Învățare Automată în Vedere Artificială

Curs 1: Introducere

Echipă

Daniel Sociu

Alexandru Enache

Mihail Chirobocea

Andrei Janca

Vlad Păunescu

Structură curs

- 9 saptamani (1 mai e miercuri, e in pre-sesiune)
- 8 cursuri x 2h
- 7 laboratoare x 2h (4 pct.)
  - 6 punctate, 1 bonus cu LLM-uri
  - se prezinta saptamana urmatoare (1 pct penalizare pt fiecare sapt)
- Proiect - 2 prezentări (6 pct.):
  - săptămâna 5 - prezentare temă proiect + SOTA + prezentare abordare aleasă 2.5 p
    - SOTA = State of the Art
  - săptămâna 10 - prezentare finală + demo 3.5 p
  - https://paperswithcode.com/
- Tehnologii: Python + PyTorch + Google Colaboratory

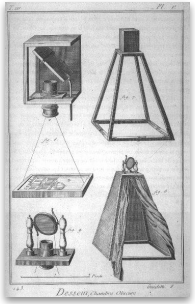
Prerequisites

- Maths
  - Algebră Liniară
  - Analiză
  - Statistică și Probabilități
- Programming
  - Python
- Nice to have
  - Inteligență Artificială/Învățare Automată

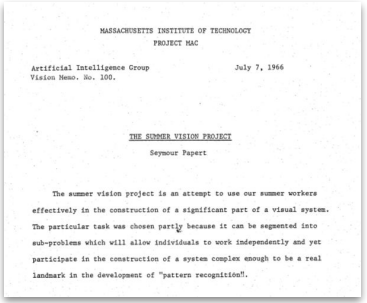
Subiecte curs

1. Introducere în Vederea Artificială
2. Clasificarea Imaginilor & Optimizare
3. Backpropagation & Convoluții
4. Rețele Convoluționale
5. Detecția de Obiecte
6. Segmentarea Semantică
7. Vision Transformers and GPT
8. Auto-encoders and diffusion models

Istoria Vederii (Artificiale) - Timeline



Camera Obscura



MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group
Vision Memo. No. 100.

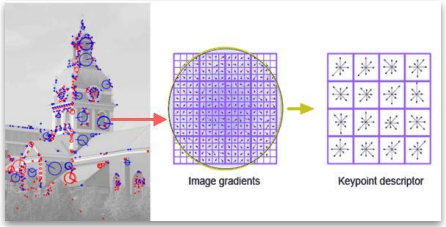
July 7, 1966

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

The Summer Vision Project



SIFT, David Lowe

Sec. XVI

1959

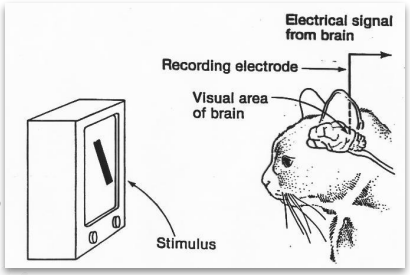
1966

1998

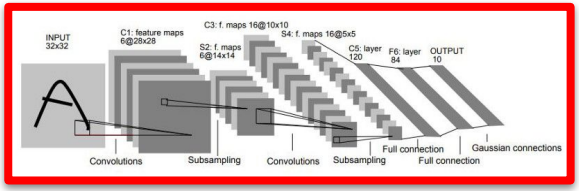
1999

2001

Hubel & Wiesel



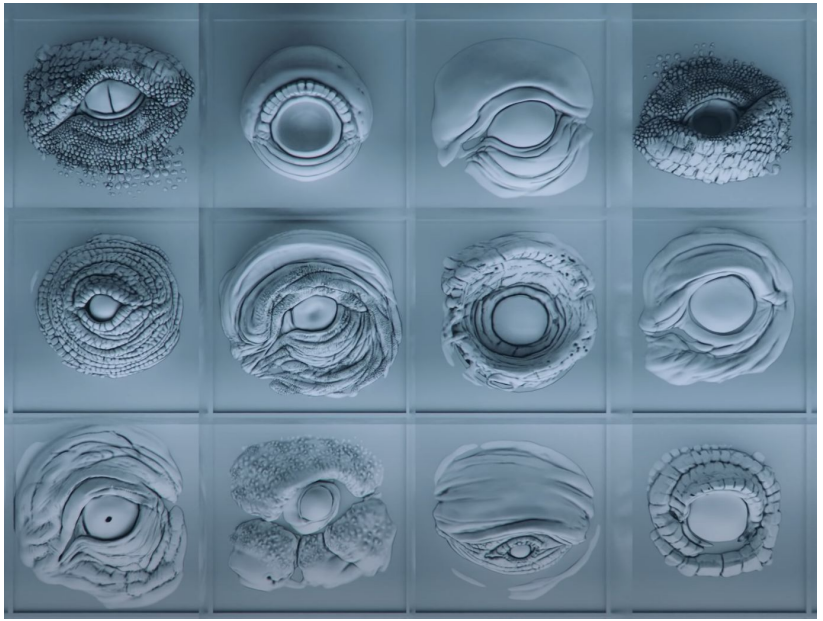
LeNet



Face Detection, Viola & Jones

Istoria Vederii (Artificiale) - **Explozia cambriană**

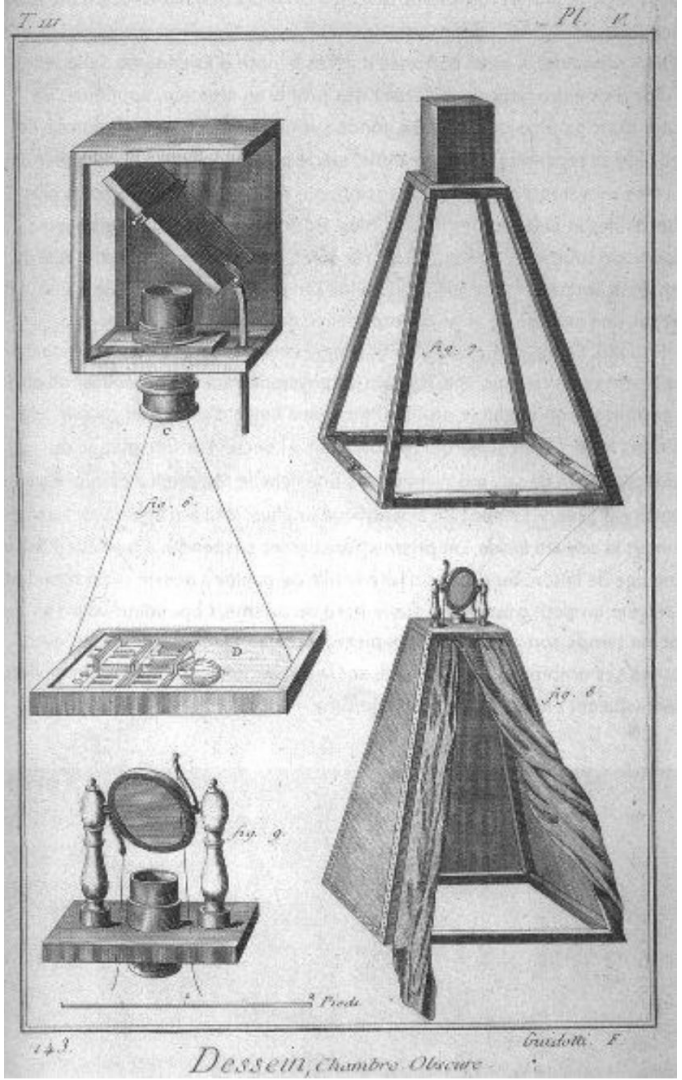
Eyesight have evolved multiple times, independently, in different species



