

**UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI**

**FACULTATEA DE ȘTIINȚE APLICATE**

**Matematică și Informatică Aplicată în Inginerie**

# **PROIECT DE DIPLOMĂ**

**CONDUCĂTOR ȘTIINȚIFIC,  
Lect.univ.dr. Iuliana MUNTEANU**

**ABSOLVENT,  
Soare Robert Daniel**

**București**

**2021**



UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI  
FACULTATEA DE ȘTIINȚE APLICATE  
Matematică și Informatică Aplicată în Inginerie



Aprobat Decan,  
Prof.dr. Emil PETRESCU

# PROIECT DE DIPLOMĂ

Învățarea automată pentru un agent într-un mediu 2D

.....

CONDUCĂTOR ȘTIINȚIFIC,  
Lect.univ.dr. Iuliana MUNTEANU

ABSOLVENT,  
Soare Robert Daniel

București

2021

# Cuprins

|   |           |
|---|-----------|
| <b>Introducere</b>  | <b>4</b>  |
| <b>1 Învățare automată</b>                                | <b>7</b>  |
| 1.1 Istoric . . . . .                                     | 7         |
| 1.2 Clasificare . . . . .                                 | 8         |
| 1.3 Industrie . . . . .                                   | 9         |
| 1.4 Programe software pentru dezvoltare . . . . .         | 10        |
| 1.5 Big Data . . . . .                                    | 10        |
| <b>2 Rețele neuronale artificiale</b>                     | <b>13</b> |
| 2.1 Introducere . . . . .                                 | 13        |
| 2.2 Structură . . . . .                                   | 13        |
| 2.3 Funcții de activare și metode de optimizare . . . . . | 14        |
| 2.4 Tipuri de rețele . . . . .                            | 15        |
| <b>3 Metode de învățare</b>                               | <b>17</b> |
| 3.1 Q-Learning . . . . .                                  | 17        |
| <b>4 Aplicație</b>  | <b>19</b> |
| 4.1 Structură . . . . .                                   | 19        |
| 4.2 Simulator . . . . .                                   | 19        |
| 4.3 Interfață . . . . .                                   | 19        |
| 4.4 Model de învățare . . . . .                           | 19        |
| 4.5 Agent . . . . .                                       | 19        |
| <b>Concluzii finale</b>                                   | <b>19</b> |
| <b>Bibliografie</b>                                       | <b>22</b> |
| <b>Index</b>  | <b>23</b> |



# Introducere

Învățarea automată a devenit un subiect de interes din ce în ce mai important, această fiind utilizată în vaste domenii, precum: industria auto, alimentară, agricolă, bancară, aerospațială și mai cu seamă în industria tehnologiei informației. Unul din rolurile ei cele mai importante constă în analiza și clasificarea datelor, predicția unor evenimente în baza unor fapte deja întâmplare, crearea unui profil virtual pentru un grup de utilizatori, etc.

Datorită marelui conectivități dintre oamenii din ziua de astăzi; sistemele politice, economice și relațiile interumane au devenit extrem de complexe. Totul a devenit interconectat. O idee a unui singur individ poate fi transmisă pe tot globul pământesc, aceasta idee putând afectând milioane de oameni în diverse locuri și a cărui impact politic și economic poate fi greu de estimat. De asemenea, un incident economic local, un dezastru natural, sau un conflict politic dintre două țări pot avea efecte devastatoare asupra economiei globale și a structurii geopolitice curente.

Fiecare eveniment din ziua de astăzi are o influență mai mică sau mai mare asupra acestei mari rețele de sisteme ale civilizației umane. Întrebarea naturală la această dilemă este: putem face o estimare asupra acestor evenimente și ale cazurilor lor speciale? Se poate, și asta datorită faptului că multe evenimente sunt monitorizate și înregistrate, precum: tranzacțiile bancare, documente legislative și juridice, vremea, traseele și destinațiile mașinăriilor de transport marfă (automobile, avioane, vapoare), discursuri și opinii în rețele sociale, date medicale din dispozitive inteligente (telefoane smart, ceasuri și brățari smart), date provenite din simulări virtuale sau experimente.

