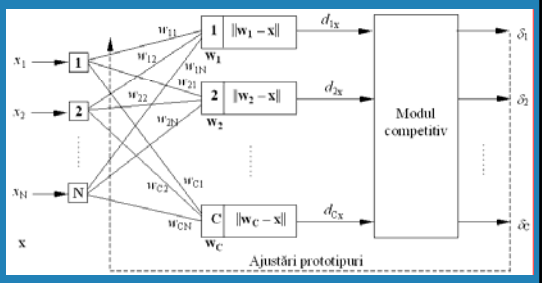
SISTEME CU IA

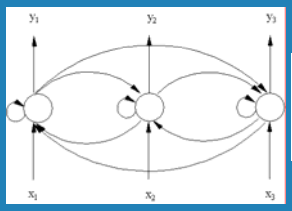
RNA PENTRU CLASIFICARE SI RECUNOASTERE

Retele competitive

Arhitectura de principiu



Modulul competitiv – rețele MaxNet / MinNet



$$y_j^{(t)} = max\left(0, x_j^{(t-1)}\right)$$

$$x_j^{(t)} = y_j^{(t-1)} - \alpha \cdot \sum_{\substack{k=1 \\ k \neq j}}^{N} y_k^{(t-1)}$$

Algoritm generic pentru antrenarea rețelelor competitive

1 Date de intrare: modelele de antrenare, sub forma vectorilor de intrare $\mathbf{x}^{(m)}$ (m = 1,...,M); numărul de clase C; rata de învățare η_c; factorul de adaptare F_c.
2 Inițializarea prototipurilor rețelei:
for c = 1 to C do
$\mathbf{w}_c$ = random( )
3 Adaptarea prototipurilor neuronilor-clasă:
for m = 1 to M do
// Pentru fiecare model de antrenare ...
for c = 1 to C do
$D_{m,c} = || \mathbf{x}^{(m)} - \mathbf{w}_c ||$

Algoritm generic pentru antrenarea rețelelor competitive

// c^* este neuronul clasă câștigător
// Adaptarea prototipurilor
for c = 1 to C do
$\mathbf{w}_c = \mathbf{w}_c + \eta_c \cdot F_c \cdot (\mathbf{x}^{(m)} - \mathbf{w}_c)$

4. Criteriul de oprire: dacă criteriul de oprire este satisfăcut, algoritmul se încheie; în caz contrar, se revine la pasul 3.

RNA pentru clasificare si recunoastere

RETELE KOHONEN

Retele Kohonen

Trei tipuri importante

(a) rețeaua *cuantificator vectorial* sau VQ (*vector quantization*);
(b) rețeaua LVQ (*Learning Vector Quantization*) și
(c) rețeaua pentru *hărți de trăsături cu auto-organizare* sau SOFM (*Self-Organizing Feature Maps*).

Retele Kohonen - reteaua VQ

Algoritmul VQ se aplică unor rețele competitive in care fiecare neuron corespunde unei clase şi este caracterizat printr-un *prototip* format din ponderile conexiunilor care îl leagă de neuronii de intrare.

Pentru fiecare model de antrenare aplicat pe inrarea rețelei se determină prototipul *câştigător* şi se deplasează acest prototip către modelul respectiv:

$$\mathbf{w}_c = \mathbf{w}_c + \eta \cdot (\mathbf{x}^{(m)} - \mathbf{w}_c)$$

Retele Kohonen - reteaua VQ

Algoritmul VQ este foarte asemănător cu *algoritmul c-medii* , cea mai importantă excepție fiind că, în ultimul caz rata de învățare este înlocuită de inversa numărului de modele care au fost asociate prototipului câştigător. În plus, calculul prototipului se face direct, folosind toate modelele asociate lui:

$$\mathbf{w}_c = \frac{1}{n(c)} \cdot \sum_{m=1}^{M} \mathbf{x}^{(m)} \Big|_{Clasa(m)=c}$$

Retele Kohonen - reteaua LVQ

Pentru prima data, a*lgoritmul LVQ* a fost folosit pentru comprimarea imaginilor si a informatiei, in general.

