

Retele neuronale artificiale

Istoric

Ideea de a căuta în structura sistemului nervos o sursă de inspirație pentru concepție sistemelor de tratare a informației nu este nouă. În anul 1943, în încercarea de a explica modul în care creierul uman realizează funcții booleene, W. McCulloch și S. Pitts, (McCulloch 43) au propus modelarea celulei nervoase (neuronul), printr-un automat cu prag denumit neuron formal. Acest model este, în linii generale, acceptat și astăzi.

Progresele înregistrate în neurologie și psihologie au determinat apariția unor modele matematice ale învățării. Un astfel de model a fost propus în 1949 de către D.O.Hebb. Modelul Hebb a reprezentat punctul de plecare în tentativele întreprinse în deceniul următor de realizare a unor rețele neuronale artificiale capabile să învețe.

În 1958 Frank Rosenblatt a publicat prima sa carte despre perceptron. Acesta era un model probabilistic pentru memorarea și organizarea informației în creier. Un alt model de rețea neurală, cunoscut ca ADALINE (Adaptive Linear Neuron) a fost propus la începutul anilor 60 de către Bernard Widrow. Rețelele realizate în această perioadă au fost aplicate pentru rezolvarea unor probleme cum ar fi recunoașterea unor structuri specifice în electrocardiograme sau percepția artificială. Astăzi se poate spune că acestea sunt aplicații de recunoaștere a formelor.

Primele succese ale rețelelor neurale au determinat o mare emoție în lumea științifică. Specialiștii erau aproape unanimi în a considera că și alte procese asociate cu inteligența și memoria umană pot fi modelate prin astfel de rețele eventual mai complicate.

Marvin Minsky a întreprins o analiză lucidă a posibilităților și limitelor modelelor neurale existente în epocă. El a demonstrat imposibilitatea principală a rețelelor neuronale cu un singur strat de a rezolva unele probleme simple. De exemplu, funcția logică SAU EXCLUSIV (XOR) nu poate fi calculată pe o astfel de rețea.

A urmat apoi un deceniu de stagnare, în care puținii fideli ai domeniului au fost lipsiți de fondurile necesare unor cercetări de anvergură.

Apariția în 1986 a cărții *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, de David Rumelhart, James McClelland și grupul PDP se consideră a fi evenimentul care a marcat relansarea cercetărilor privind modelele conexioniste. Interesul crescând a condus la organizarea în 1987 a primei conferințe internaționale de rețele neurale. (Mai multe informații despre istoricul și evoluția RNA sunt prezentate în “Momente importante în apariția rețelelor neurale”).

Calculul neuronal reprezintă acum un domeniu de cercetare fascinant și o provocare intelectuală și tehnologică majoră. Rețelele neurale au modificat imaginea noastră asupra proceselor de calcul și aspectelor algoritmice ale Inteligenței Artificiale (IA) și au furnizat psihologiei un nou model al proceselor mentale.

Prin aplicațiile în rezolvarea unor probleme dificile, cum sunt cele de estimare, identificare și predicție, a unor probleme de optimizare complexe, rețelele neuronale capătă o pondere și un impact tot mai mare, nu doar în știință și tehnologie, ci și în domeniul mult mai sensibil al vieții sociale. Rețelele neuronale pot reprezenta instrumente puternice în luarea deciziilor economice și chiar politice.

Teoria mulțimilor nuanțate (fuzzy), al cărui început poate fi situat în 1965 prin lucrările lui L.A. Zadeh, reprezintă un instrument matematic foarte puternic pentru tratarea ambiguității de natură nestică și pentru modelarea raționamentului nuanțat și aproximativ. Aceasta teorie a furnizat calculului neuronal o serie de modele foarte interesante. Utilizând mulțimile nuanțate, se pot realiza rețele neuronale capabile să învețe din date de instruire incomplete, ambigue sau contradictorii. Este previzibilă o creștere a rolului modelelor de învățare bazate pe utilizarea conceptului mulțimilor nuanțate și pe raționamentul aproximativ. Rețelele neuronale bazate pe astfel de modele vor avea un grad sporit de robustețe și, desigur, un grad de inteligență sporit.

Calculul neural reprezintă acum un domeniu de cercetare fascinant și o provocare intelectuală și tehnologică majoră. Rețelele neurale au modificat imaginea noastră asupra proceselor

de calcul și aspectelor algoritmice ale Inteligentei Artificiale și au furnizat psihologiei un model al proceselor mentale.

Prin aplicațiile în rezolvarea unor probleme dificile, cum sunt cele de estimare, identificare, de optimizare a unor probleme complexe, rețelele neurale capătă o pondere și un impact tot mai mare, nu doar în știință și tehnologie, ci și în domeniul mult sensibil al vieții sociale. Rețelele neurale pot reprezenta instrumente puternice în luarea deciziilor economice și chiar politice.

Dar, în încheiere, trebuie menționat că, până în prezent, tehnicile neuronale nu sunt complet stăpânite din punct de vedere matematic, ceea ce face dificilă cuantificarea tuturor avantajelor pe care le poate prezenta o asemenea abordare.

Prezentare generală în analogie cu creierul uman

Rețelele neurale au prezentat în ultimii ani o explozie de interes, și sunt aplicate cu succes într-o foarte largă sferă de domenii, în arii diverse, cum ar fi finanțele, medicina, ingineria, geologia și fizica. Într-adevăr, oriunde sunt probleme de predicție, clasificare sau control, rețelele neurale au fost introduse cu succes.

O Rețea Neurală Artificială (RNA) (ANN - Artificial Neural Network) este, în principiu, un sistem modelat după creierul uman, deși câteva rețele neurale nu sunt chiar așa de apropiate de structura creierului, iar altele nu au la mijloc o bază biologică a creierului. Oricum, majoritatea rețelelor neurale, au o mare similaritate cu creierul biologic. Rețelele neurale s-au dezvoltat în cadrul domeniului Inteligentei Artificiale; în special, atacând imitativ toleranța la greșeli și capacitatea de a învăța a sistemelor neurale biologice, modelând la un nivel foarte jos structura creierului (Pettersen, 1996).

Analogie cu neuronul biologic

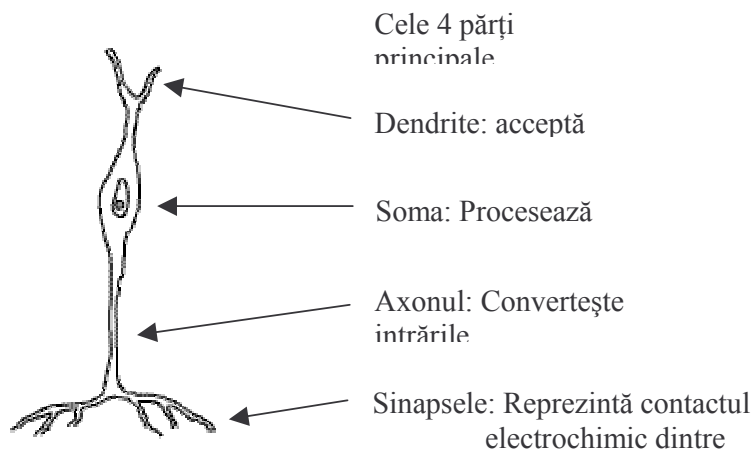
Încă de la începutul abordării domeniului, s-a simțit nevoia unui model care să capteze trăsăturile principale ale unui sistem neuronal real, în scopul de a prezenta comportări similare. Modelul trebuie să ignore deliberat aspectele secundare, adică să fie suficient de simplu pentru a putea fi implementat.

Elementul ce stă la baza creierului uman, este un anumit tip de celulă, care ne permite să ne aducem aminte, să gândim, și să putem aplica informații din anumite experiențe anterioare fiecărei acțiuni pe care încercăm să o îndeplinim. Aceste celule sunt cunoscute cu numele de neuroni, fiecare dintre neuroni, putându-se conecta cu mai mult de 200.000 de astfel de neuroni. Puterea creierului, provine exact din acest fapt, adică din cauza numărului mare de neuroni, și a multiplelor conexiuni dintre ei. Creierul este alcătuit în principal dintr-un număr foarte mare (circa 10.000.000.000) de neuroni, masiv interconectați (cu o medie de câteva mii de interconexiuni pentru fiecare neuron în parte, deși acest lucru variază enorm).

Fiecare neuron este o celulă specializată care poate propaga un semnal electrochimic. Un neuron prezintă o structură de intrare ramificată (dendritele), un corp al celulei (soma), și o structură de ieșire ramificată (axonul). Terminațiile axonului unei celule sunt conectate cu dendritele unui alt neuron, printr-o sinapsă. Toți neuronii naturali, prezintă 4 componente de bază, care sunt: dendritele, soma, axonul și sinapsele. Pentru început, neuronul biologic primește informații de intrare, de la alte surse, le combină într-un anumit mod, realizează în general o operație neliniară asupra rezultatului obținut înainte, și furnizează o ieșire care este de fapt rezultatul final. Figura de mai jos ne prezintă un neuron biologic simplificat și relațiile dintre cele 4 componente ale sale.

