NurosLab

Budişteanu Ionuţ – elev în clasa a X-a la CN„Mircea cel Batrân”,
ibudisteanu@acm.org
Mlisan Mirela – profesor îndrumător – CN „Mircea cel Batrân”,
mirela_mlisan@yahoo.com

Abstract

Softul permite crearea cu drag&drop(vizual), simularea, invăţarea de Reţele Neuronale Artificiale, Multi-Layer-Perceptron şi Kohonen cât şi poate genera un fişier DLL care să conţină reţeaua învăţată şi algoritmul de simulare, pentru a fi folosit în orice mediu de programare prentru dezvoltarea ulterioară a unei aplicaţii de către programator. Se pot genera retele neuronale şi în Pascal şi Delphi. Permite importarea şi exportarea din XML.

1. Introducere

Pentru a întelege, mai bine proiectul, s-au creat mai multe exemple:

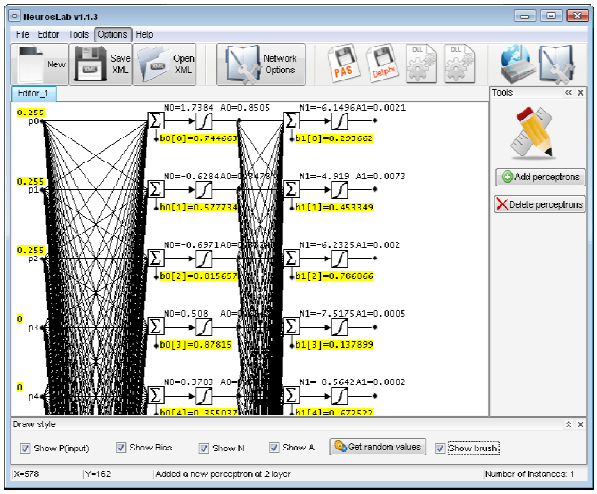
Optical Character Recognition pentru recunoaşterea scrisului de mână, Face recognition, Face classification, Speech recognition, Aproximări de funcţii elementare şi continue, Recunoaşterea persoanelor cu cagulă, ce intră intr-o bancă, amanent, Eye-Tracking, Interfata vocal (VUI) ,Clusterizări. Exemplele sunt funcţionale şi complete.

Ce alte exemple s-ar putea face?

Fingerprint recognition, retina recognition, boli din radiografii de exemplu: Cancerul, predictie, Human-Computer interface(Brain Computer Interface)

Ideea este, dacă vrem sa dezvoltăm o aplicaţie cu AI, nu trebuie să ştim să scriem o aplicaţie cu inteligenţă artificială, noi numai trebuie să scriem algoritmul de utilizare. Vom folosi NeurosLab, pentru a exporta ponderile reţelei şi să genereze coduri sursă.

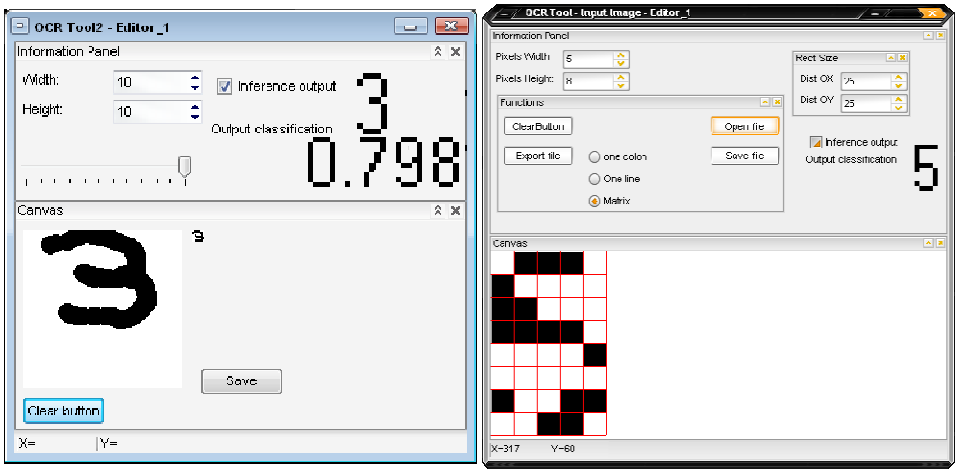
Algoritmi de învăţare implementaţi: BackPropagation, BackPropagation with Momentum, DarkenMoody, Kohonen-varianta pălăria mexicană.



