

Ministerul Educației, Culturii și Cercetări al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Informatică și Ingineria Sistemelor

**Lucrare de laborator nr.2**

La disciplina

Sisteme de inteligență artificială

A efectuat: st.gr. IA-182

Ulmanu Cristian

A verificat: Stratulat Ștefan

Chișinău 2020

Tema: **Generații de Rețele Neurale Artificiale.**

Scopul:

* descrierea caracteristicilor (rețelelor) neurale de tip Perceptron, Recurente și Spiking;
* prezentarea unor funcții de activare pentru neuroni;
* descoperirea parametrilor unor neuroni care aproximează funcții logice elementare și reprezentarea regiunilor de decizie ale acestora.

### Sarcină:

Proiectarea și implementarea unui neuron în 3 generații.

* Neuron de tip McCulloch-Pitts
* Neuron Recurent
* Neuron Spiking

**Neuron de tip McCulloch-Pitts**

Un neuron artificial ( neuron matematic McCulloch - Pitts , neuron formal ) este un nod de rețea neuronală artificială , care este un model simplificat al unui neuron natural . Matematic, un neuron artificial este de obicei reprezentat ca o funcție neliniară a unui singur argument - o combinație liniară a tuturor semnalelor de intrare. Această funcție se numește funcția de activare sau funcția de preluare , funcția de transfer... Rezultatul este trimis la o singură ieșire. Astfel de neuroni artificiali sunt combinați în rețele - conectează ieșirile unor neuroni cu intrările altora. Neuronii și rețelele artificiale sunt elementele de bază ale unui neurocomputer ideal .

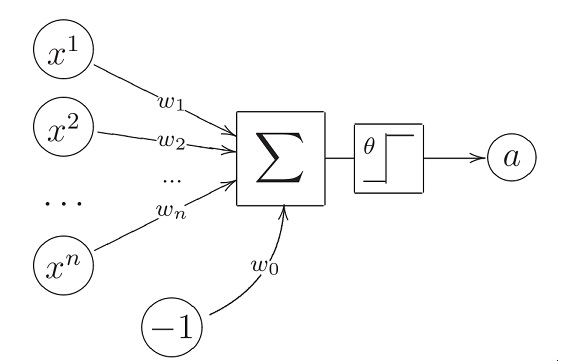
Matematic, un neuron este un sumator ponderat, a cărui singură ieșire este determinată prin intrările sale și o matrice de greutate după cum urmează:



Aici {\ displaystyle x\_ {i}} și {\ displaystyle w\_ {i}} - respectiv, semnalele de la intrările neuronului și greutățile intrărilor, funcția u se numește câmp local indus, iar f (u) este funcția de transfer. Valorile posibile ale semnalelor la intrările unui neuron sunt considerate date în interval{\ displaystyle [0,1]}... Ele pot fi digitale (0 sau 1) sau analogice. Intrare suplimentară{\ displaystyle x\_ {0}} și greutatea corespunzătoare {\ displaystyle w\_ {0}}sunt utilizate pentru *inițializarea unui* neuron [[8]](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD#cite_note-%D0%A2%D0%B5%D1%80%D0%B5%D1%85%D0%BE%D0%B2-8) . Inițializarea înseamnă o deplasare a funcției de activare a unui neuron de-a lungul axei orizontale, adică formarea pragului de sensibilitate al unui neuron [[5]](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD#cite_note-_12f5d3d17fa9584b-5) . În plus, uneori se adaugă la ieșirea unui neuron o variabilă aleatorie numită schimbare. Schimbarea poate fi privită ca un semnal la o sinapsă suplimentară, mereu încărcată.

Practic, neuronii sunt clasificați în funcție de poziția lor în topologia rețelei. Acțiune:

* *Neuroni de intrare*  - luați vectorul original care codifică semnalul de intrare. De regulă, acești neuroni nu efectuează operații de calcul, ci transmit pur și simplu semnalul de intrare primit la ieșire, posibil prin amplificarea sau slăbirea acestuia;
* *Neuronii de ieșire*  sunt ieșirile rețelei. În neuronii de ieșire, pot fi efectuate orice operații de calcul;
* *Neuroni intermediari*  - efectuează operații de calcul de bază .



