

Ministerul Educației al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare Informatică și Microelectronică
Catedra Automatică și Tehnologii Informaționale

RAPORT

Lucrare de laborator nr 2

Disciplina: proiectarea sistemelor informaționale

Tema: Modelarea funcțională a sistemului în notația IDEF0.

A efectuat:

Vovc Artemie st. TI-133

A verificat:

Cojocaru Svetlana lector universitar

Chișinău 2016

Cuprins

1 Sarcina.....	3
2 Descrierea domeniului	4
3 Determinarea contextului de modelare	6
4 Proiectarea diagramelor	7
Concluzia	11
Bibliografia	12

1 Sarcina

Elaborarea diagramelor de context în notația IDEF0, decompoziție de nivelul unu în notația IDEF0 și decompoziția de nivelul doi în notația IDEF0.

Scopul lucrării:

- de a alege domeniul obiectiv;
- descrierea domeniului;
- determinarea contextului de modelare;
- elaborarea diagramei de context în notația IDEF0;
- elaborarea diagramei de decompoziție de nivelul unu în notația IDEF0;
- elaborarea diagramei de decompoziție de nivelul doi în notația IDEF0.

2 Descrierea domeniului

Cercetările din domeniul Inteligenței artificiale au vizat dezvoltarea conceptului de calcul neuronal, un instrument folosit în generarea de sisteme cu inteligență artificială.

Un loc aparte îl constituie studiile privind învățarea automată și mașinile de învățare sau auto-instruibile („machine learning”). Învățarea automată studiază sistemele capabile de a-și îmbunătăți performanțele, bazându-se pe o serie de date de instruire.

Calculul neuronal încearcă să genereze, să dezvolte sisteme instruibile pentru scopuri generale, folosind o cantitate mică de cunoștințe inițiale. Astfel de sisteme se mai numesc rețele neuronale sau sisteme conexioniste. În literatura de specialitate, se mai găsesc denumirile alternative de sisteme cu auto-organizare, sisteme de prelucrare paralelă distribuită, modele neurodinamice, calculatoare neuronale, etc.

Modelul conexionist presupune memorarea informației în mod difuz, în toată rețeaua, diferind de modul de memorare la anumite adrese de memorie, în cazul calculatoarelor electronice.

Calculatoarele electronice oferă performanțe în rezolvarea problemelor din diverse domenii, precum inginerie, economie, medicină, cercetare științifică, etc, însă au limitări referitoare la problemele de percepție și învățare din experiență, specifice naturii umane. De aceea, cercetările s-au concentrat pe lărgirea spectrului posibilităților de programare ale unui calculator convențional, folosind cunoștințele neurobiologice.

Primele cercetări realizate de McCulloch și Pitts în anul 1943, au pus bazele calculului neuronal, prin definirea modelului neuronului. McCulloch și Pitts au dezvoltat un model logic bazat pe predicate și au clasificat neuronii în două categorii: de tip excitator sau inhibitor, enunțând propoziții cu condiții necesare și suficiente. Principalele rezultate au fost demonstrate în cazul rețelelor neuronale neregulate, în care sinapsele sau conexiunile dintre neuroni nu formează cicluri.

Teoria calculului neuronal, caracteristică anilor '40, a fost reformulată de Kleene în 1956, care a dezvoltat modelul rețelelor regulate. Rezultatele matematice se referă la activarea neuronilor de intrare și definesc starea rețelei neuronale, după procesarea tuturor semnalelor.

Rosenblatt a propus în 1958 un tip de rețea bazată pe perceptroni, obținută prin interconectarea unei mulțimi de neuroni, definind astfel primul model de rețea neuronală artificială. Conform teoriei lui Rosenblatt, perceptronul conține cinci elemente de bază: un vector cu intrări, ponderile (conexiunile dintre neuroni), funcția de însumare, dispozitivul de detecție a pragului și o ieșire. Ieșirea este 1 (sau -1) și reprezintă valoarea funcției de activare, aplicată combinației semnalelor de intrare, luându-se în considerare cazul depășirii valorii prag.

Rețelele neuronale cu mai multe straturi de percepționi, propuse de Rosenblatt, erau capabile să rezolve probleme simple de clasificare, prin modificarea ponderilor conexiunilor dintre neuroni. Tehnica de antrenare folosită se numește regula de învățare a perceptronului.

