



UNIVERSITATEA DE VEST DIN TIMIȘOARA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
PROGRAMUL DE STUDII DE LICENȚĂ Informatică

REZOLVAREA SISTEMELOR DE ECUAȚII LINIARE FOLOSIND TEHNICI DIN INTELIGENȚA ARTIFICIALĂ

COORDONATOR:

Prof. Conf. Maftciu-Scai Liviu Octavian

ABSOLVENT:

Pal Adrian

TIMIȘOARA

2021

Abstract

Artificial intelligence can be defined as the simulation of human intelligence processes executed by machines and is one of the most highly growing fields in Computer Science, now having serious applicability in our day to day lives, even in the solving of mathematical equations. When it comes to solving systems of linear equations there are many very well known direct and iterative methods, however, it is yet quite unknown how effective it is to solve these with the help of AI (Artificial intelligence). The purpose of this thesis paper is to investigate and describe the impact that AI has on solving systems of equations, in particular linear equations. In the following pages, there will be presented, step by step, a technique to create a neural network that learns how to solve these types of systems. The study examines the accuracy, effectiveness, and meaningfulness of which neural networks have on solving squared, deterministic systems of linear equations, with unique solutions, within the range dimension between two and ten. There will be presented a supervised learning algorithm, meaning there will be a dataset of ten thousand or more systems alongside with their respective solutions from which the algorithm will be able to learn. The architecture of the neural network that will be used to solve these systems of equations will be of kind perceptron, to be more specific, feedforward neural networks, these are just multi-layer perceptrons that have one or more hidden layers. There will be presented different methods to solve these types of systems of equations along with Python code implementation, using different known libraries like: TensorFlow, Keras and scikit-learn.

Cuprins

Introducere	4
1 Partea teoretică	6
1.1 Definirea unui sistem de ecuații liniare	6
1.2 Inteligența artificială - Rețele neuronale artificiale	6
1.3 Metode de rezolvare a sistemelor de ecuații pentru a crea setul de date, pentru învățare a rețelei	6
1.4 Arhitectura rețelelor neuronale de tip feedforward (multi-layer percep- ton)	6
2 Partea Practică	7
2.1 Introducere (Limbaaj, tehnologii, librării)	7
2.2 Creare seturi de date (condiții - sisteme (dim între 2-6) deterministice pătrate cu soluții unice)	7
2.3 Rețele neuronale de tip perceptron si feedforward - metode de a crea rețelele	7
2.4 Implementarea propriu-zisa (python - Tensorflow, Keras	8
2.5 Design, arhitectura, Diagrame	8

Introducere

Un sistem de ecuații este un grup de două sau mai multe ecuații matematice care conțin aceleași variabile. Sistemele de ecuații liniare au apărut pentru prima oară în Europa, odată cu introducerea a coordonatelor în geometrie, de către René Descartes, în 1637. De fapt, în această nouă geometrie, numită acum geometrie carteziană, liniile și planele sunt reprezentate prin ecuații liniare, iar calcularea intersecțiilor acestora echivalează cu rezolvarea sistemelor de ecuații liniare. Sistemele de ecuații sunt extrem de importante cu o semnificativă aplicabilitate în viața reală, fiind foarte ușor de remodelat pe mai multe tipuri de probleme.

Problema rezolvării sistemelor de ecuații este în continuare una de mare interes în domeniul științific. Sistemele de ecuații de dimensiuni reduse se pot rezolva ușor și relativ repede folosind metode directe, cum ar fi Cramer, Gauss, Gauss-Jordan ETC sau folosind metode iterative precum metoda Jacobi, Gauss-Seidel ETC. Rezolvarea acestor sisteme devine dificilă atunci când sistemul este compus din multe ecuații, cu cât mai multe ecuații, cu atât crește complexitatea rezolvării sistemului. Odată cu apariția sistemelor de calcul, a început să se urmărească și eficientizarea rezolvării sistemelor de ecuații, mai ales unde numărul de ecuații depășește numărul zecilor. Prin metode iterative, cu ajutorul metodelor performante de calcul, se pot rezolva sistemele de ecuații cu o viteză mult mai mare, dar tot nu este suficient. Se întâmpină probleme de timp la sisteme care au un număr foarte mare de ecuații. Aici rolul inteligenței artificiale este unul foarte important.

