**UNIVERSITATEA DE MEDICINĂ, FARMACIE, ȘTIINȚE ȘI TEHNOLOGIE “GEORGE EMIL PALADE”**

**DIN TÂRGU-MUREŞ**

**FACULTATEA DE INGINERIE ȘI TEHNOLOGIA INFORMAȚIEI SPECIALIZAREA: INFORMATICĂ**

**Recunoașterea vorbitorului pe baza calculului neuronal**

**Îndrumător științific: Absolvent:**

Prof. univ. dr. Călin Enăchescu Hălmaciu Ioan Cosmin

**– 2020 –**

A close up of text on a white background

Description automatically generated

Cuprins

[1 INTRODUCERE 4](#_Toc44569598)

[1.1 MOTIVAȚIE 4](#_Toc44569599)

[1.2 STRUCTURA LUCRĂRII 5](#_Toc44569600)

[2 NEURONUL 6](#_Toc44569601)

[2.1 INTRODUCERE IN NEURON 6](#_Toc44569602)

[2.2 PROPRIETĂȚILE FUNCȚIONALE ALE NEURONULUI 7](#_Toc44569603)

[2.3 DESPRE NEURONUL ARTIFICIAL 9](#_Toc44569604)

[2.4 CALCULUL STIMULULUI 11](#_Toc44569605)

[3 REȚELE NEURONALE 16](#_Toc44569606)

[3.1 INTRODUCERE 16](#_Toc44569607)

[3.2 PRINCIPALELE DIFERENȚE DINTRE REȚELELE NEURONALE BIOLOGICE ȘI ARTIFICIALE 17](#_Toc44569608)

[3.3 STRUCTURA REȚELEI NEURONALE 21](#_Toc44569609)

[3.4 TIPURI DE REȚELE NEURONALE 22](#_Toc44569610)

[4 RECUNOAȘTEREA VORBITORULUI 37](#_Toc44569611)

[4.1 INTRODUCERE 37](#_Toc44569612)

[4.2 TRĂSĂTURILE VORBITORULUI 37](#_Toc44569613)

[4.3 INTEGRAREA TRĂSĂTURILOR ÎN APLICAȚIE 38](#_Toc44569614)

[5 IMPLEMENTARE 40](#_Toc44569615)

[5.1 INTRODUCERE 40](#_Toc44569616)

[5.2 EXTRAGEREA TRĂSĂTURILOR 41](#_Toc44569617)

[5.3 ARHITECTURA REȚELEI NEURONALE 43](#_Toc44569618)

[5.4 PROCESUL DE VERIFICARE 47](#_Toc44569619)

[5.5 CLASELE 49](#_Toc44569620)

[6. MANUAL DE INSTRUCȚIUNI 51](#_Toc44569621)

[7. CONCLUZII 57](#_Toc44569622)

[BIBLIOGRAFIE 58](#_Toc44569623)

# 1 INTRODUCERE

## 1.1 MOTIVAȚIE

Încă din copilărie am fost fascinat de complexitatea calculatoarelor. Fascinat în legătură cu câtă ușurință poți stoca și manevra date într-o cantitate numeroasă, cum poți crea o aplicație lăsându-ți imaginația să zboare, de la cele mai practice și folositoare aplicații, la cele mai complexe sisteme de operare sau cele mai creative și animate jocuri. Inteligența artificială, pe de o parte, a devenit elementul față de care m-am simțit, în mod inedit, atras. O imitație de inteligență susținută de un procesor cu funcții matematice complexe, capabile să egaleze, ba chiar să depășească, cei mai buni jucători de șah și specialiști din diverse domenii.

Potențialul inteligenței artificiale, pentru mine, la aceea vârstă, era ceva incontestabil. Astfel, am dorit să descopăr cum ar putea o aplicație, care face același lucru de 100 de ori, să obțină rezultate din ce în ce mai bune când însăși Albert Einstein declara definiția nebuniei ca fiind să faci același lucru de mai multe ori în speranța unui rezultat diferit.

Mai târziu însă, am reușit să aflu, la nivel teoretic, principiile de funcționare din spatele unei inteligențe artificiale. Acest lucru m-a îndemnat să concluzionez că sunt martorul unui eveniment egal, dacă nu superior, revoluției industriale și anume: o eră a progresului tehnologic care va marca pe veci omenirea. Acest lucru mă motivează să particip, în mod direct, în calitate de programator, la această revoluție a tehnologiei.

Recunoașterea vocală reprezintă un domeniu de interes la nivel internațional. Aplicațiile de recunoaștere vocală sunt folosite în domeniul securității, fiind mult mai practic să te identifici cu o voce decât printr-un PIN. PIN-ul sau parola necesită memorare, în timp ce identificarea printr-o aplicație de recunoaștere vocală nu necesită acest lucru.

În această ordine de idei, sunt motivat să lucrez și descopăr noi utilizări și domenii de lucru în care inteligența artificială este direct implicată, în speranța eficientizării și ușurării procesului de lucru uman.

## 1.2 **STRUCTURA LUCRĂRII**

În **Capitolul 2** sunt prezentate elementele generale ale neuronului artificial. În primul subcapitol sunt prezentate pe scurt elementele unui neuron biologic. În subcapitolul 2 e prezentat modul de funcționare a neuronului și ce determina această funcționare. În subcapitolul 3 e prezentat neuronul artificial, modul în care neuronul artificial imită neuronul biologic și diferențele dintre aceștia. În subcapitolul 4 este prezentat modul în care neuronul artificial reacționează la un stimul extern și cum acesta generează un stimul.

În **Capitolul 3** sunt prezentate elementele generale ale unei rețele neuronale. Primul subcapitol conține o introducere explicativă asupra conceptului de rețea neuronală artificială . În subcapitolul 2 sunt prezentate diferențe între o rețea neuronală artificială și una biologică. În subcapitolul 3 sunt prezentate diferite structuri ale unei rețele neuronale. În subcapitolul 4 este prezentat procesul de învățare al rețelei neuronale.

În **Capitolul 4** sunt prezentate elementele generale de procesare a datelor pentru recunoașterea vorbitorului. Subcapitolul 1 conține o introducere în care o să fie explicat mai în detaliu acest proces. În subcapitolul 2 e prezentată modalitatea de extragere a trăsăturilor vorbitorului și ce reprezintă acestea. În subcapitolul 3 e prezentată aplicabilitatea acestor elemente într-o aplicație.

În **Capitolul 5** este prezentat procesul de implementare al aplicației și algoritmii din spatele acesteia. În subcapitolul 1 sunt prezentați primii pași parcurși pentru implementare. În subcapitolul 2 e prezentată implementarea procesului de extragere a trăsăturilor. În subcapitolul 3 e prezentată rețeaua neuronală în ansamblul ei, cu toate funcționalitățile sale. În Subcapitolul 4 sunt prezentate clasele și diagramele de implementare a aplicației.

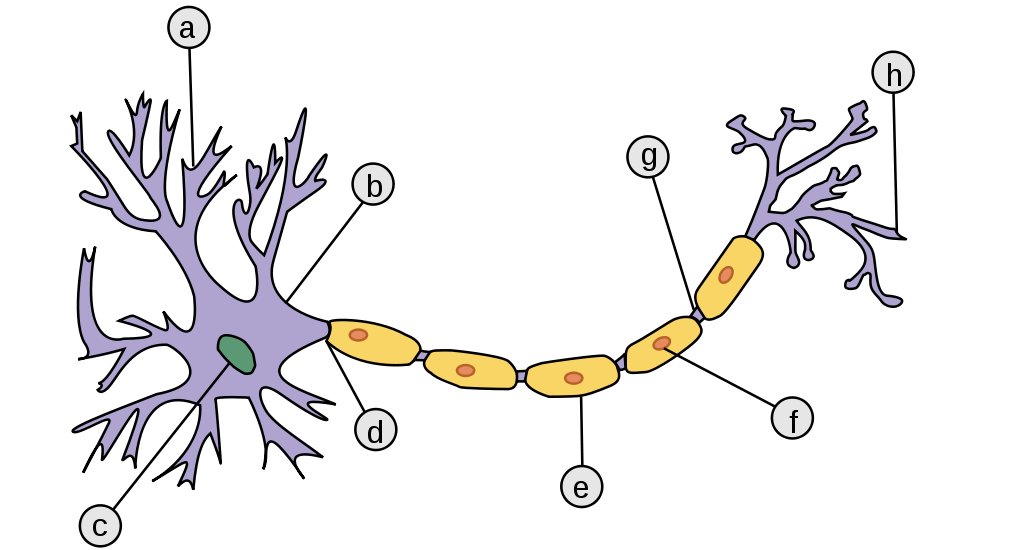
2 NEURONUL

## 2.1 **INTRODUCERE IN NEURON**

Pentru a putea utiliza aceste informații în domeniul inteligenței artificiale este necesar să menționez ce componente are un neuron biologic și, în special, să fac înțeleasă funcționalitatea acestuia.

Conform definiției dicționarului limbii române neuronul este “elementul de bază al sistemului nervos”. Neuronii sunt clasați în trei mari categorii. Conform funcționalității acestora: neuron senzorial, neuron motor, și în ultimul rând tipul de neuron care reprezintă cel mai mare interes în tratarea acestui subiect, inter-neuronul, care are rolul de a conecta ceilalți neuroni formând un circuit neuronal. Neuronul senzorial răspunde la stimuli precum atingerea, sunetele sau lumina, care afectează organele de simț, pielea, și răspunde prin a trimite un semnal la creier. Neuronul motor primește semnale de la creier, cu rolul de a coordona tot ceea ce înseamnă contracții musculare. **[6]**

Un neuron are un corp celular denumit „soma” și un număr mare de prelungiri.



**Figura 2-1 Reprezentarea biologică a unui neuron [6]**

În imaginea alăturată, se observă o reprezentare a unui neuron, unde punctul A reprezintă dendrite, punctul B pericarion, punctul C nucleu, punctul D conul de emergență al axonului, punctul E teacă de mielină, punctul F celula Schwann, punctul G strangulație Ranvier și punctul H butoni terminali.[6]

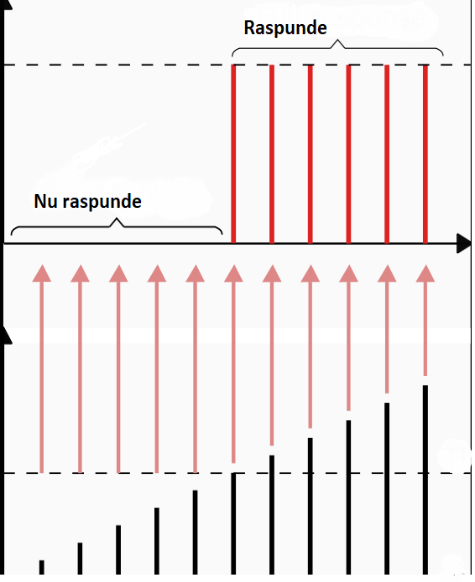
Aceștia mai pot fi clasificați și după numărul de prelungiri în neuroni multipolari având: un număr mare de prelungiri, neuroni bipolari cu două ramificații la extremități, neuroni pseudo-unipolari cu o singură prelungire și în neuroni unipolari cu o singură prelungire axonică.

În general, un om deține între 40 de miliarde și 100 miliarde de neuroni, cu un număr cuprins între 7000 și 10000 de conexiuni sinaptice. Sinapsele sunt cele care asigură conectivitatea neuronilor. Dimensiunea unui neuron este cuprinsă între 100-200 μm și 4-8 μm, unde μm reprezintă un micrometru, un micrometru are 0.0001 cm.**[6]**

## 2.2 **PROPRIETĂȚILE FUNCȚIONALE ALE NEURONULUI**

Neuronul biologic are cinci proprietăți funcționale, și anume: excitabilitatea reprezintă proprietatea de a reacționa la un stimul prin activare, membrana jucând un rol important în această funcție deschizându-și sau închizându-și canalele în funcție de impulsul energetic. În cazul neuronului biologic, acesta are cinci proprietăți funcționale, și anume: excitabilitatea, conductibilitatea, degenerescență, regenerarea, activitatea sinaptică. Excitabilitatea reprezintă proprietatea de a reacționa la un stimul prin activare, membrana jucând un rol important în această funcție, deschizându-și sau închizându-și canalele în funcție de impulsul energetic. Degenerescența e proprietatea care asigură degradarea în condiții de lezare. Conductibilitatea este proprietatea care asigură conducerea impulsurilor electrice. Regenerarea reprezintă capacitatea neuronului de a se reface după anumite lezări. A cincea proprietate, activitatea sinaptică, asigură codarea chimică a informației și transmiterea acesteia prin sinapse.**[6]**

Un alt element important al neuronilor este respectarea principiului denumit “totul sau nimic”. Acest principiu se referă la faptul că, dacă un neuron răspunde la un impuls trimis din exterior sau trimis de către un alt neuron, în eventualitatea în care intensitatea semnalului trimis depășește un anumit prag, acesta va răspunde oferind la rândul său un impuls la o intensitate maximă.



**Figura 2-2 Răspunsul unui neuron în funcție de semnalul primit[6]**

În imaginea alăturată se poate observa cum, un neuron primește impulsuri electrochimice și răspunde la aceeași intensitate maximă, doar după ce răspunsul a depășit un prag minim, specific fiecărui neuron. Indiferent de intensitatea impulsului, dacă condiția de depășire a pragului este îndeplinită, neuronul va răspunde la intensitate maximă. Valoarea pe care stimulii trebuie să o depășească pentru a se propaga se numește “valoare prag”. Stimulii sub “valoarea prag” se numesc stimuli subliminali și produc potențiale locale.

Intensitatea minimă a unui stimul electrochimic aplicat pe un neuron și care produce un răspuns din partea neuronului se numește reobază. Dacă avem un neuron A și un neuron B, timpul util reprezintă timpul de la aplicarea impulsului electrochimic de către neuronul A către neuronul B până la obținerea unui răspuns din partea neuronului B, sau mușchiului pe care neuronul B îl deservește.**[6]**

Pentru domeniul inteligenței artificiale cele mai importante proprietăți funcționale sunt cele de excitabilitate și de conductibilitate. Conectivitatea neuronilor, capacitatea lor de a transmite impulsuri și de a răspunde acestor impulsuri constituie elementul de bază al calculului neuronal cât și al neuronului biologic.