Când un neuron este activat, el generează prin axon un semnal electrochimic. Acest semnal parcurge sinapsele altor neuroni, care la rândul lor vor putea avea posibilitatea să se activeze. Un neuron se activează numai în cazul în care semnalul receptat de corpul celulei, prin dendrite, depășește un anumit nivel (prag de activare).

Puterea semnalului primit de neuron (și prin urmare urmând posibilitatea activării) depinde foarte mult de eficacitatea sinapselor. Fiecare sinapsă, conține o deschizătură, cu un neuro-transmițător chimic care poate transmite un semnal prin respectiva deschizătură.



Unul dintre cei mai influenți cercetători în sistemele neurologice (Donald Hebb) a revendicat că învățarea constă, în principal, în a alterna "tăria" conexiunilor sinaptice. În concluzie, dintr-un foarte mare număr de elemente simple de procesare (fiecare creând o pondere din suma intrărilor sale, și apoi trimițând un semnal dacă toate intrările trec peste un anumit prag), creierul poate executa acțiuni foarte complexe. Desigur, o mare importanță o are complexitatea creierului, care nu a fost discutată aici, dar este interesant cum aceste rețele neurale artificiale pot achiziționa asemenea rezultate remarcabile utilizând un model nu chiar foarte complex în comparație cu creierul.

Neuronul artificial, elementul de bază al rețelelor neurale, simulează cele 4 funcții de bază ale neuronului natural.

Trebuie precizat că sistemul neuronal biologic nu se naște cu toate cunoștințele și abilitățile pe care le va putea, eventual, dobândi ulterior. În timp, are loc un proces de învățare, în cursul căruia se produc modificări atât în structura rețelei neuronale cât și în conținutul ei, pentru a încorpora noi informații.

Modelul neuronului artificial

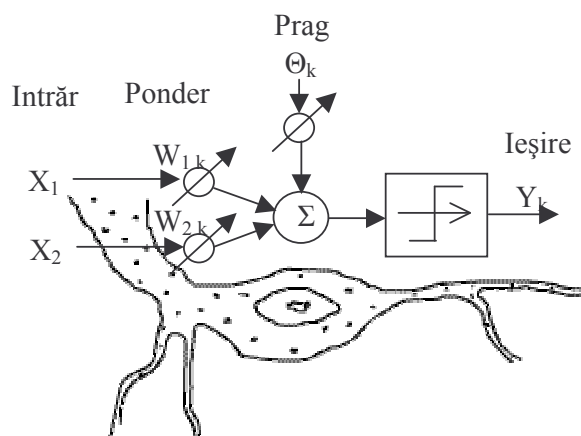
Neuronul artificial

Primul model formal al neuronului, privit ca element de procesare a informației a fost propus de W. McCulloch și S. Pitts în anul 1943, (McCulloch 43). Acest model pornește de la ipotezele următoare:

- Activitatea neuronului este de tipul "totul sau nimic", adică ieșirea acestuia poate fi zero sau unu.
- Pentru activarea unui neuron este necesar ca la un moment dat cel puțin una din sinapsele (legăturile) sale să fie excitată.
- Singura întârziere în sistemul nervos este cea sinaptică (ceea ce permite modelarea în timp discret, cu o unitate de timp echivalentă cu timpul modificării valorii unei sinapse).
- Activitatea oricărei conexiuni inhibitoare determină trecerea în starea inactivă a neuronului receptor (ieșire egală cu zero).
- Structura rețelei (configurația conexiunilor dintre neuroni) este invariantă în timp.

Neuronul artificial, elementul de bază al rețelelor neurale, simulează cele 4 funcții de bază ale neuronului natural. Neuronii artificiali sunt mult mai simpli decât neuronii biologici; figura de mai jos prezintă componentele de bază a unui neuron artificial.

De urmărit este faptul că intrările sunt reprezentate cu x_{ik} . Fiecare dintre aceste intrări sunt multiplicat de câte o pondere, corespunzătoare unei conexiuni, reprezentate cu w_{ik} . În cel mai simplu caz, acestea sunt însumate, și trecute printr-o funcție de transfer pentru a se genera un rezultat, care apoi va fi furnizat ieșirilor.



Pentru a putea prinde esența sistemelor neurale biologice, un neuron artificial este definit, după cum urmează:

- el primește un număr de intrări (de la datele de intrare originale, sau de la ieșirile altor neuroni din rețeaua neurală). Fiecare intrare vine printr-o conexiune ce a fost întărită (sau slăbită); aceste ponderi corespund unei activități sinaptice din neuronul biologic. Fiecare neuron are propriul și singurul său prag. Suma ponderilor intrărilor este formată, iar pragul este scăzut, pentru a se compune activarea neuronului (cunoscută și sub denumirea de Potențial Post-Sinaptic, sau PSP - Post-Synaptic Potential al neuronului).
- semnalul de activare este trecut printr-o funcție de activare (cunoscută și sub numele de funcție de transfer) pentru a produce ieșirea unui neuron.

Dacă se utilizează un pas pentru funcția de activare (ex: ieșirea neuronului este 0, dacă intrarea este mai mică ca 0, și 1, dacă intrarea este mai mare sau egală cu 0) atunci neuronul acționează ca un neuron biologic, descris anterior (scăzând pragul din suma ponderilor și comparând cu 0, este echivalent în a compara suma ponderilor cu pragul). Momentan, pasul funcției este utilizat rar în rețelele neurale artificiale. De notat este faptul că ponderile pot fi negative, ceea ce implică faptul că sinapsa are un efect mai degrabă inhibitoriu decât excitatoriu pentru neuron: neuronii inhibitori se află în creier.

Aceasta descrie fiecare neuron individual. Următoarea întrebare este: cum trebuie să fie conectați neuronii? Dacă rețelei nu i se cunoaște viitoarea utilizare, ea trebuie să conțină intrări (care vor primi valorile variabilelor care ne interesează) și ieșiri (cum ar fi semnale de control sau predicție). Intrările și ieșirile corespund, de exemplu, unor senzori sau unor nervi ai motorului, la fel ca cei care vin de la ochi și controlează mâinile. Oricum, pot exista și neuroni ascunși, care vor juca un rol important în rețea. Straturile de neuroni de la intrare, ascunse și de la ieșire trebuie să fie conectate împreună.

Deși majoritatea rețelelor neurale artificiale sunt construite pe aceeași structură de bază, unele părți fundamentale pot să difere foarte mult de la o rețea la alta.

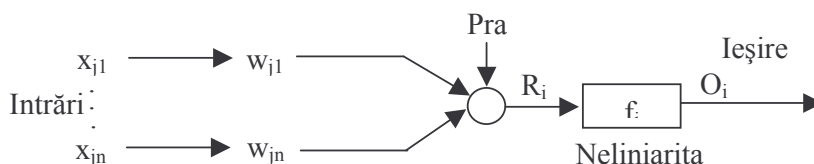
Structura unui neuron artificial

Un neuron are un set de n intrări x_j , unde indicele j ia valori de la 1 la n , indicând sursa semnalului de intrare. Fiecare intrare x_j este ponderată prin intermediul unor ponderi sinaptice, înainte de a intra în corpul principal al elementului de procesare. Această pondere se notează cu w_j , astfel că, o intrare x_j a neuronului va fi multiplicată cu ponderea w_j . Dacă $w_j > 0$, atunci ponderea se numește pondere excitatoare, iar dacă $w_j < 0$, ponderea se numește inhibitorie.

În plus, un neuron are un prag de activare (sau de excitare) precum și o valoare de prag θ care trebuie atinsă sau depășită pentru ca neuronul să producă semnal. De asemenea, neuronul are o funcție de activare f și un semnal de ieșire O (ieșirea funcției neliniare f). Semnalul de ieșire O constituie intrarea către un alt neuron.

Atunci când un neuron face parte dintr-o rețea neuronală, el este referit ca un nod. Pentru m noduri într-o rețea, în descriere se va introduce un nou indice i, necesar pentru a distinge un neuron. Intrările, ponderile, pragurile de activare, funcția neliniară, etc. se vor scrie ca: x_{ij} , w_{ij} , O_{ij} , θ_{ij} , f_{ij} , etc.

Modelul de bază al neuronului este prezentat în figura următoare:



Funcția de transfer a modelului de bază este descrisă de relația:

$$O_i = f_i \left(\sum_{j=1}^n w_{ij} x_{ij} \right)$$

iar condiția de activare este:

$$\sum_{j=1}^n w_{ij} x_{ij} \geq \theta_i$$

unde indexul i reprezintă neuronul în discuție iar j reprezintă intrările de la alți neuroni.

Acest model a fost prezentat în 1943 de către Warren McCulloch și Pitts. Este modelul unui singur neuron idealizat. Este un model foarte simplu, fără învățare și fără adaptare și s-a constituit blocul de bază pentru celelalte modele elaborate ulterior.