Se spune că o categorie de informații caracterizată de un vector cu N_1 componente este *comprimată* dacă unei clase de informații de acest tip îi poate fi asociată o clasă descrisă de un alt vector cu N_2 componente, cu proprietatea $N_2 < N_1$.

Retele Kohonen - reteaua LVQ

Rețelele de tip LVQ sunt similare rețelelor Kohonen SOFM, cu deosebirea că folosesc o antrenare supravegheată**.**

Pentru fiecare vector de intrare $\mathbf{x}^{(m)}$ se cunoaște clasa căreia aparține acesta c^* sau un vector de ieșire dorit $\mathbf{d}^{(m)}$. Astfel, se realizează comprimarea vectorului $\mathbf{x}^{(m)}$ de dimensiune N_1, în vectorul $\mathbf{d}^{(m)}$ de dimensiune N_2 ($N_2 < N_1$).

Retele Kohonen - reteaua LVQ

Principiu

Pentru fiecare model de antrenare $\mathbf{x}^{(m)}$ se calculează distanțele față de prototipurile existente:

$$D_{m,c} = \|\mathbf{x}^{(m)} - \mathbf{w}_c\| \quad (c = 1,\ldots,C)$$

și se alege distanța minimă D_{m,c^*}, corespunzătoare neuronului clasă c^* căreia aparține modelul $\mathbf{x}^{(m)}$.

Clasa c^* este comparată cu valoarea dorită $d^{(m)}$ şi pe baza acestei comparatii se realizeaza adaptarea prototipurilor.

Retele Kohonen - reteaua LVQ

Principiu - continuare

Dacă $c^* = d^{(m)}$, prototipul $\mathbf{w}_{c^*}$ este adaptat pozitiv. Dacă dimpotrivă, $c^* \neq d^{(m)}$, prototipul $\mathbf{w}_{c^*}$ este adaptat negativ.

Adaptarea pozitiva (apropierea prototipului $\mathbf{w}_{c^*}$ de modelul curent $\mathbf{x}^{(m)}$):

$$\mathbf{w}_{c^*} = \mathbf{w}_{c^*} + \eta \cdot (\mathbf{x}^{(m)} - \mathbf{w}_{c^*})$$

Adaptarea negativa(indepartarea prototipului $\mathbf{w}_{c^*}$ de modelul curent $\mathbf{x}^{(m)}$):

$$\mathbf{w}_{c^*} = \mathbf{w}_{c^*} - \eta \cdot (\mathbf{x}^{(m)} - \mathbf{w}_{c^*})$$

Retele Kohonen - reteaua LVQ

Algoritm

1. Date de intrare: modelede antrenare - vectori de intrare $\mathbf{x}^{(m)}$ şi clase asociate $d^{(m)}$, $m = 1,\ldots,M$; numărul de clase C; rata de învăţare η.
2. Iniţializare prototipuri reţea - primele C modele din setul de antrenare:
   for $c = 1$ to C do $\mathbf{w}_c = \mathbf{x}^{(c)}$
3. Adaptarea prototipurilor neuronilor-clasă:
   for $m = 1$ to M do
   // Calculul distanţelor de la modelul $\mathbf{x}^{(m)}$ la prototipurile $\mathbf{w}_c$
   for $c = 1$ to C do
   $D_{m,c} = ||\mathbf{x}^{(m)} - \mathbf{w}_c||$
   // Selectarea neuronului-clasă câştigător
   $D_{min} = D_{m,1}$; $c^* = 1$;
   for $c = 2$ to C do

Retele Kohonen - reteaua LVQ

Algoritm - continuare

for $c = 2$ to C do

if $D_{m,c} < D_{min}$ then
$D_{min} = D_{m,c}$
$c^* = c$

// Compară c^* cu $d^{(m)}$ şi adaptează prototipurile

if $c^* = d^{(m)}$ then $\mathbf{w}_{c^*} = \mathbf{w}_{c^*} + \eta \cdot (\mathbf{x}^{(m)} - \mathbf{w}_{c^*})$
else $\mathbf{w}_{c^*} = \mathbf{w}_{c^*} - \eta \cdot (\mathbf{x}^{(m)} - \mathbf{w}_{c^*})$

4. Criteriul de oprire: dacă criteriul de oprire este satisfăcut, algoritmul se încheie; în caz contrar, se revine la pasul 3.