Diferenţa dintre alte softuri şi NeurosLab este că dupa ce s-a antrenat o RNA, dacă se doreşte sa scrieţi o aplicaţie, care simuleaza reţeua învăţată, putem exporta ponderile reţelei şi softul poate să genereze(exporte) o clasa în Delphi sau Pascal sau Dynamic Link Library(DLL), care simulează RNA, deci putem folosi NeurosLab, în orice mediu de programare, care permite importări de funcţii din DLL. Putem importa/exporta din XML reţeaua şi ponderile şi bias-urile reţelei.

Optical Character Recognition(OCR)

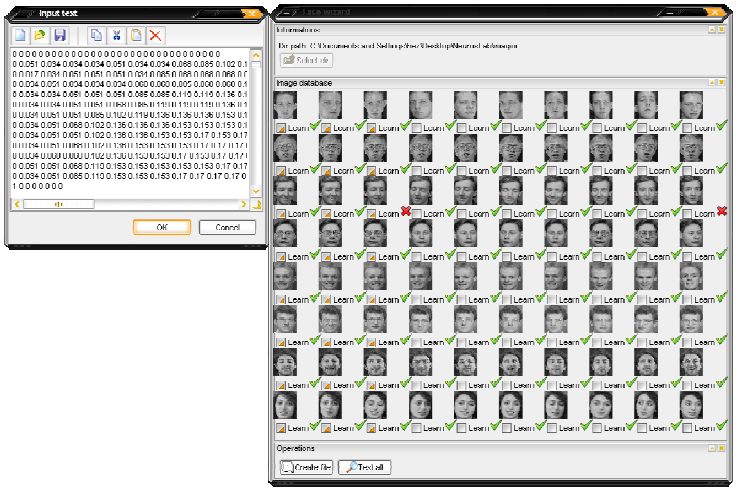
A fost creat pentru a testa rețeaua. Se poat desena caracterele şi pe urmă dacă se va bifa „Inference output", atunci softul crează o inferență logică pentru a clasifica automat în ce clasa este. Bineînțeles este creată o rețea neuronală, creat setul de antrenament(format din input şi target) şi învățată. Simularea este in timp real. Practic se pot vedea clasificările datelor cu erori(zgomot).



Face classification

Un alt exemplu este clasificarea fețelor umane.

Deoarece in imaginile alese există 39*32=1248 de pixeli şi matricile de pe stratul ascuns vor fi urişe, s-a defragmentat(impartit) matricea inițială, în matrici mai mici. Astfel se obțin 3 matrici de 13*32=416 pixeli, astfel matricile s-au micşorat,in acest fel simulează, învață mult mai repede şi necesită mai puțină memorie.



Fişierul de antrenament,
generat.

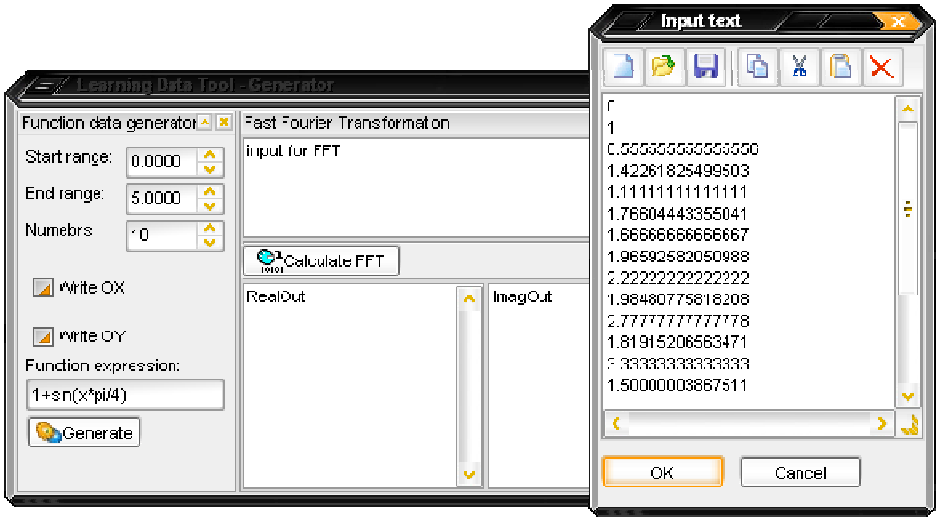
După ce a fost creat fişierul de antenament, reţeaua neuronală învăţă, şi reuşeste să clasifice imaginile şi din alte poziţii. Cel mai interesant e că pe omul #4 s-a învăţat cu ochelari, şi l-a recunoscut fără ochelari, şi chiar privind dintr-o anumit unghi diferit.