Dezavantajul acestui model este chiar modelul funcției de tranziție a tipului „prag” al neuronului. În formalismul lui W. McCulloch-Pitts, neuronii au stări 0, 1 și logică prag de tranziție de la stat la stat. Fiecare neuron din rețea determină o sumă ponderată a stărilor tuturor celorlalți neuroni și îl compară cu un prag pentru a-și determina propria stare.

Tipul de prag al funcției nu oferă rețelei neuronale suficientă flexibilitate în antrenament și reglare pentru o sarcină dată. Dacă valoarea produsului cu punct calculat, chiar și nesemnificativ, nu atinge pragul specificat, atunci semnalul de ieșire nu este generat deloc și neuronul „nu trage”. Aceasta înseamnă că intensitatea semnalului de ieșire (axon) al unui neuron dat se pierde și, prin urmare, se formează o valoare de nivel scăzut la intrările ponderate din următorul strat de neuroni.

În plus, modelul nu ia în considerare multe caracteristici ale funcționării neuronilor reali (natura impulsivă a activității, însumarea neliniară a informațiilor de intrare, refractaritatea).

Pur și simplu, un neuron McCulloch-Pitts constă din următoarele:

Un nod având o valoare prag specificată.

Un număr finit, dar nelimitat, de intrări cu valori binare către nod. Fiecare intrare este una dintre cele două tipuri:

Intrările de excitare contribuie la excitația globală a unui nod. Dacă suma valorilor de intrare excitative ale nodului este egală sau depășește pragul nodului, neuronul este considerat a fi activat și „declanșează”, oferind o valoare de 1. Dacă, totuși, excitația totală scade sub pragul nodului, unitatea nu se declanșează și scoate 0.

Intrările inhibitorii împiedică excitația neuronală. Dacă oricare dintre intrările inhibitoare transmite o valoare de 1, neuronul este considerat a fi inhibat și nu trage așa cum este indicat de valoarea de ieșire 0.

class MPNeuron(object):

def \_\_init\_\_(self, threshold, inputs):

self.threshold = threshold

self.inputs = inputs

def activate(self):

excitations = 0

for trigger in self.inputs:

if trigger.excitatory:

excitations += trigger.value

else:

if trigger.value:

return 0

if excitations >= self.threshold:

return 1

return 0

class MPInput(object):

def \_\_init\_\_(self, excitatory):

self.excitatory = excitatory

self.value = 0

def trigger(self, value):

self.value = value

def AND(x1, x2):

inputs = [MPInput(True), MPInput(True)]

gate = MPNeuron(2, inputs)

inputs[0].trigger(x1)

inputs[1].trigger(x2)

return gate.activate()

def OR(x1, x2):

inputs = [MPInput(True), MPInput(True)]

gate = MPNeuron(1, inputs)

inputs[0].trigger(x1)

inputs[1].trigger(x2)

return gate.activate()

def NOT(x):

inputs = [MPInput(False)]

gate = MPNeuron(0, inputs)

inputs[0].trigger(x)

