

Rezolvarea problemelor de optimizare cu rețele neuronale

octombrie 2003

Ideea utilizării rețelelor neuronale în rezolvarea problemelor de optimizare dificile (din clasa celor NP-complete) a fost sugerată prima oară de către Hopfield și Tank în 1985. Ei au demonstrat că problema comis voiajorului (TSP - "Travelling Salesman Problem") poate fi rezolvată utilizând o rețea total interconectată (ulterior rețelele cu o astfel de arhitectură au fost denumite generic ca modele Hopfield). Deși abordarea neuronală a problemelor de optimizare permite, în general, obținerea doar a unei soluții sub-optimale și unele implementări sunt mai puțin eficiente decât unele metode meta-euristice ("simulated annealing", algoritmi genetici, "tabu search" etc.) este important faptul că oferă o cale alternativă de rezolvare ce conduce la posibilitatea proiectării unor sisteme hard dedicate.

O altă abordare neuronală a problemelor de optimizare de tipul TSP o reprezintă rețeaua de tip Kohonen cu auto-organizare, cunoscută sub denumirea de *rețea elastică* ("elastic net"). Se bazează pe ideea că nodurile (orașele) problemei sunt amplasate într-un plan iar "distanțele" dintre ele sunt reprezentate chiar de distanța euclidiană. Rețeaua are o arhitectură de tip Kohonen unidimensională, unitățile fiind amplasate în nodurile unei grile circulare. Prin procesul clasic de auto-organizare de tip Kohonen se ajunge ca "inelul" unităților să "treacă" prin sau cât mai aproape de pozițiile orașelor păstrând lungimea traseului cât mai mică.

1 Caracteristicile modelului Hopfield

Arhitectura. Modelul Hopfield este o rețea constituită din N unități total interconectate, în care fiecare unitate joacă simultan rol de unitate de intrare și de ieșire (vezi fig. 1). El poate fi utilizat atât pentru simularea *memoriilor asociative* (sisteme dinamice ce permit stocarea informațiilor prin intermediul parametrilor lor și regăsirea acestora pornind de la exemplare incomplete sau afectate de zgomot) cât și pentru rezolvarea problemelor de optimizare combinatorială. În continuare ne referim doar la al doilea tip de aplicație.

Funcționare. Pentru a descrie relațiile de funcționare ale rețelei pentru fiecare neuron i vom folosi următoarele notații:

- $x_i(t)$ - potențialul neuronului la momentul t (suma ponderată a semnalelor provenite de la celelalte unități);
- $y_i(t) = f_i(x_i(t))$ - semnalul de ieșire produs de neuron (f_i este funcția de activare);
- $I_i(t)$ - semnal de intrare primit din partea mediului;
- w_{ij} - ponderea conexiunii dintre unitatea j și unitatea i . Ansamblul tuturor ponderilor formează o matrice pătratică $W = (w_{ij})_{i=\overline{1,N}, j=\overline{1,N}}$.

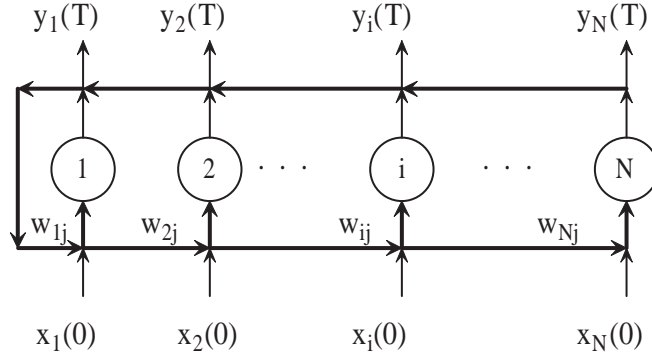


Figura 1: Arhitectura modelului Hopfield

Întrucât în majoritatea aplicațiilor toate unitățile au același tip de funcție de activare vom considera că $f_i = f$ pentru $i = \overline{1, N}$. Prezența conexiunilor inverse face ca semnalul produs de către o unitate să revină direct (prin bucle) sau indirect (transformat prin intermediul altor unități) la unitatea de la care a pornit. Din acest motiv funcționarea constă de fapt în evoluția stării rețelei până în momentul atingerii unei stări staționare (care va fi interpretată ca răspunsul rețelei). Orice rețea recurentă poate fi modelată printr-un *sistem dinamic* a cărui evoluție poate fi descrisă în timp discret (prin intermediul unui sistem de ecuații cu diferențe - relații de recurență) sau în timp continuu (prin intermediul unui sistem de ecuații diferențiale).

Evoluție în timp discret. Este adecvată situației în care rețeaua va fi simulată soft. Să considerăm că la un moment dat t , starea rețelei este determinată de către vectorul $Y(t) = (y_1(t), \dots, y_N(t))$ al semnalelor pe care le produc unitățile. Există două strategii de modificare a stării rețelei:

- *Asincronă.* La un moment dat o singură unitate (i^*) își schimbă starea, astfel că evoluția stării rețelei poate fi descrisă de:

$$\begin{cases} y_i(t+1) = f\left(\sum_{j=1}^N w_{ij}y_j(t) + I_i(t)\right) & i = i^* \\ y_i(t+1) = y_i(t) & i \neq i^* \end{cases} \quad (1)$$

Alegerea unității i^* se poate efectua într-o manieră aleatoare (prin selecție fără revenire a unui indice din $\{1, \dots, N\}$) sau într-o manieră secvențială (prima dată se selectează unitatea 1, după aceea unitatea 2 ș.a.m.d).