Interesul actual pentru acest domeniu este justificat în primul rând de posibilitățile ultimelor generații de calculatoare, care oferă puterea de calcul necesară cercetării RNA. RNA are ca punct de inspirație sistemul nervos uman. Specialiștii consideră că la ora actuală sistemul biologic este insuficient explorat și de aceea modulele utilizate pentru conceperea unei RNA reprezintă introducerea într-un model biologic simplificat.

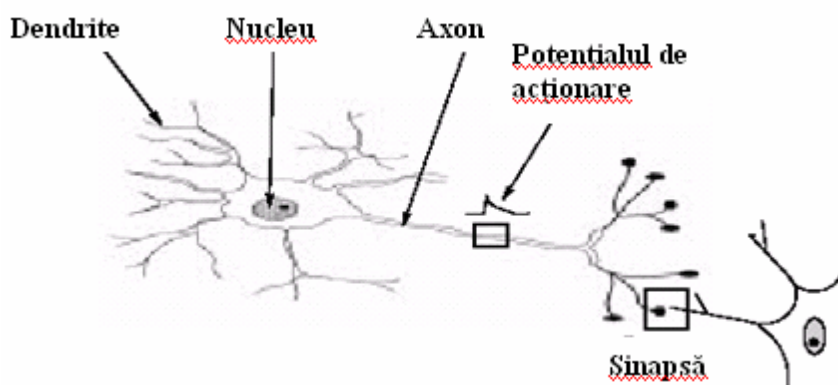


Figura 2.1 – Neuronul biologic

Unitatea de organizare a sistemului nervos este neuronul, (fig. 2.1) o celulă care reprezintă un număr de dendrite și un axon, prin intermediul căruia se conectează la alți neuroni. Dendritele constituie intrările în celula neuronală. Funcția dendritelor este aceea de a recepționa excitații și de a le conduce până la corpul neuronului. Axonul reprezintă ieșirea. Funcția axonilor este aceea de a conduce influxul nervos de la corpul celular la dendritele sau corpul celular al altui neuron. RNA este alcătuită dintr-o mulțime de noduri în care se află neuronii artificiali, elemente de procesare neliniară care operează în paralel. Prin analogie cu neuronul biologic, un neuron artificial are mai multe intrări și o ieșire, care se poate conecta la intrările altor neuroni.

3 Determinarea contextului de modelare

Subiectul modelării

Ca subiectul modelării este luat sistemul informațional „SDK rețele neuronale” care este la rândul său un mecanism ce permite formarea altor rețele.

Scopul modelării

Pentru ce este nevoie de acest sistem?

Pentru a genera rețele neuronale artificiale (RNA) care vor fi capabile să prezică soluții la toate problemele care apar.

Ce trebuie să facă sistemul?

- generarea RNA;
- antrenarea RNA;
- performarea;
- publicarea RNA care rezolvă o problemă anumită.

Ce obținem la final pentru utilizatori?

În acest caz sunt două tipuri de utilizatori, simplu utilizator și un alt dezvoltator. Dacă de luat din punct de vedere a unui dezvoltator sistemul dat va reprezenta ca un mecanism pentru dezvoltarea altor produse performante ce necesită calcul neuronal. Din viziunea unui utilizator simplu va fi o soluție la o problemă de zi cu zi, așa cum rețelele neuronale pot prevedea viitorul cu o eroare destul de mică, oamenii o pot utiliza în interesele sale în dependență cum a fost învățată și ce date a evoluat rețeaua.

Punctul de vedere

Viteza de dezvoltare a părții hard în calculatoare formează un impuls la dezvoltarea rețelelor neuronale. În ultimele decenii se evidențiază o succesiune de aplicații mobile care folosesc în baza sa rețelele neuronale ca de exemplu „Prisma” (reprezintă o aplicație ce este în stare să facă o poză și să-i atribuie niște schimbări care o face dintr-o poză obișnuită o pictură de artă). Prisma este doar un simplu exemplu de sistem informațional în care este utilizat rețelele neuronale și este un exemplu destul de apropiat pentru utilizator simpli de a înțelege forța rețelelor neuronale.

4 Proiectarea diagramelor

În ceea ce urmează vom analiza diagrama de context vezi figura 4.1.