## 2.3 **DESPRE NEURONUL ARTIFICIAL**

La fel ca neuronul biologic, neuronul artificial reprezintă cea mai importantă componentă dintr-o rețea neuronală artificială. Proprietățile funcționale ale neuronului biologic vor fi înlocuite cu funcții matematice, iar componentele acestuia vor fi înlocuite cu diferite variabile sau proprietăți. Există mai multe tipuri de neuroni artificiali într-o rețea neuronală: neuron de intrare, neuron ascuns, neuron de ieșire.**[5]**

Neuronii de intrare reprezintă neuronii senzoriali, cei care sunt conectați la exterior și propagă stimulul mai departe imitând conectivitatea neuronilor biologici. Captarea și stocarea acestor stimuli se face cu ajutorul unei liste de intrare. Lista de intrare reprezintă primul element pe care un neuron artificial îl deține. În această listă sunt valori numerice cuprinse între -1 și 1, în majoritatea cazurilor, aduse la această formă prin normalizare. În cazul neuronilor de intrare, valorile din interiorul listei vor reprezenta stimulii primiți din exterior, singura lui sarcină fiind de a propaga mai departe celorlalți neuroni aceste valori.

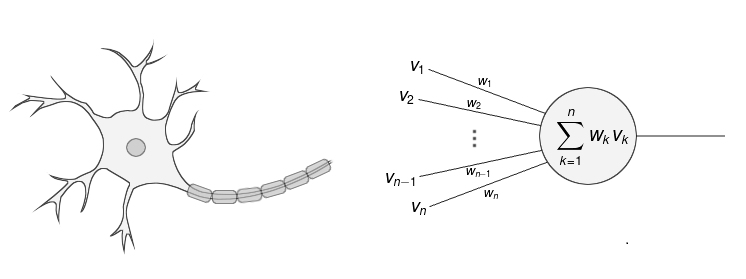
Neuronul ascuns este cel care imită comportamentul unui neuron inter-neuronal. Acesta primește de la neuronul de intrare un stimul față de care va genera un răspuns. Răspunsul va fi propagat mai departe celorlalți neuroni din interiorul rețelei neuronale. În comparație cu neuronul de intrare, care are rolul doar de a propaga stimulii, în interiorul neuronului ascuns se efectuează calcule pentru stabilirea pragului minim.

Neuronul de ieșire se comportă la fel ca un neuron ascuns, cu excepția faptului că, acesta propagă stimulii în exteriorul rețelei neuronale cu ajutorul unei variabile de ieșire în care este stocat sub formă numerică intensitatea stimulului generat. În interiorul neuronului de ieșire au loc aceleași calcule ca la neuronul ascuns.

Neuronul artificial nu imită doar cele trei tipuri de neuroni, imită și componentele acestora, de exemplu: Neuronul de ieșire, cu ajutorul listei de intrare, reușește să imită comportamentul dendrite. În cazul neuronului biologic, dendrita e o componentă a neuronului cu rolul de a transmite impulsurile nervoase către alte celule primite de la corpul celular soma. Corpul celular soma este imitat cu ajutorul funcției de activare a neuronului. Soma reprezintă funcția care calculează valoarea intensității semnalului generat de către neuron, și mai apoi, determină dacă pragul minim a fost depășit sau nu. **[4]**

Axonul este reprezentat de către o variabilă de ieșire ajutată împreună cu mai multe funcții. În neuronul biologic axonul reprezintă terminația nervoasă asemănătoare cu dendrita, însă total opusă, în sensul în care, dacă dendrita are rolul de captare a stimulilor, axonul are rolul de propagare a stimulilor noi generații.

Neuronii artificiali, în funcție de datele propagate, mai pot fi binari sau reali. Neuronii reali trimit valori numerice cuprinse între -1 și 1, în timp ce neuronii artificiali, pot trimite doar 2 valori numerice, anume 0 sau 1. **[4]**



**Figura 2-3 Reprezentare a neuronului biologic și al neuronului artificial [7]**

În imaginea de deasupra putem observa cum arată un neuron biologic lângă un neuron artificial.

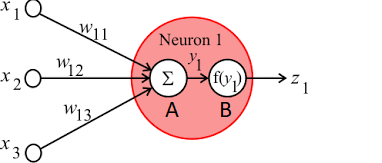
Cu literă V sunt reprezentate dendritele care captează stimulii, cu litera W este reprezentată tăria sinaptică. În interiorul neuronului se calculează intensitatea semnalului cu ajutorul funcției de activare, imitând comportamentul corpului celular soma din neuronul biologic. La extremitate se află axonul care propagă semnalul mai departe, dacă condițiile sunt îndeplinite.

## 2.4 **CALCULUL STIMULULUI**

Neuronul artificial va genera un stimul, la o anumită intensitate, reprezentată de o valoare numerică. Determinarea acestei intensități se realizează printr-o serie de calcule în interiorul neuronului, care, în majoritatea cazurilor, va fi reprezentat cu ajutorul programării orientate pe obiect, sub forma unui obiect.

Am stabilit anterior că neuronul artificial conține o listă și o variabilă, o listă în care sunt stocați stimulii recepționați de către neuron, și variabila care stochează stimulul generat de neuron pentru a fi trimis mai departe. Neuronul artificial mai conține o a doua listă în care sunt stocate legăturile sinaptice cu ceilalți neuroni. Aceste legături vor fi reprezentate la fel, numeric, cu valori cuprinse între –1 și 1. Tăriile sinaptice vor fi discutate în detaliu în capitolul Rețele Neuronale. **[4]**

Pentru a putea calcula stimulul avem nevoie de 2 funcții: funcția de integrare și funcția de activare.



**Figura 2-4 Reprezentarea neuronului biologic[8]**

În imaginea alăturată x1, x2, x3 reprezintă stimulii captați care vor fi stocați în lista de intrare. W11, W12, W13 reprezintă tăriile sinaptice prin care stimulul x1, x2, x3 ajunge la neuron. Aceștia vor fi stocați în lista tăriilor sinaptice.

Punctul A reprezintă funcția de integrare,Y1 reprezintă rezultatul funcției de integrare numită global input. Punctul B reprezintă funcția de activare care va folosi ca argument y1. Z1 reprezintă intensitatea stimulului stocat și trimis prin variabila de ieșire. Când construim un neuron artificial putem alege oricare dintre cele patru funcții de integrare. Cele patru funcții de integrare sunt: sumă, produs, minim, maxim. **[2]**

*Sumă*



*Produs*





Minim



*Maxim*

Pentru a ușura calcularea variabilei global input cu ajutorul funcției de integrare putem folosi un nou calcul în care calculăm produsul dintre stimul și tăria sinaptică prin care stimulul ajunge la neuron, mai exact înmulțim fiecare element din input cu fiecare element corespunzător din lista tăriilor sinaptice (xi \* wji). De exemplu, dacă avem un neuron artificial care primește doi stimuli x1, x2 unde x1 e reprezentat numeric 0.4 și x2 0.6, și acești stimuli ajung în neuron prin legăturile sinaptice w11 și w12, unde w11 e reprezentat numeric 0.75 și w11 0.85, putem crea o listă auxiliară în care adăugăm produsul x1 \* w11 și x2 \* w12, și anume lista va conține elementele [0.4 \* 0.75, 0.6\*0.85]-> [0.3,0.51]. Apoi indiferent de funcția aleasă putem aplica rezultatele obținute anterior, de exemplu:

Pentru Sumă, global input = 0,3 + 0.51 = 0.81.

Pentru Produs, global input = 0.3 \* 0.51 = 0.153

Pentru Minim, global input = min (0.3,0.51) = 0.3.

Pentru Maxim, global input = max (0.3,0.51) = 0.51.

Global input va fi folosit ca argument în funcția de activare, la fel cum, lista auxiliară a fost folosită ca argument în funcția de integrare. **[2]**

Funcția de activare calculează starea neuronului, în cazul neuronului biologic starea poate fi: activ sau inactiv. În cazul neuronului artificial starea e reprezentată, în mod real, sub formă numerică cuprinsă între [-1 ; 1] sau [0 ; 1], sau în formă binară, imitând comportamentul neuronului biologic, având doar 2 stări, 1 sau 0, activ sau inactiv. **[12]**

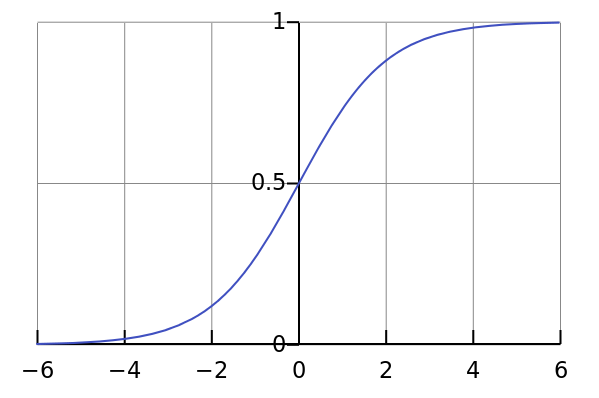
Funcția de activare este cea care decide dacă neuronul devine activ. Având doar funcția de integrare prin însumarea produselor dintre intrări și tării sinaptice obținem global input care poate să aibă orice valoare, atât negativă cât și pozitivă. Având doar această valoare, neuronul nu poate realiza dacă trebuie sau nu să devină activ. **[11]** Din această cauza avem nevoie de o nouă funcție, funcția de activare, care verifică valoarea la global input și decide dacă neuronul este sau nu activ. Există și funcții de activare ineficiente precum: funcție de tip treaptă, care funcționează conform principiului prag minim, și anume dacă global input trece de o anumită valoare, de un anumit prag, atunci neuronul să fie activ, în caz contrar să fie inactiv, respectiv să genereze valoarea 1 dacă trece de prag, și 0 dacă nu trece de prag. Această funcție este ineficientă deoarece rețelele neuronale pe care le folosesc sunt dificile de antrenat. De asemenea și funcțiile de tip liniare, unde activarea e proporțională cu intrările, sunt ineficiente.

Conform definiției, funcția liniară este o funcție care formalizează o relație dintre două spații vectoriale, o funcție este liniară dacă respectă condiția de omogenitate și de aditivitate, și anume, oricare ar fi: **[9]**

x, y aparține U , f(x + y) = f(x) + f(y) ( aditivitate)

x aparține U, astfel încât a aparține K , f(a \* x) = a \* f(x) (omogenitate)

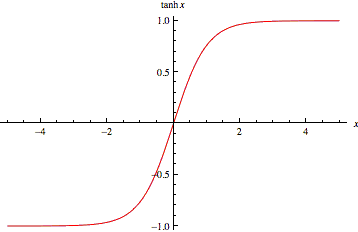
În acest caz, neuronul va genera valori cuprinse între (0, 1) și nu doar 2 posibile valori. Funcțiile liniare nu au comportament binar. Problema cu funcțiile liniare o reprezintă derivata, care este o constantă. În cazul funcțiilor liniare, deoarece rezultatul va fi mereu o altă funcție liniară, indiferent de câte straturi de neuroni avem, aceasta se va comporta ca și când avem un singur strat. Este de precizat că, neuronul artificial poate fi programat astfel încât să aibă un comportament binar indiferent de funcția aleasă. Funcția de activare trebuie să fie o funcție care nu este lineară deoarece în general rețeaua neuronală urmărește să producă o decizie nelineară, decizie din urma unor valori de intrări și tării sinaptice nelineare. Funcția sigmoidală este nelineară și nebinară. **[9]**



**Figura 2-5 Graficul Funcției sigmoidale [9]**

După cum putem observa în poza de mai sus, între valorile X -2 și 2, panta generată de valorile lui Y este foarte abruptă. Acest lucru înseamnă că, dacă schimbăm valoarea lui X în acea zonă, atunci Y o să sufere modificări semnificative. De asemenea faptul că valorile sunt generate în intervalul (0;1) reprezintă un avantaj **[9]**. Prin această proprietate se va face o distincție clară în predicții. Dezavantajul funcției sigmoidale, după cum se poate observa în poză, este valoarea lui Y care tinde să fie influențată tot mai puțin de către X. Acest aspect va reprezenta un dezavantaj în procesul de învățare, deoarece o să devină mult mai lent.

Funcția tangentă hiperbolică este o funcție non-lineară asemănătoare cu funcția sigmoidală și are valori cuprinse între (-1; 1).



**Figura 2-6 Graficul funcției tangentă hiperbolică[10]**

Avantajul acestei funcții este că, dacă X este negativ atunci și Y va fi negativ, iar dacă X este aproape de 0 atunci și Y va fi aproape de 0. Totodată influența lui X asupra lui Y scade într-un ritm mult mai lent în comparație cu funcția sigmoidală. Funcția sigmoidală și tangenta hiperbolică sunt cele mai des folosite funcții. **[9]**

Funcțiile de activare sunt:

1. Sigmoidală
2. Tangentă Hiperbolică
3. Rampă
4. Signum
5. Treaptă

3 REȚELE NEURONALE

## 3.1 INTRODUCERE

Rețeaua neuronală reprezintă o multitudine de neuroni conectați între ei. Un singur neuron poate fi conectat la o multitudine de neuroni. Conexiunile în neuronul biologic se fac de la axoni către dendrite și se numesc sinapse. În neuronul artificial conexiunile poartă aceeași denumire. Rețelele neuronale artificiale sunt o ramură a inteligenței artificiale. La fel ca neuronul artificial, acestea urmăresc să imite comportamentul rețelelor neuronale biologice, dar într-un mod mult mai limitat și mai simplificat. Rețelele neuronale artificiale dețin neuroni artificiali, care sunt conectați între ei prin sinapse, și sunt repartizați pe mai multe straturi. Structura neuronilor artificiali dintr-o rețea neuronală sunt clasați în funcție de natura datelor folosite, tipul funcției de activare, memoria și resursele disponibile.

Cu toate că rețeaua neuronală imită o rețea neuronală biologică există diferențe evidente și incontestabile între cele două. Cele mai mari deosebiri vor putea fi observate în următoarele caracteristici.

Obiectivul rețelelor neuronale este acela de a crea o inteligență artificială capabilă să rezolve diferite probleme într-un mod asemănător cu cel al creierului uman. În timp, obiectivul rețelei neuronale s-a axat pe rezolvarea unor probleme specifice. Unele dintre aceste probleme sunt: analizarea și înțelegerea pozelor precum și a videoclipurilor, recunoaștere vocală, recunoașterea cuvintelor rostite de către o persoană, oferirea de diagnostice medicale, și uneori probleme ce sunt considerate a fi în totalitate specifice omului precum pictură sau interpretarea artei. **[4]**

Conform dicționarului, inteligența reprezintă: "Capacitatea de a înțelege fenomene și lucruri pe baza experienței acumulate anterior". Deși rețeaua neuronală este mult mai simplă față de creierul uman, acestea funcționează pe același principiu de învățare, învățare prin experiență, iar această capacitate de învățare este cea care o face atât de specială față de alte sisteme de prelucrare a informațiilor.

Rețeaua neuronală trebuie să aibă ca date de intrare o reprezentare corectă a informațiilor, care să permită interpretarea, predicția sau un răspuns la un stimul extern. Informațiile trebuie să fie perechi de date intrare-ieșire, astfel încât, rețeaua neuronală să fie capabilă să stabilească o relație de tip cauză-efect. O rețea neuronală poate să construiască un model propriu de procesare și analizare a datelor. **[4]**

Rețelele neuronale pot fi folosite în aplicații de business precum: previziuni financiare, controlul proceselor industriale, cercetări pe piață, validări de date pe baza clasificării, management de risc, previziuni de marketing.