- *Sincronă.* Unitățile își schimbă simultan starea:

$$y_i(t+1) = f\left(\sum_{j=1}^N w_{ij}y_j(t) + I_i(t)\right), \quad i = \overline{1, N} \quad (2)$$

O astfel de evoluție poate fi implementată cu ușurință în paralel.

Pentru a putea implementa funcționarea rețelei, relațiile (1) și (2) trebuie completate cu starea inițială ($Y(0) = Y^0$) și cu o condiție de oprire de tipul $\|Y(t+1) - Y(t)\| < \epsilon$ ($\|\cdot\|$ reprezintă o normă în R^N iar $\epsilon > 0$ este o măsură a erorii apriori acceptată pentru aproximarea *punctului fix* al procesului iterativ).

Evoluție în timp continuu. Este adecvată situației în care rețeaua va fi implementată hard. În implementarea prin circuite electrice fiecare unitate este realizată dintr-un circuit care conține un amplificator (modelează funcția de transfer, f) precum și un rezistor (r_i) conectat în paralel cu un condensator (C_i) (vezi figura 2). Starea fiecărei unități, x_i , reprezintă tensiunea de intrare, iar y_i reprezintă tensiunea de ieșire din amplificator. Unitatea i este conectată cu unitatea j prin intermediul unui rezistor, a cărui rezistență, R_{ij} , determină ponderea conexiunii.

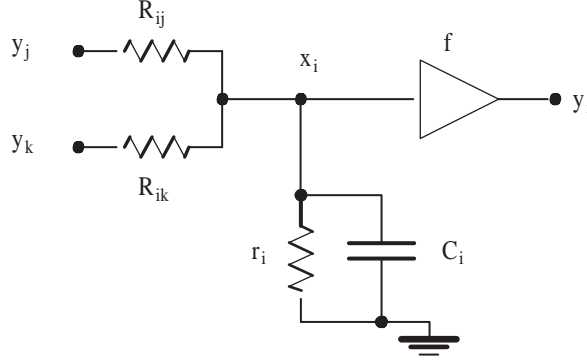


Figura 2: Circuit pentru implementarea unei unități și a conexiunii cu alte unități [2]

Cu aceste notații ecuațiile circuitului pot fi scrise:

$$\tau_i \frac{dx_i(t)}{dt} = -x_i(t) + \sum_{j=1}^N w_{ij} f(x_j(t)) + I_i(t), \quad i = \overline{1, N} \quad (3)$$

unde $\tau_i = R_i C_i$, $1/R_i = 1/r_i + \sum_{i=1}^N 1/R_{ij}$ și $w_{ij} = R_i/R_{ij}$. Pentru a simplifica relațiile se consideră că $\tau_i = 1$.

În ambele variante, pentru a putea determina răspunsul rețelei (astfel încât rețeaua să poată rezolva o sarcină computațională) este necesar ca procesul de evoluție să se *stabilizeze* într-o *stare staționară*, $X = (x_1, \dots, x_N)$, care verifică

$$x_i = \sum_{j=1}^N w_{ij} f(x_j) + I_i, \quad i = \overline{1, N}. \quad (4)$$

Dacă se consideră ca starea rețelei este reprezentată de vectorul semnalelor de ieșire, $Y = (y_1, \dots, y_N)$ atunci starea staționară verifică

$$y_i = f\left(\sum_{j=1}^N w_{ij} y_j + I_i\right), \quad i = \overline{1, N}. \quad (5)$$

Problema care apare este de a stabili în ce condiții rețeaua evoluează către o stare staționară și cum este influențat acest lucru de starea inițială. În domeniul sistemelor dinamice probleme de acest tip sunt abordate utilizând metoda Liapunov.

2 Proprietăți de stabilitate și funcții Liapunov

Comportarea unei rețele recurente, descrisă prin evoluția stării sale (de exemplu $X(t)$) se poate încadra într-un dintre situațiile:

- $X(t)$ tinde către o stare staționară X^* (numită și *punct fix* al dinamicii rețelei).
- $X(t)$ oscilează permanent între două sau mai multe stări posibile (comportare *periodică*).
- $\|X(t)\|$ devine din ce în ce mai mare pe măsură ce t crește.
- $X(t)$ are o comportare haotică.

Comportarea cea mai utilă, atât din perspectiva utilizării ca memorii asociative cât și pentru a rezolva probleme de optimizare este cea corespunzătoare primei situații. Din punct de vedere practic o proprietate utilă a stării staționare X^* este cea de *asimptotic stabilitate*. Din punct de vedere intuitiv, proprietatea de stabilitate poate fi ilustrată prin starea de echilibru a unei bile aflată pe o suprafață convexă (stare asimptotic stabilă), orizontală (stare stabilă) respectiv concavă (stare instabilă) (vezi figura 3).