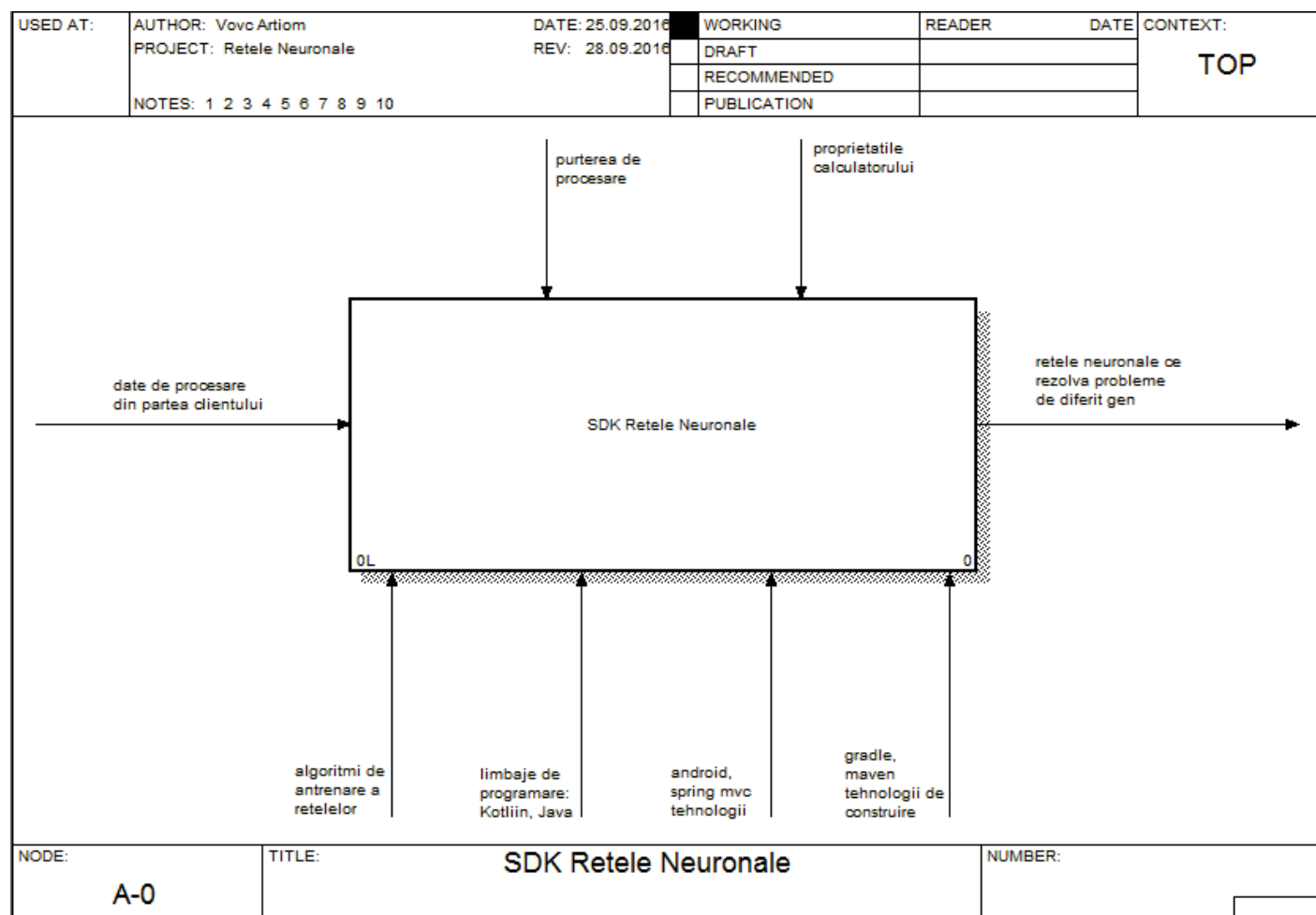


Figura 4.1 – diagrama de context „SDK rețele neuronale”

Intrările:

- datele de procesare din partea clientului.

Ieșirile:

- rețele neuronale ce rezolvă diferite probleme.

Reglări:

- puterea de procesare;
- proprietățile calculatorului.

