

Spiking Neural Networks for Classification

Formal Methods in Software Engineering

Craciun Florin, Dan Alexandru, Dumitrescu
Nicolle, Vizinteanu Teodora

Ce este un ANN? (Artificial Neural Network)



Artificial Neural Networks

- Rețele de noduri, reprezentând neuronii din creierul nostru, conectate prin muchii – sinapsele din creierul nostru
- Neuronii primesc datele de input care sunt folosite ca parametrii unei funcții unde sunt luate în calcul și ponderile
- Sinapsele sunt muchiile dintre neuroni, fiecare având o pondere care influențează următorul nod

The background is a solid orange color. In the top-left corner, there are three vertical bars of varying heights, each composed of three overlapping circles. In the bottom-right corner, there are four vertical bars of increasing height, each also composed of three overlapping circles.

Ce este un SNN? (Spiking Neural Networks)



Ce este un SNN? (Spiking Neural Network)

- **Definiție:**

Un SNN este un tip de rețea neuronală artificială inspirată de funcționarea creierului biologic. Informația este transmisă prin **impulsuri electrice discrete** numite *spikes*, nu prin valori continue.

- **Caracteristici cheie:**

- Neuronii trimit semnale doar când o anumită valoare de prag este atinsă.
- Timpul dintre impulsuri este esențial pentru codificarea informației.
- Simulează mai realist comportamentul creierului uman.



Ce este un SNN? (Spiking Neural Network)

- **Diferența față de rețelele tradiționale (ANN):**
 - ANN: Semnal continuu (valori reale)
 - SNN: Semnal discret în timp (spikes \rightarrow 0 sau 1, cu semnificație temporală)
- **Relevanță:**
 - Mai eficient energetic
 - Permite procesare asincronă, bazată pe evenimente
 - Ideal pentru hardware neuromorfic și aplicații în timp real

Limitari ale SNN-urilor

- Complexitatea antrenarii din cauza sincronizarii spike-urilor si modificarea ponderilor
- Necesita putere mare de calcul, comparativ cu celelalte tipuri de retele neuronale
- Ajustarea valorilor parametrilor pot schimba foarte usor performanta modelului



Diferența față de rețelele tradiționale (ANN):

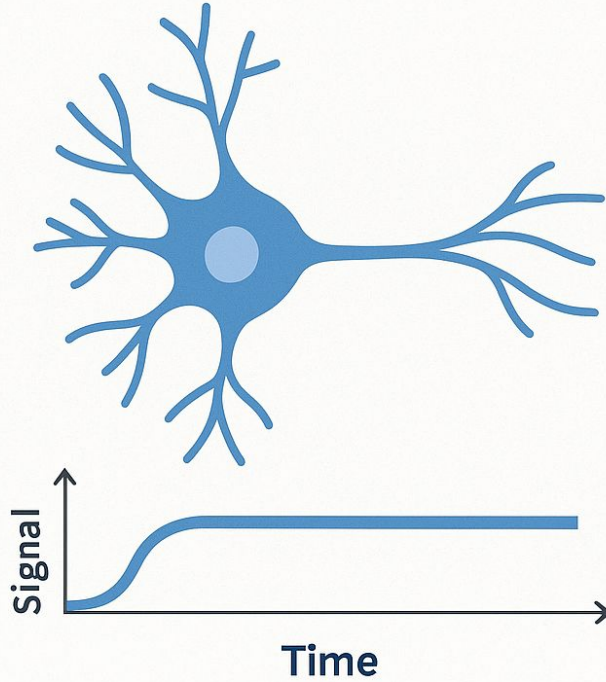
Neuron ANN (Artificial Neural Network)

- **Neuron cu ramificații:** Reprezintă un neuron artificial tradițional, inspirat de cel biologic, care procesează informația în mod continuu.
- **Săgeată orizontală:** Indică direcția fluxului de informație – *Intrare* → *Neuron* → *Ieșire*.
- **Graficul de jos (semnal continuu):**
 - Axă orizontală: Timp
 - Axă verticală: Nivelul semnalului
 - Linia netedă arată cum ieșirea neuronului ANN este o valoare continuă (ex: între 0 și 1), calculată la fiecare pas.