Tot acest mare volum de informații și metodele de manipulare, stocare întră în așa numita categorie *Big Data*. Analiză acestui volum imens de date devine o sarcină foarte dificilă și laborioasă în cazul metodelor convenționale de analiză a datelor folosind statistică clasică. În esență, învățarea automată se folosește atât de teoria clasică cât și de noile descoperiri în calculul numeric pentru a crea modele matematice dinamice care pot acumula cunoștințe și acționa în baza lor folosind toate datele pe care le primește ca set de învățare.

În această lucrare se va analiza cum algoritmi de învățare automată pot fi folosiți în crearea unui agent autonom care să îndeplinească sarcinile într-un spațiu 2-dimensional. Problema va consta în crearea unui mediu 2-dimensional în care un agent trebuie să ghideze o mașină astfel încât această să parcurgă cu succes un traseu.

În primul capitol este descris termenul de învățare automată, care sunt subdomeniile

sale, cum este folosit în industrie. În al doilea capitol este o mică introducere pentru rețelele neuronale artificiale.

În al treilea capitol vor fi prezentați câțiva algoritmi de învățare, iar al patrulea descrierea aplicației.

# Capitolul 1

## Învățare automată

### 1.1 Istoric

Învățarea automată este o ramură a inteligenței artificiale care se ocupă cu studiul tehnicilor și metodelor prin care se oferă unui calculator abilitatea de a învăța. Prin învățare ne referim la posibilitatea de a oferi o decizie în baza unor cunoștințe deduse din experiențe anterioare.

Multe tehnici din învățarea automată au la bază modelul de interacțiune al neuronilor, descris de către Donal Hebb în cartea sa *The Organization of Behavior* [1]. Termenul de învățare automată (în engleză *machine learning*) a aparut în anul 1953, dat de Arthur Samuel, creatorul unui program de jucat checker, capabil să ia decizii bazate pe experiențele anterioare [2]. În anul 1957, Frank Rosenblatt crează Perceptron-ul - utilizat în crearea unui calculator capabil să recunoască forme într-o imagine - folosindu-se de observațiile din lucrările lui Donald Hebb și Arthur Samuel. Perceptron-ul de unul singur are o putere destul de limitată, dar odată cu descoperirea utilizării sale în combinații de mai multe straturi a dat naștere la termenul de rețea neuronală.

De-a lungul timpului, acest domeniu a avut o evoluție înceată, un factor important fiind capacitățile limitate de procesarea ale calculatoarelor. Dar odată cu avansurile tehnologice, cercetarea în acest domeniu a început să fie din ce în ce mai activă, în ultimii ani culminând cu evenimente care au atras interesului publicului general, precum: IBM's Deep Blue, IBM's Watson, Google's Deepmind și Google's AlphaGo.

## 1.2 Clasificare

Fiind un domeniu foarte vast și cuprinzător, aceasta se împarte în 3 mari categorii:

- Învățare supervizată
- Învățare nesupervizată
- Învățare prin recompensă

În învățarea supervizată, procesul de antrenare se bazează pe analiza unor date formate din perechi de valori intrare-ieșire (set de date etichetat) pentru calibrarea funcțiilor de deducere. Este folosit pentru rezolvarea problemelor de clasificare.

Exemple de algoritmi:

- Support-vector machines
- Regresia liniară
- Regresia logistică
- Arbori de decizie
- Rețele neuronale
- Clasificator bayesian naiv

Pentru învățarea nesupervizată, procesul de antrenare constă în crearea unor modele interne de recunoaștere a unor tipare în urma analizei unui set de date neetichetat. Este deseori folosit în descoperirea similarităților și diferențelor într-un set de date.

Exemple de algoritmi:

- K-means clustering
- Autoencoders
- Analiza componentei principale
- Descompunerea valorilor singulare

În învățarea prin recompensă, procesul de antrenare constă în maximizarea unei funcții de recompensă, modelul calibrându-se astfel încât deciziile luate să ducă spre obținerea unei recompense cât mai mari.