Dacă la OCR un algoritm euristic precum o distanţă Hamming, distanţă Levenshtein, distanţa metrica între matricea de intrare şi matricea învăţată fără zgomot fiind mica , ar funcţiona. Pe când la clasificarea feţelor umane nu ar funcţiona, deoarece dacă se modifică poziţia capului cu un singur milimetru, toţi pixeli din imagine se modifică – un algoritm gen Hamming esueaza.

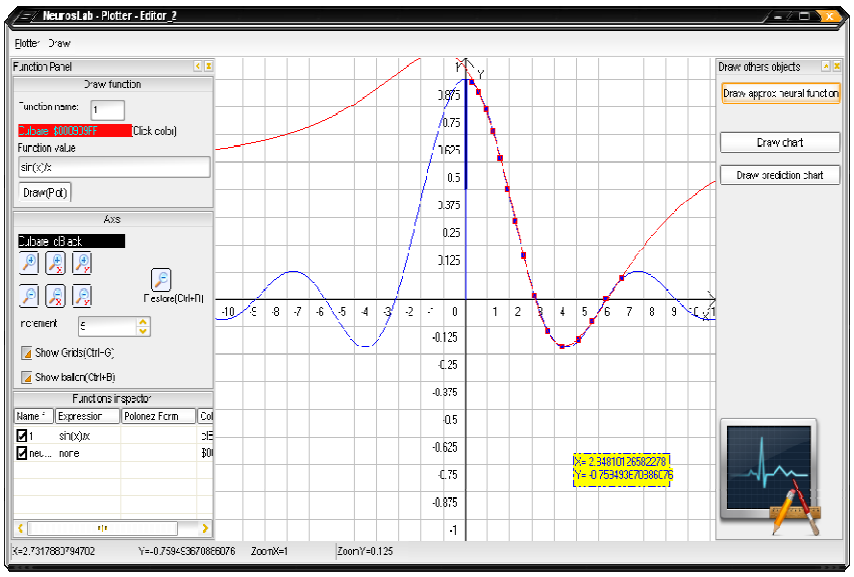
De reţinut, că ANN nu clasifică nu numai pozele învăţate, ci şi alte mii sau zeci de mii de poze, foarte asemănătoare cu acestea, sunt clasificate, către cea mai apropiată poză. Marele dezavantaj este că faptul ca o poză foarte „depărtată" de acestea, este totuşi clasificată într-una dintre clase, neexistand un mecanism de rejecţie a acesteea, practic ANN calculează o distanţa metrică, faţa de clase, şi o „aruncă" în clasa ceea mai apropiată.

Aproximări de funcţii elementare şi continue

Pentru a demonstra, că o reţea neuronală artificiala Multi-Layer-Perceptron poate să aproximeze funcţii elementare continue, s-a creat un tool care desenează funcţii. S-a creat un tool care permite luarea de eşantioane automat, dintr-o funcţie.



Se pot alege şi de mână eşantioanele selectând punctele din graficul funcţiei. In următoarea poză, se pot vedea eşantioanele(bulinele) cât şi functia aproximată de reţea(roşu). Practic reţelele neuronale sunt aproximatori universali şi pot aproxima din eşantioane funcţii, de sute sau mi de variabile. Acestea nu pot fi plotate, dar aplicaţiile sunt imense: meteorologie, prognoză consumului energetic. Se poate modela orice proces cu multe variabile



