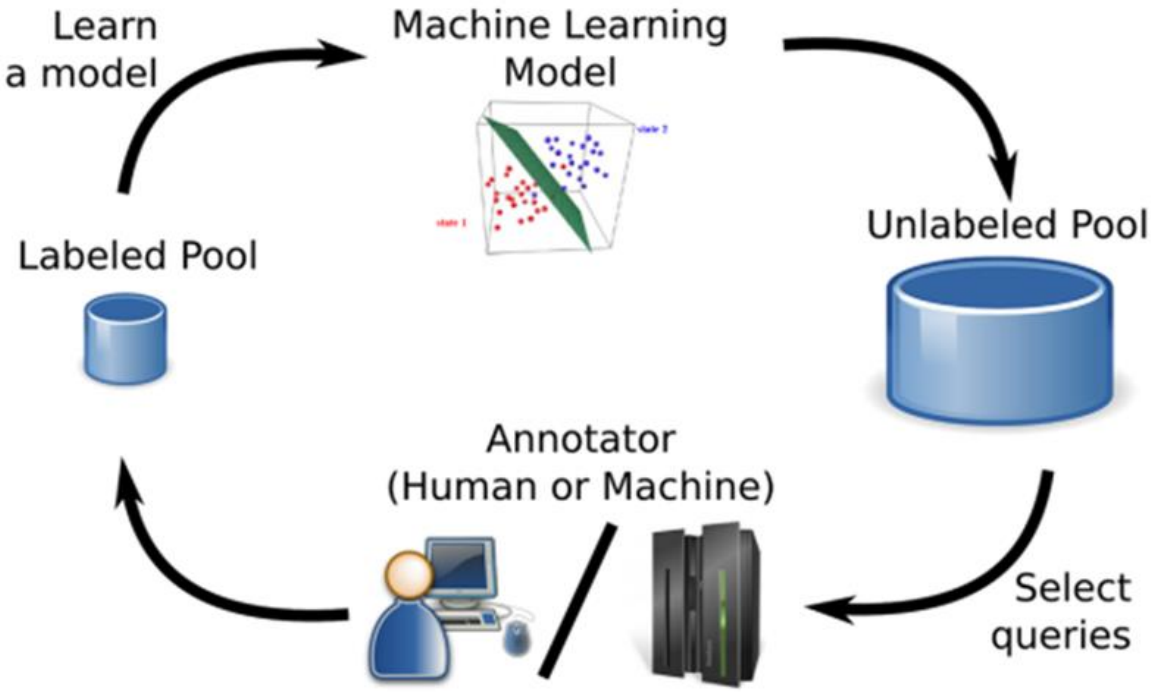


Paradigma Reinforcement Learning



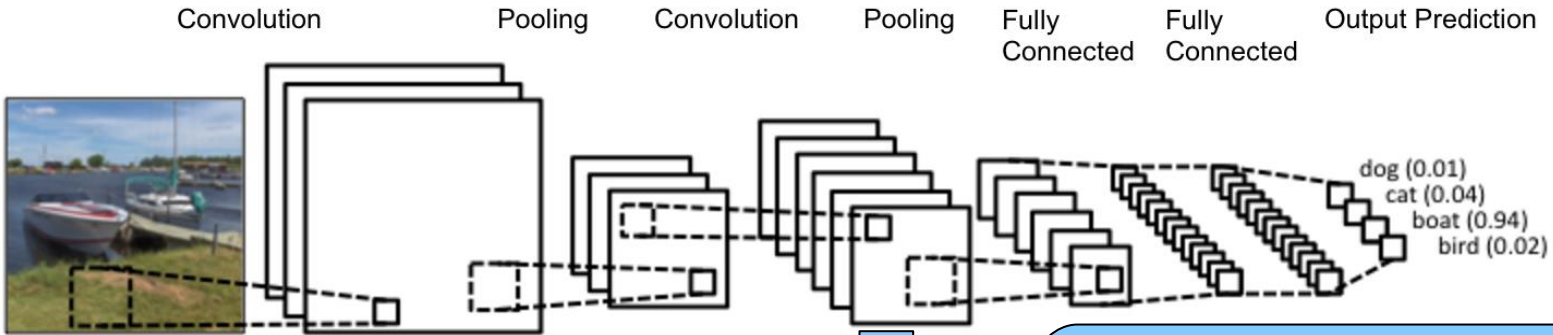
Învățare activă

- Având un set mare de exemple netichetate, trebuie să alegem un subset mult mai mic pe care să îl etichetăm pentru a obține un clasificator cât mai bun