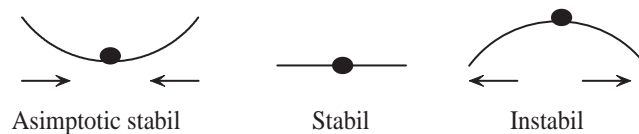


Figura 3: Proprietatea de stabilitate a unui punct de echilibru

Pentru a defini formal aceste noțiuni să considerăm un sistem descris prin

$$\frac{dX(t)}{dt} = F(X(t)), \quad t \geq 0 \quad (6)$$

iar $X^* = F(X^*)$ un punct fix al acestuia. Pentru sistemul (6) completat cu o condiție inițială $X(0) = X^0$ vom nota prin $X(t; X^0)$ unica soluție ce corespunde condiției. Asimptotic stabilitatea constă din două proprietăți:

Stabilitate:

X^* este stabilă dacă pentru orice $\epsilon > 0$ există $\delta(\epsilon) > 0$ astfel încât pentru orice X^0 cu $\|X^0 - X^*\| < \delta$ are loc $\|X(t; X^0) - X^*\| < \epsilon$ pentru orice $t > 0$.

Atractivitate:

X^* este atractivă dacă există $\delta > 0$ astfel încât pentru orice X^0 cu $\|X^0 - X^*\| < \delta$ are loc $\lim_{t \rightarrow \infty} X(t; X^0) = X^*$.

Mulțimea $\mathcal{A}(X^*) = \{X^0 \mid \lim_{t \rightarrow \infty} X(t; X^0) = X^*\}$ se numește *regiune de atracție* a lui X^* . Dacă $\mathcal{A}(X^*) = R^N$ atunci X^* este *global atractivă* (dacă are și proprietatea de stabilitate este *global asimptotic stabilă*) altfel este *local atractivă* (respectiv *local asimptotic stabilă*). Global asimptotic stabilitatea este o proprietate utilă în cazul problemelor de optimizare pe când local asimptotic stabilitatea este utilă în modelarea prin rețele recurente a memoriilor asociative.

Verificarea proprietăților de asimptotic stabilitate pe baza definițiilor este dificilă când soluția nu este cunoscută explicit (cum este cazul rețelelor neuronale). În acest caz un instrument util este metoda lui Liapunov la baza căreia stă următoarea teoremă.

Teorema de stabilitate a lui Liapunov. Dacă există o funcție $V : R^N \rightarrow R$ mărginită inferior astfel încât $\frac{dV(X(t))}{dt} < 0$ pentru orice $t > 0$ atunci orice soluție staționară, X^* , a lui (6) este asimptotic stabilă.

Funcția V se numește funcție Liapunov și este într-un fel analoagă noțiunii de energie din fizică, în sensul că X^* este un punct de minim al lui V iar o poziție de echilibru stabil (vezi figura 3) are asociată o energie minimă.

Din perspectiva utilizării rețelelor neuronale pentru rezolvarea problemelor de optimizare relevanța rezultatului de mai sus este următoarea: punctele de minim ale lui V sunt soluții staționare asimptotic stabile ale lui (6), ceea ce înseamnă că starea sistemului tinde în mod *natural* către ele. În felul acesta problema determinării minimelor lui V poate fi rezolvată *lăsând* sistemul să se stabilizeze (să se apropie de starea staționară).

Problema care apare, însă, este că Teorema de stabilitate a lui Liapunov nu asigură faptul că stările staționare sunt *singurele* soluții asimptotic stabile ale lui (6) existând posibilitatea ca sistemul să posede, de exemplu, soluții asimptotic stabile periodice. Aceasta înseamnă că nu am avea garanția că sistemul evoluează către ceea ce dorim. Această dificultate poate fi rezolvată utilizând un alt rezultat teoretic, Principiul de invarianță al lui LaSalle.

Pentru a enunța principiul vom considera următoarele definiții:

Funcție Liapunov în $G \subset R^n$. O funcție continuu diferențiabilă $V : R^n \rightarrow R$ se numește funcție Liapunov în G pentru (6) dacă derivata ei în virtutea sistemului

$$dV(X) \stackrel{not}{=} (\nabla V(X))^T F(X)$$

are semn constant pe G .

Mulțime invariantă. O mulțime $A \subset R^n$ se numește invariantă în raport cu (6) dacă pentru orice condiție inițială $X^0 \in A$ soluția corespunzătoare, $X(t; X^0)$ rămâne în A ($X(t; X^0) \in A$ pentru orice $t > 0$).

Principiul de invarianță al lui LaSalle. Dacă V este o funcție Liapunov pentru (6) în G atunci orice soluție $X(t; X^0)$ care rămâne în G fie devine nemărginită, fie tinde către cea mai mare mulțime invariantă inclusă în $Z = \{X \in \overline{G} \mid dV(X) = 0\}$, unde \overline{G} este închiderea lui G .

Rezultatele de mai sus au corespondent și pentru sistemele discrete de forma:

$$X(t+1) = -X(t) + F(X(t))$$

diferența principală fiind dată de faptul că derivata funcției Liapunov este înlocuită cu o diferență de forma: $V(X(t+1)) - V(X(t))$.