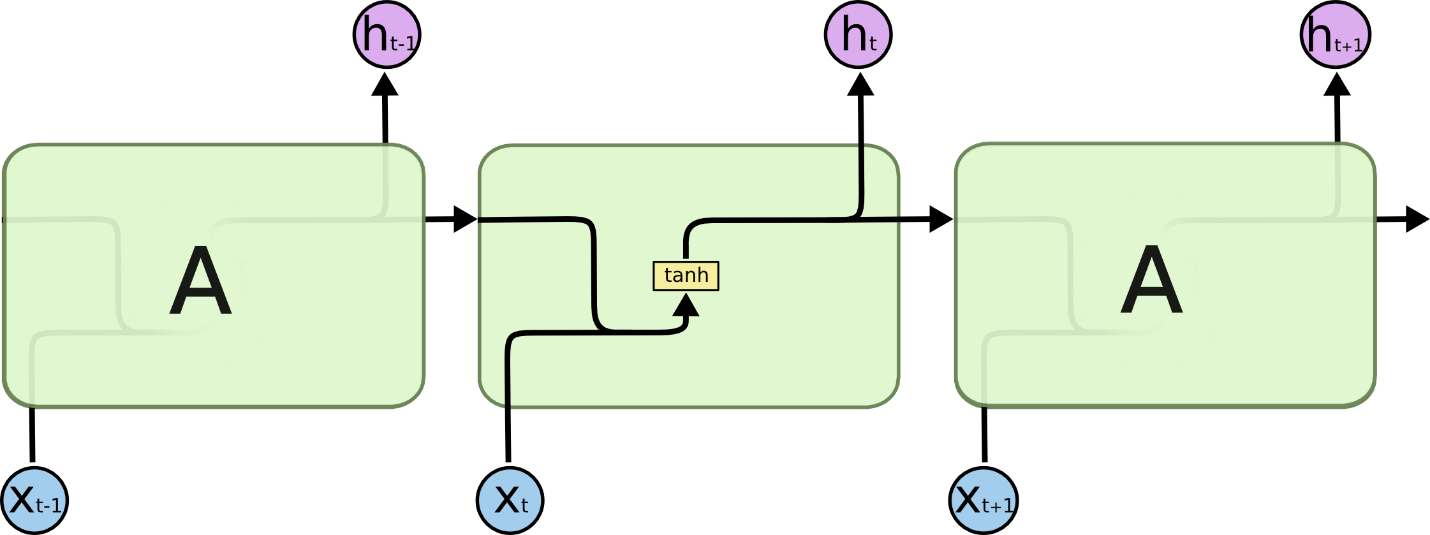
return gate.activate()

**Rețeaua neuronală recurentă** ( **RNS** , [Engl.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Rețea neuronală recurentă ; RNN ) - [rețele neuronale de](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) tip , în care comunicațiile dintre elemente formează o secvență direcționată. Acest lucru face posibilă procesarea unei serii de evenimente în timp sau lanțuri spațiale succesive. Spre deosebire de [perceptronii](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%86%D0%B5%D0%BF%D1%82%D1%80%D0%BE%D0%BD) multistrat , rețelele recurente își pot folosi memoria internă pentru a procesa secvențe de lungime arbitrară. Prin urmare, RNN-urile sunt aplicabile în sarcini în care ceva holistic este împărțit în părți, de exemplu: [recunoașterea scrisului de mână](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%80%D1%83%D0%BA%D0%BE%D0%BF%D0%B8%D1%81%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B2%D0%B2%D0%BE%D0%B4%D0%B0) sau [recunoașterea vorbirii](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%80%D0%B5%D1%87%D0%B8) . Multe soluții arhitecturale diferite au fost propuse pentru rețelele recurente, de la simple la complexe. Recent, cele mai răspândite sunt [rețeaua cu memorie pe termen lung și pe termen scurt](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BB%D0%B3%D0%B0%D1%8F_%D0%BA%D1%80%D0%B0%D1%82%D0%BA%D0%BE%D1%81%D1%80%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C) (LSTM) și[unitatea recurentă controlată](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D1%80%D0%B5%D0%BA%D1%83%D1%80%D1%80%D0%B5%D0%BD%D1%82%D0%BD%D1%8B%D0%B9_%D0%B1%D0%BB%D0%BE%D0%BA) (GRU).

Rețelele de memorie pe termen scurt - denumite de obicei „LSTM” - sunt un tip special de RNN, capabil să învețe dependențe pe termen lung. Au fost introduse de Hochreiter & Schmidhuber (1997) și au fost rafinate și popularizate de mulți oameni în urma lucrărilor. 1 Lucrează extrem de bine la o mare varietate de probleme și sunt acum utilizate pe scară largă.

