

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

Introducere	4
1 Învățare automată	7
1.1 Istoric	7
1.2 Clasificare	8
1.3 Industrie	9
1.4 Programe software pentru dezvoltare	9
2	11
2.1	11
2.2	11
Concluzii finale	11
Bibliografie	14
Index	15

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, industrială, 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 întâmplate, crearea unui profil virtual pentru un grup de utilizatori.

În această lucrare se va analiza cum algoritmi de învățare automată pot fi folosiți în crearea unui agent autonom care să îndeplinească sarcinii într-un spațiu 2-dimensional.

În primul capitol este descris termenul de învățare automată, cum este folosit în industrie.

/newpage

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 neurale
- 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

În ultimii ani, 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.

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 librării pentru procesarea datelor, crearea de rețele neurale, 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 și comercializării.

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

Exemple de librării sau aplicații software:

- Tensorflow - librărie dezvoltată de către Google în vederea utilizării cu ușurință algoritmilor de învățare, cât și funcții utilitare pentru manipularea datelor.
- PyTorch - librărie dezvoltată de către Facebook pentru protipirea aplicațiilor de viziune computerizată, procesarea limbajului natural, etc.
- ML.NET - librărie dezvoltată de Microsoft pentru crearea rapidă a unor aplicații de procesare a datelor folosind algoritmi de învățare.
- scikit-Learn - librărie care conține funcții statistice folosite pentru analiza datelor.
- Apache Spark - colecție 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 - librărie pentru dezvoltarea aplicațiilor pentru medii de lucru care nu dispun de o putere de procesare foarte mare, precum dispozitivele mobile.
- Keras - librărie pentru dezvoltarea rețelelor neurale
- H2O.ai - platformă de procesare și analiză a datelor pentru mediul comercial

Capitolul 2

.....

2.1

2.2

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] Cormen, T.H., Leiserson, C.E., Rivest, R.L. *Introducere în algoritmi*, Cluj-Napoca,
Editura Computer Libris Agora, 2000.
- [5] <https://www.mathworks.com/products/computer-vision.html>

Index

capitol

 C1, 7

 C2, 9

concluzii, 11

sectiune

 S1.1, 7