Mecanisme:

- algoritmi de antrenare a rețelelor;

- limbajele de programare kotlin și java;
- android, rest api server;
- gradle, maven tehnologii de construire a proiectelor.

Mai jos este reprezentat diagrama de decompoziție nivelul unu. În ea se observă procesele de bază ce se produc în sistem. Vezi figura 4.2 detalii despre procesele de baza a sistemului.

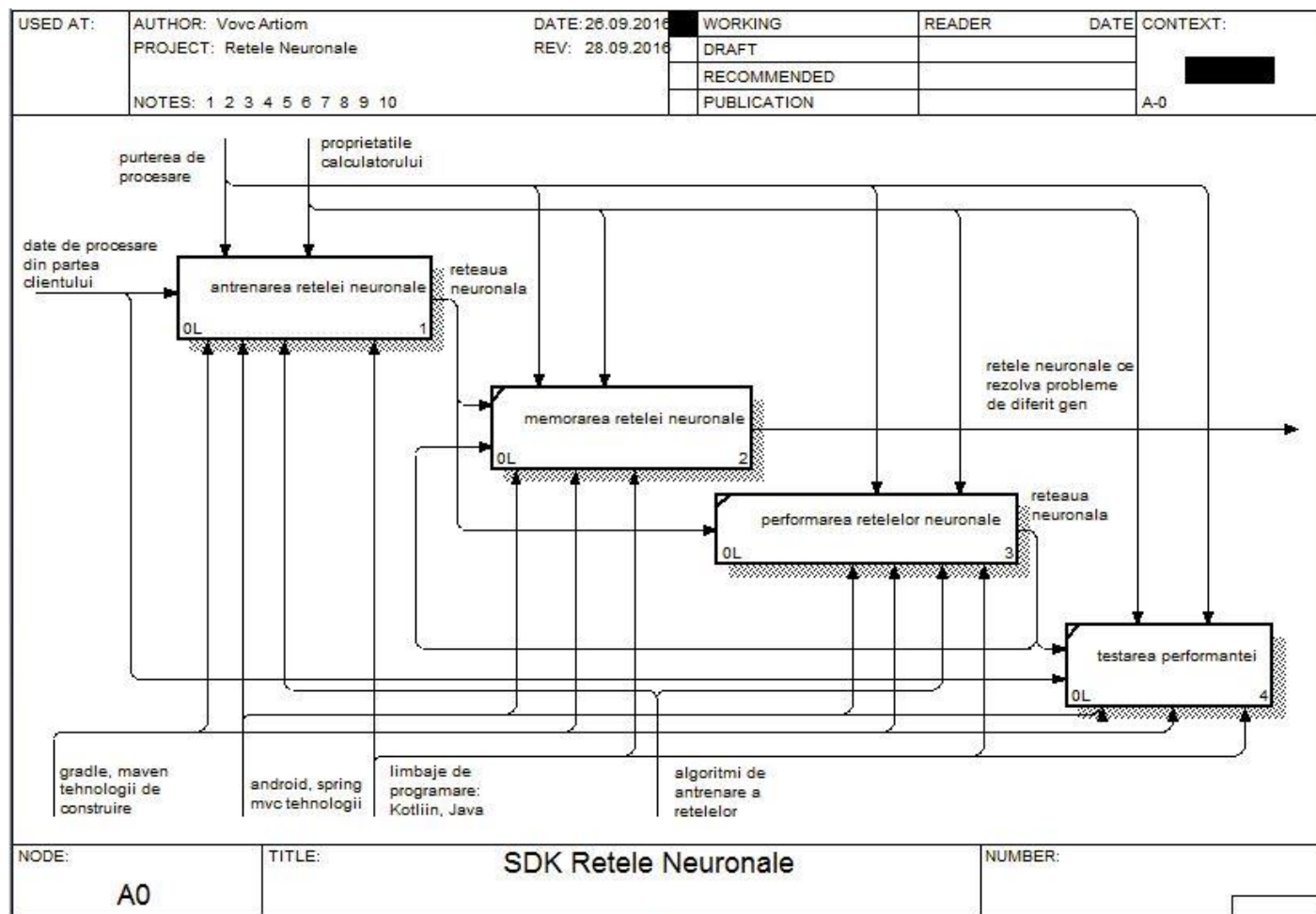


Figura 4.2 – diagrama de decompoziție nivelul unu pentru sistem

Sistemul informațional implementat va conține două componente principale care vor utiliza în comun o componentă numită SDK(software development kit) rețele neuronale. Acest SDK în ansamblu reprezintă nucleul sistemului în care se va afla toate instrumente logice necesare pentru a crea diferite tipuri de rețele neuronale și a le suporta în diferit mod. Mai jos sunt enumerate funcționalitățile minime a SDK-ului.