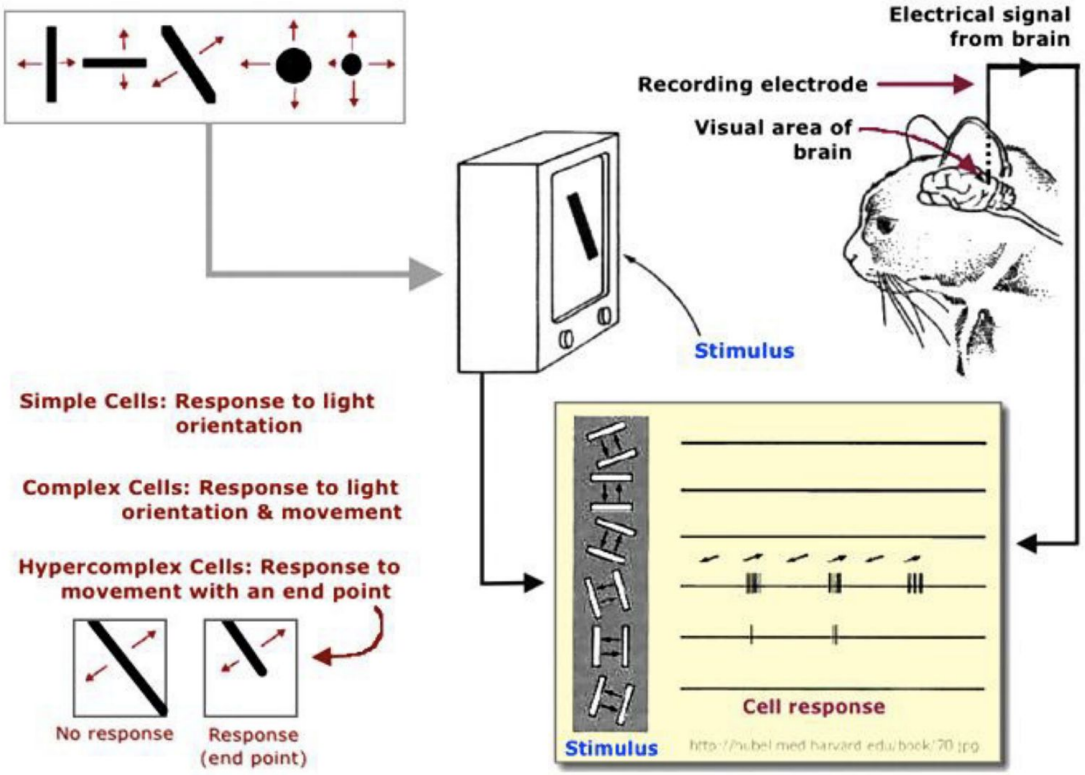
Istoria Vederii (Artificiale)

Camera Obscura

Leonardo da Vinci
16th Century, A.D.



Istoria Vederii (Artificiale) - Neuroștiințe




's did it before it was cool

Hubel & Wiesel, 1959

video

Istoria Vederii (Artificiale)

The Summer Vision Project **MIT** - 1966

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

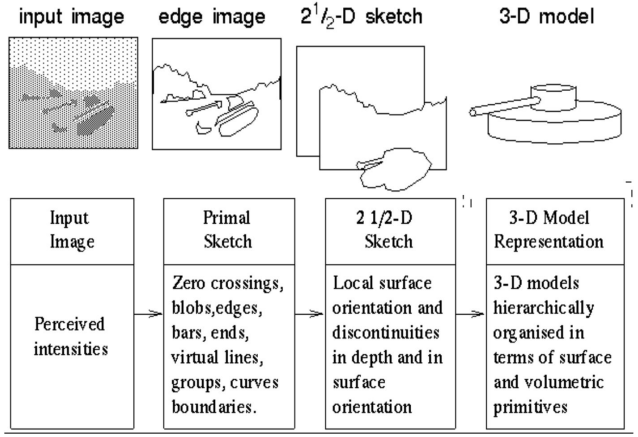
Artificial Intelligence Group — July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

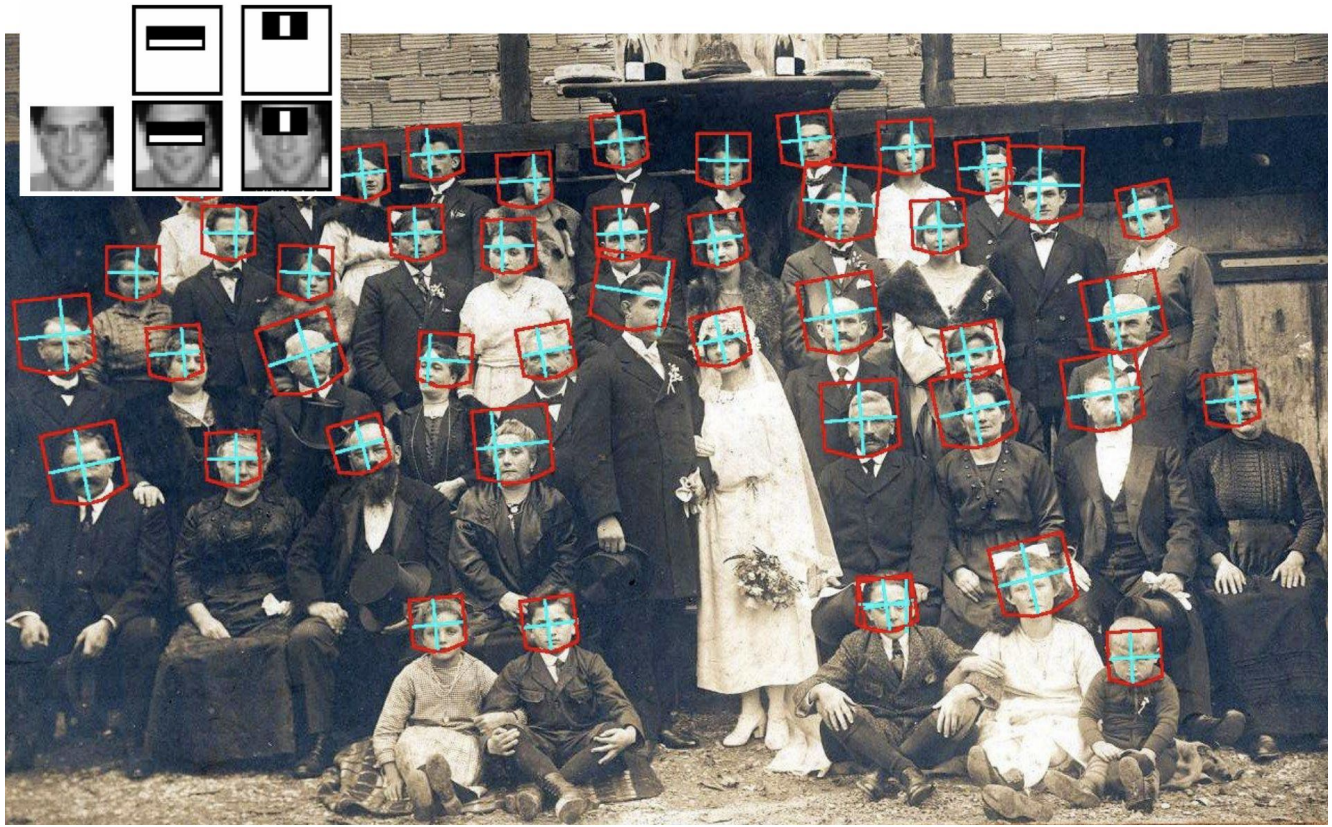
The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Vision - David Marr, 1970s