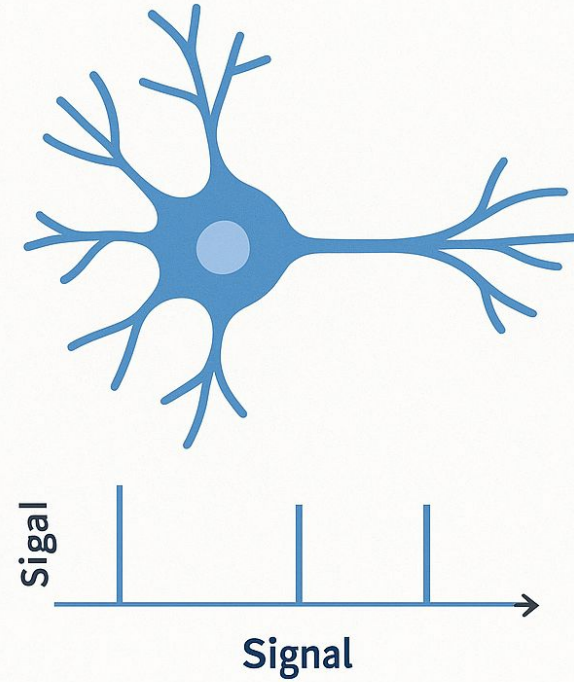
Dreapta – Neuron SNN (Spiking Neural Network)

- **Neuron cu ramificații:** Reprezintă un neuron SNN, care funcționează mai asemănător cu un neuron biologic.
- **Săgeată orizontală:** Direcția semnalului – *Intrare* → *Neuron* → *Ieșire*, dar semnalul este transmis doar când se produce un „spike”.
- **Graficul de jos (spikes):**
 - Axă orizontală: Timp
 - Axă verticală: Prezența unui impuls
 - Liniile verticale indică momentele în care neuronul a „emis” un spike (semnal 1), iar în rest e tăcere (semnal 0).
 - Codificarea informației este bazată pe **momentul apariției** impulsurilor, nu pe o valoare continuă.

ANN Neuron



SNN Neuron





Ce este membrana neuronală (în SNN)?



Ce este membrana neuronală (în SNN)?

- **Definiție:**
 - Membrana neuronală este *componenta care separă interiorul neuronului de exterior* și joacă un rol esențial în generarea impulsurilor electrice (*spikes*).
- **Rol în SNN (Spiking Neural Networks):**
 - Funcționează ca un **filtru pasiv**, acumulând potențialul de membrană.
 - Când potențialul atinge un **prag de activare**, neuronul generează un spike.
 - După un spike, membrana trece printr-o perioadă de **refracție** în care nu poate răspunde imediat la noi semnale.
- **Model matematic simplificat (LIF – Leaky Integrate-and-Fire):**
 - Se acumulează semnale electrice în timp.
 - Se „scurg” parțial (leaky), simulând pierderile reale.
 - Pragul este atins \Rightarrow apare spike \Rightarrow resetare.



Ce este membrana neuronală (în SNN)?

Modelul LIF – Leaky Integrate-and-Fire

- **Ce este LIF?**
LIF este un model simplificat, dar eficient, folosit pentru a simula comportamentul unui neuron biologic într-o rețea SNN. Acesta descrie **modul în care neuronul integrează semnale și decide când să emită un spike**.

Componentele modelului LIF:

1. Integrate

(Integrare):

- Neuronul **primește impulsuri de la alți neuroni** (semnale de intrare).
- Acestea sunt acumulate sub formă de **potențial de membrană**.
- Creșterea potențialului este influențată de:
 - Intensitatea semnalelor primite și frecvența cu care sosesc

2. Leaky (Scurgere):

- Potențialul nu se păstrează la infinit.
- O parte din semnal **se pierde treptat în timp** (decădere exponențială).
- Asta simulează pierderile naturale de tensiune din biologie.

3. Fire (Declanșare):

- Când potențialul acumulat depășește un **prag de activare**, neuronul **emite un spike**.
- După spike, potențialul se **resetează la o valoare inițială**.
- Urmează o **perioadă refractară** în care neuronul nu poate „trage” din nou imediat.