Învățare prin transfer

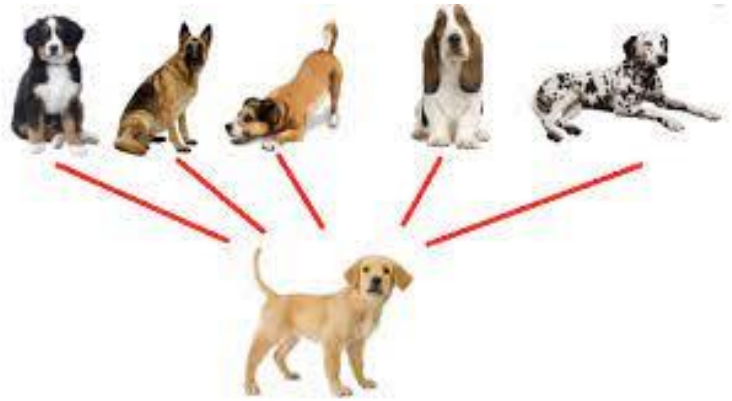
- Pornind la un model antrenat pe un domeniu / o problemă anume, doresc să îl folosesc pentru o altă problemă / alt domeniu
- Exemplu 1: rețele neuronale convoluționale



Alte clase de obiecte (mai specifice), recunoaștere facială, clasificare de texturi, etc.

Generalizare

- Ipotezele trebuie să se generalizeze pentru a clasifica corect instanțele care nu sunt în datele de antrenament.
- simpla memorare a exemplelor de antrenament este o ipoteză consistentă care nu va generalizeaza
- Briciul lui Occam:
  - găsirea unei ipoteze simple ajută la asigurarea generalizării

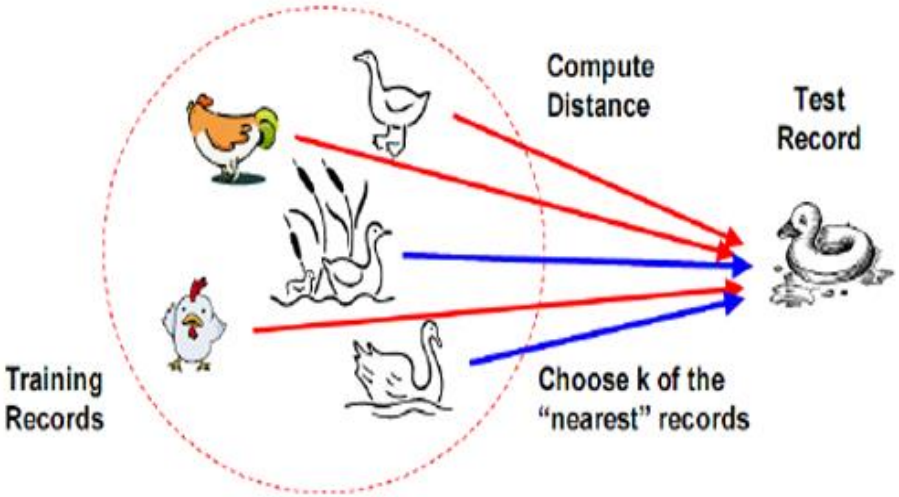


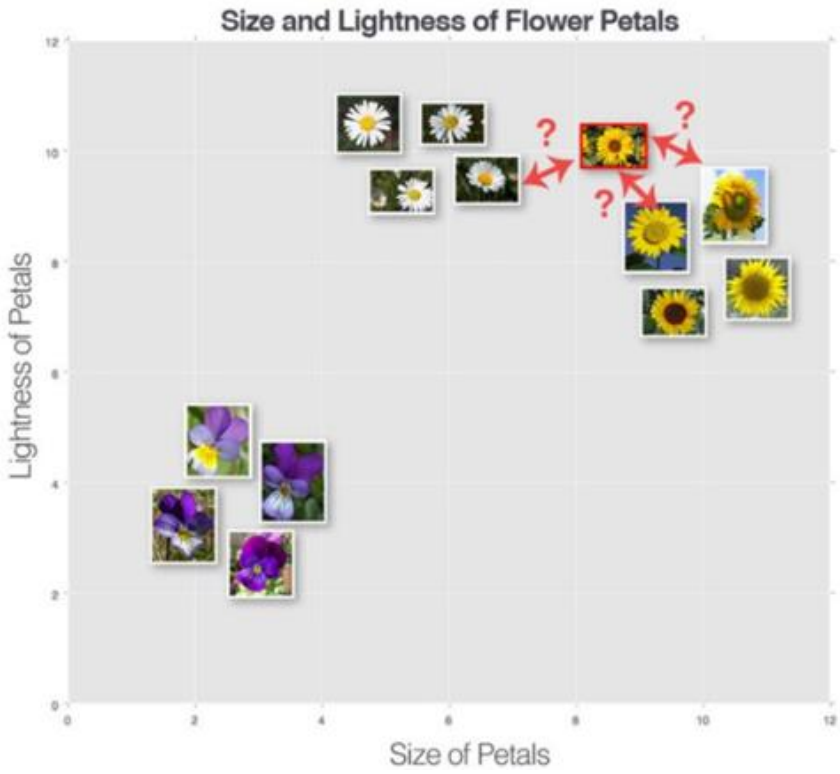
K-Nearest Neighbor (KNN)