Exemple de algoritmi:

- Monte Carlo
- Q-learning
- SARSA



- Deep Q Network
- Proximal Policy Optimization
- Deep Deterministic Policy Gradient
- Trust Region Policy Optimization

## 1.3 Industrie

Tot mai multe aplicații folosesc tehnici de învățare automată pentru optimizarea produselor, serviciilor și interacțiunilor cu utilizatorii. Cele mai notabile utilizări fiind:

- Algoritmi de căutare a știrilor în baza unor preferințe oferite explicit sau implicit de către utilizator.
- Reclame personalizate generate după profilele utilizatorilor.
- Sisteme de recomandări produse.
- Etichetarea obiectelor sau persoanelor în imagini, înregistrări audio sau video.
- Sisteme robotice autonome.
- Mașini autonome.
- Sisteme meteorologice
- Sisteme de detectare a fraudelor într-un sistem bancar.
- Clasificare și predicția evenimentelor.
- Optimizarea proceselor de producție a mărfurilor.
- Optimizarea procesului de antrenare pentru atleți.

Companiile sunt foarte interesate de modul cum interacționează și percep clienții produselor lor, ele încercând mereu să colecteze informații pentru despre modul cum sunt utilizate produsele în activitatea utilizatorului. Aceste campanii de colectare a datelor a devenit din ce în ce mai agresivă, marile companii software specializate în rețele sociale (Facebook, Twitter, Youtube, LinkedIn, Reddit) vând datele utilizatorilor în vederea oferirii unui profil al consumatorului pentru a stabili interesul pentru produs.

## 1.4 Programe software pentru dezvoltare

Interesul puternic pentru acest domeniu a venit în principal din partea marilor companii software și hardware, ele dezvoltând puternice biblioteci pentru procesarea datelor, crearea de rețele neuronale, algoritmi de învățare, etc. Pentru sprijinirea domeniului, aceste unelte sunt oferite după ca aplicații cu sursă deschisă ( în engleză *open source* ), având o licență deseori foarte permisibilă în vederea utilizării personale și comerciale.

Calitatea acestor unelte le-a făcut să devină un standard în industrie, atât comercială cât și academică.

Exemple de biblioteci sau aplicații software:

- Tensorflow - bibliotecă dezvoltată de către Google în vederea utilizării cu ușurință algoritmilor de învățare, cât și funcții utilitare pentru manipularea datelor.
- PyTorch - bibliotecă dezvoltată de către Facebook pentru protiparea aplicațiilor de viziune computerizate, procesarea limbajului natural, etc.
- ML.NET - bibliotecă dezvoltată de Microsoft pentru crearea rapidă a unor aplicații de procesare a datelor folosind algoritmi de învățare.
- scikit-Learn - bibliotecă care conține funcții statistice folosite pentru analiza datelor.
- Apache Spark - bibliotecă de aplicații destinate pentru procesarea unui volum foarte mare de date.
- Apache Kafka - aplicație care permite stocarea și distribuirea unui volum foarte mare de date în timp real către mai mulți consumatori.
- Caffe - bibliotecă pentru dezvoltare aplicațiilor pentru medii de lucru care nu dispun de o putere de procesare foarte mare, precum dispozitivele mobile.
- Keras - bibliotecă pentru dezvoltarea rețelelor neuronale
- H2O.ai - platformă de procesare și analiză a datelor pentru mediul comercial

## 1.5 Big Data

O componentă esențială pentru învățarea automată este gestionarea datelor care vor fi folosite și produse de către algoritmi algoritmi învățare. Această gestionare a informațiilor, de cele mai multe ori, va intra în cadrul domeniului de *Big Data*

Conform Uniunii Europene: „Big data se referă la volume de date colectate atât de mari și complexe încât este nevoie de noi tehnologii, cum ar fi inteligență artificială, pentru a le procesa. Datele provin din nenumărate surse diferite.”[4]

Volumul de date pe care omenirea îl produce crește de la an la an, ceea ce face analiza și înțelegerea datelor să fie o sarcină din ce în ce mai dificilă. Tot mai mulți oameni încep să

aibă acces la internet, iar numărul de dispozitive inteligente (smart phone, smart watch, smart TV) pe care un individ de dispune crește odată cu avansul tehnologic.

