**Ministerul Educaţiei al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare Informatică şi Microelectronică**

**Catedra Automatică şi Tehnologii Informaţionale**

**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](#_Toc462953507)

[2 Descrierea domeniului 4](#_Toc462953508)

[3 Determinarea contextului de modelare 6](#_Toc462953509)

[4 Proiectarea diagramelor 7](#_Toc462953510)

[Concluzia 11](#_Toc462953511)

[Bibliografia 12](#_Toc462953512)

# 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;
* descrierera domeniului;
* determinarea contextului de modelare;
* elaborarea diagramei de context în notația IDEF0;
* elaborarea diagramei de decompozișie de nivelul unu în notația IDEF0;
* elaborarea diagramei de decompozișie de nivelul doi în notația IDEF0.

# 2 Descrierea domeniului

Cercetãrile din domeniul Inteligenţei artificiale au vizat dezvoltarea conceptului de calcul neuronal, un instrument folosit în generarea de sisteme cu inteligenţã artificialã.

Un loc aparte îl constituie studiile privind învãţarea automatã şi maşinile de învãţare sau auto-instruibile („machine learning”). Învãţarea automatã studiazã sistemele capabile de a-şi îmbunãtãţi performanţele, bazându-se pe o serie de date de instruire.

Calculul neuronal încearcã sã genereze, sã dezvolte sisteme instruibile pentru scopuri generale, folosind o cantitate micã de cunoştinţe iniţiale. Astfel de sisteme se mai numesc reţele neuronale sau sisteme conexioniste. În literatura de specialitate, se mai gãsesc denumirile alternative de sisteme cu auto-organizare, sisteme de prelucrare paralelã distribuitã, modele neurodinamice, calculatoare neuronale, etc.

Modelul conexionist presupune memorarea informaţiei în mod difuz, în toatã reţeaua, diferind de modul de memorare la anumite adrese de memorie, în cazul calculatoarelor electronice.

Calculatoarele electronice oferã performanţe în rezolvarea problemelor din diverse domenii, precum inginerie, economie, medicinã, cercetare ştiinţificã, etc, însã au limitãri referitoare la problemele de percepţie şi învãţare din experienţã, specifice naturii umane. De aceea, cercetãrile s-au concentrat pe lãrgirea spectrului posibilitãţilor de programare ale unui calculator convenţional, folosind cunoştinţele neurobiologice.

Primele cercetãri realizate de McCulloch şi Pitts în anul 1943, au pus bazele calculului neuronal, prin definirea modelului neuronului. McCulloch şi Pitts au dezvoltat un model logic bazat pe predicate şi au clasificat neuronii în douã categorii: de tip excitator sau inhibitor, enunţând propoziţii cu condiţii necesare şi suficiente. Principalele rezultate au fost demonstrate în cazul reţelelor neuronale nerecurente, în care sinapsele sau conexiunile dintre neuroni nu formeazã cicluri.

Teoria calculului neuronal, caracteristicã anilor ’40, a fost reformulatã de Kleene în 1956, care a dezvoltat modelul reţelelor recurente. Rezultatele matematice se referã la activarea neuronilor de intrare şi definesc starea reţelei neuronale, dupã procesarea tuturor semnalelor.

Rosenblatt a propus în 1958 un tip de reţea bazatã pe perceptroni, obţinutã prin interconectarea unei mulţimi de neuroni, definind astfel primul model de reţea neuronalã artificialã. Conform teoriei lui Rosenblat, perceptronul conţine cinci elemente de bazã: un vector cu intrãri, ponderile (conexiunile dintre neuroni), funcţia de însumare, dispozitivul de detecţie a pragului şi o ieşire. Ieşirea este 1 (sau -1) şi reprezintã valoarea funcţiei de activare, aplicatã combinaţiei semnalelor de intrare, luându-se în considerare cazul depãşirii valorii prag.

Reţelele neuronale cu mai multe straturi de perceptroni, propuse de Rosenblatt, erau capabile sã rezolve probleme simple de clasificare, prin modificarea ponderilor conexiunilor dintre neuroni. Tehnica de antrenare folosită se numeşte regula de învãţare a perceptronului.

Interesul actual pentru acest domeniu este justificat în primul rând de posibilităţile ultimelor generaţii de calculatoare, care oferă puterea de calcul necesară cercetării RNA. RNA au ca punct de inspiraţie sistemul nervos uman. Specialiştii consideră ca la ora actuală sistemul biologic este insuficient explorat şi 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 şi un axon, prin intermediul căruia se conectează la alţi neuroni. Dendritele constituie intrările în celula neuronală. Funcţia dendritelor este aceea de a recepţiona extitaţii şi de a le conduce până la corpul neuronului. Axonul reprezintă ieşirea. Funcţia 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 mulţime 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 şi o ieşire, 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 rîndul 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 doua tipuri de utilizatori, simplu utilizator și un alt dezvoltator. Dacă de luat din punct de vede 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.