La mai puțin de un deceniu după ce a spart mașina de criptare nazistă Enigma, lucru care a ajutat forțele aliate să câștige al Doilea Război Mondial, matematicianul Alan Turing a schimbat istoria a doua oară cu o întrebare simplă: „Pot mașinile să gândească?”. Lucrarea lui Turing „Computing Machinery and Intelligence” (Mașini de calcul și inteligența) (1950) și testul său Turing au stabilit scopul și viziunea fundamentală a inteligenței artificiale. Inteligența artificială ar putea fi definită ca și capacitatea unei mașini de a imita funcții umane, cum ar fi învățarea, raționamentul, creativitatea sau planificarea. Există numeroși algoritmi în inteligența artificială care pot fi folosiți pentru rezolvarea sistemelor de ecuații. În esență, un algoritm IA este un subset extins al învățării automate care îi spune sistemului de calcul cum să învețe să opereze pe cont propriu. La rândul său, dispozitivul continuă să câștige cunoștințe pentru a îmbunătăți procesele și a rula sarcinile mai eficient. Exemple de Algoritmi artificiali ar fi: algoritmi genetici (sunt bazati pe principiile geneticii si selectiei naturale, enuntate de Darwin: “supraviețuiește cel care e cel mai bine adaptat”), Particle Swarm Optimization, glowworm swarm optimization (GSO), algoritmi mematici (AMs) ETC, acești algoritmi ușurează procesul de rezolvare a sistemelor de ecuații de acest fel. În paginile ce urmează va fi prezentat modul de rezolvare a sistemelor de ecuații liniare cu ajutorul rețelelor neuronale, mai exact rețele neuronale de tip feedforward.

O rețea neuronală reprezintă o serie de algoritmi care încearcă să recunoască relațiile care stau la baza unui set de date printr-un proces care imită modul în care funcționează creierul uman. Se va urmări și analiza în detaliu impactul și modul de rezolvare a sistemelor de ecuații liniare cu aceasta arhitectură. Rețelele neuronale de tip feedforward sunt rețele neuronale artificiale în care conexiunile dintre unități nu formează cicluri (bucle). Rețelele neuronale feedforward au fost primul tip de rețea neuronală artificială inventată. Ele sunt numite feedforward deoarece informația este transportată doar înainte în rețea (fără bucle), mai întâi prin nodurile de intrare, apoi prin nodu-

riile ascunse (dacă sunt prezente) și în final prin nodurile de ieșire. Pentru ca rețeaua neuronală să fie capabilă să învețe să rezolve sistemele de ecuații, se va folosi un algoritm de învățare supravegheat. Învățarea supravegheată, care mai este cunoscută și sub denumirea de învățare automată supravegheată, este o subcategorie a învățării automate și a inteligenței artificiale. Este definită prin utilizarea seturilor de date etichetate pentru a antrena algoritmi să clasifice datele sau să prezică rezultatele cu acuratețe.

Vor fi prezentate două metode principale de creare a setului de date. Prima metodă va fi de a stoca toate sistemele de ecuații într-un singur set de date, indiferent de dimensiune, astfel încât stratul de intrare (input layer) va fi format întotdeauna dintr-un singur neuron cu un set de liste imbricate cu lungime variabilă (ragged tensor) care conține coeficienții, soluțiile, iar pentru partea de antrenament, apare și un neuron cu valorile variabilelor sistemului. Implementarea metodei a doua este mai eficientă din punct de vedere al antrenamentului deoarece vor fi create mai multe seturi de date, în funcție de dimensiunea sistemului. Stratul de intrare este format din $n + 1$ neuroni, unde n reprezintă numărul de ecuații din sistem, fiecare neuron va conține coeficienții ecuației respective și un neuron este rezervat pentru soluții.