Principalele surse de proveniență ale acestor date sunt:

- Rețele sociale - mesaje, imagini create de utilizatori pentru ași exprima opinia la situația socială, economică și politică - datele pot fi utilizate pentru stabilirea unor tendințe sociale cu privire la activitatea și starea emoțională curentă și viitoare a oamenilor.
- Mediul și natura - date provenite de la sateliți și senzori pentru monitorizarea schimbărilor climatice - folosite pentru predicția posibilelor dezastre naturale cauzate de activitățile omului.
- Sector public - documente, certificate, atestate, adeverințe emise de către instituțiile publice - pot fi utilizate în eficientizarea serviciilor publice.
- Transport - date colectate prin GPS și de la diferiți operatori în domeniul transportului (transportul public, aeroporturi, gări) - pentru optimizarea rutelor și a curselor de transport.
- Sector Medical - fișe medicale ale pacienților - monitorizarea stării de sănătate a cetățenilor, utile pentru detectarea posibilelor amenințări de tip biologic.
- Internetul Lucrurilor (*Internet of Things*) - date provenite de la diverse aparate, precum: telefon, ceas, televizor, senzor de gaz, senzor de umiditate, camere video, etc. - utilizate la monitorizare activității individului cu scopul de a ușura anumite sarcini sau pentru a prevenii incidente.
- Sector industrial - rețele industriale de comunicații (senzori, magistrale de teren, rețele celulare), rapoarte economice - folosite pentru automatizare și îmbunătățirea produselor și a serviciilor.
- Sector bancar - tranzacții financiare, rapoarte - utilizate pentru detectarea fraudelor bancare, stabilirea ratelor la dobânzi, împrumuturi, schimb valutar, etc.

Toate aceste beneficii sunt importante pentru societatea din ziua de astăzi, companii mare concurează pentru crearea de infrastructură și servicii pentru stocarea și examinarea datelor.

Exemple de servicii:

- Amazon Web Services - cel mai mare furnizor de servicii și infrastructură cloud din lume (având peste 200 de soluții software).
- Microsoft Azure
- Google Cloud Platform

- IBM Cloud
- Oracle Cloud
- Alibaba Cloud

## Capitolul 2

# Rețele neuronale artificiale

### 2.1 Introducere

O rețea neuronală artificială este un model computațional inspirat din structura și modul de funcționare al creierului biologic. Conexiunile dintre neuronii artificiali se asemănă sinapselor, fiecare neuron se conectează cu alt neuron prin intermediul unor muchii. Semnalul trimis prin aceste muchii este ponderat de niște parametri numiți ponderi sinaptice. Mai mulți neuroni grupați formează un strat, iar mai multe straturi formează o rețea.

Procesul de învățare presupune găsirea unor valori potrivite pentru ponderile sinaptice astfel încât procesarea semnalului de intrare să ofere rezultatul dorit.

### 2.2 Structură

Structura principală al unui neuron artificial este bazat pe modelul Perceptron-ului al lui Donald Hebb, modelul matematic fiind:

$$y = \varphi \left( \sum_{k=1}^n w_k * x_k + b \right)$$

,unde  $x$  este vectorul de intrare,  $y$  vectorul de ieșire,  $w$  ponderea sinaptică,  $b$  deplasarea și  $\varphi$  este funcția de activare.

Vectorul de intrare este format din numerele reale, aceste numere putând reprezenta: imagini, frecvențe, etichete codificate, valori provenite din senzori, etc. Ponderile sinaptice au rolul de a crește sau descrește puterea semnalului reprezentat de valorile vectorului de intrare. Funcția de activare preia semnalul ponderat și oferă o valoare specifică în baza acestuia. Deplasarea ajută la deplasarea semnalului ponderat pentru o mai bună aproximare necesară pentru îndeplinirea anumitor condiții ale funcției de activare.