## 3.2 **PRINCIPALELE DIFERENȚE DINTRE REȚELELE NEURONALE BIOLOGICE ȘI ARTIFICIALE**

Deși rețelele neuronale biologice imită rețelele neuronale artificiale există unele diferențe majore între acestea, precum:

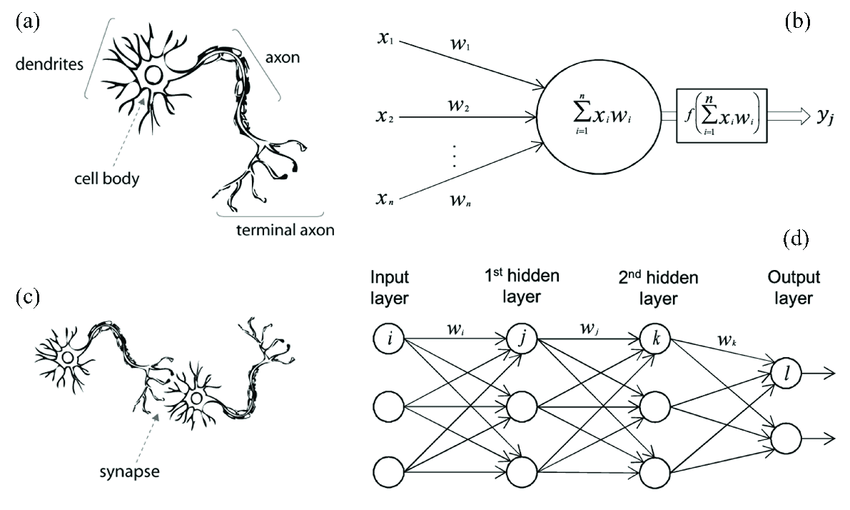
1. Dimensiunea

Creierul uman conține aproximativ 100 de milioane de neuroni, în timp ce rețeaua neuronală artificială cuprinde un număr mult mai mic, în general cuprins între 10 și 1000 neuroni. **[17]** În interiorul rețelei neuronale, de obicei, toate straturile sunt conectate cu stratul succesor acestora, ceea ce înseamnă că, legăturile sinaptice sunt în general determinate de numărul de neuroni și de straturi cu exactitate, datorită principiilor structurale la care se supun rețelele neuronale artificiale. Neuronii de pe un singur strat nu sunt conectați între ei, ci doar execută simultan aceeași sarcină. În rețelele neuronale biologice, neuronii sunt conectați într-un mod haotic.

1. Topologice

Toate straturile artificiale calculează pe rând. În schimb, în rețeaua neuronală biologică calcularea se face asincron. **[17]**  Rețelele artificiale sunt limitate deoarece neuronii de pe stratul succesor au nevoie de rezultatele neuronilor de pe stratul anterior pentru a putea calcula valoarea de ieșire.

Acest dezavantaj apare deoarece neuronii nu sunt conectați între ei dacă aceștia nu se află pe un strat vecin cu aceștia. În rețeaua biologică neuronii pot fi activi în paralel, existând porțiuni cu neuroni conectați într-o mare proporție, și porțiuni cu neuroni conectați într-o proporție mai mică.



**Figura 3-1 Reprezentarea rețelei neuronale artificiale, rețelei neuronale biologice, neuronului artificial și neuronului biologic[16]**

1. Viteza

Un neuron biologic este activat de aproximativ 200 de ori pe secundă. Semnalul traversează cu o viteză cuprinsă între 0.55 metri pe secundă, ajungând și la viteze de 120 metri / secundă. Viteza variază de la persoană la persoană în funcție de sex, vârstă, înălțime, temperatură și condiție medicală. **[6]**

Viteza neuronilor artificiali este influențată și determinată doar de puterea de calcul a sistemului. Această putere de calcul devine din ce în ce mai performantă de la an la an și este mult mai rapidă decât puterea unui creier uman. De asemenea viteza rețelei neuronale artificiale rămâne constantă, ea nu suferă de pe urma oboselii.

În concluzie, o rețea neuronală artificială este mult mai rapidă însă mult mai limitată, nu poate egala un creier uman în rezolvarea unor probleme, din punct de vedere al complexității, însă îl depășește din punct de vedere al vitezei, în cazul problemelor de o complexitate mai redusă sau bine stabilită.

1. Toleranță la erori

Datorită modului de funcționare al neuronului biologic, aceștia au o toleranță la erori. În creierul uman informația este stocată într-o cantitate imensă, într-un mod redundant, rezultând în faptul că erorile mici nu o să se definitiveze în pierderi de memorie. De asemenea creierul se poate vindeca și recupera până la o anumită limită. **[17]**

Rețelele neuronale artificiale nu au această proprietate de vindecare, și nici nu dețin o toleranță la erori. O oarecare posibilă vindecare poate fi mimată prin stocarea progresului rețelei neuronale artificiale, iar în cazul unei defecțiuni înlocuirea rețelei neuronale curente cu cea stocată. Cea mai mică eroare într-o rețea neuronală artificială poate să deterioreze rezultatul final făcându-l neutilizabil.

1. Consumul de energie

Nebăgând în seamă faptul că subiectul consumului de energie nu este atât de discutat, este de precizat că creierul uman, în comparație cu o rețea neuronală, consumă o cantitate mult mai mică de energie.

Se estimează că creierul consumă aproximativ 20 de wați pe oră, aproximativ energia necesară pentru un bec. În schimb o singură placă video poate ajunge să consume 250 wați de una singură. Totodată, nu trebuie omisă deteriorarea fizică a componentelor. Din acest punct de vedere, rețelele neuronale biologice sunt net superioare celor artificiale.

De aici putem deduce faptul că rețelele neuronale artificiale întrec capacitățile umane în oricare domeniu, dacă sunt prezente suficiente date de antrenament, iar informațiile pot fi transformate în date numerice.

În prezent, nu ne este cunoscut modul de funcționalitate al inteligenței, ceea ce înseamnă că nu cunoaștem adevărata capacitate a creierul uman, și cum de acesta este capabil să transfere și să folosească informații dintr-un domeniu în alt domeniu. Acest principiu de transfer de informații lipsește cu desăvârșire la rețelele neuronale artificiale. De exemplu, o inteligență artificială s-a dovedit capabilă să învingă un om într-un meci de șah, dar se declară nepricepută în a învinge un individ într-un meci de X și 0. Chiar dacă rețeaua neuronală artificială poate să bată anumiți oameni în anumite domenii asta nu înseamnă că acestea se comportă ca un creier uman, ci doar imită comportamentul unora.

O rețea neuronală artificială cu mai mulți neuroni nu înseamnă în mod necesar că rețeaua neuronală artificială e mai performantă ca una cu mai puțini neuroni, ceea ce poate să facă o rețea neuronală este determinat în principal de forma datelor de intrare și ieșire.

Pentru fiecare sarcină pre-stabilită a unei rețele neuronale se alege ce funcție de activare, ce tip de neuron și ce model structural trebuie să aibă rețeaua neuronală artificială. În prezent este necesară multă muncă din partea inteligenței umane pentru a crea o inteligență artificială, deoarece prin inteligența umană se stabilește cum trebuie să arate rețeaua.

3.3 STRUCTURA REȚELEI NEURONALE

Rețelele neuronale artificiale dețin un strat de intrare, unul sau mai multe straturi ascunse, și un strat de ieșire .

În general o rețea neuronală are o structură de tip total conectată, și anume fiecare neuron de pe un strat e conectat printr-o sinapsă cu toți ceilalți neuroni de pe stratul următor, excepția fiind stratul de ieșire, unde acești neuroni sunt conectați la exterior. Toți neuronii de pe un strat au aceeași funcție de integrare și funcție de activare, și același comportament binar sau real. În interiorul rețelei fiecare sinapsă are o valoare numerică numită tărie sinaptică. Aceste tării sinaptice constituie elementul de bază în procesul de învățare. **[4]**

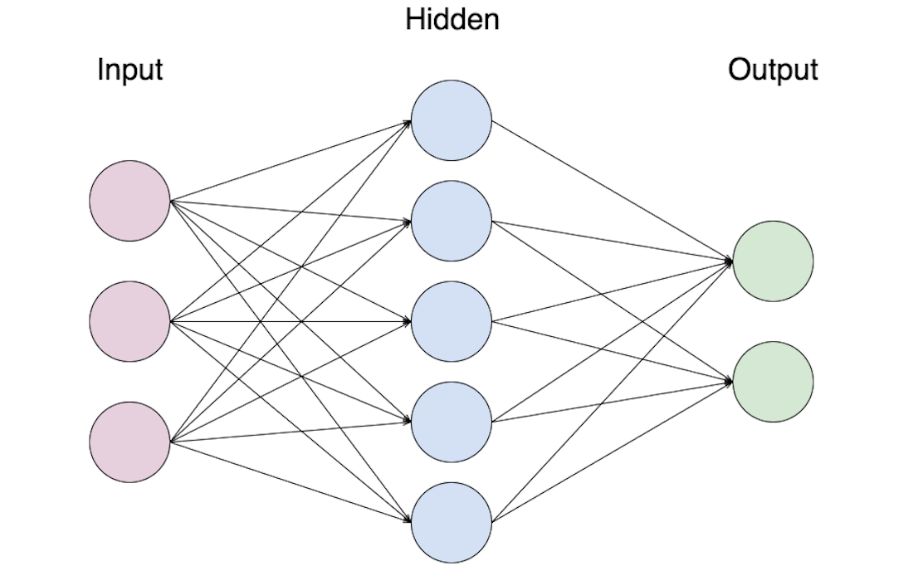
Pentru a putea comunica cu mediul exterior o rețea neuronală trebuie să aibă cel puțin 2 straturi, un strat de intrare, și unul de ieșire. Între aceste două straturi se află un număr nedeterminat de straturi ascunse.

Un strat nu deține o limită a numărului de neuroni care îi poate deține, însă în mod obligatoriu oricare strat, indiferent de tip e obligat să conțină cel puțin un singur neuron.

Stratul de intrare sau Input Layer este stratul care deține neuronii de intrare, și la fel ca neuronul de intrare acesta are rolul de a prelua informațiile de intrare și de a le transmite mai departe stratului ascuns, sau stratului de ieșire, dacă rețeaua nu deține un strat ascuns. Ca structură, funcția neuronului de intrare este f(x) = x funcția numindu-se funcție de identitate, iar valorile ei sunt reale. Datele răman nemodificate, iar tăria sinaptică cu mediul exterior, fiecare neuron având o singură sinapsă, are valoarea 1, ceea ce înseamnă că datele rămân netransformate. Totalitatea datelor propagate din stratul de ieșire vor fi stocate într-o lista de intrări, în general notată cu X. **[7]**

Stratul de ieșire sau Output Layer deține neuronii de ieșire, acest strat are rolul de a propaga rezultatul rețelei neuronale în exteriorul acesteia. Neuronii de pe stratul de ieșire vor efectua un ultim calcul cu o funcție de activare prestabilită și rezultatul va fi stocat într-o lista de ieșiri în general notată cu Y. Cele două liste, lista de intrări X și lista de ieșiri Y vor avea un număr fix de elemente. Lista X va deține un număr de elemente egal cu numărul de neuroni de intrare, iar lista Y va deține un număr de elemente egal cu numărul neuronilor de ieșire.

Stratul ascuns sau Hidden Layer este stratul pe care se află neuronii ascunși, numărul straturilor ascunse dintr-o rețea neuronală este nedeterminat. Straturile ascunse se află între stratul de intrare și stratul de ieșire.

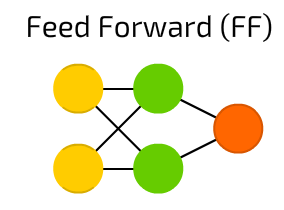
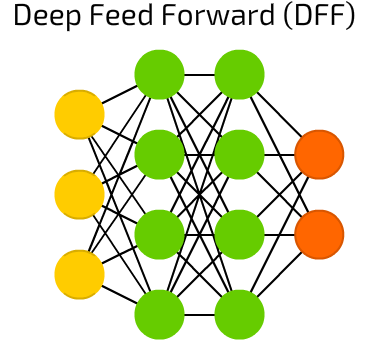


**Figura 3-2 Rețea neuronală artificială [12]**

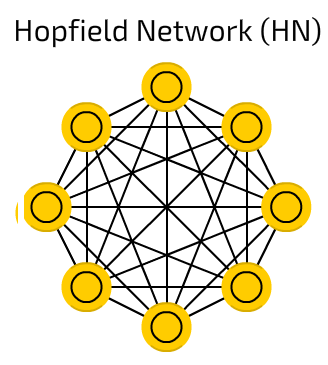
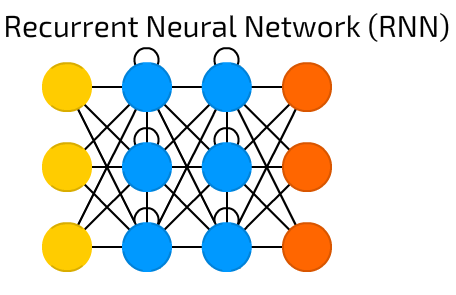
## 3.4 **TIPURI DE REȚELE NEURONALE**

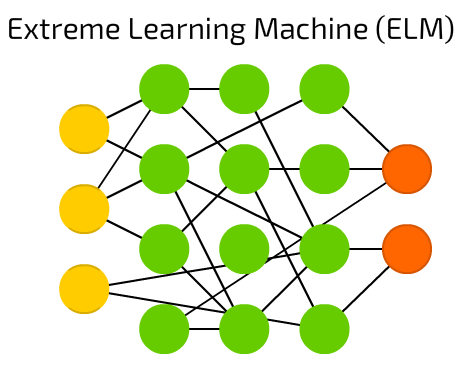
Există o multitudine de tipuri de rețele neuronale, acestea au fost implementate în funcție de operațiile matematice și setul de parametri necesari să determine outputul. Pentru fiecare tip de problemă existentă avem nevoie de o nouă rețea neuronală și, de multe ori, de implementarea unei noi structuri. În timp s-au format anumite structuri standarde pentru rezolvarea unor probleme specifice.