Capitolul 1

Partea teoretică

1.1 Definirea unui sistem de ecuații liniare

1.2 Inteligența artificială - Rețele neuronale artificiale

1.3 Metode de rezolvare a sistemelor de ecuații pentru a crea setul de date, pentru învățare a rețelei

Subiecte tratate:

1. Prezentarea matrice sistem, determinanti - cum ar trebui sa fie determinantul astfel incat sa avem un sistem de ecuatii liniare compatibile determinate cu o singura solutie unica (setul de date va avea doar astfel de sisteme).
2. metode directe: rezolvare prin Cramer, Gauss, Gauss-Jordan / metode indirecte: Jacobi, Gauss-Seidel (pentru a învăța rețeaua neuronală, trebuie rezolvat sistemul).

1.4 Arhitectura rețelelor neuronale de tip feed-forward (multi-layer perceptron)

Subiecte tratate:

1. Ce este un perceptron?
2. învățare / antrenare supravegheată
3. funcții de activare
4. funcții de pierdere / cost

Capitolul 2

Partea Practică

- 2.1 Introducere (Limbaj, tehnologii, librării)
- 2.2 Creare seturi de date (condiții - sisteme (dim
între 2-6) deterministice pătratice cu soluții unice)
- 2.3 Rețele neuronale de tip perceptron si feedforward
- metode de a crea rețelele

2.4 Implementarea propriu-zisa (python - Tensorflow, Keras)

2.5 Design, arhitectura, Diagrame

Deoarece algoritmul rețelei neuronale va folosi o metoda de antrenare supravegheata, este foarte importantă setarea unor etape de urmat în implementare.

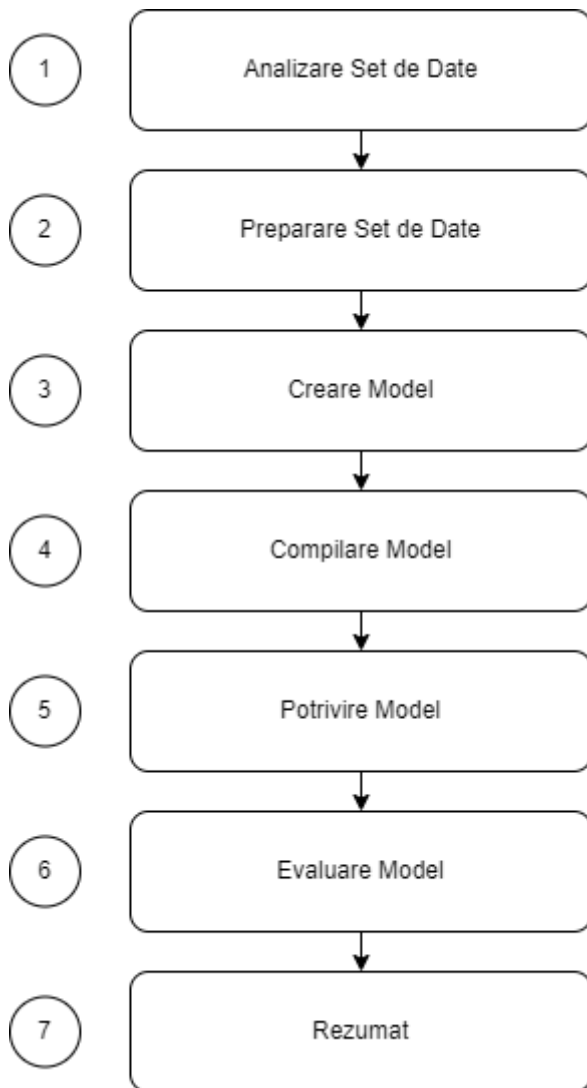


Figura 2.1: Etapele de antrenare retea neuronală

Dataset-ul este format din: Coeficientii liniari ai sistemului - matricea sistemului, Soluție și Output (necunoscute).