Fiecare intrare primește un stimul x_j care este ponderat (înmulțit) cu o valoare w_{ij} . Toate intrările ponderate sunt sumate, și, dacă această sumă egalizează o valoare de activare, este generat un răspuns, prin intermediul unei funcții neliniare f . Ieșirea este exprimată prin:

$$O_i = f \left(\sum_{j=1}^n x_{ij} w_{ij} - \theta_i \right)$$

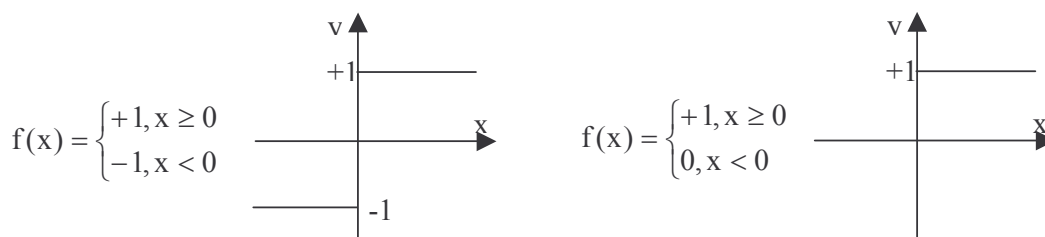
În acest model termenii θ_i și ponderile w_{ij} sunt considerate ca ajunse la valoarea lor finală. Dacă ponderile sunt constante, atunci faza de învățare a luat sfârșit.

Funcția de activare

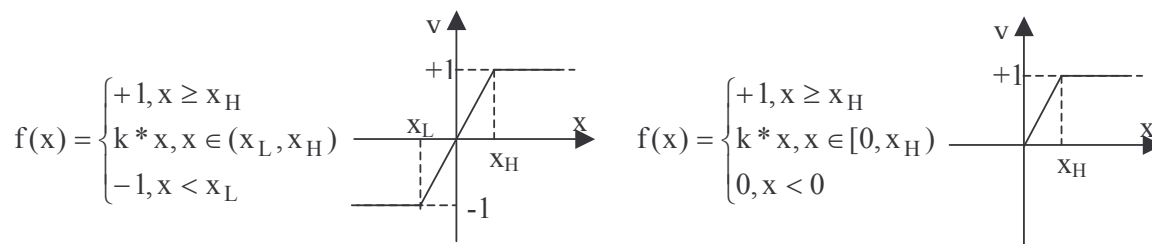
Scopul funcției de activare, numită și neliniaritate, este de a asigura un răspuns mărginit al neuronului. Prin aceasta, răspunsul unui neuron este condiționat, sau mărginit, ceea ce conduce la controlabilitatea neuronului. În lumea biologică condiționarea stimulilor este făcută asupra tuturor senzorilor de pe intrări.

Cele mai utilizate neliniarități sunt funcția treaptă, funcția rampă și funcția sigmoid. Aceste neliniarități sunt prezentate în figurile următoare:

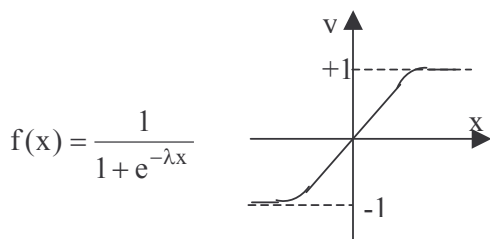
Funcția treaptă:



Funcția rampă:



Funcția sigmoid:



Nelinaritățile, prezentate în figurile de mai sus, sunt toate mărginite, ele având limită inferioară sau superioară la ± 1 , $\pm 1/2$. În rețelele actuale, utilizatorul alege aceste valori.

Cea mai utilizată funcție neliniară este totuși funcția sigmoid.

Rețele neuronale artificiale

Domeniul sistemelor modelate după creierul uman cuprinde mai multe nume, ca de exemplu: conexiionismul, procesarea paralelă distribuită, algoritmi de învățare matematici, sisteme naturale inteligente, neuro-procesarea și rețelele neurale artificiale. Este o încercare de a simula cu un hardware specializat sau cu un software sofisticat, o multitudine de straturi ce cuprind elemente simple, ce se numesc neuroni. Fiecare neuron trebuie să fie conectat cu neuronii vecini, cu un coeficient de conectivitate aleator, care reprezintă "tăria" acestor conexiuni. Învățarea se realizează ajustând aceste trăsături ale conexiunilor, care va cauza la ieșirile rețelei neurale, o serie de rezultate care vor fi în concordanță cu intrările rețelei.

Deci, o Rețea Neurală Artificială este un sistem de procesare a informației, care se bazează pe unele generalizări ale cunoașterii umane sau ale biologiei neurale. Fausett a făcut următoarele legături între cele două domenii:

- procesarea informației se realizează la nivelul unor foarte simple elemente numite neuroni;
- semnalele circulă între neuroni prin conexiunile dintre ei;
- fiecărei conexiuni dintre doi neuroni, îi este asociată o pondere, care într-o rețea neurală tipică multiplică semnalul transmis;
- fiecare neuron, folosește o funcție de activare (de obicei neliniară), asupra intrărilor rețelei (suma ponderilor corespunzătoare semnalelor de intrare), pentru a determina ieșirea rețelei.

Există trei tipuri diferite de neuroni în cadrul unei rețele neurale: neuroni de intrare (input), neuroni de ieșire (output) și în unele rețele neuroni ascunși (hidden).

- Neuronii de intrare primesc informația codată de la exterior.
- Neuronii de ieșire trimit semnale de ieșire către exterior sub formă de răspunsuri codate.
- Neuronii ascunși permit calcule intermediare între intrări și ieșiri.

O rețea neurală (NN - Neural Network) este caracterizată de următoarele elemente particulare:

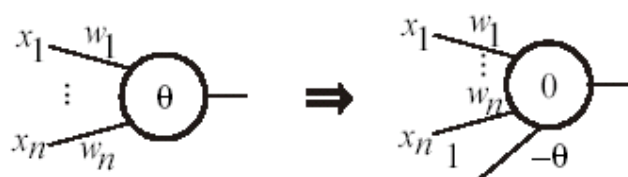
- Arhitectura - reprezintă structura rețelei, conexiunile rețelei;

- Algoritmul de învățare - este o metodă de determinare a ponderilor corespunzătoare fiecărei conexiuni în parte;
- Funcția de activare - reprezintă funcția care ne da ieșirea rețelei.

Retelele neuronale artificiale sunt constituite din numerosi neuroni artificiali interconectați între ei. Neuroni constituenți sunt de obicei organizați în mulțimi distincte numite straturi (layers). Există cazuri de rețele neuronale particulare, având o topologie specială, când neuronii nu sunt organizați pe straturi bine definite, ci pe anumite submulțimi.

O rețea neuronală clasică constă dintr-o secvență de straturi de neuroni, cu conexiuni totale sau aleatoare între neuronii conținuți în straturi succesive. Întotdeauna vom avea pentru o rețea neuronală, cel puțin două straturi de neuroni, ce au rolul de a comunica cu mediul exterior:

- un strat de intrare : care are rolul de a prelua datele de intrare x_i , $i = 0, n$ cu termen bias $x_0 = \pm 1$, sau $i = 1, n$ fără termen bias. Vectorul $x = (x_0, x_1, \dots, x_n)$ sau $x = (x_1, x_2, \dots, x_n)$ va fi numit vector de intrare al rețelei neuronale.



Tratarea pragului ca o pondere și o intrare constantă

- un strat de ieșire: unde se obțin rezultatele y_j , $j = 1, m$ furnizate de rețeaua neuronală, ca răspuns la vectorul de intrare prezentat la stratul de intrare și al propagării acestor date de la stratul de intrare către stratul de ieșire. Vectorul $y = (y_1, y_2, \dots, y_m)$ se numește vector de ieșire al rețelei neuronale.

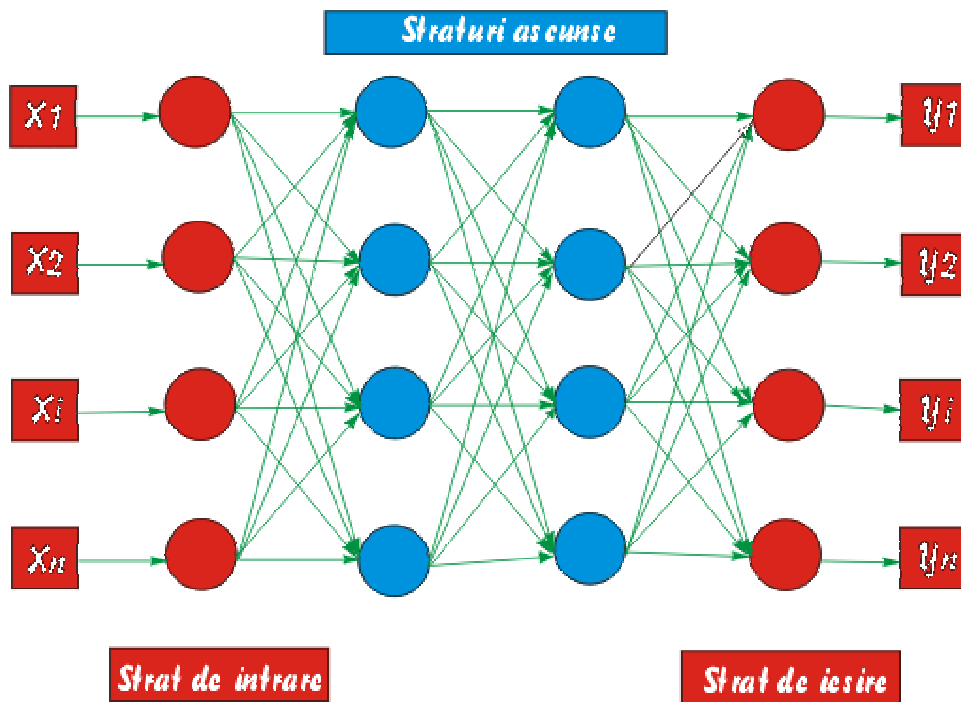


Fig. Arhitectura generală a unei rețele neuronale.