- procesarea datelor de intrare;
- formarea rezultatului;

- aplicarea algoritmilor;
- antrenarea rețelelor neuronale;
- memorarea rețelelor neuronale;
- performarea rețelelor neuronale;
- resetarea rețelelor neuronale;
- setarea tipului de procesare (performant sau în mod econom);
- determinarea algoritmului optim.

Procesarea datelor de intrare conține subproces de validare a datelor la nivel semantic și sintactic. Funcția de formarea rezultatului reprezintă o stabilire a unei forme adecvate de răspunsuri pentru un utilizator simplu, se are în vedere la baza acestui proces se ascund tabele de suprapunere a datelor calculate cu date reale înțelese de utilizatori simpli. Aplicarea algoritmilor una din cele mai importante procese deoarece un algoritm nepotrivit după care se învață o rețea neuronală poate duce la date ironate din partea rețelelor, sunt și cazuri de aplicare a mai multor tipuri de algoritmi pentru a micșora riscul de degradare a rețelelor neuronale, din motiv că la orice introducere de date la intrarea rețelelor neuronale fiecare neuron își schimbă valoarea în forma de greutate chiar dacă rețeaua de acum este antrenată (învățată) să răspundă corect la problema cutare.

Antrenarea rețelelor neuronale presupune primele etape de formare în ansamblu a rețelelor neuronale, deci doar la etapa de formare inițială se produce antrenare după care urmează mentenanța sau așa numit performarea lor. Memorarea rețelelor neuronale e o funcție de bază din motiv că sistemul informațional descris oferă niște servicii sociale ca publicarea rețelelor neuronale, acest proces conține în sine și o parte importantă și una dificilă de atribuire a greutăților fiecărui neuron într-o rețea neuronală. Resetarea rețelelor neuronale este o funcție ascunsă de utilizator ea se folosește doar de sistem pentru a performa rețelele neuronale. Setarea tipului de procesare e o funcție accesată de client, în caz dacă sistemul depistează un nivel înalt de necesitate de resurse în calcul este cedat calculele spre server. Determinarea algoritmului eficient de calcul sau optim este o funcționalitate adăugatoare pentru a oferi clienților o variantă optimală la etapa de formare a rețelelor neuronale.

Va urma ultima diagramă din modelul IDEF0 numită diagrama de decompoziție nivelul doi. În această diagramă se descrie subprocesele unui proces de bază a sistemului informațional „SDK rețele neuronale”.

Vezi figura 4.3 cu diagrama de decompoziție nivelul doi în care se vizualizează toate subprocese procesului de bază numit „antrenarea rețelelor neuronale”.