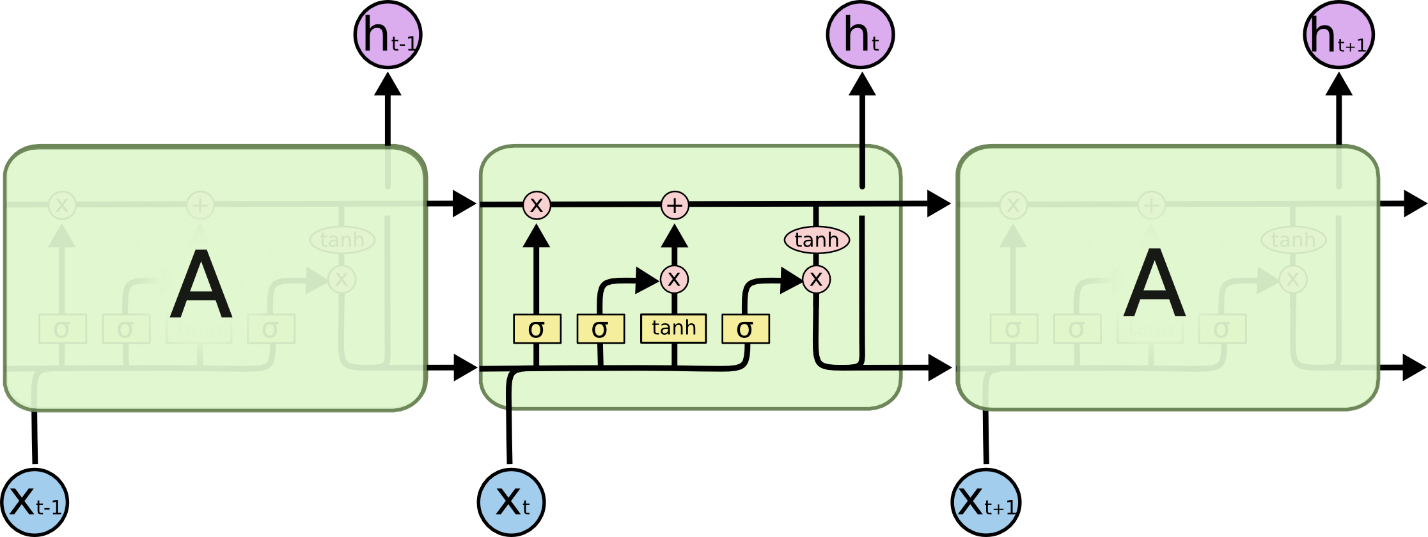
LSTM-urile sunt concepute în mod explicit pentru a evita problema dependenței pe termen lung. Amintirea informațiilor pentru perioade lungi de timp este practic comportamentul lor implicit, nu ceva care se luptă să învețe!

Toate rețelele neuronale recurente au forma unui lanț de module repetate ale rețelei neuronale. În RNN-urile standard, acest modul care se repetă va avea o structură foarte simplă, cum ar fi un singur strat tanh.



Modulul care se repetă într-un RNN standard conține un singur strat.

LSTM-urile au, de asemenea, această structură asemănătoare lanțului, dar modulul care repetă are o structură diferită. În loc să existe un singur strat de rețea neuronală, există patru, care interacționează într-un mod foarte special.



O rețea neuronală LSTM.

Modulul care se repetă într-un LSTM conține patru straturi interacționale.



Nu vă faceți griji cu privire la detaliile a ceea ce se întâmplă. Vom parcurge diagrama LSTM pas cu pas mai târziu. Deocamdată, să încercăm să ne simțim confortabili cu notația pe care o vom folosi.