**Exemplul 2.2.1** *Un neuron artificial care acționează precum o poarta logică SAU (OR) pentru două numere binare are forma:*

$$y = \varphi(x_1 + x_2 - 0.5)$$

, unde  $x = \{x_i | x_i \in \{0, 1\}\}$ ,  $y \in \{0, 1\}$ ,  $w_1 = 1$ ,  $w_2 = 1$ ,  $b = -0.5$ , iar funcția de activare este:

$$\varphi(u) = \begin{cases} 1 & u \geq 0 \\ 0 & u < 0 \end{cases}$$

**Verificare.** Pentru  $x = [1, 0]$ , avem  $u = 1 + 0 - 0.5 = 0.5$  și  $y = \varphi(u) = \varphi(0.5) = 1$  (același rezultat și pentru  $x = [0, 1]$  - datorită proprietății de comutativitate a adunării). Pentru  $x = [1, 1]$ , avem  $u = 1 + 1 - 0.5 = 1.5$  cu  $\varphi(u) = \varphi(2) = 1$ . Ultimul caz pentru  $x = [0, 0]$ , vom avea  $u =$

**Observația 2.1** Fără funcția de activare, perceptronul acționează precum o funcție liniară. Prin utilizarea unei funcții de activare potrivite, puteam aborda mai ușor problemele neliniare, precum cele pentru clasificarea datelor în diverse categorii.

Un singur perceptron oferă doar o singură valoare de ieșire. Dacă dorim să avem mai multe valori de ieșiri trebuie să mai adăugăm perceptroni. Gruparea de neuroni artificiali se numește *strat*.

Structura unui strat arată astfel în formă matriceală:

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \varphi(u_1) \\ \varphi(u_2) \\ \vdots \\ \varphi(u_n) \end{bmatrix}$$

Rezultatele acestui strat pot fi transmise către un alt strat care poate avea o altă funcție de activare, astfel putem crea modele matematice mai complexe. Această înșiruire de straturi se numește *rețea*.

## 2.3 Funcții de activare și metode de optimizare

Funcția de activare ajută rețeaua neuronală pentru înățarea de tipare complexe aflate în setul de date analizat. Alegerea unei funcții de activare este critică pentru performanța rețelei, în special cazul problemelor neliniare.

Unele din cele mai folosite funcții sunt:

$$\begin{array}{lll}
\text{Identitate} & \varphi(x) & = x \\
\text{Binary Step} & \varphi(x) & = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \\
\text{Logistic, Sigmoid} & \varphi(x) & = \frac{1}{1 + e^{-x}} \\
\text{Rectified liniar unit(ReLU)} & \varphi(x) & = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \\
\text{Softplus} & \varphi(x) & = \ln(1 + e^x)
\end{array}$$

## 2.4 Tipuri de rețele

De-a lungul anilor, au fost create foarte multe tipuri de rețele neuronale artificiale pentru a servii la rezolvarea de probleme din domenii dificile.

Exemple de tipuri de rețele:

- Feed Forward (FD)
- Deep Feed Forward (DFF)
- Radial Basis Network (RBF)
- Recurrent Neural Network (RNN)
- Long/Short Term Memory (LSTM)
- Markov Chain (MC)
- Deep Convolutional Network (DCN)
- Deconvolutional Network (DN)
- Support Vector Machine (SVM)
- Deep Belief Network (DBN)





## Capitolul 3

# Metode de învățare

### 3.1 Q-Learning



## Capitolul 4

# Aplicație

4.1 Structură

4.2 Simulator

4.3 Interfață

4.4 Model de învățare

4.5 Agent



# Concluzii finale

În această lucrare am analizat ....



# Bibliografie

- [1] Hebb, D. O. The organization of behavior : a neuropsychological theory / D.O. Hebb  
Wiley New York 1949
- [2] <http://infolab.stanford.edu/pub/voy/museum/samuel.html>
- [3] <https://www.ibm.com/cloud/learn/unsupervised-learning>
- [4] <https://www.europarl.europa.eu/news/ro/headlines/society/20210211ST097614/big-data-de>
- [5] Cormen, T.H., Leiserson, C.E., Rivest, R.L. *Introducere în algoritmi*, Cluj-Napoca,  
Editura Computer Libris Agora, 2000.
- [6] <https://www.mathworks.com/products/computer-vision.html>

# Index

capitol

    C1, 7

    C2, 9

concluzii, 11

sectiune

    S1.1, 7