Speech recognition

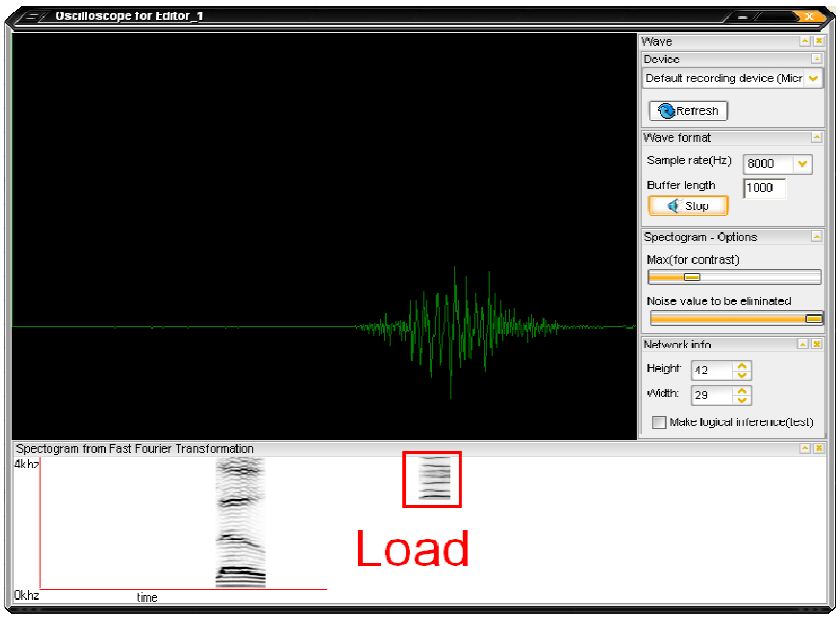
Întrucât ANN nu ştiu să facă decât clasificări, trebuie să transform, problema într-o problemă de clasificare. Sunetul trebuie transformat intr-o poza, pe care reţeua neuronală să incerce să o clasifice. Crearea unei poze în domeniul amplitudine-timp nu este utilă. Trebuie lucrat in domeniul timp-frecvenţă.

După aceea, s-a dorit să se arate dacă o retea neuronală MLP poate sa clasifice sunete. Astfel s-a creeat un osciloscop, in domeniul timp amplitudine. S-a lucrat cu Microsoft Multimedia System (mmsystem), luând astfel eşantioanele de la microfon. Eşantioanele sunt pe 8 biţi, s-a setat frecvenţa de eşantionare pe 8000 PCM. Când s-au desenat eşantioanele, nu s-a scos componenta continuă(DC). După aceea s-a scazut DC(componenta continuă) si înmulţit cu fereastra Hamming pentru a îmbunătaţi calitatea sunetului înregistrat de la microfon.

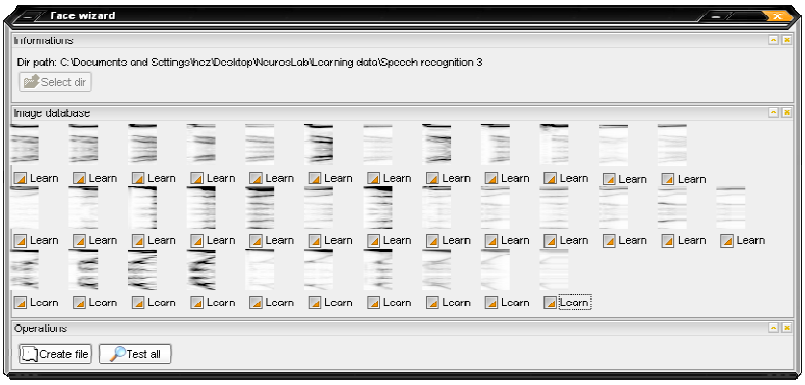
Fereastra hamming este: hamming(x)=0,54-0,46*cos(2*pi*(x/256)) Această funcţie este clasică. Funcţia este din MatLab, face corecţia de frecventă.

Algoritmul cu fereastra hamming: Se iau primele 256[1..256] de eşantioane se introduc într-un buffer şi se înmulţesc cu valorile hamming. După ce se calculează prima fereastră se introduce la analiza Transformatei Rapide Fourier, din rezultat se calculează modulul numerelor complexe după transformata, şi se obtine câte o coloana, din imaginea gray-scale, a spectogramei sunetului. Deci o să obţin 8000(numărul de eşantioane, luate intr-o secundă)/30, unde 30 este deplasamentul ferestrei hamming. După aceea, se reia procedeul, pe intervalul [30,286] şi se aplică aceeasi inmultire. Dupa aceea [60,316], etc...

Valorile care se obţin, in spectogramă, se normează cu 255, pentru a le transforma in valori de la [0..255]. Se aplică diverse metode, pentru a elimina zgomotul prin mediere, componenta continuă(DC), şi a separa, cuvintele de zonele de pauză. S-a aplicat diverse nivele de separare a zgomotului, şi a intervalului de pauză. Pentru realizarea, FFT-ului s-a utilizat singurul unit folosit din aplicaţie, scris de Dan Cross.



Spectogramele a 3 cuvinte(save, load, new), după aceea se obține fișierul de antrenament, şi dupa aceea se invață rețeaua neuronală.