În diagrama de mai sus, fiecare linie poartă un întreg vector, de la ieșirea unui nod până la intrările altora. Cercurile roz reprezintă operații punctiforme, cum ar fi adăugarea vectorului, în timp ce casetele galbene sunt straturi de rețea neuronală învățate. Liniile care fuzionează denotă concatenare, în timp ce o linie de bifurcare denotă conținutul său fiind copiat, iar copiile mergând în locații diferite.

Ideea de bază din spatele LSTM-urilor

Cheia LSTM-urilor este starea celulei, linia orizontală trecând prin partea de sus a diagramei.

Starea celulei este cam ca o bandă transportoare. Se desfășoară direct pe întregul lanț, cu doar câteva interacțiuni liniare minore. Este foarte ușor ca informațiile să curgă de-a lungul acestora neschimbate.



LSTM are capacitatea de a elimina sau adăuga informații la starea celulei, reglementate cu grijă de structuri numite porți.

Porțile sunt o modalitate de a lăsa opțional informațiile să treacă. Acestea sunt compuse dintr-un strat de rețea neuronală sigmoidă și o operație de multiplicare punctuală.

Stratul sigmoid generează numere între zero și unu, descriind cât din fiecare componentă ar trebui să fie lăsat să treacă. O valoare zero înseamnă „nu lăsa nimic să treacă”, în timp ce valoarea unuia înseamnă „lăsați totul să treacă!”



Un LSTM are trei dintre aceste porți, pentru a proteja și controla starea celulei.

Exemplu de utilizare a rețelei LTSM

<https://github.com/nlintz/TensorFlow-Tutorials/blob/master/07_lstm.ipynb>

public class Recurent {

public static void main(String []args){

int n = 20;

double input[] = {1,1,0,0,0,1,1,1,1,1,0,0,1,0,1,1,1,0,0,1};

double x1 , w1 = 0.3 ,w2=0.5, h = 0,y;

for(int i=0;i<n;i++)

{

x1 = input[i];

h = Math.tanh(h\*w2+x1\*w1);

if(h>0.4)

{

y = 1;

} else

{

y=0;

}

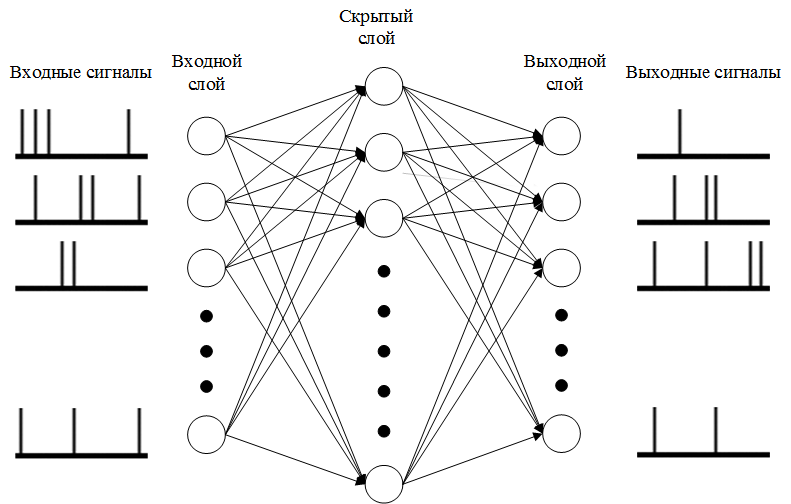
System.out.println("y["+i+"]="+y);

}

}

}

**Rețea neuronală pulsată** (IMN, [eng.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Rețele neuronale pulsate* , PNN) sau **Rețea neuronală Spike** ( SNN , [ing.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Rețea neuronală Spiking* , SNN) - a treia generație de [rețele neuronale artificiale](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) (ANN) , care diferă de cea binară (prima generație) și frecvență / viteză mare (a doua generație) ANN în care [neuronii](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD) din ea schimbă [impulsuri](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BC%D0%BF%D1%83%D0%BB%D1%8C%D1%81) scurte (pentru neuroni biologici - aproximativ 1-2 ms) [de](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BC%D0%BF%D1%83%D0%BB%D1%8C%D1%81" \o "Impuls electric) aceeași [amplitudine](https://ru.wikipedia.org/wiki/%D0%90%D0%BC%D0%BF%D0%BB%D0%B8%D1%82%D1%83%D0%B4%D0%B0) (pentru neuroni biologici - aproximativ 100 mV). Este cel mai realist, în ceea ce privește fiziologia, modelul ANN  .