Istoria Vederii (Artificiale) - Detecția fețelor

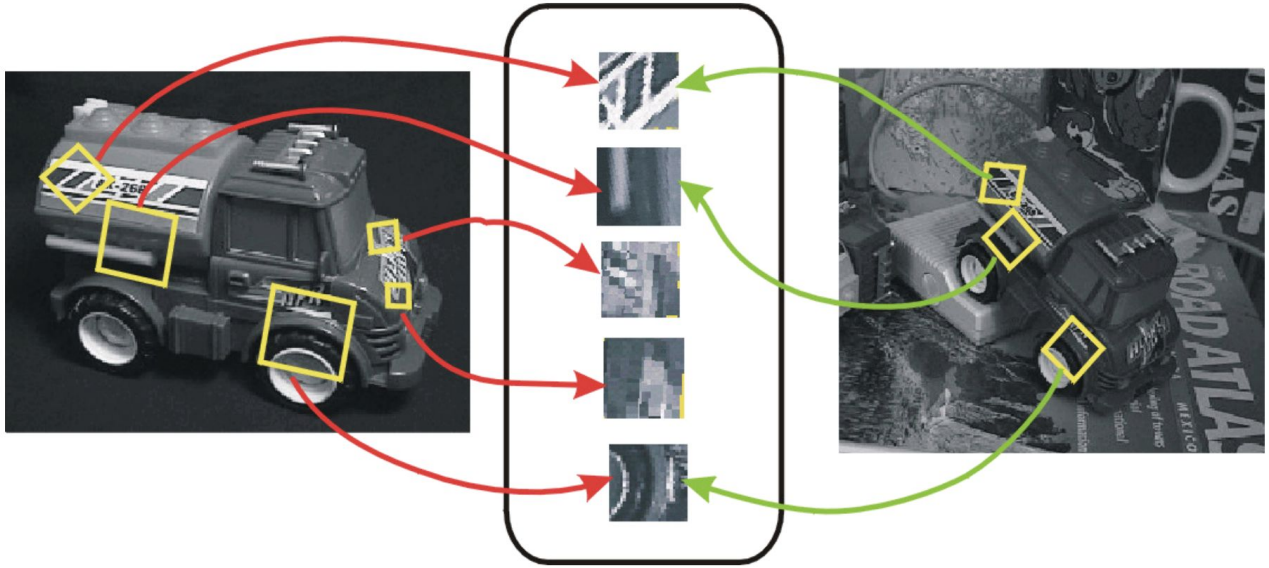
Face Detection,
Viola & Jones
2001

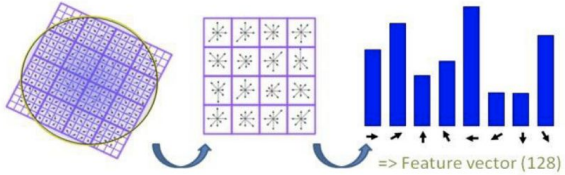


Istoria Vederii (Artificiale) - Extragerea trăsăturilor

"SIFT" & Object Recognition, David Lowe, 1999

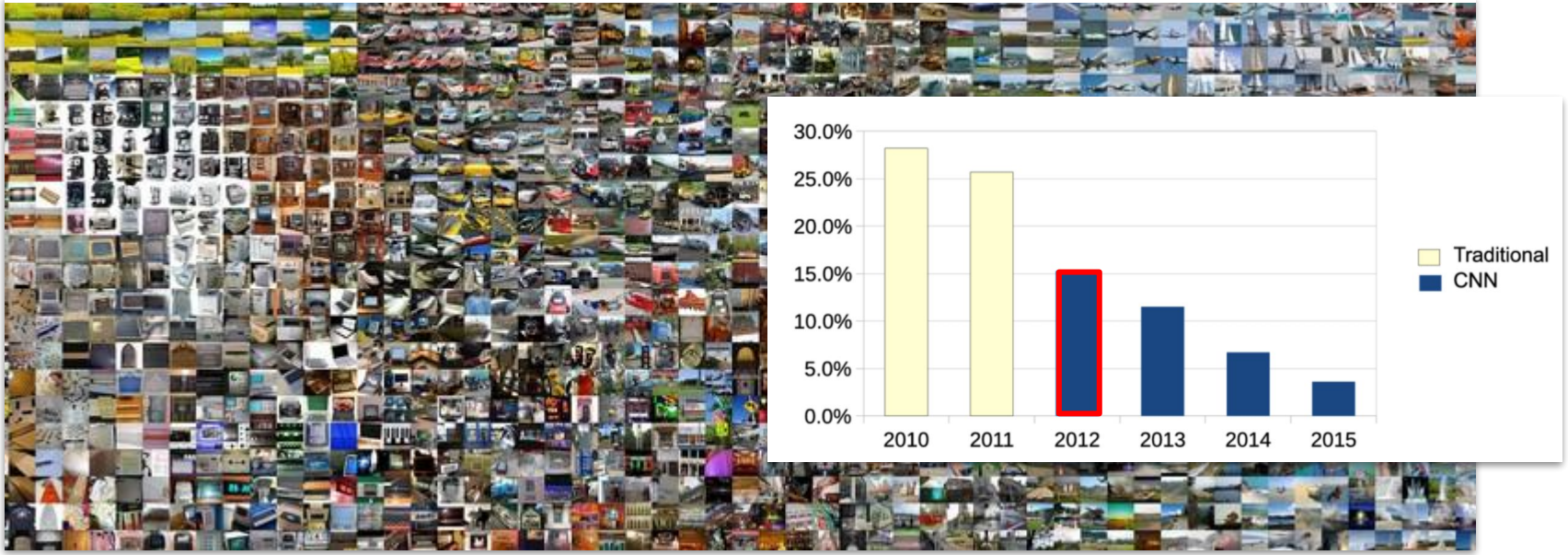
"Feature Extraction"





Rețele Neurale Convoluționale

IMAGENET Large Scale Visual Recognition Challenge

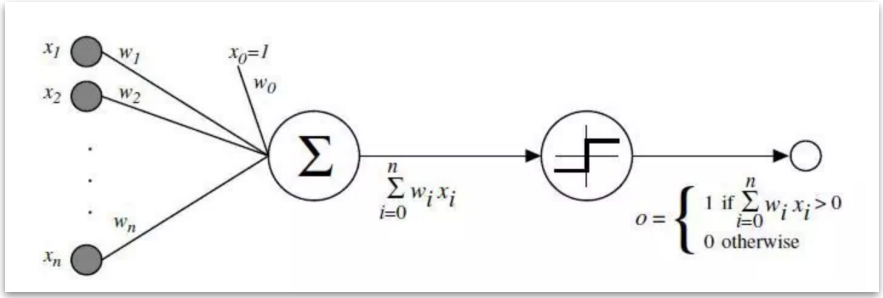


The Image Classification Challenge

- 1.4 M Images
- 1,000 object classes

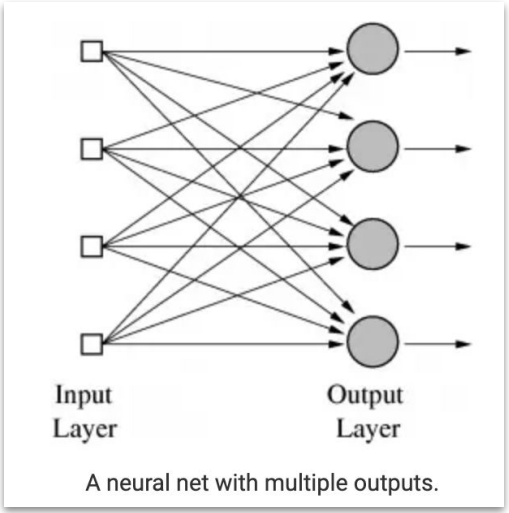
Rețele Neurale

Frank Rosenblatt's **Perceptron,** 1958



'Artificial Neural Networks' - ANNs

- Extended to work for classification tasks with many categories
- Layers of Perceptrons - or neurons, or units, as they are usually called today



A neural net with multiple outputs.

Rețele Neurale Convoluționale

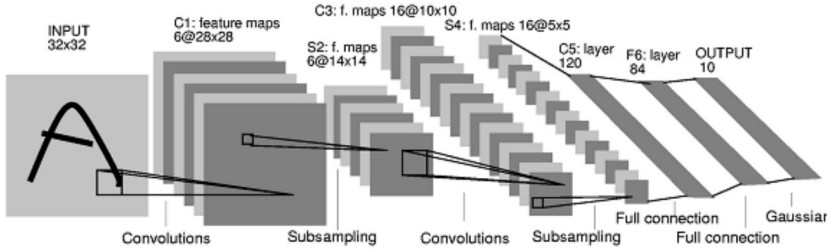
"According to the hierarchy model by Hubel and Wiesel, the neural network in the visual cortex has a hierarchy structure"

Backpropagation algorithm - Rumelhart, Hinton & Williams (1986)



1998

LeCun et al.