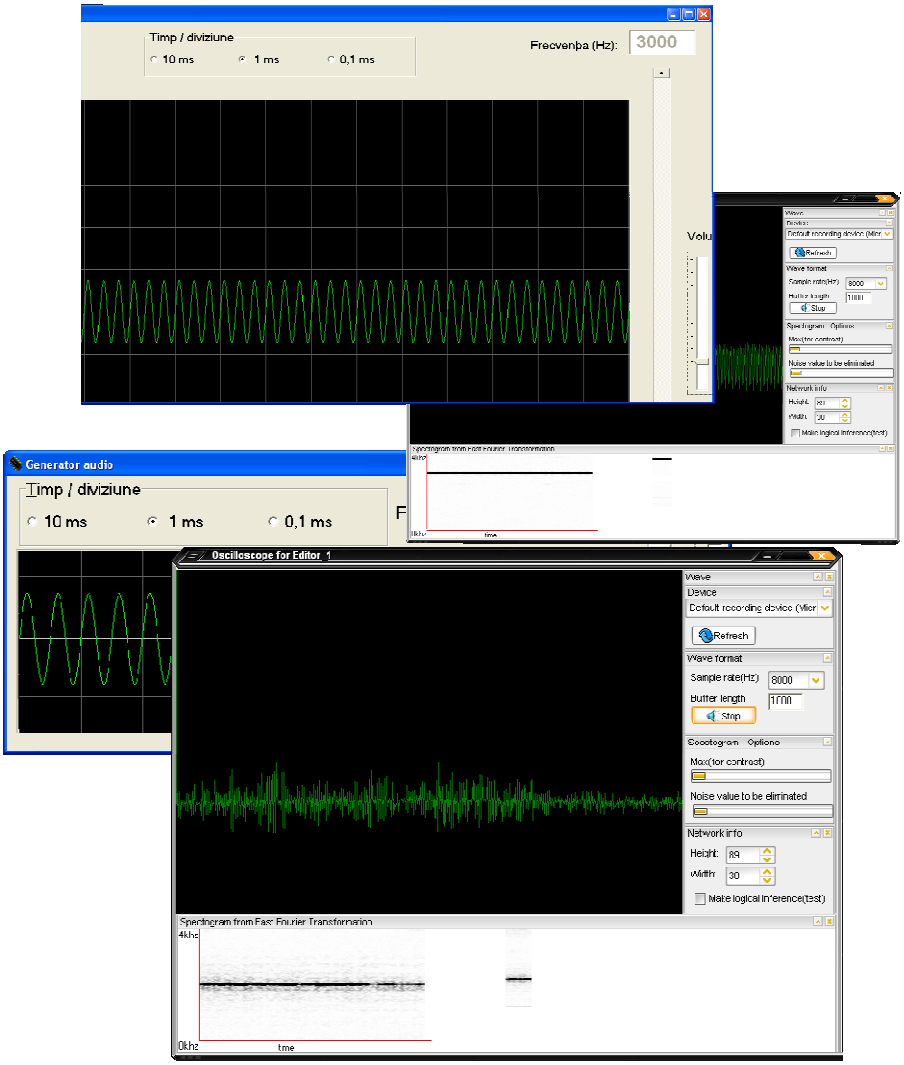


Exportarea rețelelor create şi învățate

Întrucât s-a intenționat să permită posibilitatea exportării rețelelor antrenate, altor utilizatori. În acest fel, un programator poate utiliza, partea de AI, creată în acest software, şi sa dezvolte în propriile sale aplicații, în funcție de necesitate. Programul exportă în format: Pascal, Delphi, DLL si XML. Se exportă, algoritmul de simulare, configurația rețelei, ponderile şi bias-urile rețelei(practic aici este AI). Exemplele create, sugerează capacitatea softului şi uşurința, de a exporta AI în aplicații, top-down.