### Cum funcționează

Rețeaua primește o serie de impulsuri la intrări și emite impulsuri la ieșire. În fiecare moment, fiecare [neuron](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD) are o anumită valoare (analogă cu [potențialul electric](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BF%D0%BE%D1%82%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB" \o "Potential electric) al neuronilor biologici) și, dacă această valoare depășește pragul, atunci neuronul trimite un singur [impuls](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB_%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D1%8F" \o "Potențial de acțiune) , după care valoarea sa proprie scade la un nivel sub valoarea medie (analogul procesului de reabilitare în neuroni biologici, deci numită [perioadă refractară](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D1%80%D0%B0%D0%BA%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D0%BF%D0%B5%D1%80%D0%B8%D0%BE%D0%B4" \o "Perioada refractară) ) cu 2-30 ms. Când este dezechilibrat, potențialul neuronului începe să tindă ușor la valoarea medie. Există doar doi parametri ai legăturilor de greutate ale neuronului de impuls - timpul de întârziere și valoarea greutății .

### Prezentarea informațiilor

În ANN-urile cu frecvență, se utilizează un semnal care preia o valoare care depinde de frecvența de generare a impulsurilor de către un anumit grup de neuroni (greutățile neuronilor, de fapt, sunt o formă de reprezentare a acestei frecvențe)  . Cu toate acestea, frecvența medie a impulsurilor într-o secvență este o opțiune destul de slabă pentru prezentarea informațiilor, deoarece diferite tipuri de stimulare pot duce la aceeași frecvență medie a impulsurilor  .

Pentru a scăpa de acest dezavantaj în ANN-urile cu impuls, sunt utilizate următoarele tipuri de prezentare a informațiilor  :

1. fază (timp) - informațiile despre semnal sunt stabilite de poziția exactă (sau într-o anumită fereastră) a impulsurilor în timp (în raport cu un anumit [ritm de](https://ru.wikipedia.org/wiki/%D0%A0%D0%B8%D1%82%D0%BC%D1%8B_%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BC%D0%BE%D0%B7%D0%B3%D0%B0" \o "Ritmuri cerebrale) referință general [al creierului](https://ru.wikipedia.org/wiki/%D0%A0%D0%B8%D1%82%D0%BC%D1%8B_%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BC%D0%BE%D0%B7%D0%B3%D0%B0" \o "Ritmuri cerebrale) );
2. sincron (pozițional / spațial / populație) - informațiile despre semnal sunt setate de activitatea sincronă a diferitelor grupuri de neuroni și, în consecință, apariția sincronă (sau într-o anumită fereastră) a impulsurilor la anumite ieșiri de rețea (de exemplu, receptorii auditivi ai [cohleei](https://ru.wikipedia.org/wiki/%D0%A3%D0%BB%D0%B8%D1%82%D0%BA%D0%B0_(%D0%B0%D0%BD%D0%B0%D1%82%D0%BE%D0%BC%D0%B8%D1%8F)" \o "Melc (anatomie)) care răspund la frecvențe înalte și joase) [urechea](https://ru.wikipedia.org/wiki/%D0%A3%D0%BB%D0%B8%D1%82%D0%BA%D0%B0_(%D0%B0%D0%BD%D0%B0%D1%82%D0%BE%D0%BC%D0%B8%D1%8F)" \o "Melc (anatomie)) se află în zone diferite);
3. timpul înainte de apariția primului impuls - informațiile despre semnal sunt setate de momentul apariției primului impuls la orice ieșire;
4. ordinal - informațiile despre semnal sunt setate prin ordinea primirii impulsurilor la ieșirile de rețea;
5. interval (întârziat) - informațiile despre semnal sunt setate de distanța dintre impulsurile primite la ieșirile de rețea;
6. rezonant - informațiile despre semnal sunt setate de o secvență densă de impulsuri (rafală), ducând la apariția rezonanței (impulsurile simple se degradează și nu contribuie la transmiterea informațiilor).

