Hărţile topografice Kohonen

1. Reţele neurale - Generalităţi

O reţea neurală este compusă dintr-un număr de noduri sau unități de învățare conectate între ele prin legături. Fiecare legătură are o pondere numerică asociată cu ea. Ponderea este mijlocul primar de memorare pe termen lung în reţelele neuronale şi în realitate învăţarea constă în găsirea (calcularea) acestor ponderi astfel încât reţeaua să răspundă cât mai bine la datele de antrenament. Unele noduri au legături directe cu mediul înconjurător şi pot fi considerate ca şi unităţi de intrare sau ieşire. Alte noduri, care nu au nici o legătură directă cu exteriorul sunt considerate noduri „ascunse”. Ponderile reţelei sunt modificate pe parcursul învăţării astfel încât reţeaua să reproducă prin comportamentul intrare/ieşire cât mai fidel datele de antrenament.

Fiecare unitate are un set de intrări de la alte unităţii, un set de ieşiri care se leagă la alte unităţii, un prag de activare şi un mijloc de calcul al nivelului de acţiune pentru următorul pas dat de intrări şi ponderi. Idea este că fiecare unit face un calcul local pe baza datelor primite la intrări (care pot fi de la alţi neuroni sau intrările în reţea), fără să aibă o imagine de ansamblu asupra ceea ce trebuie să înveţe reţeaua şi prezintă la ieşire rezultatul obţinut. În practică, cele mai multe reţele neuronale sunt implementate software și utilizează un controlul sincron pentru a actualiza toate uniturile din reţea la secvenţe fixe de timp.

2. Arhitectura reţelelor neurale

Există o varietate de tipuri de structuri de reţele, fiecare dintre acestea rezultă din diferite posibilităţi de calcul şi în funcţie de problema care trebuie rezolvată. O reţea neurală poate fi privită ca un graf orientat ponderat în care neuronii sunt nodurile iar arcele orientate (cu ponderile asociate) sunt legăturile între neuroni Din punct de vedere al construcţiei reţelele neurale se împart în două categorii principale: reţelele ***feed-forward*** şi reţelele ***recurente***.

În reţelele **feed-forward** (cu propagare înainte), legăturile dintre neuroni sunt unidirecţionale şi nu există bucle (legături de la un neuron de pe un strat superior la un neuron de pe un strat inferior). În reţelele feed-forward legăturile pot pleca de la topologii arbitrare, neexistând nicio legătură spre stratul anterior sau legături care sar peste un anumit strat. În aceste tipuri de reţele toţi neuronii de pe un anumit strat sunt actualizaţi la o anumită perioadă de timp. Aceste reţele sunt reţele statice ele neavând un comportament dinamic propriu-zis ieşire reţelei depinzând doar de valoarea curentă a intrării nu şi de valorile anterioare ale intrării.

Reţelele **recurente** sunt reţele în care pot exista legături înapoi, un neuron de pe un strat superior are legături cu neuronii de pe nivelurile inferioare. De asemenea în reţelele recurente pot exista legături care sar peste anumite straturi. În acest caz reţeaua reprezintă un graf orientat complet şi are un comportament dinamic propriu-zis.

3. Reguli de învăţare

Fiecare tip de reţea îşi modifică ponderile în funcţie de anumite reguli de învăţare care depind atât de tipul datelor de intrare cât şi de modul de construcţie al reţelei. Capacitatea de învăţare reprezintă o trăsătură fundamentală a inteligenţei. În cadrul reţelelor neurale învăţarea constă în modificarea ponderilor rețelei pentru a-și adapta comportamentul la necesităţile rezolvării unei probleme specifice. În general sunt modificaţii coeficienţii sinaptici (ponderile legăturilor), însă uneori se pot modifica numărul de unități de învățare și /sau configurația reţelei.

Principalul avantaj al reţelelor neuronale în comparaţie cu sistemele expert clasice este acela că, în loc de a folosi un set de reguli date de un expert uman, are loc o învățare din exemple.

Din punct de vedere al organizării datelor de intrare există două categorii de învăţare:

- învăţarea ***nesupervizată*** în care se prezintă reţelei doar datele de intrare fără a se specifica şi ieşirea dorită pentru acestea. În acest caz reţeaua este lăsată să evolueze liber urmând ca la sfârşit să vedem rezultatul învăţării.
- învăţarea ***supervizată*** în care mulţimea de exemple de antrenament sunt organizate sub forma de perechi intrare – ieşire, specificând reţelei la fiecare pas care trebuie să fie ieşirea corectă, urmând ca reţeaua să generalizeze datele de intrare.

Există patru tipuri de reguli de învăţare:

- **regula de învăţare competitivă** – în cazul acestei reguli între unităţile de ieşire are loc o competiţie pentru activare. În final o singură unitate va fi activă la un moment dat. Acest fenomen este cunoscut şi sub numele „*winner takes all*”.