# of transistors

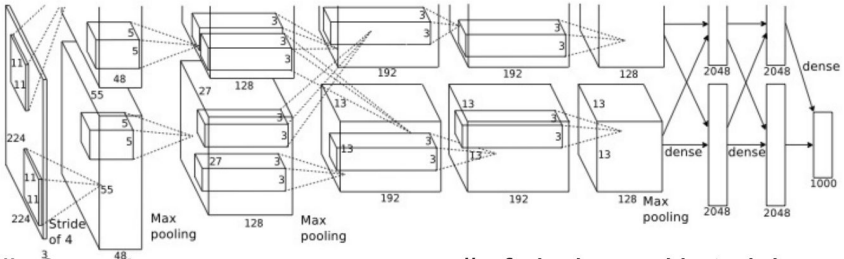
10^6

# of pixels used in training

10^7 NIST

2012

Krizhevsky et al.



# of transistors

10^9

GPUs

# of pixels used in training

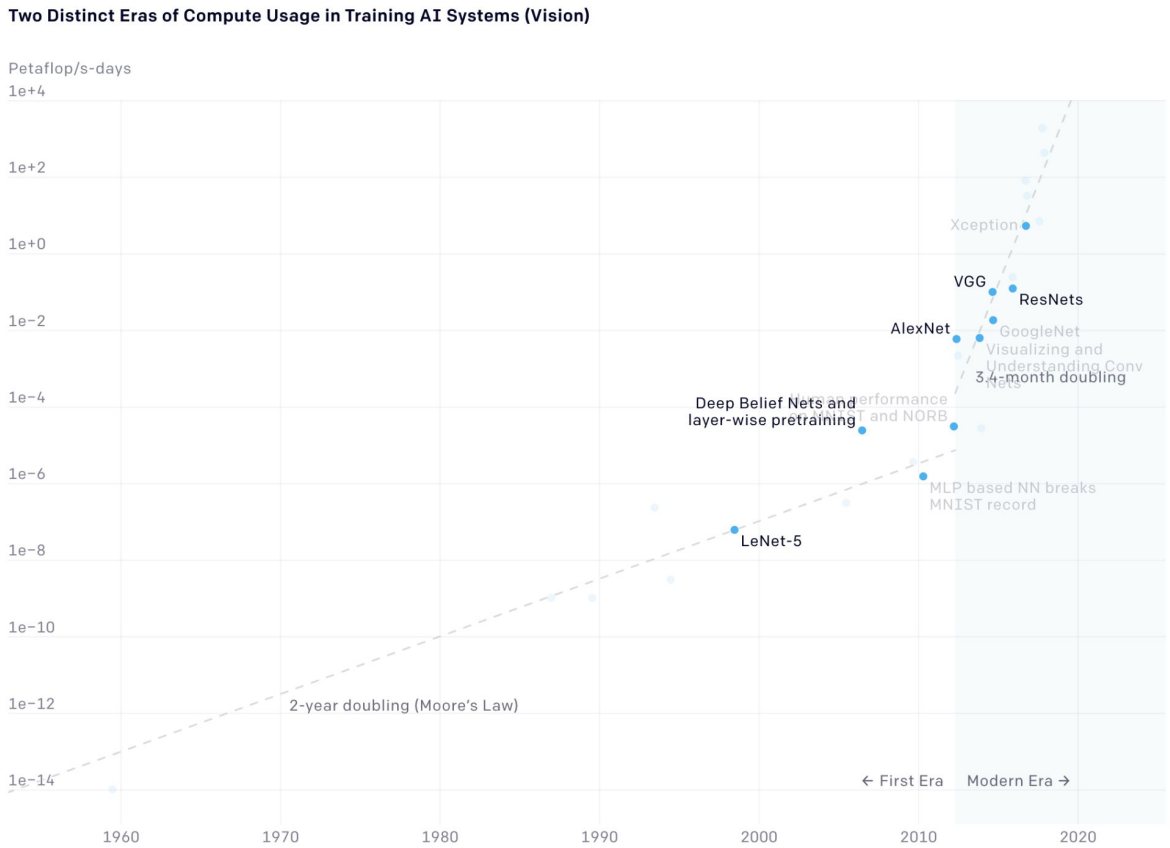
10^{14} IMAGENET

Legea lui Moore in AI

- the amount of compute used in the largest AI training runs has been increasing exponentially with a 3.4-month doubling time
- by comparison, Moore's Law had a 2-year doubling period
- Since 2012, this metric has grown by more than 300,000x
    - a 2-year doubling period would yield only a 7x increase
- Improvements in compute have been a key component of AI progress, so as long as this trend continues, it's worth preparing for the implications of systems far outside today's capabilities.

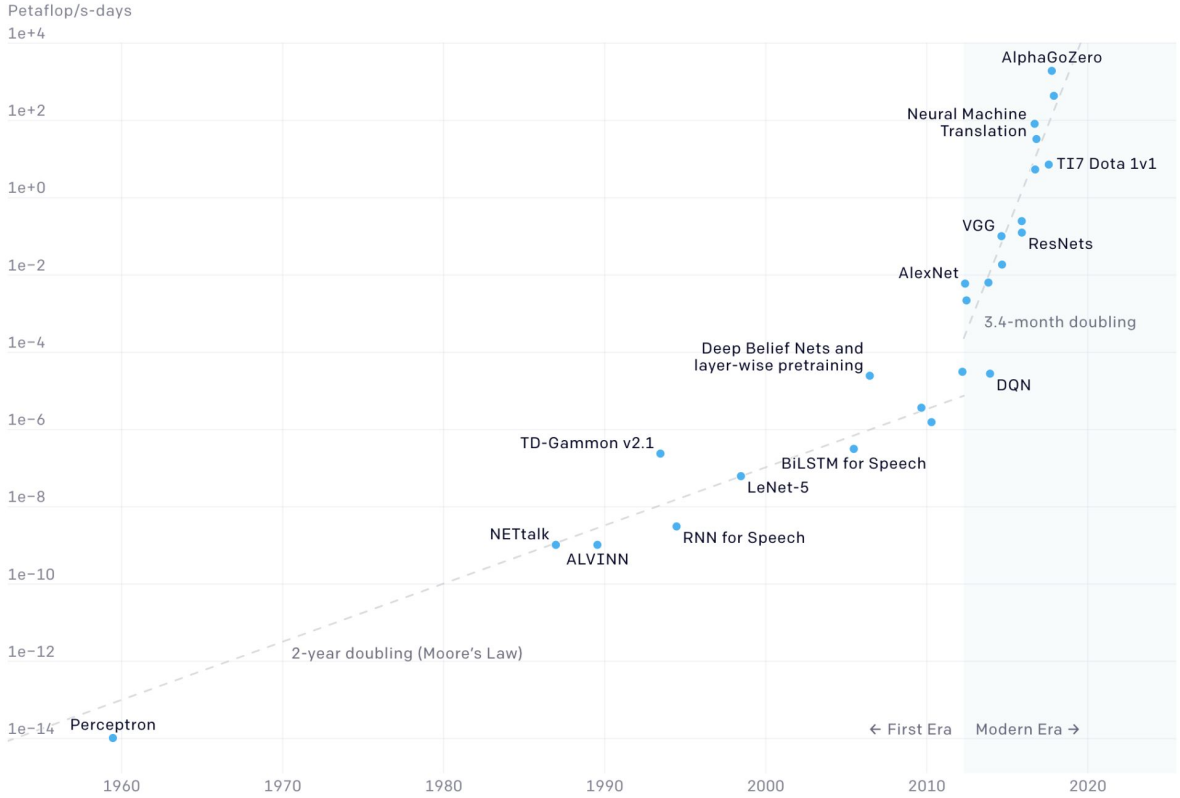
Source: https://openai.com/blog/ai-and-compute/