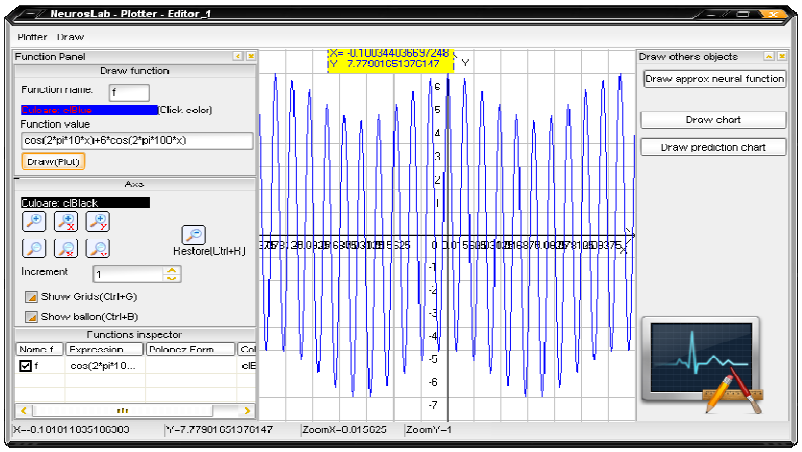
Testarea FFT-ului

Ca să putem vedea daca functionează corect FFT s-au creat sunete, compuse doar din anumite frecvențe. In imagine este un software românesc, care generează sunete, la anumite frecvențe. Zgomotele sunt de la TV, şi s-a optat să nu se elimine zgomotele. Cu această ocazie s-au pus valori certe pe axa de frecvențe pe spectogramă,

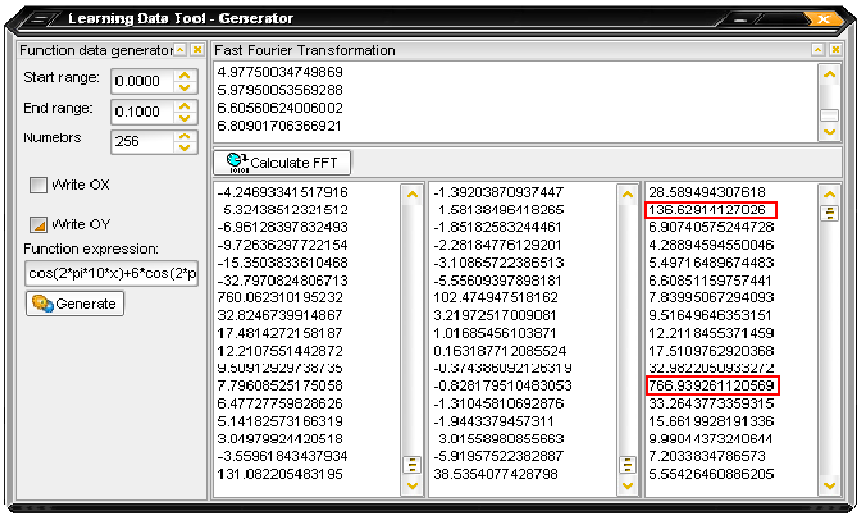


Acest tool, a redat posibilitatea sa se verifice FFT, întrucat s-a folosit un component care nu este realizat de mine, şi sunt începător în acest domeniu. S-a mai facut o verificare şi in Matlab pentru aceeasi funcție, concluzia a fost: componentul lucrează bine, şi a permis, sa etalonez axa Oy in domeniul frecvenței, a spectogramei sunetului.

Funcția pe care s-a testat a fost: **cos(2*pi*10*x)+6*cos(2*pi*100*x)**



S-au luat 256 de eşantioane, de la funcția anterioară, şi introdus la FFT



Alte opțiuni:

- Schimbarea limbi, este un algoritm recursiv importă XML, care permite schimbarea în orice limbă
- Permite reactualizarea fişierelor aplicației, de pe un server http cu noi exemple sau algoritmi de învățare.
- Crează statistici despre comportamentul aplicației(loguri), util în debug mode
- Deseneaza rețele kohonen bidimensionale, pentru a putea vizualiza clusteri. Multidimenisional clusterii nu pot fi vizualizați, dar pot fi separați în clase.

9. Informații

Site-ul softului: http://www.neuroslab.com/

O documentație mai completă se poate găsi la adresa:
http://www.neuroslab.com/Downloads/Doc/Romana%202.0.doc

Prezentări mai vechi ale softului:
Speech recognition http://www.trilulilu.ro/gigasoftware/d9d052d2a1ffa9
OCR prima versiune http://www.trilulilu.ro/gigasoftware/15acae86286e37
Face classification http://www.trilulilu.ro/gigasoftware/0e6e4708e063be
Crearea în Delphi al unui OCR cu NeurosLab
http://www.trilulilu.ro/gigasoftware/bbb199f1babd86

Bibliografia

[1] Virgil Tiponut, Catalin Caleanu, Rețele neuronale –arhitecturi si algoritmi, Univ de Vest
[2] Virgil Tiponut, Catalin Caleanu, Rețele neuronale. Aplicații, Universitatea de Vest
[3] Rețele neuronale - Nicolae Tăndăreanu
[4] Razvan Andonie , Inteligența artificială-„Universitatea „Babes-Bolyai" Cluj-Napoca
[5] MatLab help
[6] ACM Digital Library