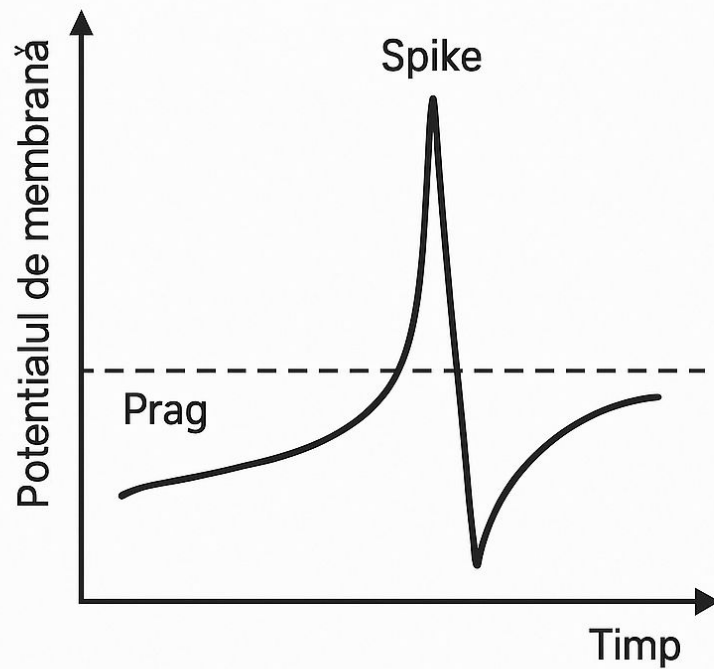
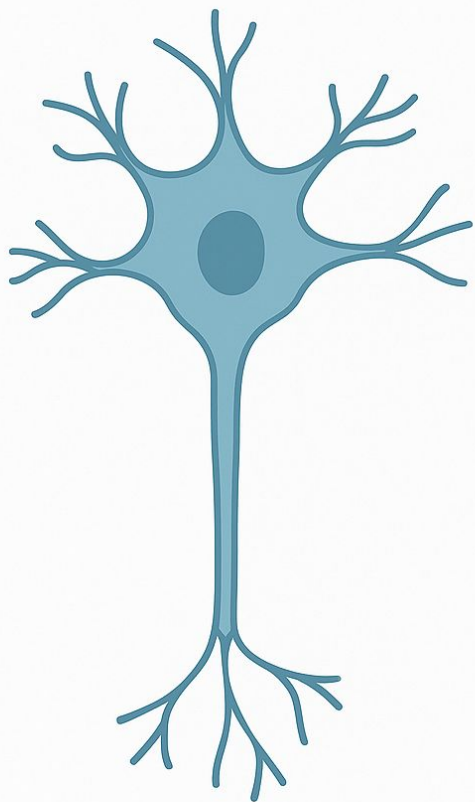
Ce este membrana neuronală (în SNN)?

Neuron ilustrat

- Un neuron biologic/simulat în SNN.
- Semnalul electric se propagă prin membrana acestuia.
- Rolul membranei este de a **acumula și controla** potențialul electric interior.

Grafic: Potențialul de membrană în timp

- **Axa verticală:** *Potențialul de membrană* – măsoară „tensiunea” electrică internă.
- **Axa orizontală:** *Timpul* – evoluția potențialului în timp.
- **Curba neagră:**
 - Reprezintă **acumularea semnalelor de intrare**.
 - Când linia atinge **linia punctată** (pragul de activare), neuronul declanșează un **spike** (impuls).
- **Spike:**
 - Reprezentat ca o creștere bruscă a curbei.
 - După spike, urmează o scădere (resetare), care simulează **perioada refractară** a neuronului.
- **Linia punctată:**
Pragul de activare – dacă potențialul ajunge aici, neuronul „focusează” (spikează).



Antrenarea unui SNN



- Se parcurg mai multe epoci de antrenare
- Se face rate encoding pe imagini adica le transformam in impulsuri
- Se calculeaza pierderea, adica diferenta dintre predictii si etichetele reale
- Folosim algoritmul de backpropagation pentru a ajusta ponderile

```
1  for epoch in range(num_epochs):
2      net.train()
3      running_loss = 0
4      correct = 0
5      total = 0
6
7      for batch_idx, (images, labels) in enumerate(train_loader):
8          images = images.to(device)
9          labels = labels.to(device)
10
11         images = images.view(batch_size, -1)
12         images = rate_encode(images, time_steps)
13
14         optimizer.zero_grad()
15         outputs = net(images)
16
17         out_sum = outputs.sum(0)
18         loss = loss_fn(out_sum, labels)
19         loss.backward()
20         optimizer.step()
21
22         running_loss += loss.item()
23         _, predicted = out_sum.max(1)
24         total += labels.size(0)
25         correct += predicted.eq(labels).sum().item()
26
```