Legea lui Moore in AI



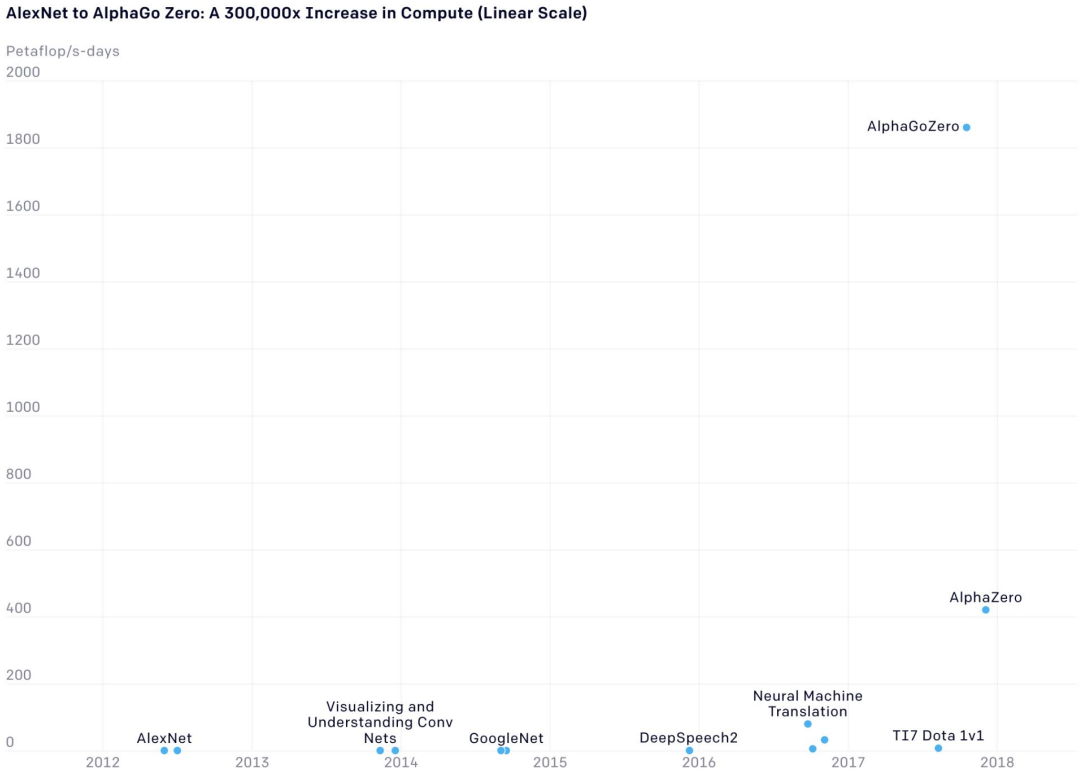
The total amount of compute, in petaflop/s-days,[2] used to train selected results that are relatively well known, used a lot of compute for their time, and gave enough information to estimate the compute used. Source https://openai.com/blog/ai-and-compute/

Legea lui Moore in AI



The total amount of compute, in petaflop/s-days,[2] used to train selected results that are relatively well known, used a lot of compute for their time, and gave enough information to estimate the compute used. Source https://openai.com/blog/ai-and-compute/

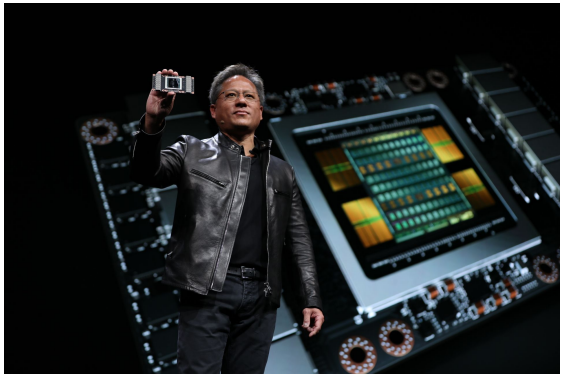
Legea lui Moore in AI



The total amount of compute, in petaflop/s-days,[2] used to train selected results that are relatively well known, used a lot of compute for their time, and gave enough information to estimate the compute used. Source https://openai.com/blog/ai-and-compute/

Modern Era of AI

- **How much compute to train the largest GPT-3(175B parameters)** ?
    - **3640** petaflop/s-day - compute required to train largest GPT-3 model
    - **1 petaflop/s-day is equivalent to 8 Volta V100 GPUs at full efficiency of a day.**
        - **32GB x8 GPU Memory HBM2 source**
- How many days to train GPT-3.5?
    - As an example, it requires approximately 9.2 days on 512 V100 GPUs to train a 8.3B GPT-2 (Shoeybi et al., 2019),
    - 14.8 days on 10000 V100 GPUs to train a 175B GPT-3 (Patterson et al., 2021).
    - 3640 days on8 V100 GPUs at full efficiency
    - OpenAI used Microsoft Azure cloud to train and scale



Modern Era of AI

Mode	Parameters (B)	VRAM Requirement (GB)	Recommended GPU
DeepSeek-R1-Zero	671B	~1,543 GB	Multi-GPU setup (e.g., NVIDIA A100 80GB x16)
DeepSeek-R1	671B	~1,543 GB	Multi-GPU setup (e.g., NVIDIA A100 80GB x16)
DeepSeek-R1-Distill-Qwen-1.5B	1.5B	~3.9 GB	NVIDIA RTX 3060 12GB or higher
DeepSeek-R1-Distill-Qwen-7B	7B	~18 GB	NVIDIA RTX 4090 24GB or higher

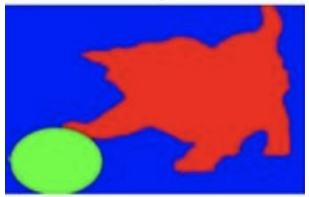
Aplicații

Clasificare

Cat

Detectie

Segmentare



Red: cat
Green: ball
Blue: background

Subtitrare

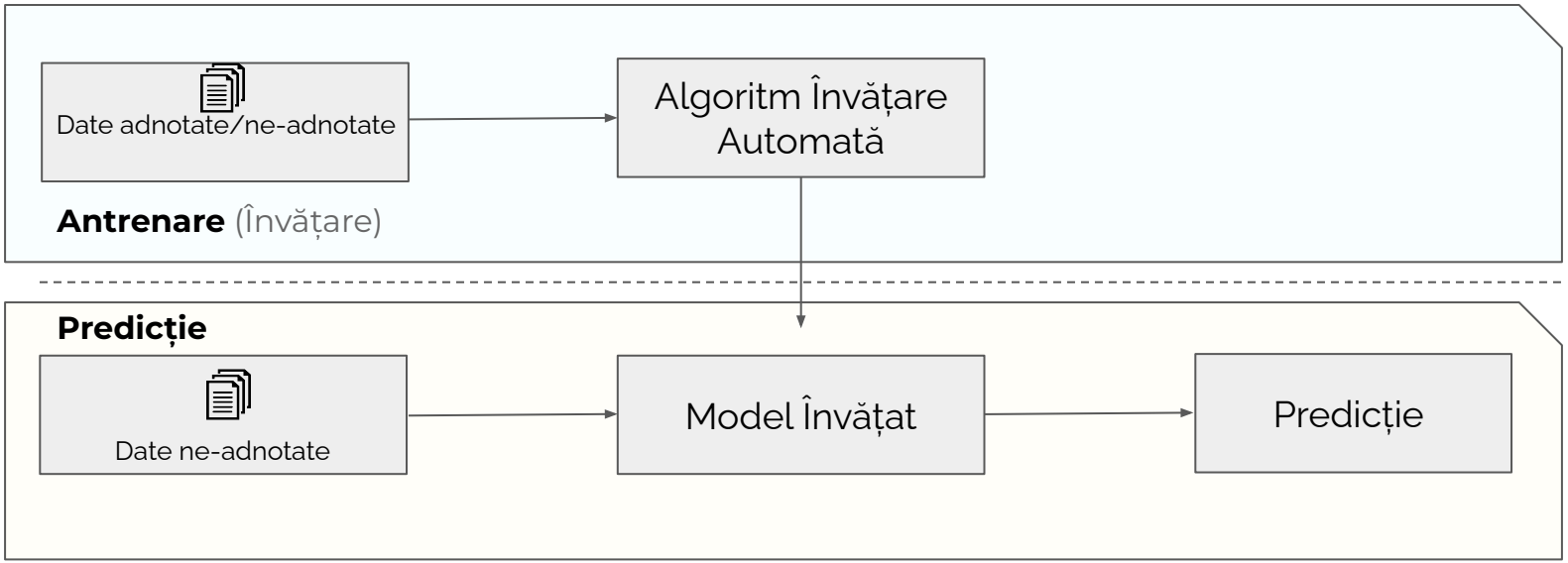
A cat is playing a ball.