Principalele rezultate privind existența funcțiilor Liapunov pentru rețele neuronale sunt:

Funcție Liapunov pentru rețele cu dinamică discretă asincronă (1). Dacă $w_{ij} = w_{ji}$ pentru orice $i \neq j$, $w_{ii} = 0$ pentru orice i și f este funcția signum (cu valori în $\{-1, 1\}$) sau Heaviside (cu valori în $\{0, 1\}$) iar semnalul de intrare este constant ($I_i(t) = I_i$) atunci

$$V(y_1, \dots, y_N) = -\frac{1}{2} \sum_{i,j=1}^N w_{ij} y_i y_j - \sum_{i=1}^N y_i I_i \quad (7)$$

este funcție Liapunov pentru (1), singurele soluții asimptotic stabile fiind punctele staționare.

Funcție Liapunov pentru rețele cu dinamică continuă (3). Dacă $w_{ij} = w_{ji}$ pentru orice $i \neq j$, $\tau_i = 1$ pentru orice i , f este funcție diferentiabilă strict crescătoare, iar semnalul de intrare este constant ($I_i(t) = I_i$) atunci

$$V(x_1, \dots, x_N) = -\frac{1}{2} \sum_{i,j=1}^N w_{ij} f(x_i) f(x_j) - \sum_{i=1}^N f(x_i) I_i + \sum_{i=1}^N \int_0^{f(x_i)} f^{-1}(z) dz \quad (8)$$

este funcție Liapunov pentru (3), iar singurele soluții asimptotic stabile sunt punctele staționare.

3 Exemple de probleme de optimizare rezolvabile cu rețele neuronale

Problema comis voiajorului. Se consideră n orașe și un comis voiajor care trebuie să le viziteze pe toate, trecând o singură prin fiecare oraș. Se pune problema determinării unui circuit de cost cât mai mic (costul total al circuitului este determinat de suma costurilor trecerii de la un oraș la altul). Această formulare intuitivă corespunde de fapt problemei determinării unui circuit hamiltonian de cost minim într-un graf etichetat. Aplicații practice ale acestei probleme formale sunt: planificarea traseelor într-o firmă de mesajerie, stabilirea deplasărilor unui braț de robot sau proiectarea automată a circuitelor integrate.

Dacă orașele sunt numerotate de la 1 la n , o soluție a problemei este de fapt o permutare de ordin n . Două soluții care pot fi obținute una din alta prin transformări circulare nu sunt de fapt distincte. Pentru a avea o singură reprezentare a unui traseu este suficient să se fixeze orașul de pornire. Dacă acesta este fixat și nu are importanță sensul de parcurgere a orașelor numărul de trasee distincte este $(n-1)!/2$. O abordare exhaustivă iese din discuție pentru valori mari ale lui n .

Pentru a pregăti rezolvarea problemei cu rețele neuronale vom folosi pentru specificarea unui traseu o matrice, $Y = (y_{ij})_{i=\overline{1,n}, j=\overline{1,n}}$, cu n linii și n coloane. Fiecare linie corespunde unui oraș, iar fiecare coloană unei etape a traseului. Pe fiecare linie și pe fiecare coloană se află o singură valoare egală cu 1 celelalte fiind egale cu 0. Dacă $y_{ij} = 1$ atunci prin orașul i s-a trecut în etapa j . Pentru primul traseu din figura 4 matricea asociată este:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

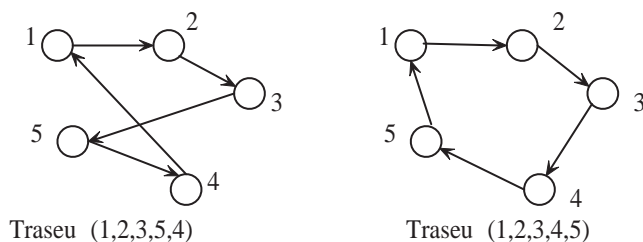


Figura 4: Trasee în problema comis voiajorului

în timp ce pentru al doilea matricea asociată este diagonală (matricea identitate).

Cu notațiile de mai sus problema revine la a determina configurația Y care satisface restricțiile:

$$\sum_{i=1}^n \left(\sum_{j=1}^n y_{ij} - 1 \right)^2 = 0 \quad \text{fiecare oraș } i \text{ este vizitat o singură dată} \quad (9)$$

$$\sum_{j=1}^n \left(\sum_{i=1}^n u_{ij} - 1 \right)^2 = 0 \quad \text{în fiecare etapă } j \text{ este vizitat un singur oraș} \quad (10)$$

și minimizează funcția de cost

$$C(Y) = \sum_{i=1}^n \sum_{k=1, k \neq i}^n \sum_{j=1}^n c_{ik} y_{ij} (y_{k,j-1} + y_{k,j+1}) \quad (11)$$

unde c_{ik} este costul trecerii de la orașul i la orașul k iar indicele j ia valori în $\{1, \dots, n\}$ într-o manieră circulară (dacă $j = 1$ atunci $j - 1 = n$ iar dacă $j = n$ atunci $j + 1 = 1$).