- simplu, dar un algoritm de clasificare foarte puternic
- clasifică pe baza unei măsuri de similitudine
- neparametric
- Învățare “leneşă” (*lazy learning*)
  - nu „învață” până când nu este dat exemplul de testare
  - ori de câte ori avem date noi de clasificat, găsim K vecinii săi cei mai apropiați din datele de antrenament

K-Nearest Neighbor (KNN)

- Se folosesc cele mai apropiate k instanțe pentru a realiza clasificarea





Calcularea distanței

- Distanța Euclidiană
  - Pentru a calcula distanța dintre 2 puncte

x = $(x_1, x_2, x_3, \ldots, x_n)$

y = $(y_1, y_2, y_3, \ldots, y_n)$

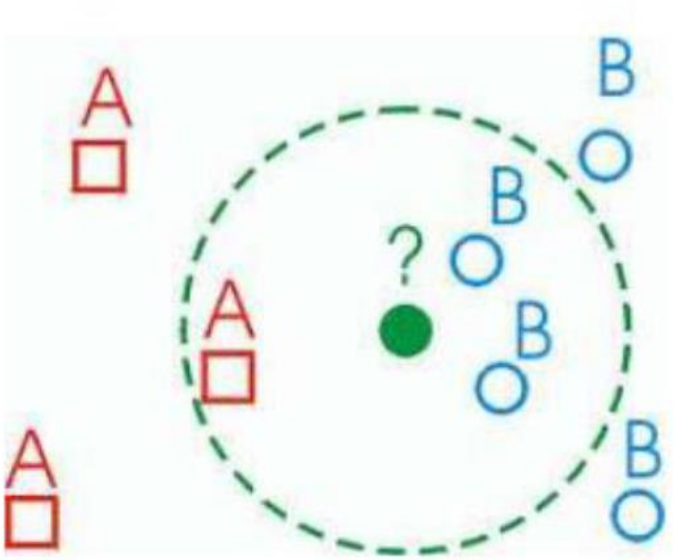
$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}$$

- Observație
  - Pentru n = 1: $d(x,y) = \sqrt{(x - y)^2}$
  - Pentru n = 2: $d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$
  - Pentru n = 3: $d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$

KNN: Clasificare

- clasificat prin „MAJORITY VOTES” pentru clasele vecine
  - atribuit celei mai comune clase dintre cei K vecini cei mai apropiați ai săi (prin măsurarea „distanței” între date)



Pașii KNN

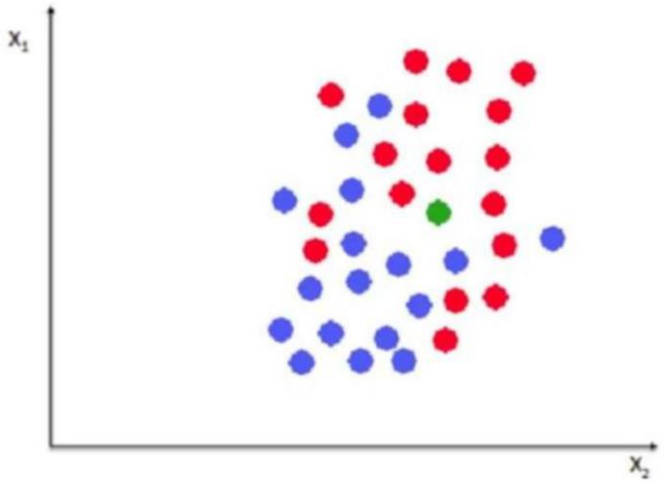
Step 1: Determine parameter K = number of nearest neighbors

Step 2: Calculate the distance between the query-instance and all the training examples.

Step 3: Sort the distance and determine nearest neighbors based on the k-th minimum distance.

Step 4:Gather the category Y of the nearest neighbors.