- **regula de învăţare prin corecţia erorii** – în învăţarea supervizată reţeaua dispune de ieşirea dorită pentru fiecare vector de intrare. În timpul învăţării ieşirea reţelei nu este egală cu valoarea dorită. Principiul corecţiei erorii foloseşte semnalul de eroare pentru modificarea ponderilor în scopul minimizării erorii.

- **regula de învăţare Boltzman** – maşinile Boltzman sunt reţele recurente simetrice şi acestea învaţă asocieri între vectorii de intrare şi vectorii de ieşire.

- **regula de învăţare Hebb** – este cea mai veche regulă de învăţare, în acest domeniu, care respectă postulatul lui Hebb apărut în 1949 având la bază observaţia neurobiologică: *„Dacă ambii neuroni legaţi într-o sinapsă sunt activi simultan coeficientul sinaptic al acestei legături creşte”.* Această

regulă este plauzibilă biologic şi prezintă avantajul că învăţarea are loc local, modificarea ponderii unei sinapse depinzând doar de neuronii alăturaţi.

4. Învăţare online/ offline

Din punct de vedere al timpului când se face actualizarea ponderilor reţelei există două timpuri de învăţare: învăţarea on-line şi învăţarea off-line.

1 off-line - se determină, pentru fiecare pereche intrare/ieşire, modificarea ce trebuie adusă coeficienţilor sinaptici. Aceste modificări se sumează pe parcurs și se aplică rețelei numai după ce au fost prezentate toate datele de antrenament (perechile intrare/ieşire).

2 on-line - modificarea coeficienţilor, calculaţi pentru o pereche intrare/ieşire se aplicată reţelei imediat după prezentarea acestei perechi. Prezintă, în raport cu precedentul, avantajul că este în general este mai rapid şi poate părăsi unele așa zise minime locale ale funcţiei de eroare totală.

5. Reţelele competitive

În multe reţele neurale există un strat de neuroni care evoluează pe baza competiţiei între neuroni. Cale mai cunoscute reţele sunt „CounterPropagation” şi „Hărţile Topografice ale lui Kohonen”.

În procesul „winner takes all” neuronul care este iniţial cel mai activat va domina treptat şi va deveni cu activare maximă în timp ce ceilalţi neuroni din competiţie se vor dezactiva. În multe reţele neurale procesul „winner takes all” este necesar pentru a selecta, în timpul învăţării sau a procesării, neuronul cu activarea maximă.

6. SOM (Self-Organizing Maps – hărţi cu auto-organizare)

Această reţea cunoscută şi sub numele de Reţeaua Kohonen după numele profesorului care a dezvoltat-o (Theo Kohonen) prezintă o reţea de învăţare nesupervizată în care unităţile de intrare/ieşire sunt dispuse sub formă de grilă (în general rectangulară) în plan, iar coeficienţii conexiunilor între unităţi din stratul de ieşire depind de „distanţele” între unităţi. Astfel dacă unităţile sunt apropiate acestea se influenţează reciproc iar dacă acestea sunt depărtate influenţa este inhibatoare. Se defineşte astfel pentru fiecare neuron o vecinătate a sa. Forma acestei vecinătăţi poate varia în funcţie de datele de intrare şi poate fi rectangulară sau circulară. În timpul învăţării competitive se actualizează nu numai vectorul asociat unităţii învingătoare ci şi (uneori cu un coeficient mai redus) unităţile din vecinătatea acesteia. Acest mod de modificare încurajează unităţile vecine să răspundă în mod similar la vectori de intrare similari. Deoarece T. Kohonen a lucrat la dezvoltarea teoriei competiţiei, elementele de procesare competitivă sunt adesea referite ca unităţi Kohonen.

Reţeaua este utilizată în majoritatea cazurilor în aproximarea distribuției vectorilor de intrare, în reducerea adimensionalității datelor menţinând (pe cât posibil) vecinătatea şi în „clustering”.

Arhitectura reţelei SOM constă într-o matrice bidimensională de unităţi de ieşire, fiecare conectată cu toate cele *n* unităţi de intrare. Fie $\mathbf{w_{ij}}$ vectorul *n*-dimensional asociat unităţii din poziţia (i,j) a matricei bidimensionale. Fiecare neuron de ieşire calculează distanţa dintre vectorul de intrare **x** şi vectorul pondere $\mathbf{w_{ij}}$ memorat. Ponderea neuronului învingător (şi a celor din vecinătatea acestuia) pentru pasul următor se calculează după următoarea formulă (formula lui Kohonen), acesta modificându-se după fiecare vector de intrare fiind astfel din categoria reţelelor cu învăţare online:

$$\overrightarrow{w_i}(t+1) = \begin{cases} \overrightarrow{w_i}(t) + \alpha(t)(\vec{x} - \overrightarrow{w_i}(t)) & pentru\ i \in Vecinatatii \\ \overrightarrow{w_i}(t) & rest \end{cases} \quad (1)$$

unde

$w_i(t+1)$ reprezintă ponderea de la pasul următor a unui neuron

$w_i(t)$ reprezintă ponderea de la pasul curent a unui neuron

x – vectorul de intrare

$\alpha(t)$ – rata de învăţare

Idea acestui algoritm este că se caută unitatea „învingătoare” pentru fiecare vector de intrare (la fel ca la algoritmul anterior cel bazat pe asociere pe baza similarităţiilor), dar modificarea coeficienţilor sinaptici se realizează nu numai pentru unitatea învingătoare ci (în general cu un coeficient α mai redus) şi pentru unităţile aflate în vecinătatea acestei unităţi, vecinătate definită de *v*. Acest mod de modificare a coeficienţilor încurajează unităţile vecine să răspundă în mod similar la vectori de intrare similari; reţeaua este astfel o „hartă” a mulţimii vectorilor de intrare.