Problema alocării resurselor. Se consideră o mulțime de n locații care trebuie afectate la m concentratoare. Problema constă în găsirea unei afectări a locațiilor la concentratoare astfel încât să fie minimizat costul total și să nu fie depășită capacitatea nici unui concentrator. De exemplu, locațiile pot reprezenta clienți (terminale), concentratoarele pot reprezenta servere (mașini gazdă) iar costul conexiunii între locația i și concentratorul j poate fi distanța d_{ij} dintre cele două. Problema de optimizare enunțată mai sus este una cu restricții, acestea fiind:

- Fiecare locație $i \in \{1, \dots, n\}$ este conectată la un singur concentrator.
- Fiecare concentrator $j \in \{1, \dots, m\}$ este conectat la maxim c_j locații (c_j este numită capacitatea concentratorului j).

Să considerăm pentru fiecare conexiune posibilă o variabilă binară $y_{ij} \in \{0, 1\}$, definită astfel:

$$y_{ij} = \begin{cases} 1 & \text{dacă } i \text{ este conectat la } j \\ 0 & \text{dacă } i \text{ nu este conectat la } j \end{cases}$$

Dacă costul afectării locației i la concentratorul j este p_{ij} atunci costul total al unei afectări este:

$$T = \sum_{i=1}^n \sum_{j=1}^m y_{ij} p_{ij}.$$

Problema este să se determine mn valori $y_{ij} \in \{0, 1\}$ astfel încât restricțiile (a) și (b) să fie satisfăcute și costul total T să fie minimizat. Evident, există m^n afectări posibile care satisfac (a) (numărul funcțiilor definite pe $\{1, \dots, n\}$ cu valori în $\{1, \dots, m\}$) deci tehnicile de căutare exhaustivă sunt nepractice dacă m și n au valori mari.

4 Formularea neuronală a problemelor de optimizare

Idea de bază în minimizarea unei funcții obiectiv este de a proiecta o rețea de tip Hopfield (se aleg funcțiile de transfer adecvate, ponderile conexiunilor și semnalele de intrare) astfel încât funcția Liapunov asociată rețelei să coincidă cu funcția obiectiv. În felul acesta dinamica rețelei va conduce în mod natural către un punct de minim al funcției obiectiv.

Să considerăm cazul mai general al unei probleme de minimizare cu restricții. Presupunem că funcția obiectiv este $C : D^N \rightarrow R$ cu D o mulțime finită (de exemplu, $D = \{0, 1\}$) iar variabilele y_1, \dots, y_N verifică restricțiile

$$R_k(y_1, \dots, y_N) = 0, \quad k \in \{1, \dots, r\}$$

cu $R_k : D^N \rightarrow R$ funcții nenegative.

Pentru a o rezolva cu ajutorul rețelelor neuronale, problema trebuie formulată ca una fără restricții. Pentru aceasta se poate folosi *metoda penalizării* prin care se construiește funcția obiectiv ca o combinație liniară a funcției obiectiv inițiale și a funcțiilor R_k care definesc restricțiile. Coeficienții combinației liniare se aleg astfel încât să reflecte importanța satisfacerii fiecărei restricții în raport cu optimizarea funcției obiectiv. Astfel se construiește

$$C^*(y_1, \dots, y_N) = aC(y_1, \dots, y_N) + \sum_{k=1}^r b_k R_k(y_1, \dots, y_N), \quad a > 0, \quad b_k > 0. \quad (12)$$

Pentru a determina parametrii rețelei se reorganizează (12) sub forma unei funcții Liapunov (7) și se procedează la identificarea coeficienților termenilor de același tip. Acest lucru este posibil doar pentru problemele de optimizare în care funcția obiectiv și restricțiile conțin termeni de grad cel mult doi.

Astfel prin rescrieri de forma

$$C(y_1, \dots, y_N) = -\frac{1}{2} \sum_{i,j=1}^N w_{ij}^{obj} y_i y_j - \sum_{i=1}^N I_i^{obj} y_i,$$

$$R_k(y_1, \dots, y_N) = -\frac{1}{2} \sum_{i,j=1}^N w_{ij}^k y_i y_j - \sum_{i=1}^N I_i^k y_i, \quad k = \overline{1, r}$$

parametrii rețelei vor fi obținuți prin identificare:

$$w_{ij} = a w_{ij}^{obj} + \sum_{k=1}^r b_k w_{ij}^k,$$

$$I_i = a I_i^{obj} + \sum_{k=1}^r b_k I_i^k.$$

Proiectarea rețelei pentru problema comis voiajorului. Se va utiliza o rețea cu $N = n^2$ unități de tip Heaviside. Funcția obținută prin introducerea restricțiilor este:

$$C^*(Y) = \frac{a}{2} \sum_{i=1}^n \sum_{k=1, k \neq i}^n \sum_{j=1}^n c_{ik} y_{ij} (y_{k,j-1} + y_{k,j+1}) + \frac{b}{2} \left(\sum_{i=1}^n \left(\sum_{j=1}^n y_{ij} - 1 \right)^2 + \sum_{j=1}^n \left(\sum_{i=1}^n y_{ij} - 1 \right)^2 \right) \quad (13)$$