Unele din cele mai populare rețele neuronale sunt:

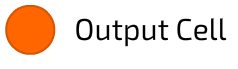
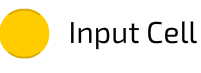
 

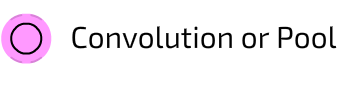
**Figura 3-3 Rețea neuronală de tip FF[15] Figura 3-4 Rețea neuronală de tip DFF[15]**

**Figura 3-4 Rețea neuronală de tip RNN[15] Figura 3-5 Rețea neuronală de tip HN[15]**

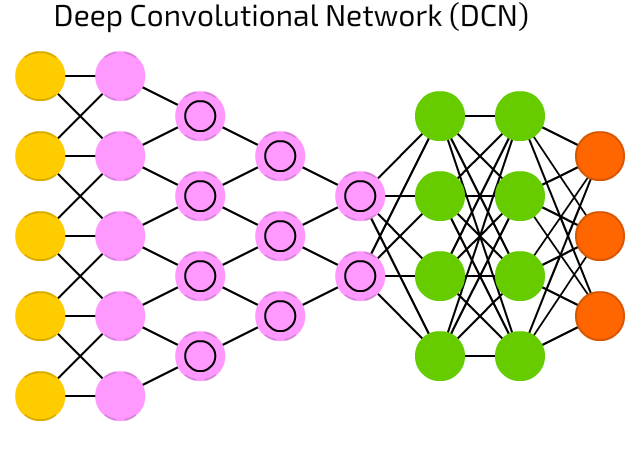


**Figura 3-6 Rețea neuronală de tip ELM [15]**



**Figura 3-7, Termeni explicativi pentru figurile 3-3,3-4,3-5,3-6,3-8 [15]**

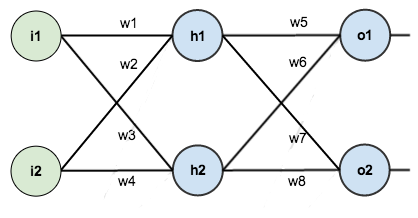


**Figura 3-8 Rețea neuronală de tip DCN [15]**

Putem observa cât de diferite pot fi structurile rețelelor neuronale și cât de complexe. Într-o rețea neuronală complexă apar mai multe tipuri de neuroni, aceștia funcționează la fel ca neuronul ascuns, doar că are componente și funcționalități suplimentare, determinate în funcție de contextul problemei pe care o rezolvă rețeaua neuronală.

Rețelele neuronale directe sunt unele dintre cele mai simple rețele neuronale. Acestea sunt caracterizate prin faptul că datele se propagă într-o singură direcție, fără bucle (recurențe), și anume doar înainte. După cum se poate observa din celelalte exemple de rețele neuronale , rețelele cu propagare directă ( feed forward) sunt cele mai ușor de implementat.

În continuare putem observa cum funcționează o rețea neuronală directă (feed forward).



**Figura 3-9 Rețea neuronală cu 2 neuroni intrare, 2 neuroni ascunși, 2 neuroni ieșire, 8 legături sinaptice [14]**

Deasupra avem o rețea neuronală cu doi neuroni de intrare notați i1 și i2, un strat ascuns cu doi neuroni ascunși h1 și h2, doi neuroni de ieșire, o1 și o2. De asemenea avem legăturile sinaptice dintre neuroni, în număr de 8, notate cu w1,w2,...,w8.

Presupunem că valorile tăriilor sinaptice sunt următoarele: w1 = 0.15, w2 = 0.17, w3 = 0.11, w4 = 0.36, w5 = 0.47, w6 = 0.55, w7 = 0.50 , w8 = 0.28.

Funcția de activare a tuturor neuronilor va fi funcția sigmoidală, și funcția de integrare va fi funcția sumă, și comportamentul va fi unul real. Dacă introducem în rețeaua neuronală prin neuronii de intrare o listă de intrare care să conțină numerele 0.35 și 0.45 calculele pentru determinarea listei de ieșire se vor efectua în următorul fel. În primul rând, datele vor fi propagate în cei doi neuroni ascunși h1 și h2 prin neuronii de intrare i1 și i2, unde i1 va propaga valoarea 0.35, iar i2 valoarea 0.45, valori aflate în lista de intrare. Neuronul h1 este conectat de neuronul de intrare i1 prin legătura sinaptică w1, tărie sinaptică având valoarea 0.15 și este conectat de neuronul de intrare i2 prin legătură sinaptică w3 având valoarea 0.11. Neuronul h1 o să calculeze care este valoarea pe care trebuie să o propage mai departe.

Prima oară folosind funcția de integrare sumă, o să calculeze global input, GPh1 = (0.35 \* 0.15) + (0.45 \* 0.17), deoarece i1 propagă valoarea 0.35 prin w1 cu valoarea 0.15, și i2 propagă valoarea 0.45 prin w2 având valoarea 0.17.

Prin urmare global input în h1 este GPh1= 0.0525 + 0.0765 = 0.6015. În continuare ieșirea neuronului h1 va fi F(GPh1), unde F este funcția sigmoidală, și anume F(0.6015) = 0.6459.

Urmând aceeași pași pentru h2, GPh2 = (0.35 \* 0.11) + (0.45 \* 0.36) = 0.0385 + 0.162 = 0.2005 și ieșirea neuronului h2 va fi F(0.2005) = 0.5499. În continuare, h1 și h2 o să propage mai departe rezultatele obținute, și, deoarece nu mai avem alte straturi de neuroni ascunși stratul succesor este stratul neuronilor de ieșire cu neuronii o1 și o2.

Urmând aceeași procedură, dar cu alte date de intrare în interiorul neuronului o1

GPo1 = (0.6015 \* 0.47) + (0.5499 \* 0.55) = 0.2827 + 0.3024 = 0.5851, și F(0.5851)=0.6422 și pentru o2 GPo2 = (0.6015 \* 0.50) + (0.5499 \* 0.28) = 0.3007 + 0.1539 = 0.4546 și F(0.4546) = 0.6117. Rezultatele la și și anume 0.6422 și 0.6117 vor reprezenta datele de ieșire a rețelei neuronale folosite în exemplu pentru datele de intrare 0.35 și 0.45.

Cu cât o rețea neuronală este mai complexă, cu atât algoritmii de obținere a rezultatului final devin mai complecși.

**3.5** PROCESUL DE ÎNVĂȚARE ÎN REȚELE NEURONALE ARTIFICIALE

La fel ca în rețeaua neuronală biologică, cunoștințele în rețeaua neuronală artificială sunt dobândite în urma procesului de învățare. Similar creierului uman, procesul de învățare are loc prin modificarea tăriilor sinaptice dintre neuroni. Acest lucru va face neuronii mai receptivi sau mai puțin receptivi la semnalele venite prin anumite tării sinaptice.

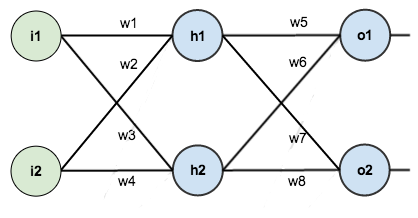
Rețelele neuronale pot folosi trei metode de învățare: învățare supervizată (supervised learning), învățare nesupervizata (unsupervised learning) și învățare întărită ( reinforcement learning).

Majoritatea rețelelor neuronale folosesc metoda învățării supervizate. Putem deduce din nume faptul că această metodă folosește un supervizor, care determină dacă rezultatele rețelei neuronale sunt sau nu corecte. Acest tip de învățare folosește o lista de intrări și o listă de ținte. Țintele reprezintă rezultatul dorit al neuronilor de ieșire dacă rețelei neuronale i se prezintă vectorul de intrare specific. **[4]**

Această rețea neuronală se comportă în felul următor. Se introduc datele în lista de intrare, intrările sunt prezentate rețelei neuronale care va returna lista de ieșire. Valorile obținute de către rețeaua neuronală stocate în lista de ieșire vor fi comparate cu o lista de ținte. În continuare, modificarea tăriilor sinaptice va avea loc calculând diferența dintre valorile din lista de ieșire cu valorile din lista de ținte. Această modificare se va realiza pe baza unui algoritm specific numit lege de învățare.

Învățarea nesupervizată se folosește în situațiile în care nu cunoaștem rezultatul dorit. În cazul învățării nesupervizate nu avem ținte și nici un supraveghetor care să spună dacă rezultatul rețelei neuronale este sau nu corect. Rețelele neuronale care folosesc învățarea nesupervizată au rolul de a găsi corelații între date folosite în general în statistici.

Învățarea întărită e derivată din învățarea supervizată. Diferența dintre învățarea supervizată și învățarea întărită e că modificarea tăriilor sinaptice nu se produce cu ajutorul diferenței dintre lista de ieșire și lista de ținte, ci o clasificare a listei de intrări.



**Figura 3-9 Rețea neuronală cu 2 neuroni de intrare, 2 neuroni ascunși, 2 neuroni ieșire, 8 legături sinaptice [14]**

În continuare putem observa cum învață o rețea neuronală cu propagare directă folosind exemplul anterior.

Rețeaua neuronală are doi neuroni de intrare, doi neuroni ascunși, și doi neuroni de ieșire, tăriile sinaptice sunt w1 = 0.15, w2 = 0.17, w3 = 0.11, w4 = 0.36, w5 = 0.47, w6 = 0.55, w7 = 0.50, w8 = 0.28. Comportamentul tuturor neuronilor este real, funcția de activare este funcția sigmoidală și funcția de integrare este sumă. Introducând valorile 0.35 și 0.45 rețeaua neuronală va returna rezultatul 0.6422 și 0.6117.

În continuare, să presupunem că metoda de învățare este învățare supervizată și legea de învățare va fi back-propagation sau propagare inversă, având o listă de ținte cu valorile 0.08 și 0.95. Ordinea calculelor va fi executată astfel:

Întâia oară se calculează Eroarea totală, adică suma erorilor neuronilor de ieșire.

**[12]**

Eroarea la neuronul de ieșire = = 0.5 \* [(0.08 - 0.6422) \* (0.08 - 0.6422)] = 0.1580

= 0.5 \* [(0.95 – 0.6117) \* (0.95 – 0.6117)] = 0.0572

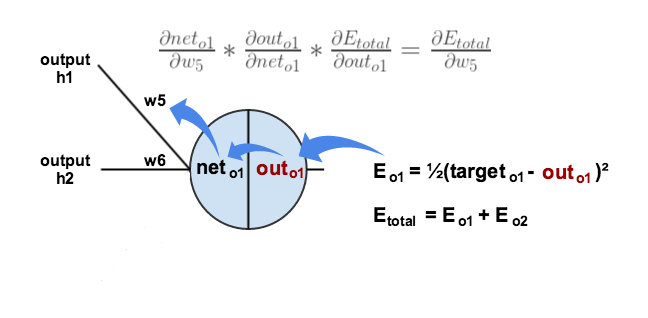
= + = 0.1580 + 0.0572 = 0.2152

Avem o eroare totală de 21.52%.

Ca să aflăm cu cât trebuie modificat w5, cât să micșorăm eroarea avem nevoie să calculăm:

=

Unde e valoarea de ieșire a neuronului și reprezintă global input al neuronului .



**Figura 3-10 Reprezentarea grafică a parcurgerii datelor pentru calculare[14]**

Este de precizat: În formulă utilizăm si deoarece legătura sinaptică w5 e conectată la neuronul .

În continuare calculăm fiecare din cele trei porțiuni împreună.



La punctul 3 este vorba de deoarece legatura sinpatică w5 leagă neuronul h1 de neuronul o1.

În continuare o să calculăm noua valoarea a legăturii sinaptice w5.

Unde rata de învățare (learning rate) este o constantă pre-stabilită în rețeaua neuronală, aceasta are rolul de a optimiza rețeaua neuronală influențând dimensiunea pasului în fiecare iterație de învățare a rețelei neuronale. O rată de învățare prea mare poate rezulta ca tăria sinaptică să sară peste valoarea ei optimă, iar o rată de învățare prea mică poate rezulta într-o rețea neuronală care învață mult prea încet.

De asemenea rata de învățare ajută rețeaua neuronală să nu rămână blocată într-o minimă locală. Ea poate fi de mai multe feluri, dar pentru exemplul curent aceasta este o constantă egală cu 0.20. Deci: **[14]**

Urmând aceeași pași calculăm noua valoarea a legăturii sinaptice pentru w6, w7 și w8.

Pentru w6 :

Pentru w7 :

Pentru w8:

Am aflat noile valori ale tăriilor sinaptice care leagă stratul ascuns de stratul de ieșire. În continuare avem nevoie să aflăm tăriile sinaptice w1, w2, w3, w4, care leagă neuronii de intrare de neuronii de pe stratul ascuns

Pentru w1:

Formulele de bază sunt aceleași:

=

Unde:

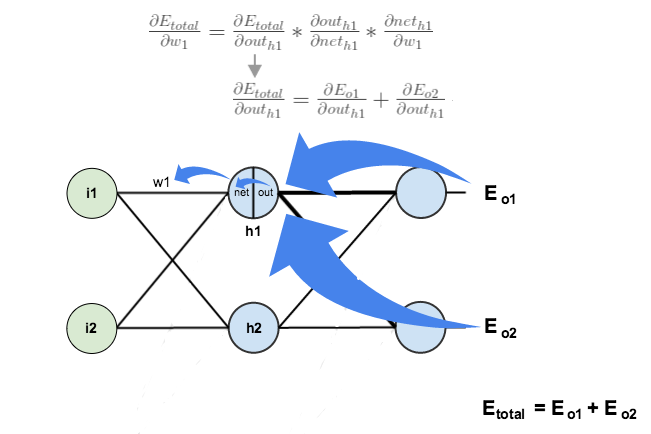
unde n reprezintă numărul neuronilor de ieșire, în cazul nostru n = 2.

\* ( 1 – outo2)

= 0.47 = 0.55

, reprezintă vechea valoare a tăriei sinaptice dintre neuronul 01 cu

neuronul h1 si 02 cu h1, respectiv w5 si w6, care au fost schimbate anterior.



**Figura 3-11 Reprezentarea grafică a parcurgerii datelor pentru calculare[14]**

= [- (targeto1 – outo1) ] \* [ outo1 – ( 1 – outo1) ]

= [- (0.08– 0.6422) ] \* [ 0.6422 – ( 1 – 0. 0.6422) ] = 0.5622 \* 0.2297 = 0.1297

= [- (targeto2 – outo2) ] \* [ outo2 – ( 1 – outo2) ]

= [- (0.95– 0.6117) ] \* [ 0.6117 – ( 1 – 0.6117) ] = -0.3383 \* 0.2375 = -0.08

Aceste date vor fi refolosite. În continuare:

= 0.47

= 0.1297 + 0.47 = 0.5997

outh1 \* (1 -outh1) = 0.5851 \* (1 – 0.5851) = 0.2427

= outi1 = 0.35

Din asta, rezultă că noua valoare w1 = 0.15 – 0.20 \* 0.09 = 0.132

Pentru w2:

=

= 0.50

= 0.1297 + 0.50 = 0.6297 0.2

= 0.5499 \* ( 1 – 0.5499) = 0.2475

= 0.8297 \* 0.2475 \* 0.35 = 0.071

W2 = 0.17 – 0.20 \* 0.071 = 0.156

Pentru w3:

=

= 0.1297 + 0.47 = 0.599

*outh1 \* (1 -outh1) = 0.5851 \* (1 – 0.5851) = 0.2427*

45

= \* 0.2427 \* 0.45 = 0.1168

W3 = 0.11 – 0.20 \* 0.1168 = 0.087

Pentru w4:

=

= 0.50

= 0.1297 + 0.50 = 0.6297 0.2

= 0.5499 \* ( 1 – 0.5499) = 0.2475

= 0.8297 \* 0.2475 \* 0.45 = 0.092

W4 = 0.36 – 0.20 \* 0.092 = 0.342

Acestea sunt noile valori ale tăriilor sinaptice după ce rețeaua neuronală învață. Rețeaua neuronală va învăța până când eroarea va fi sub un anumit prag, ea devenind mai mică la fiecare generație.