Toate celelalte straturi ale rețelei neuronale, dacă există, diferite de stratul de intrare și de stratul de ieșire vor fi denumite straturi ascunse (hidden layers).

Obs. Când vom spune despre o rețea neuronală ca are n straturi, nu vom lua în calcul și stratul de intrare, ceea ce înseamnă ca vom avea $n - 1$ straturi ascunse + stratul de ieșire. Neuronii din stratul de intrare au întotdeauna ca și funcție de activare funcția liniară, rolul lor fiind doar de-a propaga către stratul succesor vectorul de intrare. ceilalți neuroni, din celelalte straturi, pot avea orice funcții de activare.

Într-o RNA se pot distinge două etape de funcționare: etapa de învățare (antrenare) și etapa de recunoaștere (generalizare). În cazul învățării supervizate, în etapa de antrenare, parametrii rețelei (ponderile conexiunilor sinaptice, w_{ij}) sunt ajustați pe măsura prezentării perechilor de vectori intrare/ieșire. Etapa de recunoaștere constă în exploatarea rețelei, adică pentru date de intrare ce nu figurau în lotul de antrenare, RNA oferă un răspuns la ieșire. Cele două etape de funcționare se întrepătrund adesea.

Există mai multe posibilități de clasificare a rețelelor neuronale, două dintre acestea fiind:

- după sensul de transfer al informației se disting: rețele statice și dinamice.
- după modul de învățare (antrenare), există rețele cu învățare supravegheată și nesupravegheată.

În cazul rețelelor statice, datele primite din exterior tranzitează rețeaua întotdeauna în sens unic, de la intrare către ieșire, de unde și denumirea de rețele feedforward. Rețelele dinamice conțin bucle de reacție, ceea ce determină o evoluție mai complexă în timp.

Referitor la cel de-al doilea criteriu de clasificare, precizăm că învățarea supervizată se desfășoară pe baza unor exemplare care comportă perechi univoce intrare/ieșire, cu ajutorul cărora algoritmi specializați stabilesc măsura influențelor între neuroni. În cazul învățării nesupravegheate are loc, de fapt, un proces de autoorganizare a rețelei, fără a mai fi necesară precizarea unor corelații intrare/ieșire de tipul menționat mai înainte.

Rețelele neurale artificiale sunt, astfel, un instrument de analiză, inspirat din structura paralelă a creierului uman, simulând un model de prelucrare paralel, puternic interconectat, compus din multe elemente de procesare relativ simple. Rețelele neurale artificiale actuale sunt foarte departe de structura neurală biologică și de simularea proceselor cognitive. Diferențele existente se referă atât la arhitectură, cât și la proprietăți, de exemplu:

- rețelele neurale artificiale nu copiază structurile neurale biologice, ci reprezintă modele mult simplificate ale acestora, după cum afirma și T. Sejnowski "nici cele mai realiste modele ale creierului s-ar putea să nu reușească evidențierea funcționării țesutului nervos, studiul trebuie realizat pe modele simplificate, care să conțină trăsături esențiale".
- rețelele neurale sunt realizate de obicei pe baza câtorva tipuri de neuroni artificiali, în timp ce există o mare varietate de neuroni biologici;
- neuronii artificiali sunt mai rapizi decât cei biologici.

Avantaje

Succesul covârșitor al aplicării rețelelor neurale în diverse domenii, poate fi atribuit câtorva factori:

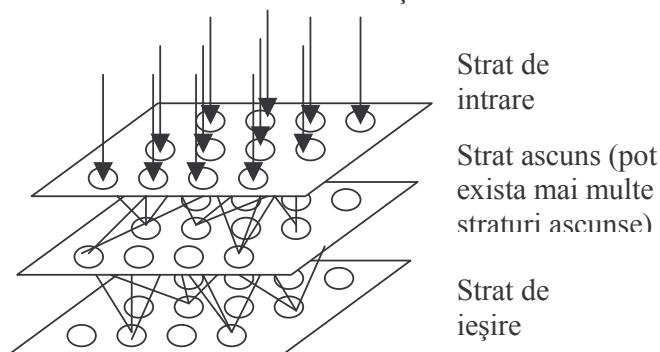
- *Puterea.* Rețelele neurale folosesc tehnici foarte sofisticate de modelare, capabile să modeleze funcții extrem de complexe. În particular, rețelele neurale sunt neliniare. Pentru mulți ani modelarea liniară a fost utilizată ca o tehnică comună în marea majoritate a domeniilor în care era necesară modelarea, deoarece modelele liniare erau singurele strategii de optimizare cunoscute. Unde aproximarea liniară nu este pe deplin corectă (un caz foarte întâlnit) modelele suferă corespunzător. Rețelele neurale țin în frâu problema cursei de dimensionalitate, care dă peste cap încercările de a modela funcții neliniare cu un număr foarte mare de variabile.
- *Folosirea ușoară.* Rețelele neurale învață după un anumit exemplu. Un utilizator de rețele neurale adună datele reprezentative, și apoi le invocă algoritmilor de antrenare să învețe automat structura

de date dată de utilizator. Cu toate că utilizatorul nu are cunoștințe euristice despre cum se selectează și se construiesc datele, cum se selectează o rețea neurală, și cum se interpretează rezultatele, nivelul de cunoștințe al utilizatorului pentru a lucra cu rețele neurale este mult mai mic în comparație cu cazul în care s-ar fi folosit metode de statistică tradiționale.

Rețelele neurale se apelează intuitiv, și sunt bazate pe un nivel foarte jos de modelare a sistemelor biologice. În viitor, dezvoltarea acestor modele neuro-biologice vor conduce sigur la apariția computerelor inteligente.

Straturile rețelelor neurale

Biologic, rețelele neurale sunt construite într-un spațiu tridimensional, format din componente microscopice. Acești neuroni par capabili să se conecteze cu vecinii fără anumite legi de conectare. Acest fapt nu este adevărat în orice rețea construită de om. Rețelele neurale artificiale sunt alcătuite din neuroni artificiali primitivi. Această alcătuire de neuroni, este produsă creându-se straturi, fiecare fiind conectat la alt strat. Modul cum sunt conectate aceste straturi poate să varieze de la o rețea la alta. Toate rețelele neurale artificiale au o structură și o topologie aproximativ similară. Câțiva neuroni, vor interacționa cu lumea de afară, prin a primi datele de intrare, iar alți neuroni vor avea rolul de a furniza lumii de afară datele de ieșire. Restul neuronilor vor fi ascunși.



Din figura de mai sus, se observă ca neuronii sunt grupați în straturi. Stratul de intrare este format din neuroni care primesc date de intrare, din mediul extern. Stratul de ieșire este format, la rândul său, din neuroni care vor comunica în afara sistemului, datele de ieșire. De obicei, există între aceste două straturi un anumit număr de straturi ascunse, figura de mai sus prezentându-ne o structură simplă ce conține doar un singur strat ascuns.

În momentul în care stratul de intrare primește date, neuronii din acest strat vor furniza fiecare câte o ieșire, care va deveni o intrare pentru celelalte straturi din sistem. Procesul va continua până în momentul în care o anumită condiție este satisfăcută sau până în momentul în care stratul de ieșire furnizează către mediul exterior o ieșire.

Trebuie determinat numărul de neuroni din stratul ascuns, pentru ca rețeaua neurală să dea rezultate cât mai bune, deoarece, de exemplu, un neuron în minus poate genera erori. Dacă se mărește numărul de neuroni ascunși prea mult, se va obține o supraîncărcare a rețelei, iar setul de date pentru învățare va fi memorat, făcând rețeaua nefolositoare în cazul în care apare un nou set de învățare.

Comunicarea și tipurile de conexiuni

Neuronii sunt conectați printr-o rețea, fiecare conexiune reprezentând pentru un neuron o intrare, iar pentru celălalt cu care este conectat, o ieșire. Aceste conexiuni sunt, de obicei, unidirecționale, dar oricum pot fi și bidirecționale, din cauză că poate exista conexiune și din cealaltă direcție. Un neuron primește date de intrare de la mai mulți neuroni, dar poate produce numai o singură ieșire, care va fi comunicată altor neuroni.