Identificând C^* (fără termenul liber) cu

$$V(Y) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n w_{ij,kl} y_{ij} y_{kl} - \sum_{i=1}^n \sum_{j=1}^n y_{ij} I_{ij}. \quad (14)$$

se obțin parametrii rețelei:

$$w_{ij,kl} = -ac_{ik}(\delta_{l,j-1} + \delta_{l,j+1}) - b(\delta_{ik} + \delta_{jl} + \delta_{ik}\delta_{jl}) \\ I_{ij} = 2b,$$

cu $\delta_{ij} = 1$ dacă și numai dacă $i = j$, în celelalte situații fiind 0. Dacă $c_{ik} = c_{ki}$ pentru orice $i \neq k$ rezultă că $w_{ij,kl} = w_{kl,ij}$, adică sunt satisfăcute ipotezele teoremei de stabilitate.

Proiectarea rețelei pentru problema alocării resurselor. Să considerăm cazul când toate concentratoarele au aceeași capacitate, $c_j = c$, $j \in \{1, \dots, m\}$. În acest caz vom utiliza $N = (n + c) * m$ unități total interconectate, fiecare având funcția Heaviside ($f(u) = 1$ dacă $u \geq 0$ iar $f(u) = 0$ pentru $u < 0$). Pentru a fi mai clar modul de interpretare a ieșirilor produse de unități le vom organiza într-un tabel bidimensional (o matrice cu $(n + c)$ linii și m coloane). Astfel, fiecare unitate va fi identificată printr-un cuplu (i, j) iar ponderea conexiunii dintre (k, l) și (i, j) va fi notată cu $w_{ij,kl}$. Semnificația unităților este:

- Pentru $i \leq n$, unitatea (i, j) (aparținând liniei i și coloanei j) reprezintă ipoteza: "locația i este asignată concentratorului j ". Dacă notăm cu $y_{ij} \in \{0, 1\}$ valoarea produsă de unitatea (i, j) , o valoare egală cu 1 implică faptul că ipoteza de mai sus este adevărată, iar o valoare egală cu 0 faptul că este falsă.
- Unitățile de pe liniile $(n + 1) \dots (n + c)$ sunt unități adiționale care au fost introduse pentru a permite restricției (b) să fie exprimată printr-o egalitate: pe fiecare coloană se află exact c unități având ieșirea egală cu 1.

Dinamica rețelei prezentată mai sus poate fi descrisă prin:

$$y_{ij}(t + 1) = f\left(\sum_{k=1}^{n+c} \sum_{l=1}^m w_{ij,kl} y_{kl}(t) + I_{ij}\right),$$

unde I_{ij} specifică semnalul de intrare primit de unitatea (i, j) .

Dacă conexiunile dintre unități sunt simetrice ($w_{ij,kl} = w_{kl,ij}$) și f este funcția Heaviside, atunci rezultă că funcția Liapunov asociată sistemului este de forma:

$$V(Y) = -\frac{1}{2} \sum_{i=1}^{n+c} \sum_{j=1}^m \sum_{k=1}^{n+c} \sum_{l=1}^m w_{ij,kl} y_{ij} y_{kl} - \sum_{i=1}^{n+c} \sum_{j=1}^m y_{ij} I_{ij}. \quad (15)$$

Pe de altă parte, problema de optimizare cu restricții poate fi rescrisă ca o problemă de optimizare fără restricții introducând funcția obiectiv:

$$\begin{aligned}
V(Y) = a \sum_{i=1}^n \sum_{j=1}^m y_{ij} p_{ij} + b_1 \sum_{i=1}^n (\sum_{j=1}^m y_{ij} - 1)^2 + \\
+ b_2 \sum_{j=1}^m (\sum_{i=1}^{n+c} y_{ij} - c)^2
\end{aligned} \tag{16}$$

Semnificația celor trei termeni din relația (16) este:

- Primul termen este chiar costul asignării.
- Valoarea celui de-al doilea termen este minimă când restricția (a) este satisfăcută: fiecare locație este conectată la un singur concentrator (suma ieșirilor produse de unitățile aflate pe o linie este 1). Conține doar valorile de ieșire produse de unitățile efective.
- Valoarea celui de-al treilea termen este minimă numai atunci când fiecărui concentrator i se asociază cel mult c locații (suma ieșirilor produse de toate unitățile de pe o coloană, inclusiv cele adiționale este c).

Parametrii rețelei ($w_{ij,kl}$, I_{ij}) pot fi determinați prin identificare între relațiile (15) și (16). În urma calculelor se obține:

$$w_{ij,kl} = \begin{cases} b_1 \delta_{ik} (1 - \delta_{jl}) + b_2 \delta_{jl} (1 - \delta_{ik}) + \frac{a \delta_{jl} (1 - \delta_{ik})}{p_{ij} + p_{kl}} & i \leq n \text{ și } k \leq n \\ 0 & i > n \text{ sau } k > n \end{cases}$$

și

$$I_{ij} = \begin{cases} 2c & i \leq n \\ 2c - 1 & i > n \end{cases}$$

5 Problema minimelor locale și particularitățile variantelor stohastice

Derivând formal (7) (fără a ține cont că variabilele iau valori discrete) în raport cu câte o variabilă se observă că (1) este de fapt un proces iterativ de tip gradient asociat funcției obiectiv (7).