În plus, există tipuri de prezentare a informațiilor, care sunt o formă mixtă a mai multor tipuri simple de prezentare a informațiilor, de exemplu:

1. spațio-temporal - informațiile sunt setate nu numai de o anumită succesiune de impulsuri în timp, ci trebuie să provină și dintr-un anumit grup de neuroni;
2. populația-frecvență - informațiile sunt stabilite prin creșterea frecvenței de generare a impulsurilor de către un anumit grup de neuroni.

Exemplu de utilizare a Spiking rețea

<https://github.com/guillaume-chevalier/Spiking-Neural-Network-SNN-with-PyTorch-where-Backpropagation-engenders-STDP/blob/master/Spiking%20Neural%20Networks%20with%20PyTorch.ipynb>

public class Spiking {

public static void main(String[] args) {

int n=10;

double x[] = new double[n];

System.out.println("\nValorile lui x in timp\n");

for(int i=0;i<n;i++)

{

x[i] = (Math.random()%100)/100.0; // [0..1.0];

System.out.println(x[i]);

}

System.out.println("\nLIF feedback:\n");

double suma = 0;

double prag = 2.6;

for( int i = 0;i<n;i++)

{

suma+=x[i];

if(suma > prag)

{

System.out.println("Fire at "+Integer.toString(i)+" - "+Double.toString(suma-prag));

suma = 0.0f;

}

}

}

}

**Bibliografie:**

1. L. G. Komartsova, A. V. Maksimov "Neurocomputatoare", MSTU im. N. Bauman, 2004, [ISBN 5-7038-2554-7](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/5703825547)
2. *Wulfram Gerstner.*Spiking Neurons // [Rețele neuronale pulsate](https://books.google.com/books?id=jEug7sJXP2MC&pg=PA3&dq=%22Pulsed+Neural+Networks%22+rate-code+neuroscience&ei=FEo0ScetL4zukgSyldy8Ag) (nespecificate) / Wolfgang Maass; Christopher M. Bishop. - [MIT Press](https://ru.wikipedia.org/wiki/MIT_Press) , 2001. - [ISBN 0-262-63221-7](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/0262632217) .
3. <https://mcneela.github.io/machine_learning/2017/07/07/McCulloch-Pitts.html>
4. *Graves, A.; Liwicki, M.; Fernandez, S.; Bertolami, R.; Bunke, H.;*[*Schmidhuber, J.*](https://en.wikipedia.org/wiki/J%C3%BCrgen_Schmidhuber)*(engleză)*[*rusă.*](https://ru.wikipedia.org/w/index.php?title=Schmidhuber,_J.&action=edit&redlink=1)*...* [Connectionist Novel System A pentru îmbunătățirea recunoașterii scrierii de mână fără restricții](http://www.idsia.ch/~juergen/tpami_2008.pdf) (eng.)  // [IEEE Transactions on Pattern Analysis and Machine](https://en.wikipedia.org/wiki/IEEE_Transactions_on_Pattern_Analysis_and_Machine_Intelligence)
5. <https://github.com/nlintz/TensorFlow-Tutorials/blob/master/07_lstm.ipynb>
6. <https://github.com/guillaume-chevalier/Spiking-Neural-Network-SNN-with-PyTorch-where-Backpropagation-engenders-STDP/blob/master/Spiking%20Neural%20Networks%20with%20PyTorch.ipynb>