Super-rezoluție

Generare Imagini

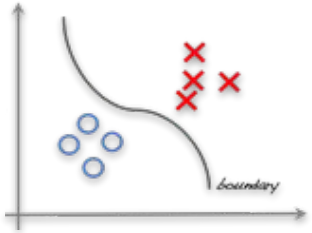
Domenii

Învățarea Automată - Concepte de Bază



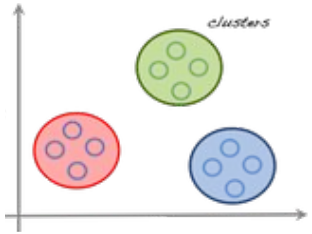
*"A computer program is said to learn from experience **E** with respect to some task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improves with experience **E**."* - Tom Mitchell

Învățarea Automată - Concepte de Bază



Învățare supervizată (Supervised Learning): Învățare dintr-un set de **date adnotate**

Exemplu: detectare spam, estimarea prețului acțiunilor, recunoaștere de caractere



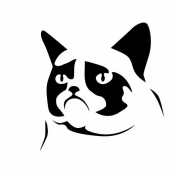
Învățare nesupervizată (Unsupervised Learning): Descoperirea tiparelor în **date neadnotate**

Exemplu: segmentare clienți, detectare anomalii

Învățare prin Recompensă (Reinforcement Learning):

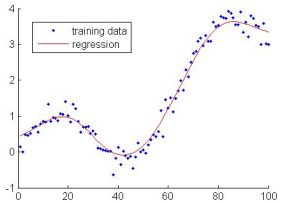
Exemplu: învață să joace Tetris, manipulare roboți

Învățarea Automată - Concepte de Bază

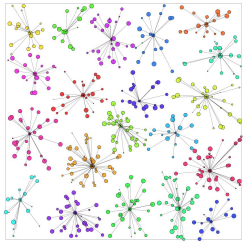


pisică 0.9
câine 0.1

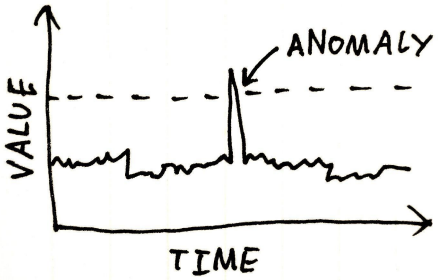
Clasificare
(Supervizată - predictivă)



Regresie
(Supervizată - predictivă)



Grupare (Clustering)
(Nesupervizată - descriptivă)



Detecția anomaliilor
(Nesupervizată - descriptivă)

Regresia liniara

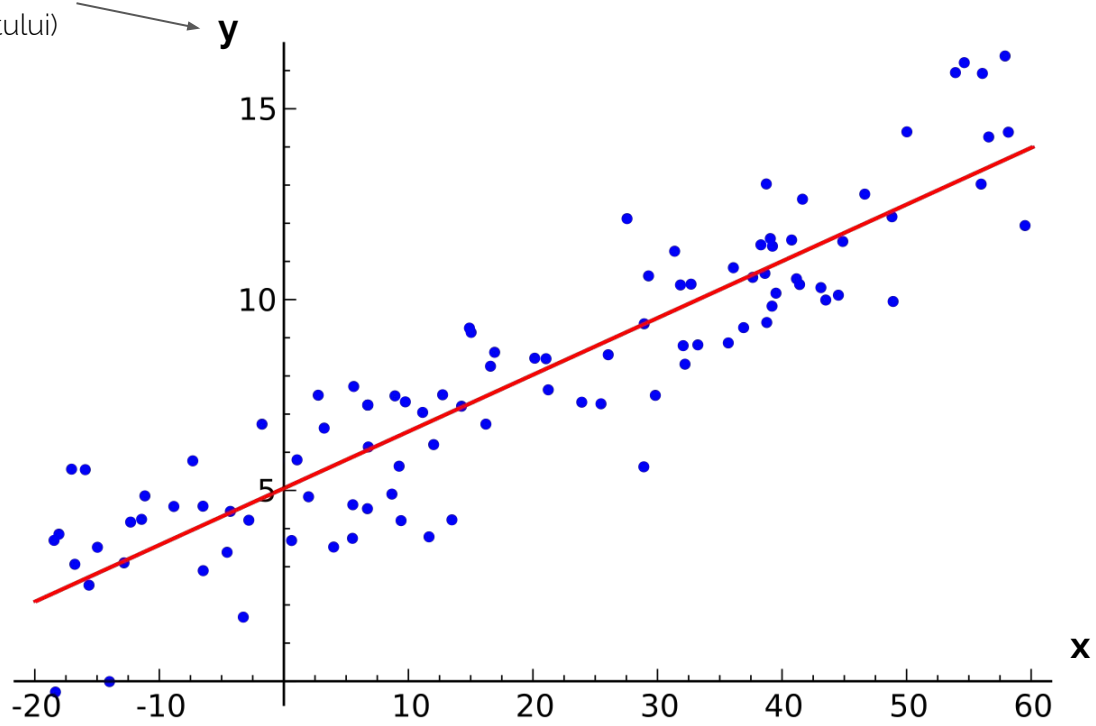
Regresia: prezicerea valorii unei variabile continue

Suprafata [mp]	Pret [* 10.000$]
40	5
35	4.7
67	12
...	...
29	3.8

Q: Cum putem prezice prețul unui apartament a cărui suprafață nu este inclusă în setul de date?

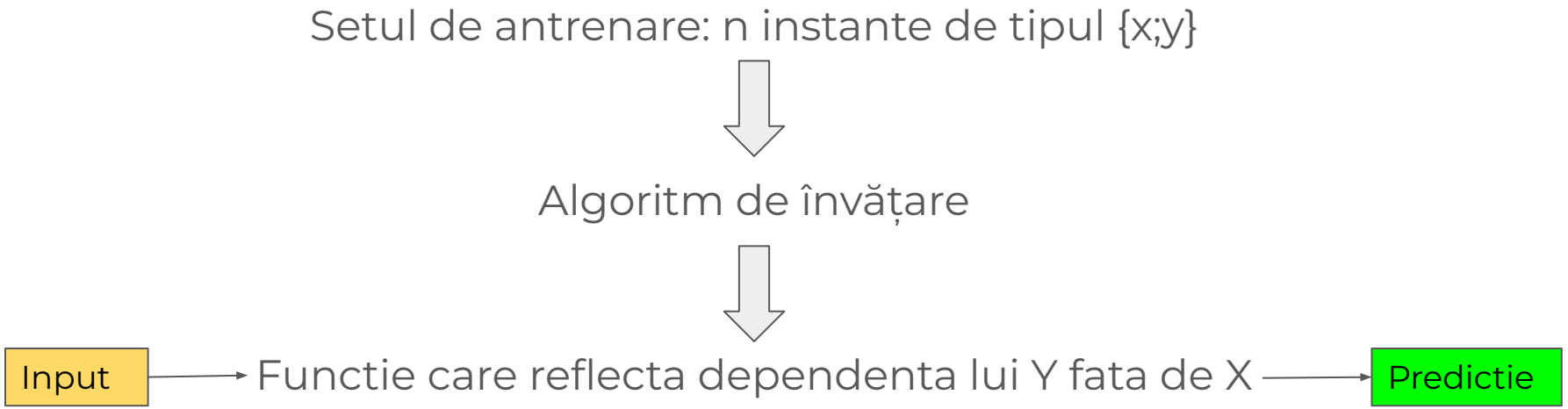
Regresia liniara

Variabila dependenta
(ex.: prețul apartamentului)