Arhitectura rețelei neuronale este de tip feedforward. Cel mai simplu tip de rețea neuronală de tip feedforward este o rețea perceptron cu un singur strat, care constă dintr-un singur strat de noduri de ieșire; intrările sunt alimentate direct la ieșiri printr-o serie de greutăți. În fiecare nod se calculează suma produselor greutăților (weights) și intrărilor (inputs), iar dacă valoarea este peste un anumit prag (de obicei 0) neuronul se declanșează și ia valoarea activată (1); în caz contrar, ia valoarea dezactivată (-1).

Neuronii cu acest tip de funcție de activare sunt numiți și neuroni artificiali sau unități de prag liniare.

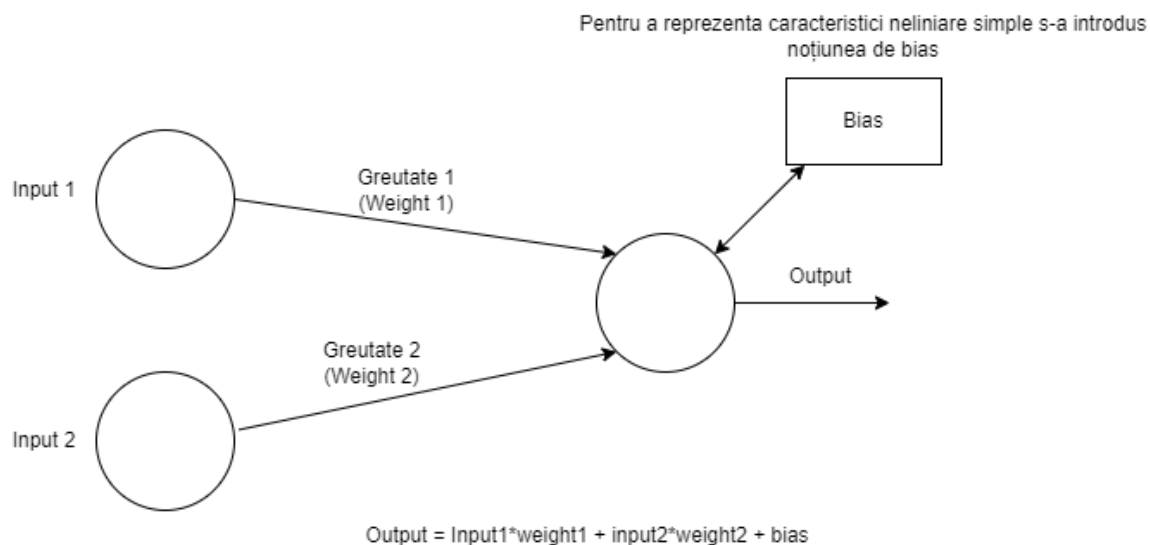


Figura 2.2: Arhitectura rețea neuronale de tip Perceptron

Rețeaua neuronală feedforward a fost primul tip de rețea neuronală artificială concepută. În această rețea, informația se deplasează într-o doar înainte de la nodurile de intrare, prin nodurile ascunse și la nodurile de ieșire. Nu există cicluri sau bucle în astfel de rețea. Rețeaua neuronală feedforward fara stratul ascuns nu este altceva decât un perceptron.