O generație reprezintă parcurgerea tuturor datelor oferite pentru învățare. O rețea neuronală care funcționează corespunzător va genera o eroare tot mai mică.

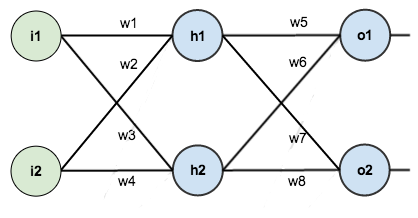
În continuare o să verificăm noua eroare a rețelei neuronale.

Noile tării sinaptice ale rețelei neuronale sunt:

w1 = 0.132, w2 = 0.156, w3 = 0.087,

w4 = 0.342, w5 = 0.455, w6 = 0.5594, w7 =0.4858, w8 = 0.2884.

Datele de intrare sunt 0.35, 0.45, țintă 0.08,0.95.



**Figura 3-9 Rețea neuronală cu 2 neuroni intrare, 2 neuroni ascunși, 2 neuroni ieșire, 8 legături sinaptice[14]**

Neth1 = 0.35 \* 0.132 + 0.087 \* 0.45 = 0.08535 , Outh1 = 0.52132

Neth2 = 0.35 \* 0.156 + 0.45\*0.342 = 0.67, Outh2 = 0.66817

Neto1 =0.08535 \* 0.455 + 0.66817 \* 0.4858 = 0.36339, Outo1 = 0.58986

Neto2 = 0.52132 \* 0.5534 + 0.66817 \* 0.2284 = 0.4442 Outo2 = 0.60926

Dacă calculăm eroarea totală obținem:

Eo1 = 0.5 \*[(0.08 – 0.058986) \* (0.08 – 0.058986)] = 0.1299

Eo2 = 0.5 \*[(0.95 – 0.60926) \* (0.95 – 0.60926)] = 0.0580

Etotal = 0.1299 + 0.0580 = 0.1879

Adică o eroare de 18.79%.

Dacă comparăm cele două erori, 21.52% și 18.79% , putem observa o diferență de 2.77%, ceea ce reprezintă o îmbunătățire majoră a rețelei neuronale după o singură generație. Cu cât eroarea totală e mai aproape de 0 cu atât eroarea o să sufere modificări mai mici.

4 RECUNOAȘTEREA VORBITORULUI

4.1 INTRODUCERE

Recunoașterea vorbitorului reprezintă identificarea persoanei care vorbește în funcție de caracteristicile vocii. Recunoașterea vorbitorului este diferită față de recunoașterea vorbirii.

Recunoașterea vorbirii se ocupă cu identificarea cuvintelor spuse de către vorbitor, și recunoașterea vorbitorului se ocupă cu identificarea persoanei care vorbește. Alte derivate ale acestui domeniu pot fi: identificarea limbi, identificarea sexului, identificarea emoției sau identificarea accentului. O aplicație de recunoaștere a vorbitorului cu ajutorul unei rețele neuronale funcționează în felul următor: Utilizatorul se înregistrează, din înregistrarea audio sunt preluate anumite trăsături specifice ale vocii și sunt stocate, aceste trăsături sunt date rețelei neuronale pentru procesorul de învățare și testare. După ce rețeaua neuronală termină procesul de învățare, aplicația este pregătită să identifice utilizatorul în viitor când îi va asculta vocea.

Recunoașterea vorbitorului poate fi dependentă sau independentă de text. O aplicație de recunoaștere a vorbitorului dependent de text poate identifica o persoană doar dacă rostește același text pe care l-a rostit în primă fază de înregistrare, iar o aplicație independentă de text este o aplicație capabilă să identifice utilizatorul indiferent de ceea ce spune.

4.2 TRĂSĂTURILE VORBITORULUI

Oamenii folosesc mulți indicatori perceptuali în recunoașterea vorbitorului. Acești indicatori pot varia de la vorbitor la vorbitor în funcție de caracteristicile lingvistice sau semantice ale vorbitorului până la caracteristicile acustice.

În prezent, nu cunoaștem în profunzime fiecare indicator, dar, totuși, de o bună bucată de vreme, creeăm și perfecționăm sisteme care imită modul în care oamenii se recunosc între ei pe baza vocii.

Indicatorii proveniți din caracteristicile semantice și lingvistice sunt indicatori dobândiți pe parcusul vieții, determinați de locație și educație, în timp ce indicatorii scazuți (precum caracteristicile acustice) sunt determinate de trăsăturile fizice ale persoanei. Până astăzi, s-a stabilit că cei mai buni indicatori pentru recunoașterea vorbitorului sunt indicatorii de nivel scăzut, aceștia putând fi măsurați cu diferite măsurători acustice ușurând extragerea semnalului.

Prin procesul de producere a vorbiri putem determina din semnalul vocal măsuratorile acustice. Sunetul se produce prin rezonanța aerului în tractul vocal și în cavitățile nazale, aerul trecând prin glotă. Glota reprezintă o regiune a laringelui, cuprinsă între vestibulul laringean și regiunea subclotica, ea find mărginită pe fiecare parte de către o coardă vocală.

Glota este cea care produce o excitare a tractului vocal în timpul pronunțării sunetelor sonore, adică vocale. Tractul vocal și glota sunt cele care determină o proporție mare din caracteristicile semnalului vocal.

## 4.3 **INTEGRAREA TRĂSĂTURILOR ÎN APLICAȚIE**

Primul pas pentru extragerea trăsăturilor este de a înregistra vocea persoanei. Prin înregistrare, obținem un fișier de tip .wav unde datele sunt stocate într-un număr foarte mare. Numărul de date este influențat de frecvența cu care înregistram. Pentru fiecare 1 Khz în frecvență fișierul o să stocheze 1000 de valori pe secundă.

Un frame reprezintă o bucată din fișierul original, secționat într-o anumită dimensiune. Între 5 și 100 ms e perioada în care majoritatea proprietăților semnalelor vocale sunt staționate[1]. Dimensiunea standard a unui frame pentru recunoașterea vorbitului este de 10-30 ms. Cu cât segmentarea se face pe dimensiuni mai mici cu atât o sa fie mai greu de procesat datele. Există mai multe metode de extragere a trăsăturilor, una dintre cele mai populare este metoda MFCC, Mel Frequency Cepstrum Coefficients. Coeficienții de tip MFC pot fi comparați într-un mod eficient cu vocea umană deoarece frecvențele sunt scalate **[1]** logaritmic. Caracteristicile fonetice vor fi exprimate în scală Mel (Mel Frequency Scale). Formula de conversie din hz în mel este mel(f) = 1127 \* ln (1+ f/700).

Pentru a putea obține MFCC executăm următorii pași: **[3]**

1) Secționăm semnalul în frame-uri;

2) Obținem transformarea Fourier;

3) Calculăm coeficienții spectrali cu aptitudinile folosite la punctul 2;

4) Se calculează log din coeficienții obținuți la punctul 3;

5) Se aplică transformarea cosinus discret pe elementele obținute la punctul 4;

6) Se păstrează din elementele obținute coeficienții de la 1 la 12.

Folosind această metoda o să obținem mai multe liste cu 12 elemente, acestea vor fi caracteristicile vocii persoanei înregistrate și datele de intrare în rețeaua neuronală.

# 5 IMPLEMENTARE

## 5.1 INTRODUCERE

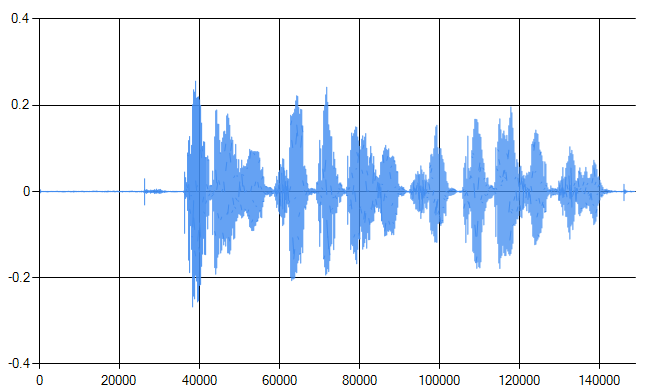
Primul pas este cel de a înregistra utilizatorul în aplicație. Prin intermediul unei interfațe grafice separate de cea principală, vocea utilizatorului va fi înregistrată

într-o dimensiune de 16 hz, adică 32000 bytes pe secundă. Datele înregistrate vor fi stocate într-un buffer, apoi vor fi salvate într-un fișier .wav în interiorul directorului Waves. Din această înregistrare se vor obține trăsăturile persoanei, MFCC.

Înregistrarea și procesarea datelor audio se va face cu ajutorul mai multor obiecte de tip WaveIn, WaveReader, WaveWriter, aflate în librăria Naudio.Waves.

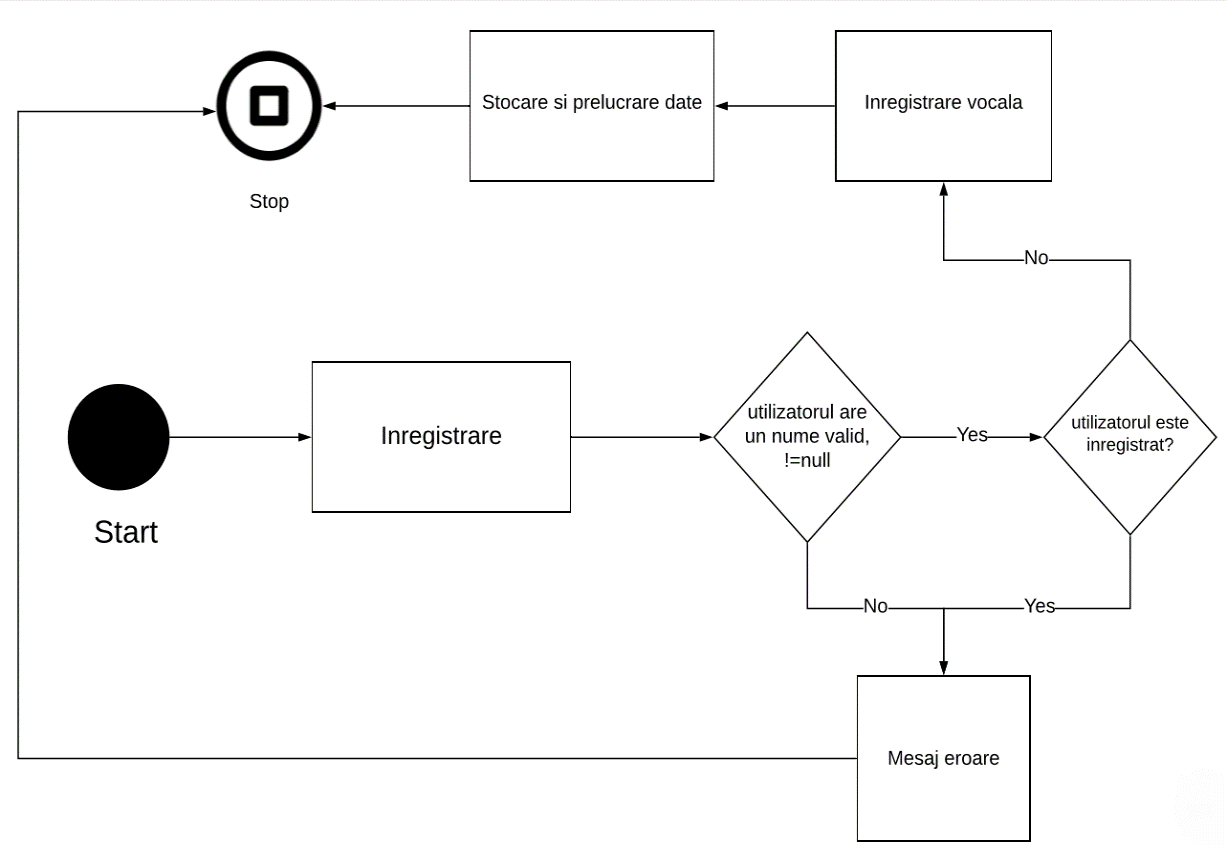
Cand interfața secundară pentru înregistrare este deschisă, se vor crea obiectele WaveIn și WaveFileWriter. La apăsarea butonului de înregistrare va fi creat un string path specific utilizatorului și obiectele WaveIn și WaveFileWriter vor fi inițializate, de asemenea WaveIn va începe să înregistreze în buffer. La o a doua apăsare a butonului se vor transfera valorile din buffer si fisierul.wav cu numele utilizatorului va fi creat.

După a doua apăsare a butonului interfața grafică secundară se va închide.



**Figura 5-1 Reprezentarea graficului unui fișier audio;**

În interiorul aplicației la selectarea unui utilizator se poate observa graficul fișierului audio. Acesta este creat prin convertirea biților din fișier în date de tip float (32 biți), care vor fi stocați într-o listă și apoi vor fi reprezentați grafic cu ajutorul unui obiect de tip System.Windows.Form DataVisualization.Charting.Chart. Aici, trebuie menționat că dimensiunea buffer-ului în care sunt citite elementele din fișierul.wave, trebuie să reprezinte o valoare a lui 2, iar în conversia de tip float, 32 biți, parcurgem buffer-ul din 4 in 4 bytes.



**Figura 5-2 Diagrama procesului de înregistrare;**

## 5.2 **EXTRAGEREA TRĂSĂTURILOR**

După finalizarea procesului de înregistrare a vocii, aplicația cu ajutorul librăriei SpeechProcesingNative o să extragă trăsăturile persoanei din aplicație și o să le salveze într-un fișier XML. Pentru început în interiorul funcției AddUser din MainForm, se va crea un string path care va indica fișierul .wav cu vocea userului, apoi se va crea o listă de liste care o să conțină caracteristicile. Pentru a putea începe extragerea caracteristicilor în interiorul clasei MFCC, avem nevoie să creăm un obiect de tip WaveFormat care să stocheze specificațiile fișierului .wave și, de asemenea, avem nevoie să știm dimensiunea frame-ului. În continuare cu ajutorul unui WaveFileReader o să citim fișierul .wave într-un buffer.