Din acest motiv una dintre principalele probleme a dinamicilor de tipul (1) și (2) este faptul că se blochează în minime locale (soluții sub-optimale ce pot fi neacceptabile). O modificare a dinamicii care conduce la "evadarea" din minimele locale constă în introducerea unor perturbații aleatoare. Acest lucru poate fi realizat prin înlocuirea unităților cu funcționare deterministă cu unități cu funcționare aleatoare. De exemplu o unitate cu valori în $\{0, 1\}$ poate fi înlocuită cu una caracterizată prin distribuția de probabilitate:

$$P(y_i = 1) = \frac{1}{1 + \exp(-\beta x_i)} \quad P(y_i = 0) = \frac{1}{1 + \exp(\beta x_i)}, \quad \beta \geq 0 \tag{17}$$

unde x_i reprezintă semnalul de intrare în unitate (obținut prin cumularea ponderată a semnalelor de ieșire de la toate unitățile) iar β este un parametru care controlează forma distribuției de probabilitate. O interpretare specifică i se poate da lui $T = 1/\beta$, considerat a fi o "pseudo-temperatură". Dacă $\beta = 0$ ($T = \infty$) atunci unitatea produce 1 sau 0 cu probabilitatea 0.5 (fără a mai conta valoarea semnalului de intrare). Corespunde situației de "agitație termică" prezentă într-un sistem de particule aflat la o temperatură foarte mare. Dacă $\beta \rightarrow \infty$ ($T = 0$) atunci $y_i = 1$

dacă $x_i > 0$ și $y_i = 0$ dacă $x_i < 0$ adică o comportare similară unităților deterministe cu funcție de transfer de tip Heaviside.

Valoarea medie a variabilei aleatoare y_i poate fi exprimată în funcție de x_i prin

$$M(y_i) = 1 \cdot P(y_i = 1) + 0 \cdot P(y_i = 0) = \frac{1}{1 + \exp(-\beta x_i)} \quad (18)$$

adică o funcție de transfer de tip logistic. Cu cât β este mai mare cu atât funcția logistică se apropie mai mult de funcția Heaviside. Într-o manieră similară se pot defini unități aleatoare ce iau valori în $\{-1, 1\}$, caz în care prin medierea semnalului de ieșire se ajunge la o funcție de transfer de tip tangentă hiperbolică ($f(x) = (\exp(2\beta x) - 1)/(\exp(2\beta x) + 1)$).

În cazul rețelelor cu unități aleatoare se poate cel mult determina distribuția de probabilitate staționară. Pentru o rețea cu unități de tipul (17) și strategie asincronă de activare se poate arăta că distribuția staționară este de forma:

$$P(Y) = \frac{1}{Z} \exp(-\beta V(Y)) \quad (19)$$

unde Z este o constantă de normalizare, iar V este dată de (7). Relația (19) sugerează că în starea de "echilibru" minimelor lui V le corespunde probabilitatea cea mai mare. O modalitate de implementare a dinamicii aleatoare este descrisă în figura 5.

Inițializări: $t = 0$, $y_i = rand(0, 1)$, $i = \overline{1, N}$

Proces iterativ:

REPETĂ

 Selectează i^* ($i^* = rand(\{1, 2, \dots, N\})$)

 Calculează $P = 1/(1 + \exp(-\beta \sum_j w_{i^*j} y_j + I_{i^*}))$

 Generează $r = rand(0, 1)$

 DACĂ $r < P$ ATUNCI $y_{i^*} = 1$ ALTFEL $y_{i^*} = 0$

 Incrementează contorul procesului iterativ: $t = t + 1$

PÂNĂ CÂND *este satisfăcut un criteriu de oprire* (de exemplu $t > t_{max}$)

Figura 5: Simularea unei rețele cu dinamică aleatoare (β constant)

O problemă delicată este alegerea lui β :

- O valoare prea mică corespunde unei comportări aleatoare necontrolată (are avantajul că permite evadarea din punctele de minim local însă nu asigură stabilizarea algoritmului într-un minim căci toate configurațiile au aceeași probabilitate).
- O valoare prea mare apropie dinamica de cea deterministă astfel că sistemul poate să rămână blocat în minime locale.

O soluție naturală pare a fi modificarea lui β pe parcursul algoritmului: o valoare inițială mică urmată de o mărire în cadrul procesului iterativ. Folosind analogia fizică o astfel de modificare a lui β corespunde aplicării unui "tratament termic" în care inițial sistemul este adus la o temperatură ridicată după care este răcit treptat. Este ideea de bază a algoritmilor ce simulează tratamente termice ("simulated annealing"). Rămâne deocamdată deschisă problema alegerii strategiei de "racire" ("cooling schedule"). O strategie posibilă de modificare a lui β este $\beta(t) = \ln(t + 1)$.