Variabila
independenta
(suprafața
apartamentului)

Regresia liniara

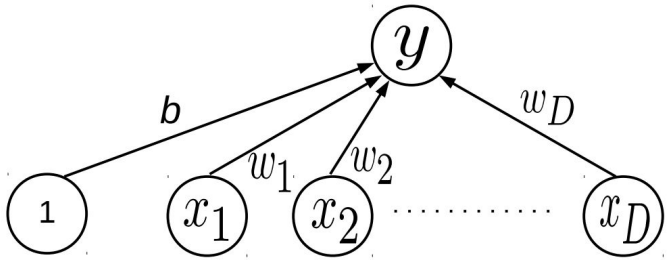


Optimizare - Regresie Liniară

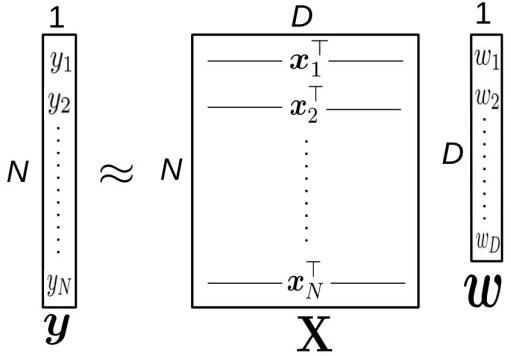
- **Modelele liniare** sunt componentele de bază ale rețelelor neuronale
- Vom formula învățarea ca o problemă de optimizare (cu privire la acești parametri)
- **Putem adăuga o constantă '1' pentru fiecare intrare (feature) și rescrie ca:*

$$\boxed{y = \boldsymbol{w}^\top \boldsymbol{x},} \quad \boldsymbol{x}, \boldsymbol{w} \in \mathbb{R}^{D+1}$$

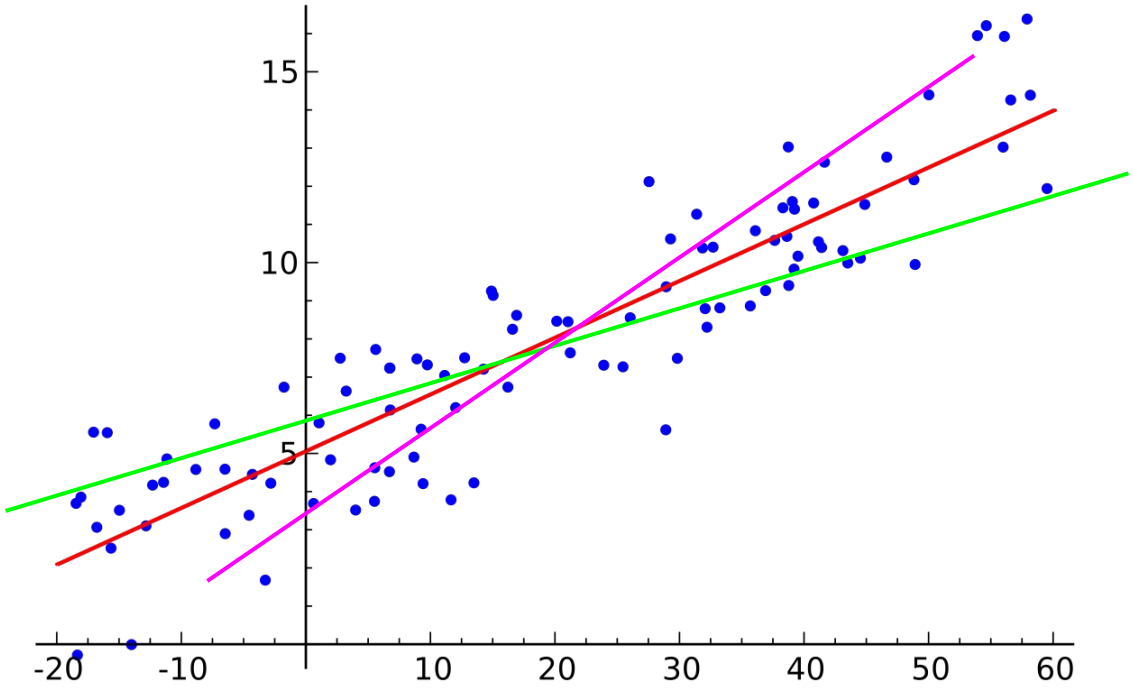
- Pentru antrenare, presupunem existența unui set de date de antrenare
  - $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N} \quad \boldsymbol{x}_n \in \mathbb{R}^D, \;\; y_n \in \mathbb{R}$



$$y = \sum_{d=1}^{D} w_d x_d + b = \boldsymbol{w}^\top \boldsymbol{x} + b$$



Regresia liniara



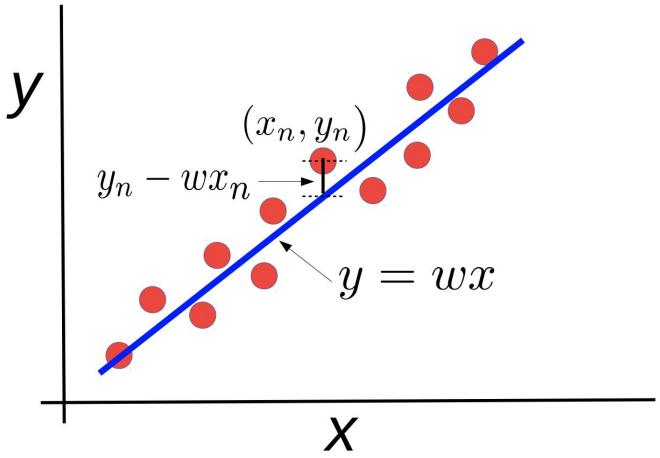
Regresia Liniară

- Eroarea modelului (reziduu) pentru un exemplu:

  $\boxed{y_n - \boldsymbol{w}^\top \boldsymbol{x}_n}$ (sau $y_n - wx_n$ pentru un input scalar - Fig)

- Definim eroarea totală sau „pierderea” pe întreg setul de antrenare (suma reziduurilor):
  - $\mathcal{L}(\boldsymbol{w}) = \sum_{n=1}^{N} (y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2$

- **Optimizarea modelului** constă în găsirea celui mai bun $\boldsymbol{w}$ - care minimizează eroarea de mai sus
  - $\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}) = \arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} (y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2$



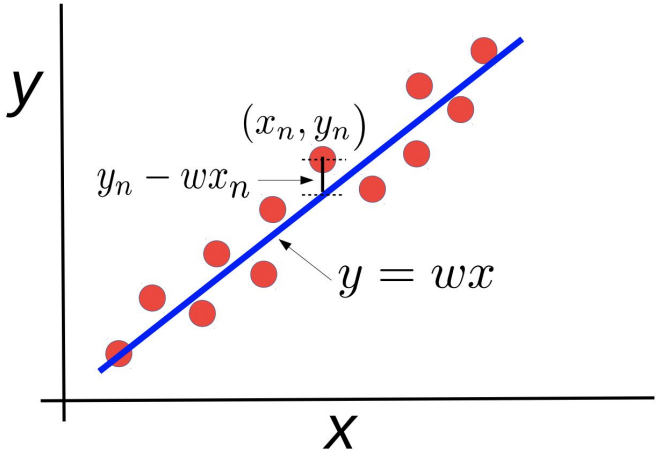
Metoda celor mai mici pătrate (“Least Squares”)

- Abordare standard în analiza de regresie pentru a aproxima soluția sistemelor nedeterminate
  - Seturi de ecuații în care există mai multe ecuații decât necunoscute
  - Minimizarea sumei pătratelor reziduurilor realizate în rezultatele fiecărei ecuații

$$\mathcal{L}(\boldsymbol{w}) = \sum_{n=1}^{N} (y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2$$