Datele din buffer vor fi convertite din bytes în integer16 cu ajutorul unei funcții denumite Int16Convertor. Această funcție conține un loop care merge din 2 în 2 până la finalul listei și va converti fiecare secvență de 16 biți într-un singur element integer.

În continuare, datele noi obținute vor fi trecute printr-un filtru de sunet cu rolul de a reduce zgomotele, cu ajutorul unei funcții DSP (Digital signal processing) și vor fi stocate într-o nouă listă unde se va face o medie a tuturor frameurilor. Dacă media valorilor din frame este mai mică de 120 asta înseamnă ca acel frame e zgomot de fundal, prin urmare nu se va lua în considerare.

Următorul pas este acela de extragere a celor 12 coeficienți MFCC pentru fiecare frame luat în considerare. În ultimul rând toți coeficienți MFCC pentru fiecare frame vor fi exportați pentru a fi salvați într-un fișier .XML cu numele utilizatorului. Exportul se va realiza cu un obiect XmlSerializer aflat în System.IO.

Exemplu coeficienți MFCC

<ArrayOfDouble>

<double>0.50157334244201124</double>

<double>-3.66866880457502</double>

<double>-2.2201321421686062</double>

<double>0.33029748241779988</double>

<double>-1.7716759761045682</double>

<double>0.6898482152938159</double>

<double>-0.26852542935595808</double>

<double>1.637531477285773</double>

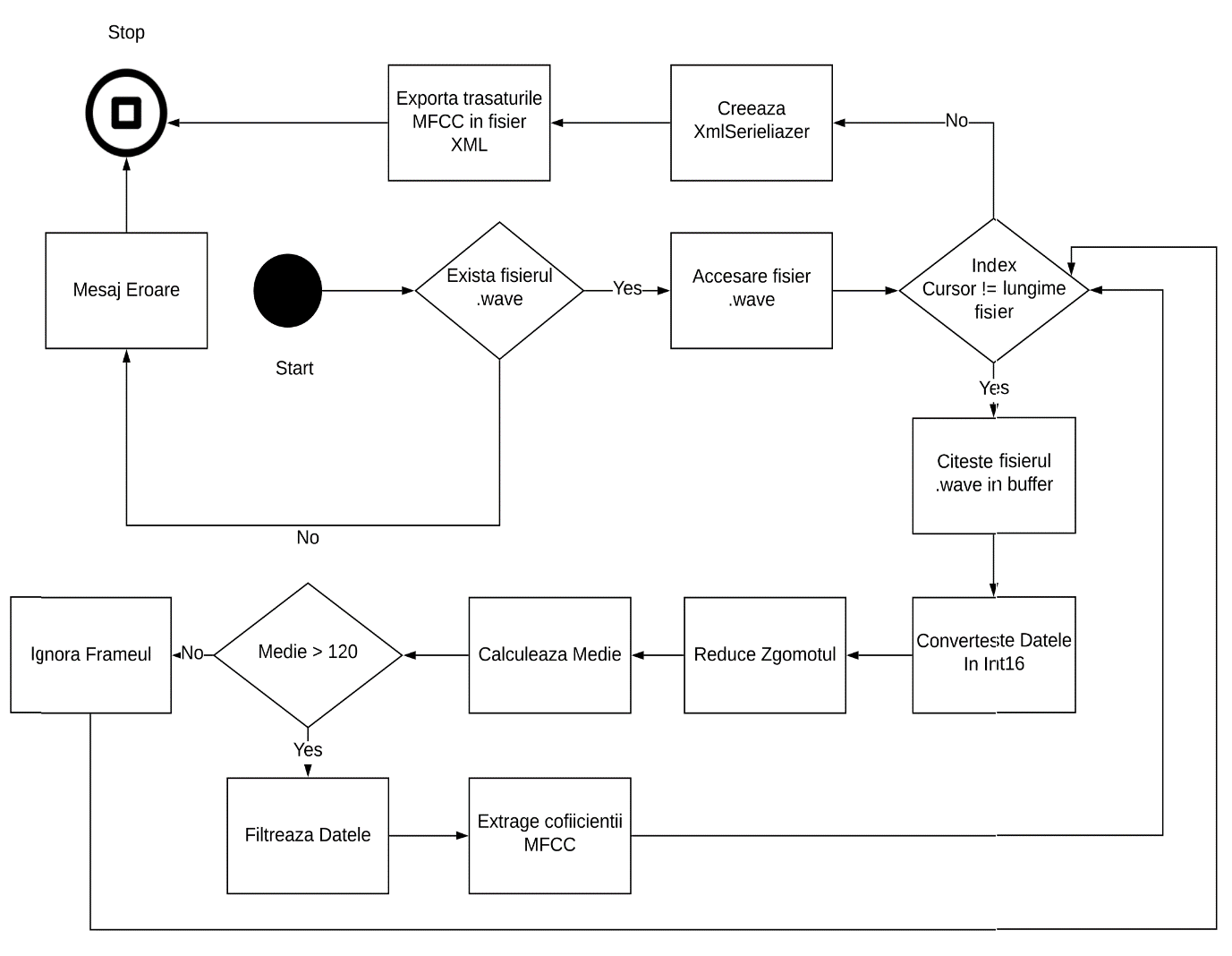
<double>0.088461937878629</double>

<double>2.6606876726899729</double>

<double>0.90202393856667529</double>

<double>-0.67238442637968543</double>

</ArrayOfDouble>



**Figura 5 -3 Diagramă de activitate a procesului de extragere a trăsăturilor;**

## 5.3 **ARHITECTURA REȚELEI NEURONALE**

Rețeaua neuronală conține 12 neuroni de intrare, unul pentru fiecare coeficient MFCC, 12 neuroni ascunși și un singur neuron de ieșire. Neuronii de intrare vor fi reprezentați printr-o listă numită Inputs unde rețeaua neuronală va primi ca argument coeficienții MFCC.

Construirea neuronilor și a tăriilor sinaptice se va produce în interiorul constructorului clasei RețeaNeuronală.

Neuronul va fi reprezentat printr-un obiect independent de RețeauaNeuronală. Neuronul deține 4 proprietăți: output-ul neuronului, o listă de input, o listă de tării sinaptice și o listă din vechile tării sinaptice. În constructor, neuronul o să primească ca argument câți neuroni are stratul predecesor, în cazul în care neuronul se află pe primul strat ascuns, atunci acest număr este 12, deoarece avem 12 neuroni de intrare reprezentați printr-o listă cu 12 valori.

Avem nevoie să cunoaștem câți neuroni are stratul predecesor pentru a putea cunoaște câte tării sinaptice avem. Tăriile sinaptice vor fi inițializate cu o valoare la nimereală aleasă din intervalul 0 și 1 cu ajutorul funcției statice NextRandom din interiorul clasei random folosindu-ne de un obiect de tip System.Random. Aceste tării sinaptice vor fi adăugate în lista Weights.

Pentru a înțelege mai bine funcționalitatea Neuronului și cum se creează legătura cu ceilalți neuroni, să presupun că avem 3 neuroni de intrare, un singur strat ascuns având un singur neuron, și un neuron de ieșire. Neuronul de pe stratul ascuns va avea 3 elemente în lista de Weigts, deoarece avem 3 neuroni de intra-reprezentanți prin 3 elemente în lista de Inputs. Neuronul de pe stratul ascuns va avea tăriile sinaptice w1, w2 și w3 și va primi ca inputs x1, x2, x3.Valoarea semnalului x1 va ajunge în Neuronul de pe stratul ascuns prin tăria sinaptica w1, semnalul x2 prin w2 și semnalul x3 prin w3.

Neuronul de ieșire va primi ca argument numărul neuronilor predecesori, adică un singur neuron pe stratul ascuns, prin urmare, acesta va avea în listă o singură tarie sinaptică w4.

În concluzie, conform acestui algoritm de implementare, neuronul va reține doar tăriile sinaptice predecesoare, și nu trebuie să se îngrijoreze de cele succesoare, deoarece acestea vor fi responsabilitatea stratului succesor. De asemenea, neuronul va conține funcția de activare împreună cu funcția de integrare, o metodă de returnare a output-ului, o metodă supra-scrisă de returnare a tăriilor sinaptice sau a unei singure tării, o metodă de returnare a unei singure tării sinaptice vechi și o metodă de modificare a tăriilor sinaptice actuale.

Metoda de modificare a tăriilor sinaptice va primi ca argument o nouă listă Weights, și va fi egalată cu o listă nouă, dar nu înainte ca lista cu tăriile sinaptice vechi să fie egalată cu cea curentă.

Avem nevoie de vechile tării sinaptice pentru procesorul de învățare, deoarece pentru a putea calcula ∂Etotal/∂outhx, avem nevoie de ∂netoy/∂outhx = cu vechea valoare a tăriei sinaptice care leagă neuronul de ieșire oy de neuronul ascuns hx, Wxy.

Constructorul rețelei neuronale deține 5 argumente: numărul neuronilor de intrare, numărul straturilor ascunse, numărul neuronilor pe strat ascuns, numărul neuronilor de ieșire și learning rate, care e declarată 0.25 din oficiu. În constructor vor fi creați toți neuronii și vor fi puși în listele potrivite: intrare, ascunși, ieșire. Rețeaua neuronală, la fel ca neuronul, mai deține o listă de output-uri vechi ale neuronului de ieșire, pentru a putea calcula ∂Eox/∂netoy=[-( targetox-outox)]\*[ outox \* ( 1 – outox)], deoarece tăriile sinaptice ale stratului ascuns cu stratul de ieșire sunt primele care se modifică.

În funcția de ForWordpropagation, care primește ca argument lista de Inputs și lista de ținte, se va efectua calculul pentru a obține output-ul neuronilor de ieșire. Funcția va începe să trimită imput-urile neuronilor de pe primul strat ascuns, urmând ca neuronul să facă legătura între datele primate și tăria sinaptică prin care acestea se deplasează.

În continuare, funcția apelează metoda de calcul a output-ului pentru fiecare neuron de pe stratul ascuns, și o stochează într-o nouă listă de input-uri, care va fi trimisă mai departe următorului strat ascuns. În cazul nostru rețeaua nu va avea mai mult de un singur strat ascuns, prin urmare datele vor fi trimise mai departe neuronului de ieșire.

După ce neuronul de ieșire efectuează calculele și obținem output-ul acestuia, rămâne stocat în neuron, lista de output-uri vechi este curățată și actualul rezultat al neuronului de ieșire este introdus în listă.

Ca ultim pas în interiorul funcției se calculează eroarea totală pintr-o funcție denumită Eroare care primește ca argument lista de target-uri și se folosește de output-ul neuronului de ieșire pentru a calcula eroarea conform formulei:

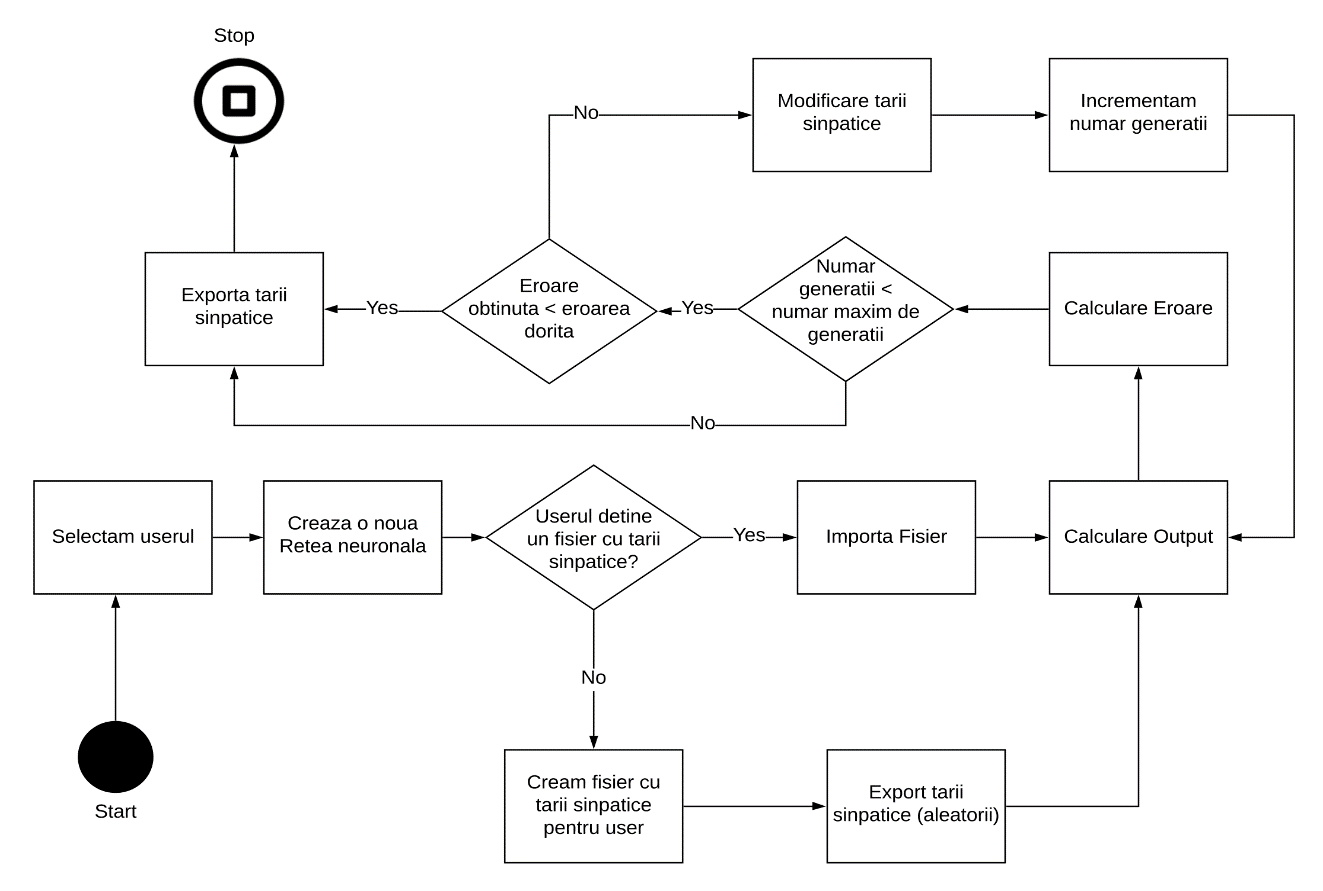
;

Funcția de învățare BackPropagation o să primească ca argument lista de target-uri și, astfel, începe prin a parcurge fiecare neuron de ieșire, în cazul nostru doar unul, și va calcula ∂Etotal/∂outhx urmând ca apoi să parcurgă fiecare neuron de pe stratul ascuns și să calculeze ∂Etotal/∂w1, modificând corespunzător tăriile sinaptice deținute de neuronul de ieșire și stocandu-le într-o listă auxiliară, următorul pas fiind să transmitem noile tării sinaptice neuronului de ieșire folosindu-ne de metoda din interiorul Neuronului.