Deși permite evitarea minimelor locale, dinamica aleatoare are dezavantajul de a fi lentă. O soluție o reprezintă înlocuirea unităților aleatoare cu variantele lor "mediate", unități cu funcții de transfer de tip logistic (sau tanh). Transformarea dinamicii aleatoare în varianta sa "mediată" se bazează pe o teorie ("mean field theory") în ale cărei detalii nu intrăm (vezi [2]). Din punct de vedere al implementării, diferențele sunt ilustrate în figura 6.

Inițializări: $t = 0$, $\beta(0) = \beta^0$, $y_i = rand(0, 1)$, $i = \overline{1, N}$
Proces iterativ:
 REPETĂ
 Selectează i^* ($i^* = rand(\{1, 2, \dots, N\})$)
 Calculează $y_{i^*} = 1/(1 + \exp(-\beta \sum_j w_{i^*j} y_j + I_{i^*}))$
 Incrementează contorul procesului iterativ: $t = t + 1$
 Calculează noua valoare a lui $\beta(t)$
 PÂNĂ CÂND *este satisfăcut un criteriu de oprire* (de exemplu $\beta(t) > \beta_{max}$)

Figura 6: Simularea unei rețele cu dinamică mediată (β variabil) - "Mean Field Annealing"

Din punct de vedere intuitiv, varianta continuă permite urmarea unei traiectorii în $[0, 1]^n$ cu scopul final de a se apropia de un "vârf" al hipercubului. În felul acesta pot fi găsite "scurtături" către soluție în loc de a urma o traiectorie în $\{0, 1\}^n$.

În momentul în care procesul iterativ s-a terminat este necesar să se verifice admisibilitatea configurației obținute întrucât ca urmare a modului de tratare a restricțiilor acestea pot fi încălcate. Dacă se lucrează cu unități continue în funcția obiectiv se mai adaugă un termen care să favorizeze configurațiile cu componente apropiate de 0 sau de 1. De exemplu la TSP sau la problema alocării resurselor se mai adaugă un termen de forma: $\sum_{i=1}^n \sum_{j=1}^n y_{ij}(1 - y_{ij})$.

6 Despre eficiența rezolvării cu rețele neuronale a problemelor de optimizare

Deși primele rezultate obținute de Hopfield și Tank păreau incurajatoare, ulterior s-a observat experimental că pentru dimensiuni mari (cazul de interes) rețelele neuronale (proiectate pe baza sugestiilor lui Hopfield și Tank) nu se comportă mai bine decât unele metode tradiționale (de exemplu algoritmul Lin-Kernighan pentru TSP). În plus includerea restricțiilor în funcția obiectiv și necesitatea alegerii coeficienților de ponderare a termenilor ridică alte probleme (se găsesc configurații de cost bun dar care nu sunt admisibile, încălcând restricțiile sau se găsesc configurații admisibile dar de cost inacceptabil). Totuși abordarea Hopfield-Tank are caracter general și a deschis posibilitatea rezolvării prin implementări hard a problemelor de optimizare.

După 1985 au fost propuse o serie de îmbunătățiri ale modelului Hopfield-Tank în vederea extinderii capacității [4]:

- Înlocuirea unităților binare cu unități ce pot lua mai multe valori. Permite rezolvarea problemelor în care configurațiile posibile nu pot (sau nu este eficient) să fie descrise în manieră binară. De exemplu la TSP în loc să se utilizeze o matrice $n \times n$ se utilizează un vector cu n elemente în care pe poziția i se află indicele orașului vizitat în etapa i . O altă problemă

este cea a partiționării mulțimii nodurilor unui graf în K submulțimi astfel încât numărul conexiunilor între nodurile aflate în submulțimi diferite să fie cât mai mic.

- Utilizarea în locul metodei penalizării pentru includerea restricțiilor în funcția obiectiv a metodei multiplicatorilor lui Lagrange. În acest caz coeficienții care controlează ponderea termenilor în funcția obiectiv sunt ajustați în cadrul procesului iterativ.
- Dezvoltarea unor modalități de tratare a restricțiilor de tip inegalitate (ca în problema alocării resurselor sau în problema rucsacului)
- Modificarea funcției obiectiv astfel încât singurele configurații asimptotic stabile să fie cele admisibile și optime [3].
- Modificarea dinamicii clasice astfel încât să fie evitate configurațiile nedorite [5]

Referințe

- [1] M.T. Hagan, H.B. Demuth, M. Beale; Neural Network Design, PWS Publ. Company, 1996.
- [2] J. Hertz, A. Krogh, R.G. Palmer; Introduction to the Theory of Neural Computation, Addison Wesley Publ. Company, 1991.
- [3] S. Matsuda; An "Optimal" Hopfield Network for Combinatorial Optimization and its Approximate Realization, IEICE Trans. Fundam, vol. E83-A, no. 6, 2000, p.1211-1221.
- [4] C. Peterson, B. Söderberg; Neural Optimization, LU TP 02-30 (preprint, Dept. of Theoretical Physics, Lund University), to appear in The Handbook of Brain Theory and Neural Networks (2nd ed.) - M.A.Arbib (ed.)
- [5] X. Zeng, T. Martinez; A New Relaxation Procedure in the Hopfield Network for Solving Optimization Problems, Neural Processing Letters 10: 211222, 1999.