Un neuron dintr-un strat poate comunica cu orice alt neuron, sau, se poate întâmpla ca acesta să nu aibă nici o conexiune. Neuronii dintr-un strat trebuie să fie întotdeauna conectați cu neuronii din cel puțin un strat.

Conexiuni inter-straturi

Există diferite tipuri de conexiuni care se pot realiza între straturi, acestea fiind denumite conexiuni inter-straturi:

- *conexiuni realizate în totalitate* - fiecare neuron aparținând primului strat este conectat cu fiecare neuron din cel de-al doilea strat;
- *conexiuni parțiale* - un neuron aparținând primului strat nu este conectat cu toți neuronii de pe cel de-al doilea strat;
- *conexiuni de tip feed forward* - neuronii de pe primul strat trimit datele de ieșire neuronilor de pe cel de-al doilea strat, dar ei nu pot primi intrări înapoi de la neuronii de pe cel de-al doilea strat;
- *conexiuni bidirecționale* - este un alt tip de conectare care poate să transporte și informațiile de ieșire (ieșirile) ale neuronilor de pe cel de-al doilea strat către neuronii de pe primul strat; Conexiunile de tip feed forward și cele bidirecționale pot fi conectate în totalitate sau doar parțial.
- *conexiuni ierarhizate* - dacă rețeaua neurală are o structură ierarhică, neuronii de pe un strat mai de jos poate comunica numai cu neuronii de pe următorul strat;
- *conexiuni rezonante* - în acest caz, straturile prezintă conexiuni bidirecționale, și ele pot trimite mesaje în rețea până când o posibilă condiție de stop este îndeplinită.

Conexiuni intra-straturi

În structurile mai complexe, există comunicare chiar și între neuronii aflați pe același strat, aceste situații fiind cunoscute sub numele de conexiuni intra-straturi. Sunt două tipuri de astfel de conexiuni:

- *conexiuni recurente*

Neuronii dintr-un strat sunt conectați total sau parțial cu alți neuroni. După ce acești neuroni primesc datele de intrare de la alt strat de neuroni, ei vor comunica date de ieșire de mai multe ori într-un anumit timp, înainte ca lor să le fie permis de a transmite date către alt strat. În general, unele condiții ce trebuiesc realizate de către neuronii de pe un strat, trebuie să fie îndeplinite înainte ca neuronii să comunice datele de ieșire către alt strat.

- *conexiuni on-center/off surround*

Un neuron dintr-un strat prezintă conexiuni excitatorii cu el însuși și cu vecinii cei mai apropiați și conexiuni inhibitorii cu ceilalți neuroni. Ne putem imagina acest tip de conexiune ca o grupare de neuroni competitivă. Fiecare astfel de grupare de neuroni trimite semnale excitatorii către ea însăși, și către toți membrii grupării, dar trimite și semnale inhibitorii către toți membrii altor astfel de grupări. După un anumit număr de iterații, de astfel de schimburi de semnale, neuronii cu valoarea ieșirii activă va câștiga, și îi este permis să reactualizeze ponderile grupării și membrilor grupării respective.

(Sunt două tipuri de conexiuni dintre doi neuroni, conexiuni excitatorii și conexiuni inhibitorii. În cazul conexiunilor excitatorii, ieșirea neuronului întărește posibila acțiune a neuronului cu care este conectat. Când conexiunea dintre doi neuroni este inhibitoare, atunci ieșirea neuronului care trimite mesajul, va reduce posibila activitate a neuronului ce primește mesajul.)

În final, menționăm că numărul neuronilor din straturile ascunse poate fi oricât de mare, însă numeroase studii efectuate până în prezent, au constatat că, pentru majoritatea problemelor practice, este suficientă folosirea a unuia sau două straturi ascunse. În ceea ce privește numărul neuronilor din fiecare strat, acesta depinde de problema specifică analizată, neexistând o procedură generală de dimensionare a unei astfel de rețele.

Caracteristicile rețelelor neuronale

Vorbind la modul general, o rețea neuronală este o rețea de comunicare, care comunică o funcție de cost pentru o ieșire dorită. Matematic vorbind, o rețea neuronală reprezintă un sistem dinamic care poate fi modelat prin intermediul unui set de ecuații cu diferențe. Un astfel de sistem depinde de valorile parametrilor ecuației. Mici schimbări ale parametrilor au efect asupra stabilității, instabilității (oscilații) sau instabilitate catastrofală (haos).

Rețelele neuronale sunt caracterizate prin:

1. Procesare colectivă
 - programul este executat în mod colectiv
 - operațiile sunt descentralizate
2. Robustețe
 - operațiile sunt insensibile la insuccesele repetate
 - operațiile sunt insensibile la intrări parțiale sau perturbate
3. Învățarea
 - Rețeaua realizează asociații în mod automat
 - Programul este creat de rețea pe durata învățării
 - Rețeaua se adaptează cu sau fără profesor, programatorul nu intervine.
4. Operații asincrone: rețelele neurale biologice nu posedă ceas pentru sincronizarea operațiilor lor.

Parametrii rețelelor neurale

Formele care se aplică la intrarea unei rețele neurale pot fi completate, parțial complete sau afectate de zgomot. O performanță 100% a rețelei înseamnă că pentru fiecare semnal de intrare aplicat la intrarea rețelei, aceasta va produce forma de ieșire dorită. Atunci când se proiectează o rețea neurală, trebuie avute în vedere următoarele:

1. Topologia rețelei
2. Numărul de straturi în rețea
3. Numărul de neuroni din fiecare strat
4. Algoritmul de învățare ce va fi adoptat (numai în cazul supervizat)
5. Numărul de iterații pentru fiecare formă pe durata antrenării
6. Numărul de calcule pe iterație
7. Viteza de recunoaștere a unei forme
8. Performanțele rețelei
9. Plasticitatea rețelei (gradul de funcționalitate a rețelei)
10. Capacitatea rețelei (numărul maxim de forme ce poate fi cunoscut de rețea)
11. Gradul de adaptabilitate a rețelei (cât de mult se poate adapta rețeaua după antrenare)
12. Valorile de prag ale neuronilor (acestea fiind fixate inițial la valori fixe: +1)
13. Valorile de activare
14. Limitarea ponderilor
15. Alegerea funcțiilor neliniare
16. Imunitatea la perturbații
17. Erorile staționare sau valorile finale ale ponderilor.

Modelul perceptronului simplu

Modelul perceptronului implementat de către Rosenblatt, reprezintă sâmburele din care s-au dezvoltat toate celelalte rețele neurale. Arhitectura perceptronului standard este cea mai simplă configurație posibilă a unei rețele și ea permite ca învățarea acesteia să se realizeze folosind un algoritm simplu și eficient. Acest algoritm este reprezentativ pentru o clasă largă de algoritmi de învățare, motiv pentru care i s-a acordat o atenție specială

Modelul neuronului artificial (McCulloch-Pitts) este un sistem în buclă deschisă. Nu dispune de nici un mecanism de a compara răspunsul actual cu cel dorit și prin urmare nu există un proces de ajustare a ponderilor. Nevoia de reacție a fost rapid recunoscută, mai ales într-o perioadă în care

teoria sistemelor se orienta către controlul adaptiv. Prin urmare, pornind de la modelul McCulloch-Pitts s-a dezvoltat următoarea generație de paradigme, cu învățare și adaptare. Un astfel de model este perceptronul.

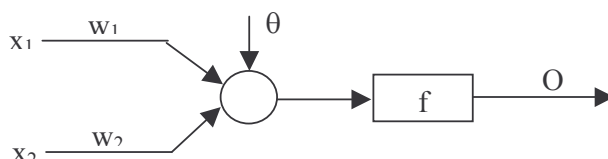
Perceptronul este o paradigmă care necesită învățarea supervizată. El este un sistem de recunoaștere a formelor care recunoaște forme abstracte și geometrice de la niște intrări optice. El poate face generalizări limitate și poate categorisi forme expuse la intrare. Mecanismul de învățare depinde de eroarea dintre țintă și ieșirea actuală.

Perceptronul simplu

Perceptronul cu un singur strat (Single Layer Perceptron) este cel mai simplu model de rețea neuronală, ce necesită învățarea supervizată și are capacități de estimare și clasificare a formelor de intrare.

Structura sa presupune, desigur, un singur strat de neuroni, iar mecanismul de învățare depinde de eroarea dintre valoarea ieșirii dorite și ieșirea curentă.

Un perceptron monostrat cu două intrări este schematic reprezentat în figura următoare:



unde vectorii $X = \{x_1, x_2\}$, $W = \{w_1, w_2\}$, reprezintă vectorii de intrare și respectiv al ponderilor, iar θ , f , reprezintă pragul și funcția de activare a acestui caz particular de perceptron (desigur că, unui caz general îi vor corespunde vectori de intrare și ponderi n-dimensionali).