Pasul următor este acela de a modifica tăriile sinaptice deținute de către neuronii ascunși, parcurgând fiecare neuron de ieșire, pentru a extrage datele deținute de neuron, cu scopul efectuării calculelor, apoi, parcurgând fiecare neuron ascuns și fiecare tărie sinaptică deținută de către acesta pentru a putea fi modificată.

Rețeaua neuronală mai deține două metode de Import și Export a tăriilor sinaptice pentru stocarea și refolosirea tăriilor după terminarea procesului de învățare. La Export toate tăriile sinaptice vor fi stocate în ordinea neuronilor într-un fișier XML. La import, metoda deține un contor al tăriilor sinaptice, în primul rand extragem din fișierul XML valorile într-o listă, în al doilea rand parcurgem fiecare neuron de pe stratul ascuns, și toți neuroni vor primi tăriile sinaptice în funcție de numărul de neuroni de pe stratul predecesor, spre exemplu, în cazul primului strat, fiecare neuron va primi elementele din listă din 12 în 12, deoarece avem 12 neuroni de intrare.

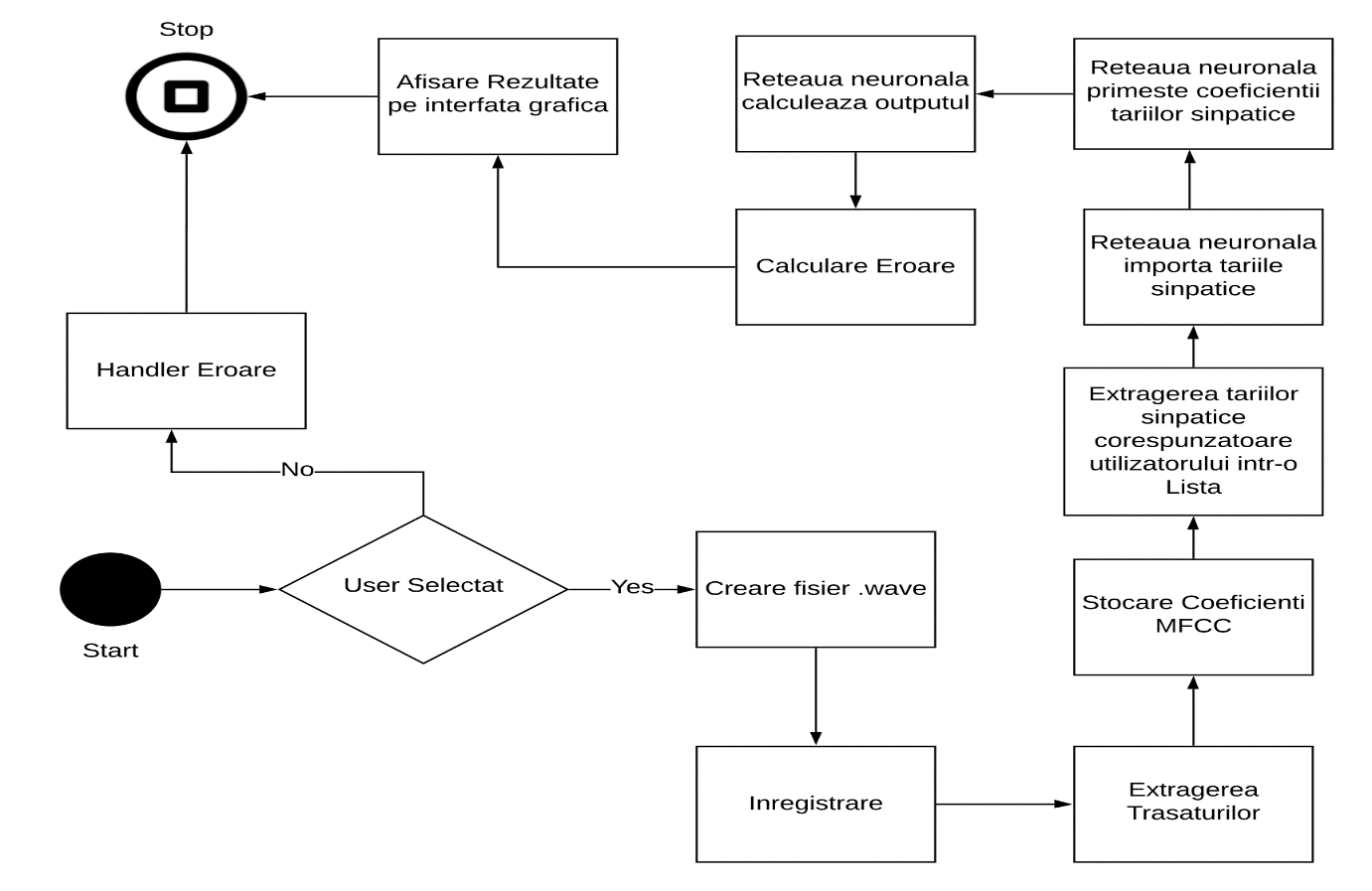
În timpul procesului de învățare, rețeaua neuronală va folosi coeficienții tuturor utilizatorilor, pentru coeficienții utilizatorului curent targetul va fi 1 și pentru ceilalți coeficienți targetul va fi 0.



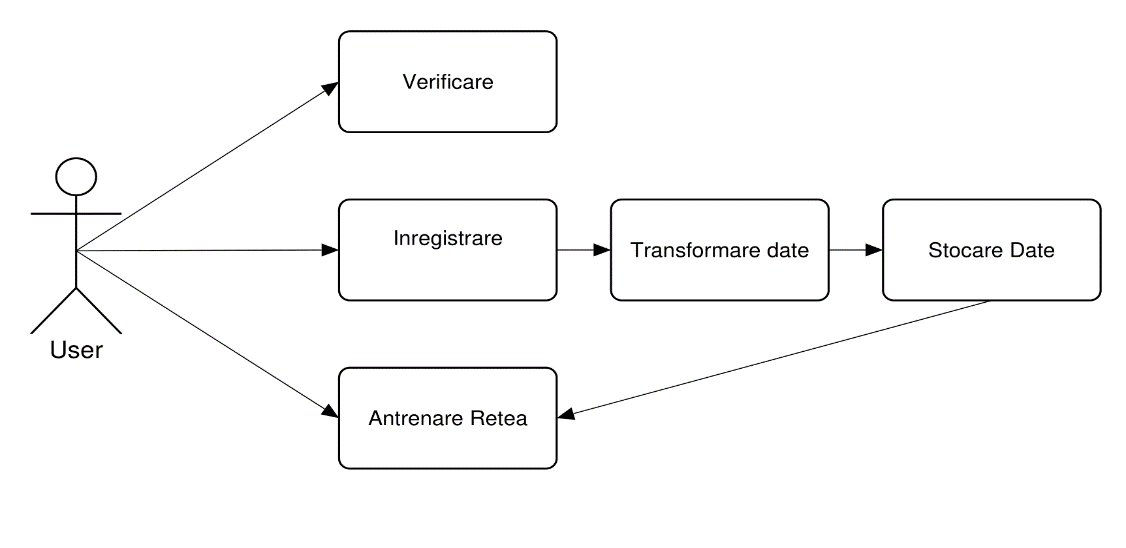
**Figura 5-4 Diagramă de activitate a procesului de invațare**

## 5.4 **PROCESUL DE VERIFICARE**

În procesul de verificare utilizatorul selectează un xpl din userii înregistrați, se creează un nou fișier .wave, utilizatorul se înregistrează audio apelând la o metodă de înregistrare explicată la primul subcapitol, din acest fișier sunt preluate trăsăturile explicate la subcapitolul 2 și se creează o nouă rețea neuronală care o să importe tăriile sinaptice ale utilizatorului și o să primească ca input caracteristicile persoanei, adică coeficienții MFCC. Rezultatul final va fi outputul neuronului de ieșire exprimat în procente. După finalizarea calculelor fișierele noi create vor fi distruse.



**Figura 5-5 Diagrama de activitate a procesului de verificare**



**Figura 5-6 Diagrama cazurilor de utilizare**

Utilizatorul poate interacționa cu aplicația în trei moduri. Acesta poate să antreneze rețeaua neuronală pentru fiecare utilizator în parte, acesta poate efectua verificări asupra rezultatelor obținute de către rețeaua neuronala pentru orice utilizator și acesta poate să se înregistreze în aplicație astfel încât aplicația să-i poată procesa datele

## 5.5 **CLASELE**

Unde:

1) MainForm este clasa principală care face conexiunea dintre toate celelalte elemente.

2) AddUserForm este clasa care se ocupă de adăugarea utilizatorilor noi.

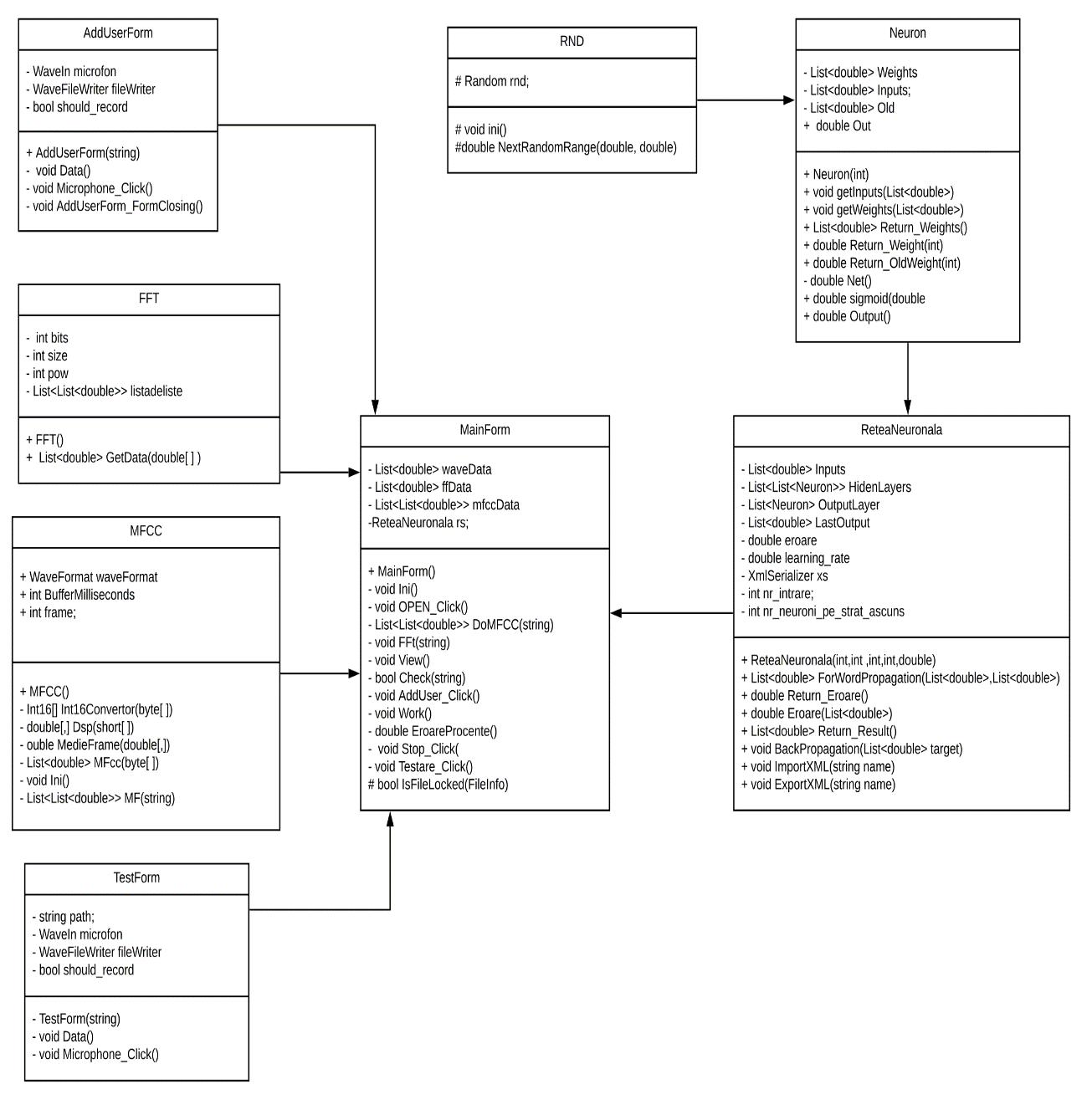
3) FFT este clasa care transformă datele din fisierul .wave în FFT ca apoi să fie prezentate graficului din MainForm.

4) MFCC este clasa care se ocupă de preluarea trăsăturilor, a coeficienților MFCC.

5) RND este o clasă statică cu folosință în generarea numerelor aleatorii

6) Neuron este clasa care implementează neuronul

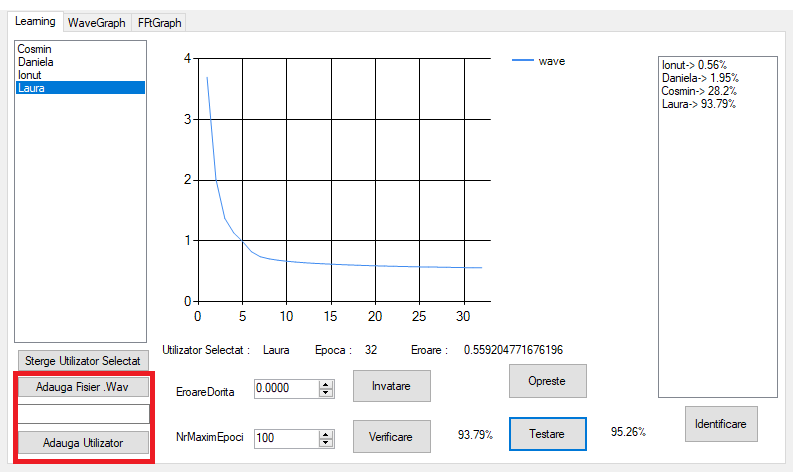
7) RețeaNeuronală este clasa care construiește și se ocupă de funcționalitatea rețelei neuronale.