6.1 Algoritmul de învățare SOM

Idea de învăţare constă în a plasa reţeaua undeva în plan şi a o antrena la început cu o vecinătate şi un coeficient de învăţare mare care pe parcursul învăţării să scadă treptat. Paşii algoritmului SOM sunt:

1. se dispun unităţile de ieşire sub formă de grilă în plan (această dispunere reprezintă de fapt şi iniţializarea ponderilor unităţilor cu valori aleatoare mici). Se recomandă memorarea acestor unităţi într-o matrice pentru a se putea calcula ulterior mai ușor vecinătatea unui neuron. In cazul reţelei SOM ponderea unui neuron este considerată poziţia acelui neuron în spaţiul de reprezentare al datelor de intrare.
2. se stabileşte o anumită formulă pentru calculul vecinătăţii şi al coeficientului de învăţare (formulă care să depindă de pasul la care s-a ajuns în învăţare).
3. se ia un vector de intrare, şi se calculează distanţa dintre el şi toţi neuroni din reţea.
4. Neuronul care este cel mai apropiat este considerat neuronul „învingător” (ca în formula 2) şi la acest neuron se modifică ponderea după formula lui Kohonen (formula 1)

$$\left\|\mathbf{x}-\mathbf{w}_{\mathbf{m},\mathbf{n}}\right\| = \min_{i,j}\left\|\mathbf{x}-\mathbf{w}_{\mathbf{ij}}\right\| \quad (2)$$

5. La toţi neuroni care sunt vecini (la pasul respectiv) cu neuronul „învingător” se recalculează ponderea după aceeaşi formulă (formula 1).
6. dacă coeficientul de învăţare (α) a ajuns la 0 se consideră că reţeaua a învăţat deoarece nu vor mai avea loc modificări în reţea (conform formulei lui Kohonen)
7. dacă coeficientul de învăţare nu a ajuns la 0 se sare la pasul 3

În cadrul SOM vecinătatea joacă un rol esenţial. În cazul dispunerii neuronilor pe o grilă pătrată vecinătatea unui neuron este definită ca fiind toţi neuronii vecini din cadrul matricei care sunt în domeniul [i-*v*, i+*v*] şi [j-*v*, j+*v*] pentru neuronul învingător de pe poziţia (i,j) din matricea de start iar *v* este vecinătatea la acel moment dat. Există mai multe formule pentru calculul vecinătăţii şi al coeficientului de învăţare la pasul t, acestea depinzând foarte mult de aplicaţie. Un exemplu de astfel de formulă pentru aplicaţia curentă ar fi:

$$Vecinatate(t) = 6.1 * e^{-\frac{t}{N}} + 1$$

$$\alpha(t) = 0{,}6 * e^{-\frac{t}{N}}$$

unde *t* reprezintă pasul curent

N – numărul total de paşi în care am vrea să înveţe reţeaua

Obs:

- Se consideră un pas numai în momentul în care s-a terminat de prezentat reţelei tot setul de date de antrenament.
- Coeficientul 6.1 pentru vecinătate este ales deoarece vrem să pornim cu o vecinătate mare la început (se recomandă vecinătatea să fie aproximativ 60-70% din numărul de neuroni de pe o dimensiune a reţelei). Coeficientul 6.1 a fost ales pentru o reţea de 10x10 neuroni.
- Coeficientul de 0,6 la rata de învăţare (α) este ales deoarece vrem ca rata de învățare să fie mică încât să tindă repede la 0.

Problemă. Să se implementeze algoritmul Kohonen prezentat anterior pentru gruparea automată a puntelor aflate în fişierul generat în cadrul primului laborator. Acesta va funcţiona ca un algoritm de clustering (grupare nesupervizată a datelor de intrare) iar pe ecran, pentru a putea avea o analiză vizuală a învăţării realizate de acesta, se vor afişa atât punctele de intrare cât şi evolutiv (la fiecare pas) poziţia neuronilor din reţea. Pentru o vizualizare mai bună a modului de „aşezare” a reţelei SOM se recomandă pe lângă afişarea neuronilor din reţea pe ecran să se afişeze şi legăturile între aceştia pe primul nivel de vecinătate (Adică pentru neuronul de pe poziţia ***(i,j)*** se vor afişa şi legăturile acestuia cu neuronii de pe poziţiile: ***(i-1,j), (i+1,j), (i,j-1), (i,j+1)***.