Pe durata antrenării perceptronului, fiecare set de intrare X excită intrările și este produs un răspuns O . Acest răspuns este comparat cu ieșirea dorită, iar eroarea rezultată este utilizată pentru ajustarea ponderilor.

Eroarea poate fi exprimată ca:

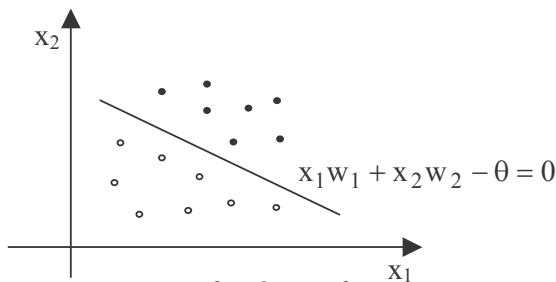
$$E = T - O$$

unde, T este valoarea ieșirii țintă, iar O valoarea curentă a ieșirii.

Se dovedește că, folosind modelul perceptronului se rezolvă problema clasificării mulțimii unor vectori de intrare n-dimensionali, în două clase A și B. Ecuția:

$$\sum_{i=1}^n w_i x_i - \theta = 0$$

determină frontiera celor două semispații în R^n , care separă elementele clasei A, de cele ale clasei B. Pentru cazul $n = 2$, această ecuație reprezintă o dreaptă în planul (x_1, x_2) , în care fiecare vector de intrare este reprezentat printr-un punct. În figura următoare este reprezentată separarea semispațiilor, pentru acest caz.



Din ecuația dreptei $x_1 w_1 + x_2 w_2 - \theta = 0$, rezultă:

$$x_2 = -\left(\frac{w_1}{w_2}\right)x_1 + \frac{\theta}{w_2}$$

unde $-\frac{w_1}{w_2}$ = panta dreptei de separație, iar $\frac{\theta}{w_2}$ = intersecția dreptei de separație cu axa x_2 .

Prin modificarea ponderilor w_1 și w_2 se modifică poziția dreptei în planul respectiv: problema constă în găsirea unui algoritm de învățare, care să determine ponderile w_1 și w_2 astfel ca, frontiera să ajungă să separe corect elementele clasificate, în clasele A, respectiv, B.

Învățarea

Perceptronul monostrat (Single Layer Perceptron) necesită învățarea supervizată. Rolul profesorului în învățarea supervizată este de supervizare. Prin profesor se înțelege că, pe durata învățării, există un circuit adițional care compară ieșirea dorită cu cea actuală. Aceasta înseamnă că, atunci când la intrare se aplică un stimul, ieșirea dorită este cunoscută apriori. Ieșirea actuală a rețelei neuronale și ieșirea dorită sunt aplicate unui comparator care produce un semnal de eroare. Această eroare este procesată în concordanță cu algoritmul de învățare, care va încerca să minimizeze această eroare ajustând valorile ponderilor sinaptice w_{ij} . Acest proces necesită un număr mare de iterații până la obținerea unei erori dorite.

Modelarea procesului de învățare se face prin explicitarea evoluției în timp a ponderilor. Forma matricei w_{ij} determină întotdeauna topologia rețelei, dacă considerăm că ponderile w_{ij} nule din această matrice formalizează lipsa unei legături sinaptice între neuronii i și j . Pragul de activare θ este asimilat uneori cu ponderea unei conexiuni dinspre un neuron formal, cu activarea constantă $x_0 = 1$ și $w_{0j} = \theta$.

În general, schimbarea ponderilor este proporțională cu semnalul de învățare și cu stimulul neuronului.

Prin urmare, regula de învățare poate fi exprimată ca:

$$w_{ij}(k+1) = w_{ij}(k) + \text{termen corector}$$

unde k reprezintă iterația curentă. Învățarea se termină când termenul corector este 0.

Algoritmul de învățare

În rețelele neuronale artificiale, descrierea modului în care ponderile conexiunilor w_{ij} se modifică pe durata sesiunii de învățare poartă denumirea de algoritm de învățare. Valorile finale ale ponderilor w_{ij} definesc programul rețelei neuronale.

Înainte de antrenarea perceptronului trebuie să se verifice următoarele:

1. Se alege un set de vectori de intrare $\{x_{ij}\}$.
2. Se alege un set de vectori țintă doriți (unul pentru fiecare vector de intrare).
3. Se alege o valoare mică și pozitivă pentru rata de învățare μ și, dacă este necesar, un criteriu de modificare a acesteia (inclusiv un pas de modificare).
4. Se selectează neliniaritatea (dacă se alege funcția sigmoid se alege și factorul ei de amplificare).
5. Se decide procedura de terminare: se definește o valoare acceptată a erorii (poate fi chiar nulă) pentru ieșirea actuală. Când această eroare este atinsă, procesul de învățare este terminat. O altă procedură de terminare poate fi impusă prin numărul de iterații. Totuși, ultima nu garantează obținerea unor valori adecvate pentru ponderi.

După executarea operațiilor de mai sus algoritmul poate fi implementat urmând pașii :

P1. Se inițializează ponderile și pragul de activare: se setează $w_j(0)$ și θ la valori mici, aleatoare.

P2. Se aplică o formă de intrare X_p și ieșirea dorită corespunzătoare T_p , unde p este numărul formei din set.

P3. Se calculează ieșirea actuală:

$$O(k) = f \left(\sum_{j=0}^n w_j(k) x_j(k) \right)$$

P4. Se adaptează ponderile utilizând relația:

$$w(k+1) = w(k) + \mu[T(k) - w(k)x(k)]x(k), \text{ pentru } 0 \leq k \leq n-1$$

Ieșirea dorită este atinsă când ponderile nu se mai schimbă.

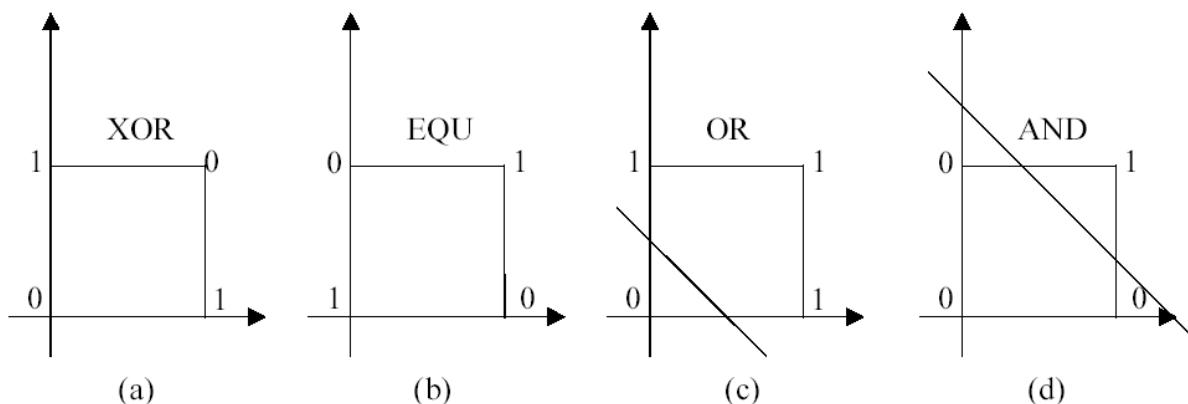
P5. Se repetă pașii P2 - P4.

Operații logice cu perceptroni simpli

Un perceptron simplu (Single Layer Perceptron) poate fi privit ca:

1. O funcție pentru rezolvarea problemei de recunoaștere a formelor (două clase de forme).
2. O unitate binară (logică). Același perceptron, funcție de valorile ponderilor, poate implementa câteva funcții logice. Pentru $n = 2$, pot fi implementate funcțiile: AND, OR, NOT. Funcția XOR nu poate fi implementată în acest caz.

x_1	x_2	$f(\text{XOR})$	$f(\text{EQU})$	$f(\text{OR})$	$f(\text{AND})$
0	0	0	1	0	0
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	1	1	1



În multe probleme concrete de clasificare (și de învățare) intervin clase de obiecte care nu sunt separabile liniar. Cele mai celebre, dar și mai simple probleme care nu pot fi rezolvate de un perceptron, sunt problemele calculării valorilor funcțiilor logice sau exclusiv (XOR) și de echivalență (EQU).

$$\begin{array}{llll}
 x_1 = 0 & x_2 = 0 & w_1x_1 + w_2x_2 = 0 & \Rightarrow 0 \geq \theta \\
 x_1 = 1 & x_2 = 0 & w_1x_1 + w_2x_2 = w_1 & \Rightarrow w_1 < \theta \\
 x_1 = 0 & x_2 = 1 & w_1x_1 + w_2x_2 = w_2 & \Rightarrow w_2 < \theta \\
 x_1 = 1 & x_2 = 1 & w_1x_1 + w_2x_2 = w_1 + w_2 & \Rightarrow w_1 + w_2 \geq \theta
 \end{array}$$

Conform primei condiții, θ este negativ. Din inegalitatea a doua și a treia rezultă

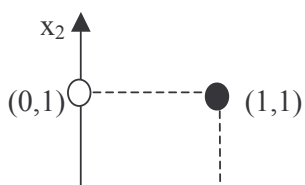
$$w_1 + w_2 < 2\theta,$$

iar din ultima inegalitate rezultă

$$w_1 + w_2 \geq \theta,$$

evident contradicție. Prin obținerea acestei contradicții se arată că funcția de echivalență nu este calculabilă de perceptron.