Fie urmatorul sistem patratice, deterministic, cu solutie unica:

$$\begin{cases} 3X + 5Y + Z = 23 \\ 7X - 2Y + 4Z = 8 \\ -6X + 3Y + 2Z = 15 \end{cases}$$

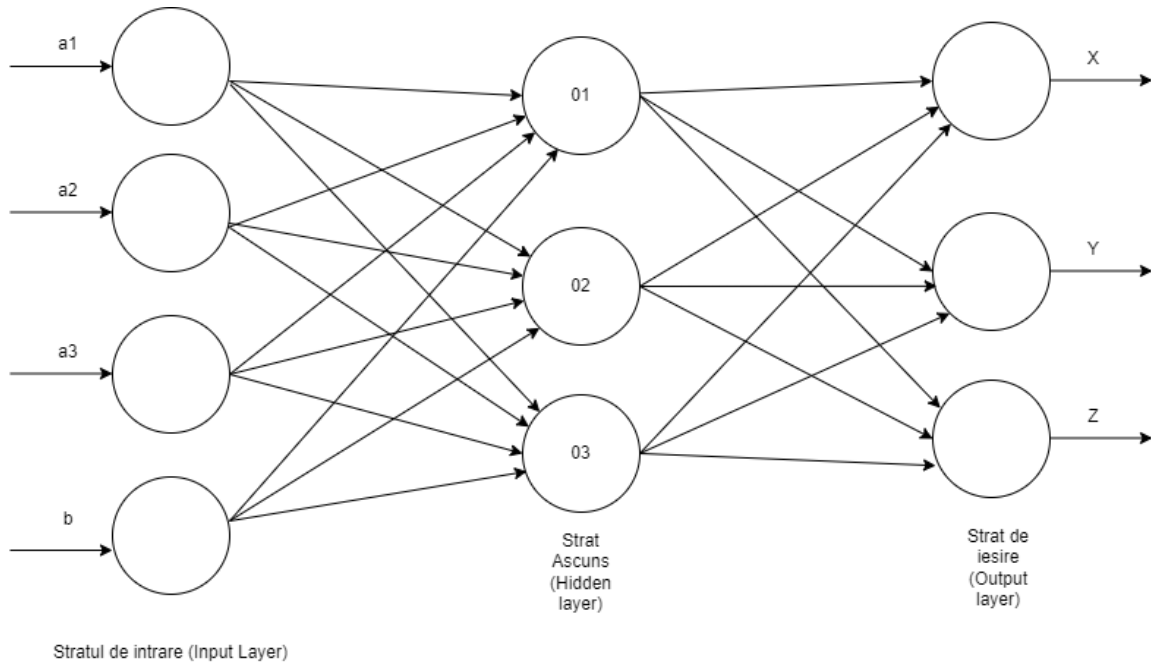


Figura 2.3: Arhitectura rețea neuronală de tip feedforward pentru rezolvarea sistemelor de ecuatii liniare patratice - 4/3/3

a) Input Layer

Contine 3 neuroni, unde n reprezinta numarul de ecuatii liniare din sistem, in cazul mentionat mai sus 3, fiecare neuron reprezinta un tensor cu coeficientii ecuatiilor. Input1 - a1: [3,5,1], Input2 - a2: [7,-2,4], Input3 - a3: [-6,3,2] iar acel + 1 neuron (b) reprezinta rezultatul sistemului de ecuatii adica tensorul [23,8,15].

b) Hidden Layer

Fiecare neuron reprezinta rezultatul obtinut de la expansiunea permutarii procesului. Funcția de activare aplică ponderi intrărilor și le direcționează printr-o funcție de activare ca ieșire. Pe scurt, straturile ascunse efectuează transformări neliniare ale intrărilor intrate în rețea.

c) Output Layer

Contine n neuroni , fiecare neuron reprezinta valoarea unei variabile. X,Y si Z

Bibliografie

- [1] François Chollet (2021) *Deep Learning with Python, Second Edition*
- [2] Giovanni Landi, Alessandro Zampini (2018) *Linear Algebra and Analytic Geometry for Physical Sciences*
- [3] Aurélien Géron (2017) *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd Edition*
- [4] Morgan Maynard (2020) *Neural Networks: Introduction to Artificial Neurons, Backpropagation and Multilayer Feedforward Neural Networks with Real-World Applications*