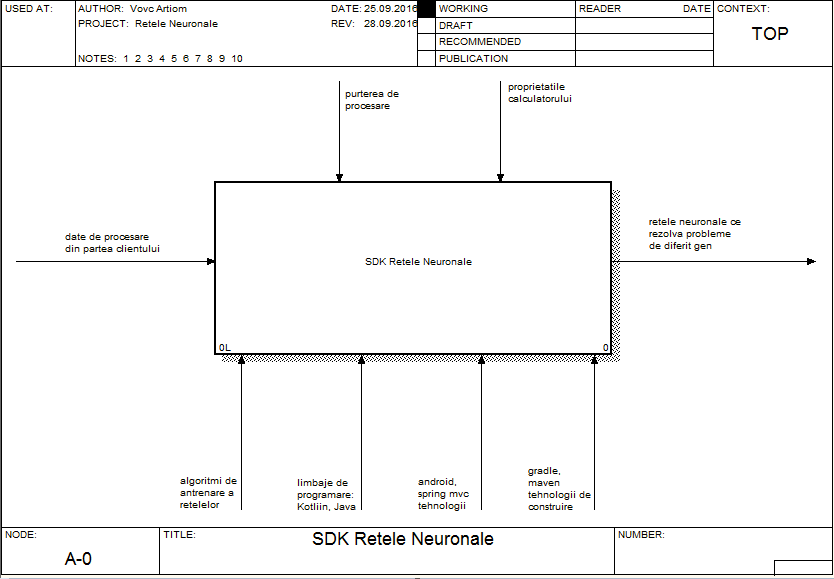


Figura 4.1 – diagrama de context „SDK rețele neuronale”

Intrările:

- datele de procesare din partea clientului.

Ieșirile:

- rețele neuronale ce rezolvă diferite probleme.

Reglări:

- puterea de procesare;

- priprietățile 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.

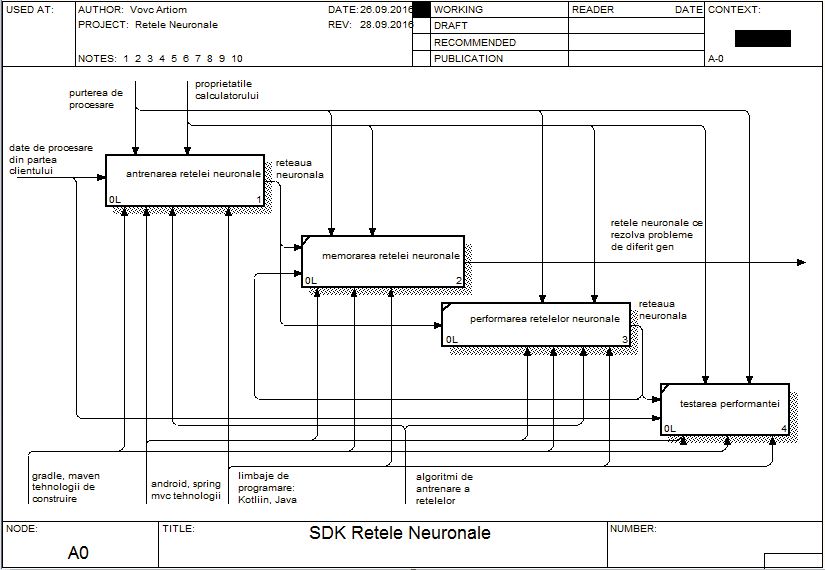


Figura 4.2 – diagrama de decompoziție nivelul unu pentru sistem

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 intsrumente 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-aului.

* 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 inrare conține subprocese de validare a datelor la nivel semantic și sintaxic. 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 deacum este antrenată (învățată) să raspundă 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 înnalt 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 neurnale”.

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

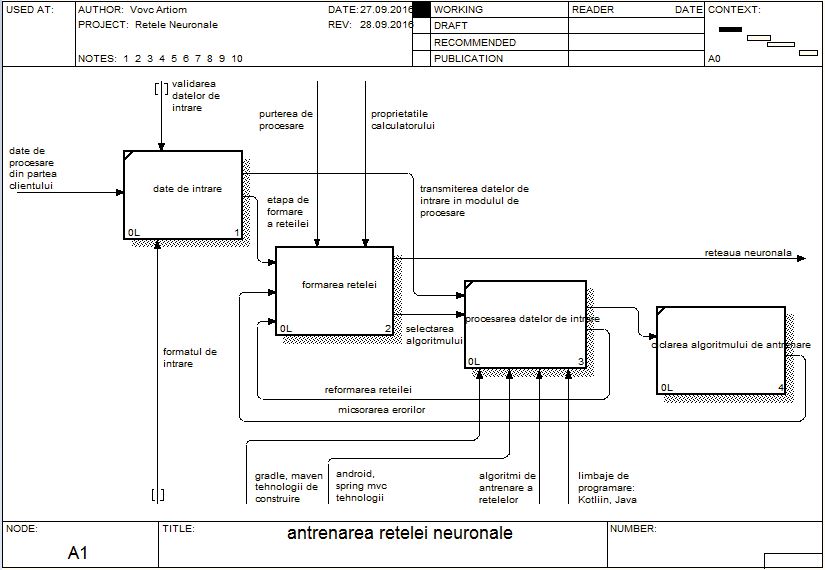


Figura 4.3 – diagrama de decompoziție nivelul doi pentru sistem

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

Datele de intrare sunt un set de informații ce este furnizat de un utilizator care așteptă 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 greutăților 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 subprocesele 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