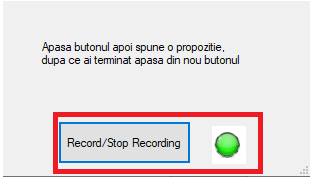
# 

# 6. MANUAL DE INSTRUCȚIUNI

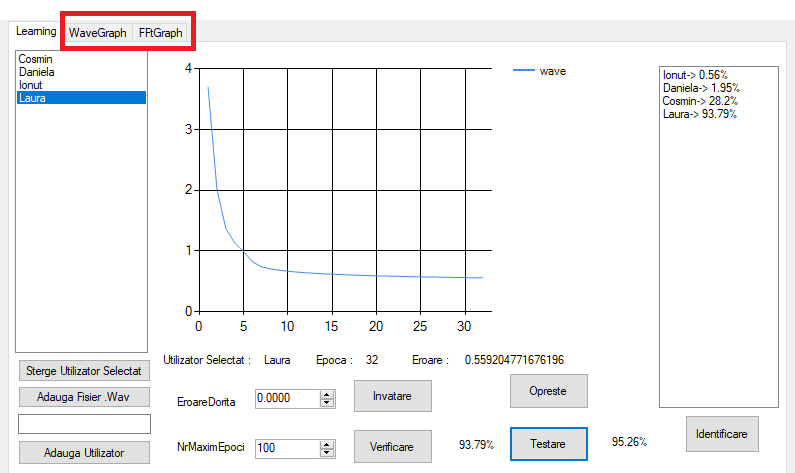
Pentru un utilizator nou, primul pas este de a se înregistra. Pentru a se înregistra, trebuie să-și scrie numele în căsuța de jos și să apese pe butonul înregistrare. Numele trebuie să fie diferit față de ceilalți utilizatori înregistrați. Inregistrarea se mai poate face si prin adaugarea unui fisier have apasand butonul.

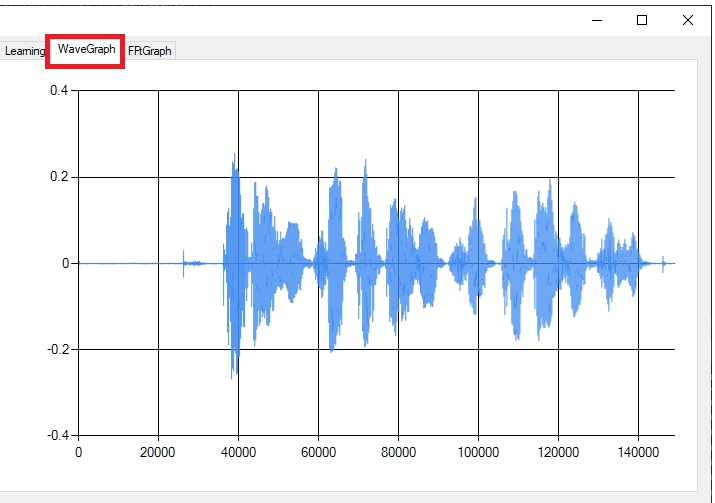


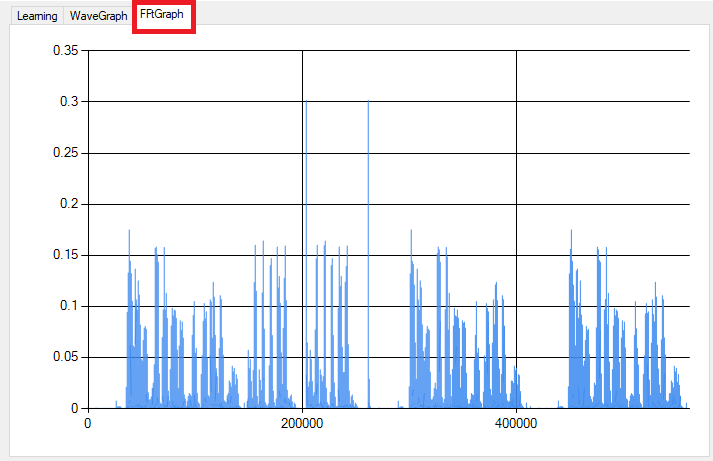
După apăsarea butonului de înregistrare, utilizatorul va apăsa pe butonul de Record, va spune o propoziție sau mai multe, indicat să se vorbească 30 de secunde, apoi va apăsa din nou butonul și va închide căsuța.



În continuare utilizatorul poate să vadă cum arată graficul fișierului .wave și graficul FFT al acestuia în celelalte 2 tab-uri.



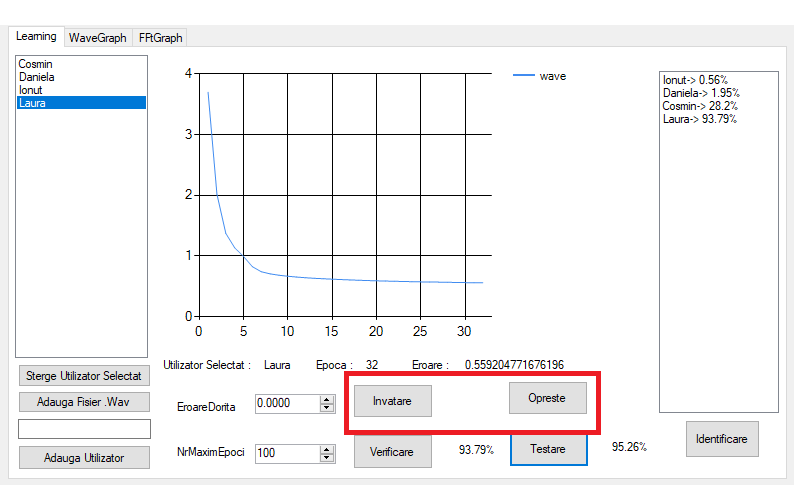




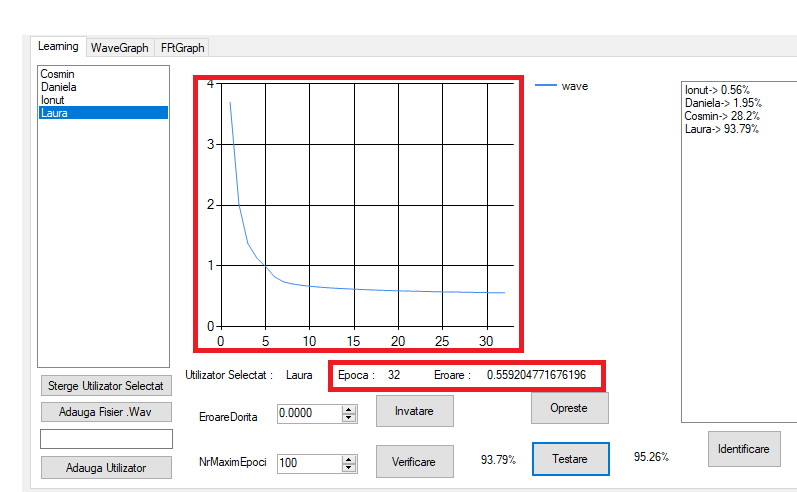
Utilizatorul trebuie să aleagă numarul de generații, sau eroarea dorită pană la oprirea procesului de învățare după care să apese pe butonul de învățare. Utilizatorul poate opri oricând procesul de învățare prin apăsarea butonului Oprire.



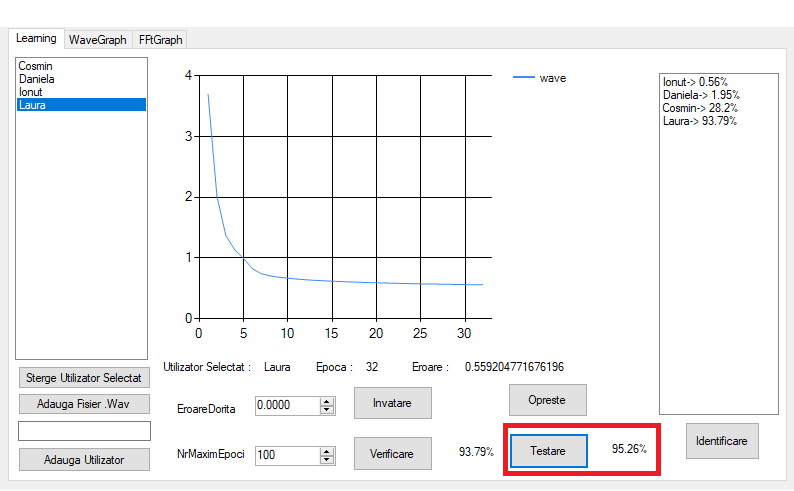
După ce selectează numărul de generații maxim și eroarea dorită, poate începe procesul de învățare prin apăsarea butonului Învățare, de asemenea procesul poate fi oprit oricând prin apăsarea butonului Oprire.



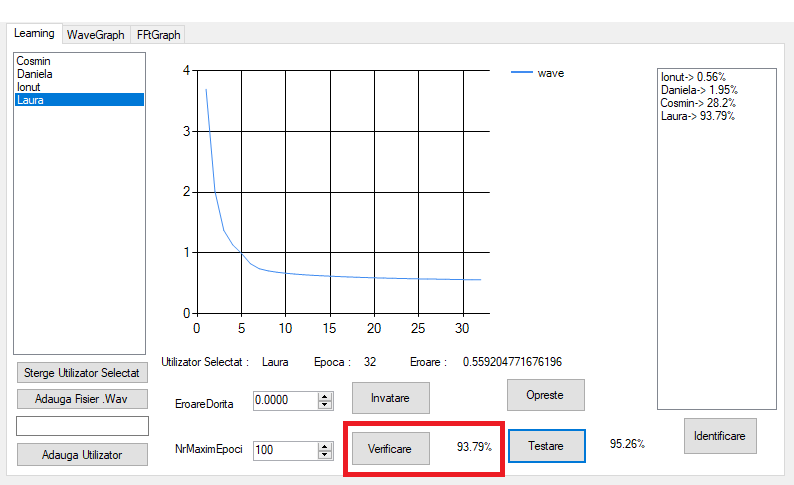
Valoarea erorii poate fi vizualizată din grafic.

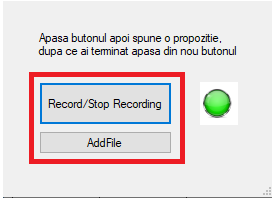


La apasarea butonului Testare utilizatorul poate sa testeze eficienta programului, procentul din partea dreapta a butonului reprezinta eficienta aplicatiei

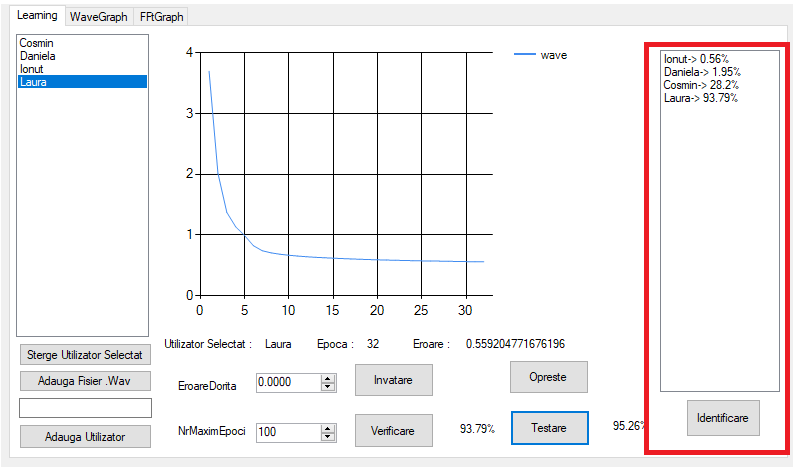


Cand utilizatorul apasa butonul verificare acesta poate sa adauge o inregistrare sau sa se inregistreze, procentul din partea dreapta a butonului reprezinta sansele ca inregistrarea sa apartina utilizatorului selectat





Prin apasarea butonului Identificare aplicatia o sa analizeze o inregistrare si o sa afiseze care sunt sansele ca inregistrarea sa apartina fiecarui utilizator determinand astfel a carui utilizator apartine inregistrarea.



# 

# 7. CONCLUZII

Scopul acestui proiect este de a construi o aplicație prin intermediul rețelelor neuronale care să fie capabilă să recunoască vorbitorul prin intermediul vocii. Numărul rețelelor neuronale folosite este echivalent cu numărul de utilizatori, fiecărui utilizator îi corespunde o rețea neuronala. Rețelele neuronale își dovedesc eficiența în această problema prin obținerea unui rezultat de 85-95% corect în urma testării după doar câteva sute de epoci procesate în mai puțin de 2 minute.

Pornind de la această aplicație pot fi dezvoltate diferite sisteme de securitate, de identificare prin intermediul voci.

Punctele slabe ale acestei aplicații sunt sensibilitatea la zgomote adiționale, cu cât numărul de utilizatori înregistrați e mai mic cu atât rețeaua e mai ineficienta, daca numărul de utilizatori este mai mic de 2 atunci aplicația este incapabilă să execute procesul de învățare. Un alt dezavantaj ar fi faptul că aplicația este incapabilă să distingă vocea unei persoane dacă în timpul înregistrării sau verificării vorbesc mai multe persoane.

Un avantaj al acestei aplicații este faptul că oricând poți adăuga sau șterge un utilizator fară nicio dificultate. Un alt avantaj este timpul de execuție scurt datorat simplificării calculelor de extragere a trăsăturilor.

Ca îmbunătățiri ar putea fi recunoașterea mai multor persoane din aceeași înregistrare audio.

# BIBLIOGRAFIE

[1]Mustafa Jamil, Md. Golam Rabbani Md. Saifur Rahman Md. Rashidul Hasan, "Speaker Identification Using Mel Frequency Cepstrum Coefficients," in 3rd International Conference on Electrical & Computer Engineering, Dhaka, 2004.

|  |
| --- |
| [2] John Moody and Christian J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," in *Neural Computation*., 1989, vol. 1. |
|  |

[3]​Magnus Rosell, "An Introduction to Front-End Processing and Acoustic Features for Automatic Speech Recognition," 2006.

[4] <https://en.wikipedia.org/wiki/Artificial_neural_network>

[5] https://towardsdatascience.compower-of-a-single-neuron-perceptron-c418ba445095

[6] <https://en.wikipedia.org/wiki/Neuron>

[7] <https://hackernoon.com/how-do-artificial-neural-network-recognize-images-c3699af0f553>

[8] [http://sintesis.ugto.mx/WintemplaWeb/01Neural%20Lab/00Introduction/03Logic%20 Operations/index.htm](http://sintesis.ugto.mx/WintemplaWeb/01Neural%20Lab/00Introduction/03Logic%20%20Operations/index.htm)

[9] https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0

[10] <https://mathworld.wolfram.com/HyperbolicTangent.html>

[11] https://mc.ai/activation-functions-relu-softmax/

[12] <https://medium.com/@jamesdacombe/an-introduction-to-artificial-neural-networks-with-example-ad459bb6941b>

[13] <http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/>

[14] <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>

[15]<https://kbi4.org/lesson-9-machine-learning-artificial-neural-network/>

[16] <https://www.mdpi.com/2073-4441/12/2/600/htm>

[17] <https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7>