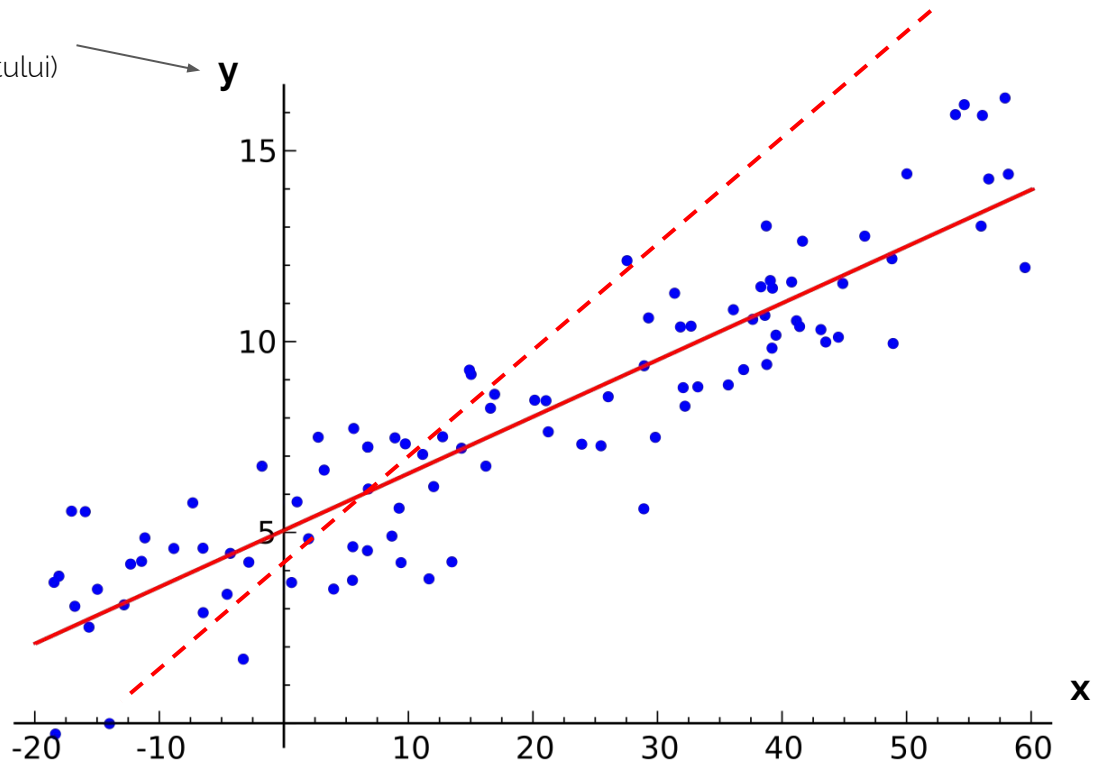
- Luând derivata (gradientul) lui L (w) în raport cu ***W*** și setand la zero (pentru a găsi valoarea minimă)
  - $\sum_{n=1}^{N} 2(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n) \frac{\partial}{\partial \boldsymbol{w}} (y_n - \boldsymbol{x}_n^\top \boldsymbol{w}) = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} \boldsymbol{x}_n (y_n - \boldsymbol{x}_n^\top \boldsymbol{w}) = 0$

- Simplificând în continuare, obținem o soluție de formă închisă pentru ***W***
  - $\boldsymbol{w} = (\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top)^{-1} \sum_{n=1}^{N} y_n \boldsymbol{x}_n = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{y}$



Regresia liniara

Variabila dependenta
(ex.: prețul apartamentului)

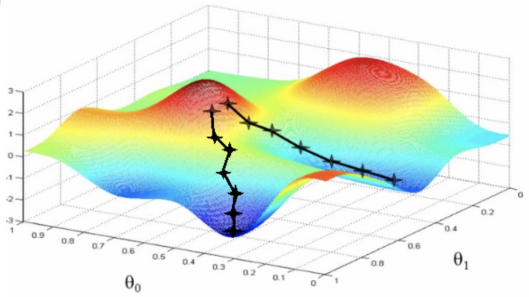


Variabila
independenta
(suprafața
apartamentului)

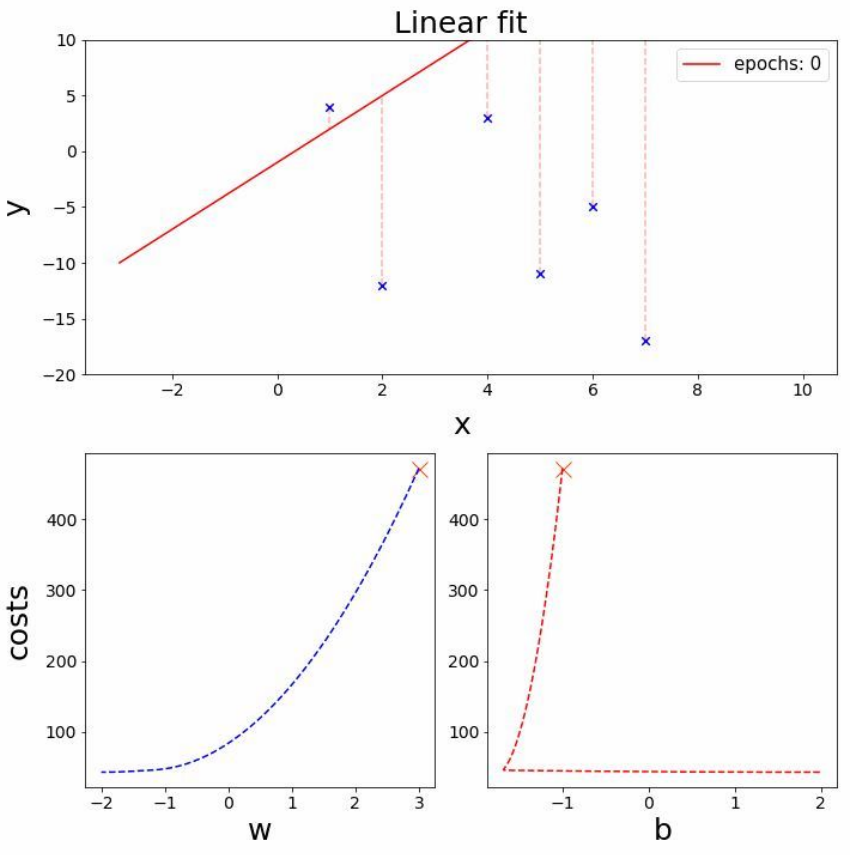
Gradient Descent

- Soluția de formă închisă are următoarele probleme: $(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top y$
  - Matricea $(\mathbf{X}^\top \mathbf{X})$ s-ar putea să nu fie inversabilă
  - Bazându-ne exclusiv pe minimizarea erorii de antrenament putem face **"overfit"** pe setul de antrenare
  - Poate fi foarte ineficientă din punct de vedere computațional atunci când D (dimensionalitatea) este foarte mare

- O modalitate mai rapidă și eficientă este de a utiliza optimizarea iterativă - gradient stocastic ("**Gradient Descent**").
  - Se porneste cu o valoare inițială (aleatoare) $\boldsymbol{w} = \boldsymbol{w}^{(0)}$
  - Actualizăm W deplasându-ne de-a lungul gradientului funcției de pierdere **L**
    - $$\boldsymbol{w}^{(t)} = \boldsymbol{w}^{(t-1)} - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}=\boldsymbol{w}^{(t-1)}}$$ $$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = -\sum_{n=1}^{N} \boldsymbol{x}_n (y_n - \boldsymbol{x}_n^\top \boldsymbol{w})$$
      - Unde, η este rata de invațare (pasul fiecărei iterații)
  - Repetăm până la convergență



Gradient Descent



[Credits]
[Gradient descent animation]

Rezumat

- Loss function
- Optimizare - antrenare (găsirea parametrilor)
- Least Squares
- Gradient Descent

Resurse

- Deep Learning course : CS230 Deep Learning
- Deep Learning course: CS231n: Convolutional Neural Networks for Visual Recognition
- Deep Learning book : https://www.deeplearningbook.org
- Mathematics book : https://mml-book.github.io/
- Machine Learning book (advanced) : Pattern Recognition And Machine Learning - Bishop, 2006
- But what is a GPT? https://www.3blue1brown.com/lessons/gpt
-

Seek out guidance and further resources from the team :)