Problema calculării valorilor funcției logice sau exclusiv (XOR) este reprezentată în figura următoare.



x_1	x_2	y
0	0	0
0	1	1

În exemplul de mai sus se utilizează un perceptron cu doi neuroni de intrare și unul de ieșire și cu două ponderi w_1 și w_2 ce leagă neuronii de intrare cu neuronul de ieșire. Dacă se asociază vectorului $(1, 0)$ de la intrare valoarea "1" de la ieșire, va rezulta că $w_1 > 0$. Asemănător, dacă asociem forme $(0, 1)$ ieșirea "1" va rezulta $w_2 > 0$. Pentru forma de intrare $(1, 1)$ perceptronul va da întotdeauna răspunsul "1", ceea ce este în contradicție cu ceea ce se dorește. Deci, este imposibil să se găsească valorile corecte pentru w_1 astfel încât problema să fie rezolvată.

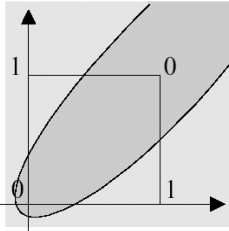


Fig. Curbă de decizie

Problema funcției XOR poate fi rezolvată de perceptron dacă este codată corespunzător.

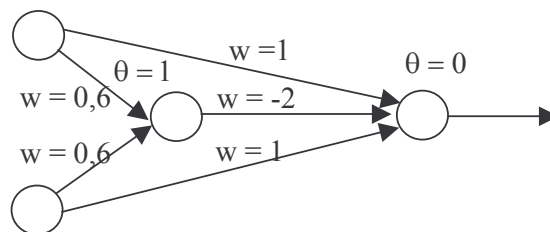
Se observă două soluții:

- Se utilizează 3 neuroni de intrare, problema devenind:

x_1	x_2	x_3	y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

A treia coloană de la intrare s-a obținut prin multiplicarea primelor două coloane. În acest fel problema este rezolvată, ponderile asociate fiind $w_1 = 1$, $w_2 = 1$ și $w_3 = -2$. Dificultatea practică nu constă atât în arhitectura rețelei cât în codarea problemei de rezolvat.

- Se creează o rețea cu mai multe straturi. Pe lângă cei doi neuroni de intrare și cel de ieșire se mai adaugă un neuron ascuns, neuronii fiind conectați ca în figura următoare.



Se observă că starea neuronului ascuns are o singură valoare diferită de zero.

Perceptronul Multistrat

Limitele învățării folosind algoritmul perceptronului nu se datorează algoritmului însuși, ci sunt legate de topologia extrem de simplă a rețelei utilizate, care asigură doar împărțirea spațiului intrărilor în două semiplane. Dacă problema de învățare necesită regiuni de decizie mai complicate, atunci trebuie mărită complexitatea rețelei.

Soluția constă în utilizarea unor rețele compuse din mai mulți perceptroni simpli organizați într-o structură ierarhică, stratificat. Datorită acestui tip de organizare, asemenea rețele neuronale

poartă numele de perceptron multistrat (Multi Layer Perceptron - MLP). Celulele elementare ale MLP se numesc neuroni și se deosebesc de perceptronul definit de Rosenblatt, prin utilizarea unor funcții de activare de tip sigmoid în locul funcției treaptă.

Într-o asemenea rețea, neuronii sunt organizați în straturi: un strat de intrare, un strat de ieșire și unul sau mai multe straturi ascunse (intermediare), denumite astfel deoarece între neuronii acestor straturi și intrările, respectiv ieșirile rețelei, nu există legături directe. De regulă, fiecare neuron al unui strat este conectat cu toți neuronii din stratul imediat superior ("fully - connected networks") și nu admit legături între neuronii ce nu aparțin unor straturi consecutive. Există, însă, și situații în care se folosesc asemenea legături, numite "short-cut connections", cu rolul de a întări influența unei anumite intrări asupra ieșirii RNA.

Numai neuronii din straturile ascunse și, eventual, cei din stratul de ieșire realizează o procesare a informației. Stratul de intrare servește doar la distribuirea intrărilor, pe care rețeaua le preia din exterior, către primul strat ascuns, fără a realiza o procesare propriu-zisă a acestora.

În structura perceptronului multistrat pot exista unul sau mai multe straturi ascunse de neuroni între stratul de intrare și cel de ieșire. Numărul de straturi ascunse și numărul neuronilor în straturi nu sunt fixate. Însă, numeroase studii efectuate până în prezent au constatat că, pentru majoritatea problemelor practice, este suficientă folosirea a unu sau două straturi ascunse. În ceea ce privește numărul neuronilor din fiecare strat, acesta depinde de problema specificată analizată, neexistând o procedură generală de dimensionare a unei astfel de rețele.

În unele situații se pot rezolva probleme de învățare complexe prin creșterea numărului de câmpuri ascunse. În realitate, aceasta nu este o soluție universală. Pentru unele probleme este necesară o creștere exponențială a numărului de straturi pentru a obține o creștere liniară a vitezei de învățare. De asemenea, există probleme pentru care viteza de învățare descreește cu mărimea numărului de straturi. Mărimea complexității rețelei este, așadar, o cale problematică în abordarea proceselor de învățare. O cale diferită este considerarea altor tipuri de neuroni. Acest lucru revine, în esență, la adoptarea unor tipuri diferite de neliniaritate în comportarea neuronilor.

Găsirea unor tipuri adecvate de neliniaritate rețelelor neurale conferă forță de calcul superioară. Comportarea neliniară a neuronilor este esențială pentru un perceptron cu mai multe straturi. Dacă funcția de ieșire ar fi liniară, considerarea mai multor straturi nu ar aduce nici un avantaj – se poate întotdeauna găsi un perceptron cu un singur strat care să realizeze aceeași funcție ca și perceptronul multistrat liniar.

Utilizarea funcțiilor de ieșire neliniare de tip prag prezintă un dezavantaj legat de faptul că aceste funcții nu sunt derivabile. Prin urmare, funcțiile de tip prag nu pot fi utilizate pentru determinarea ponderilor prin metode de optimizare standard. Acest neajuns poate fi depășit considerând funcții de ieșire (neliniare) de tip sigmoid.

Trecerea de la un singur perceptron, la un perceptron multistrat, prin adăugarea nivelelor intermediare și înlocuirea funcției treaptă cu funcția sigmoid, conduce la arhitecturi mai robuste, cu performanțe superioare, capabile să recunoască fapte mai complexe. Totodată, noua arhitectură a perceptronului multistrat este considerată o problemă de optimizare, iar cei mai folosiți algoritmi sunt cei de tipul retropropagare (Backpropagation sau Generalized Delta Rule).

Procesul de învățare

1. Noțiuni generale.

Cea mai semnificativă proprietate a rețelelor neuronale o reprezintă fără nici o îndoială capacitatea rețelelor neuronale de a învăța din mediul înconjurător și de a-și îmbunătăți performanțele pe baza acestui proces de învățare. Rețeaua neuronală învățată pe baza unui proces iterativ de ajustare a tărilor sinaptice și eventual al nivelului de activare. Dacă procesul de învățare decurge bine, atunci rețeaua neuronală acumulează tot mai multe informații, la fiecare iteratie.

Evident că atunci când folosim termenul de "proces de învățare" ne situăm într-o terminologie mult prea largă, care este dependentă de mai mulți factori. Fără a încerca să teoretizăm prea mult putem să observăm foarte simplu că termenul de "proces de învățare" este folosit în maniere total diferite

de un psiholog sau de un profesor atunci când predă. De aceea trebuie să încercăm să fim mai riguroși în definirea "învățării" și a "procesului de învățare", deoarece vom folosi de acum încolo foarte des aceste noțiuni. Pentru aceasta vom defini, în contextul calculului neuronal, "învățarea" conform cu, în felul următor:

Def.: *Învățarea este un proces prin care parametrii rețelei neuronale sunt adaptați permanent prin intermediul unor stimuli proveniți de la mediul înconjurător caruia îi aparține rețeaua neuronală. Tipul de învățare este determinat de forma de modificare a parametrilor rețelei neuronale.*

Definiția de mai sus conține următoarea secvență de evenimente:

- **Evenimentul 1:** Rețeaua neuronală primește stimuli de la mediul înconjurător;
- **Evenimentul 2:** Rețeaua neuronală se modifică ca răspuns la stimuli;
- **Evenimentul 3:** Ca urmare a acestor modificări permanente, care afectează structura sa internă, rețeaua neuronală răspunde de fiecare dată într-un nou mod mediului de la care vin stimuli.