Step 5: Use simple majority of the category of nearest neighbors as the prediction value of the query instance.



Exemplu KNN

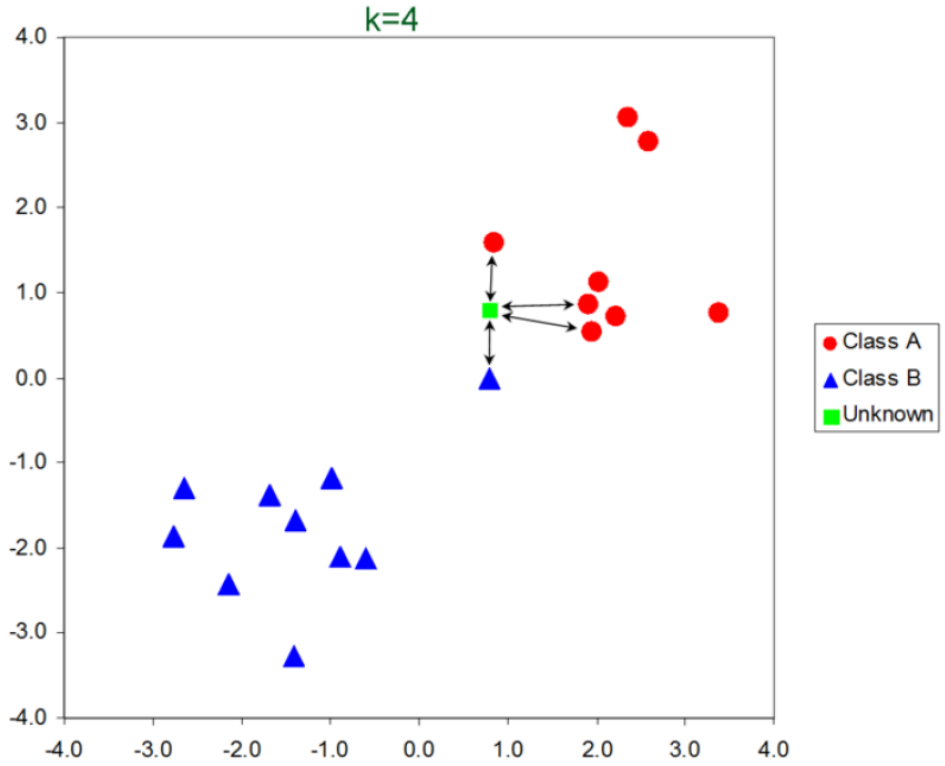


Table 1. Euclidean distance matrix D listing all possible pairwise Euclidean distances between 19 samples.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
x_2	1.5																	
x_3	1.4	1.6																
x_4	1.6	1.4	1.3															
x_5	1.7	1.4	1.5	1.5														
x_6	1.3	1.4	1.4	1.5	1.4													
x_7	1.6	1.3	1.4	1.4	1.5	1.8												
x_8	1.5	1.4	1.6	1.3	1.7	1.6	1.4											
x_9	1.4	1.3	1.4	1.5	1.2	1.4	1.3	1.5										
x_{10}	2.3	2.4	2.5	2.3	2.6	2.7	2.8	2.7	3.1									
x_{11}	2.9	2.8	2.9	3.0	2.9	3.1	2.9	3.1	3.0	1.5								
x_{12}	3.2	3.3	3.2	3.1	3.3	3.4	3.3	3.4	3.5	3.3	1.6							
x_{13}	3.3	3.4	3.2	3.2	3.3	3.4	3.2	3.3	3.5	3.6	1.4	1.7						
x_{14}	3.4	3.2	3.5	3.4	3.7	3.5	3.6	3.3	3.5	3.6	1.5	1.8	0.5					
x_{15}	4.2	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	1.7	1.6	0.3	0.5				
x_{16}	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	1.6	1.5	0.4	0.5	0.4			
x_{17}	5.9	6.2	6.2	5.8	6.1	6.0	6.1	5.9	5.8	6.0	2.3	2.3	2.5	2.3	2.4	2.5		
x_{18}	6.1	6.3	6.2	5.8	6.1	6.0	6.1	5.9	5.8	6.0	3.1	2.7	2.6	2.3	2.5	2.6	3.0	
x_{19}	6.0	6.1	6.2	5.8	6.1	6.0	6.1	5.9	5.8	6.0	3.0	2.9	2.7	2.4	2.5	2.8	3.1	0.4

$$d(x,\ y)\ =\ \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

n– numărul de dimensiuni (in cazul nostru– 2)

ÎNTREBĂRI ?