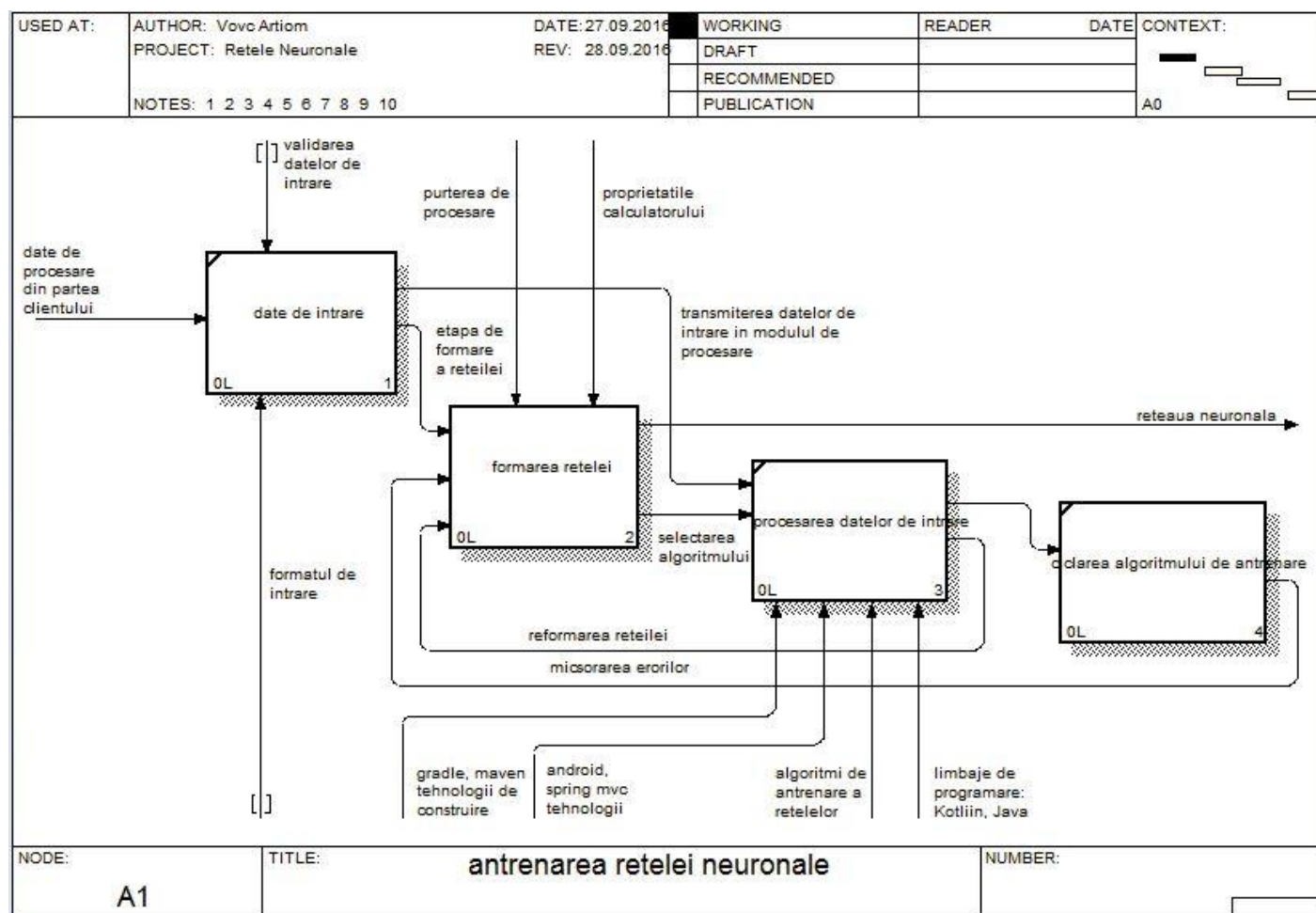


Figura 4.3 – diagrama de decompoziție nivelul doi pentru sistem

Cum se vede procesul detaliat conține în sine patru subprocese numite: date de intrare, formarea rețelei, procesarea datelor de intrare, ciclarea algoritmului de antrenare.

Datele de intrare sunt un set de informații ce este furnizat de un utilizator care așteaptă un rezultat anumit. Acestea date sunt validate de subproces pentru a nu fi erori în calcul. Prin validare se are în vedere gradul de adevăr al datelor, dacă sunt reale.

Formarea rețelei reprezintă un proces care asigură rețeaua neuronală cu o structură optimală pentru fiecare problemă în parte.

Procesarea datelor de intrare, se are în vedere procesul de schimbare a greutatea fiecărui neuron la o iterație. Aici încă se mai stabilește o metodă de învățare a rețelei, mai exact algoritmul de dezvoltare.

Concluzia

Lucrarea dată a avut ca scop să formeze o viziune de sistemul informațional pentru utilizatorii acestui sistem. În lucrare a fost prezentate diagramele cerute din sarcină pentru tema „SDK neural networks”. A fost reprezentate procesele de bază a sistemului informațional și subprocessele proceselor de bază. Allfusion process modeler oferă dezvoltatorilor un mod de a prezenta și de a forma o viziune a sistemul informațional pentru utilizatorilor cointeresați în mod direct sau indirect de sistem.

Bibliografia

1. Indrumarul metodic al universității tehnice din Moldova. AllFusion process modeler.
2. Resursă electronică bazele programului AllFusion process modeler. [regim de acces]:
<http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab02.html>