Să încercăm să dam o formulare matematică acestui proces descris mai sus. Pe baza celor prezentate anterior se poate observa faptul că în cadrul procesului de învățare se modifică taria sinaptică. De aceea avem formularea matematică cea mai generală a procesului de învățare, exprimat prin formula:

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t) \quad (1)$$

- $w_{ji}(t+1)$ și $w_{ji}(t)$ reprezintă noua și vechea valoare a tariei sinaptice w_{ji} care unește axonul neuronului i de o dendrită a neuronului j .
- reprezintă ajustarea aplicată tariei sinaptice $w_{ji}(t)$, la momentul t , obținându-se valoarea $w_{ji}(t+1)$ la momentul $t+1$, în urma procesului de ajustare.

Ecuatia (1) conține în mod evident efectele Evenimentelor 1, 2 și 3 prezentate mai sus. Ajustarea este obținută ca urmare a unor stimuli ai mediului înconjurător (Evenimentul 1), iar valoarea modificată a tariei sinaptice $w_{ji}(t+1)$ definește schimbarea din rețeaua neuronală, ca un rezultat al stimulilor prezentați rețelei neuronale (Evenimentul 2). Din momentul $t+1$ rețeaua neuronală răspunde într-un mod nou mediului înconjurător deoarece taria sinaptică s-a modificat, devenind $w_{ji}(t+1)$ (Evenimentul 3).

Def.: *Se numește algoritm de învățare, un set de reguli predefinite care soluționează problema "învățării".*

Evident, pentru o anumită rețea neuronală nu există un unic algoritm de învățare. Mai degrabă există o mulțime de legi de învățare, fiecare dintre ele având o serie de avantaje și de dezavantaje. Ceea ce face ca algoritmii de învățare să fie diferiți este modul de calcul al ajustării tariei sinaptice $\Delta w_{ji}(t)$.

Un alt factor important relativ la procesul de învățare este modul de raportare a unei rețele neuronale la mediul înconjurător. În acest context putem defini:

Def.: *Se numește paradigmă de învățare, un model al mediului înconjurător în care are loc procesul de învățare al rețelei neuronale.*

Pe această bază, putem formula taxonomia fundamentală a procesului de învățare:

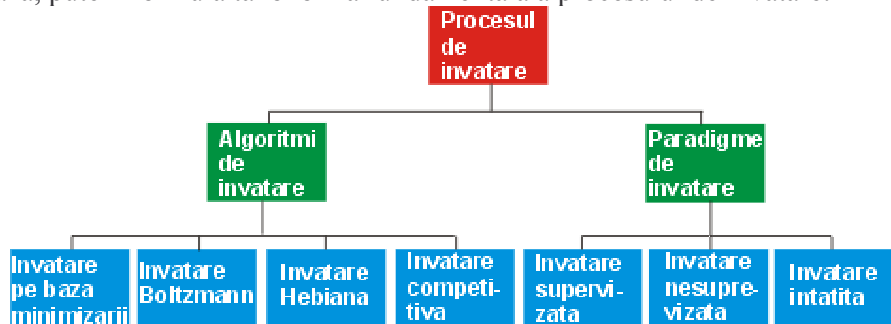


Fig. Taxonomia fundamentală a procesului de învățare.

Paradigme de învățare

1. Învățare supervizată

Modificarea tăriiilor sinaptice este făcută pe baza comparației dintre vectorul de ieșire $\mathbf{y}^\mu = (y_1^\mu, y_2^\mu, \dots, y_m^\mu)$ obținut la stratul de ieșire și vectorul tinta $\mathbf{z}^\mu = (z_1^\mu, z_2^\mu, \dots, z_m^\mu)$, $\mu = 1, \dots, P$, ce reprezintă rezultatul dorit a se obține la stratul de ieșire, când la stratul de intrare s-a prezentat vectorul de intrare $\mathbf{x}^\mu = (x_0^\mu, x_1^\mu, \dots, x_n^\mu)$, $\mu = 1, \dots, P$ din mulțimea de antrenament.

Vectorul tinta \mathbf{z}^μ este furnizat de un *profesor (antrenor)*, de unde și denumirea de învățare supervizată. Învățarea supervizată presupune prezentarea de către un antrenor a unor perechi de date de forma $(\mathbf{x}^\mu, \mathbf{z}^\mu)$, $\mu = 1, \dots, P$, ce formează o mulțime de date, numită *mulțime de antrenament*:

$$T = \{(\mathbf{x}^\mu, \mathbf{z}^\mu) | \mu = 1, 2, \dots, P\} \quad (3.16)$$

Diferența dintre răspunsul obținut y și răspunsul dorit z , reprezintă *eroarea* și este folosită pentru a modifica tăriiile sinaptice, pe baza unui algoritm specific, numit *lege de învățare*.

Putem reprezenta învățarea supervizată cu ajutorul următoarei diagrame:

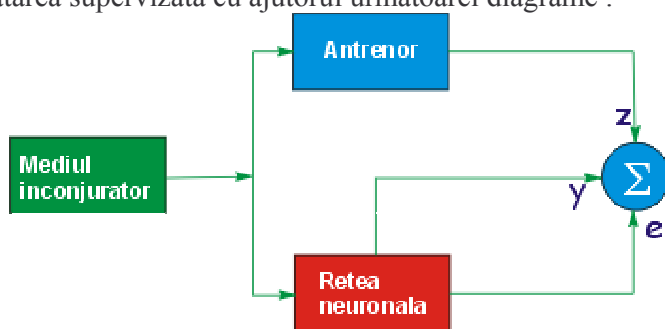


Fig. 2.: Diagrama învățării supervizate.

2. Învățare nesupervizată

În cadrul învățării nesupervizate nu există profesor (antrenor). Rețeaua neuronală trebuie să fie în stare să "descopere" singure modele, trasături, corelații sau categorii în mulțimea datelor de intrare și să le codifice sub forma unor date de ieșire. Neuronii și conexiunile rețelei neuronale trebuie să reprezinte un anumit grad de *auto-organizare (self-organization)*.

Învățarea nesupervizată poate fi utilizată doar atunci când în mulțimea datelor de intrare există *redundanță*. Fără redundanță este imposibil de a descoperi vreun model (patern) sau trasătura în mulțimea datelor de intrare. Din acest punct de vedere redundanța asigură cunoașterea.

În diagrama de mai jos este reprezentată paradigma învățării nesupervizate.



Fig. Diagrama învățării nesupervizate.

În cadrul învățării nesupervizate nu avem la dispoziție exemple ale funcției ce trebuie învățată de către rețeaua neuronală. Mai degrabă avem informații despre o măsură a calității reprezentării la care trebuie să ajungă rețeaua neuronală prin procesul de învățare, iar parametrii acesteia vor fi optimizați în raport cu această măsură. O dată ce procesul de învățare s-a terminat, și rețeaua neuronală a fost acordată la trasăturile statistice ale datelor de intrare, ea va fi capabilă să formeze reprezentări interne care să codifice trasăturile datelor de intrare și de aceea în mod automat noi clase.

Pentru ca o rețea neuronală să poată efectua o învățare nesupervizată putem utiliza un algoritm de învățare Hebbiană sau un algoritm de învățare competitivă.

3. Învățare întărită (reinforcement)

Învățarea întărită poate fi definită ca fiind învățarea on-line a unei relații de intrare-ieșire (input-output) prin intermediul unui proces de încercări și erori desemnate să maximizeze un index scalar

Definitia care pare a fi cea mai potrivita a fost data de Sutton si de Barto:

```

graph TD
    MI[Mediu Inconjurator] --> CP[Intarire primara]
    CP --> Critic[Critic]
    Critic --> IE[Intarire euristica]
    IE --> EI[Elemente de invatare]
    EI --> BC[Baza de cunostinte]
    BC --> EP[Elemente de prelucrare]
    EP --> A[Actiuni]
    A --> MI
    EI -.-> MI
    BC -.-> MI
    EP -.-> MI
    
```

The diagram illustrates the architecture of a learning system. It begins with the **Mediu Inconjurator** (Environment), which provides **Intarire primara** (Primary reinforcement) to the **Critic** module. The **Critic** module then generates **Intarire euristica** (Heuristic reinforcement), which is used by the **Elemente de invatare** (Learning elements). These elements interact with the **Baza de cunostinte** (Knowledge base) and the **Elemente de prelucrare** (Processing elements). The processing elements produce **Actiuni** (Actions), which are fed back into the **Mediu Inconjurator**. Additionally, there are feedback loops from the learning elements, knowledge base, and processing elements back to the **Mediu Inconjurator**, indicating a continuous learning process.

În diagrama, avem un element numit critic care are rolul *semnalelor primare de întarire*, recepționate de la *mediul înconjurător* într-un semnal de calitate numit *semnal euristic de întarire*. Ambele semnale de întarire sunt codificate prin intermediul unor scalari reali. Sistemul de învățare este constituit din cele trei componente normale pentru orice sistem AI (Artificial Intelligence):

- Pe baza stimulilor de intrare de la mediul înconjurator si de la baza de cunostinte, elementul de prelucrare va determina relatia:

intrare --> distributia relatiei actiunilor de iesire