Retele Kohonen - reteaua SOFM

Inspiratie biologica

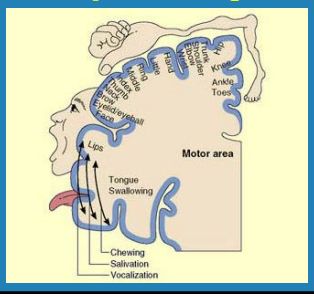
Maparea segmentelor senzoriale in creier :

Relatiilor spatiale intre stimuli le corespund relatii spetiale intre neuroni.

Pe cortex se formeaza o imagine *deformata* a corpului uman

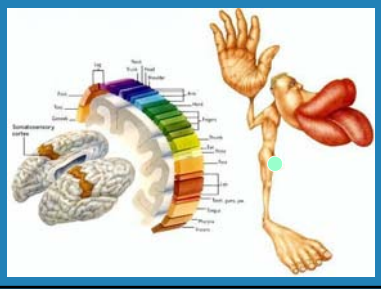
Retele Kohonen - reteaua SOFM

Inspiratie biologica



Retele Kohonen - reteaua SOFM

Inspiratie biologica



Retele Kohonen - reteaua SOFM

Rețelele SOFM au fost concepute pentru ***învățarea nesupravegheată***, numită uneori şi ***auto-organizare***.

Setul de date de antrenare conține numai mărimi de intrare, iar rețeaua SOFM învață structura datelor de intrare.

Auto-organizarea

… capacitatea unui sistem de a descoperi şi învăța structura datelor de intrare, chiar şi în absența unor informații despre această structură.

Retele Kohonen - reteaua SOFM

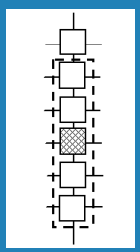
Rețeaua Kohonen (SOFM) este o extindere a învățării competitive standard la noțiunea de ***hărți de trăsături***.

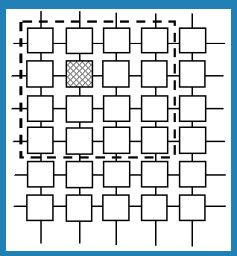
O hartă de trăsături se obține dacă la neuronii dintr-o rețea cu învățare competitivă se adaugă o anumită formă de organizare topologică.

Neuronii se dispun în nodurile unei ***grile suport***, de obicei bi-dimensionale, dar uneori uni-dimensională sau chiar tri şi chiar multi-dimensională.

Retele Kohonen - reteaua SOFM

Grila suport si neuronii -clasa





Retele Kohonen - reteaua SOFM

Particularitate SOFM vs VQ (LVQ)

Adaptarea prototipurilor are loc ***nu numai*** pentru neuronul câştigător, ci şi pentru o parte din neuronii-clasă din reţea, care se află într-o anumită ***vecinătate a neuronului câştigător***.

Vecinătatea unui neuron-clasă ***c***, notată V_c mulţimea neuronilor-clasă dispuşi în nodurile grilei-suport la o distanţă faţă de neuronul-clasă ***c*** mai mică decât o anumită valoare prag.

Retele Kohonen - reteaua SOFM

Invatarea – 2 strategii complementare:
- competitia - cooperarea

Competitia

Toţi neuronii-clasă concurează pentru dreptul de a învăţa. Implementarea competiţiei se face ca în orice alt algoritm competitiv: fiecare vector $\mathbf{x}^{(m)}$ din setul de antrenare este comparat cu prototipurile $\mathbf{w}_c$ asociate neuronilor-clasă şi se determină neuronul câştigător c^* şi poziţia acestuia pe grila-suport.

Retele Kohonen - reteaua SOFM

Invatarea – 2 strategii complementare:
- competitia - cooperarea

Cooperarea

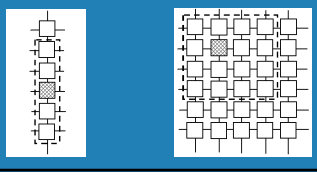
După stabilirea neuronului câştigător c^*, acesta nu îşi adaptează prototipul de unul singur, ci împreună cu neuronii-clasă care se situează în vecinătatea sa V_{c^*}.

Retele Kohonen - reteaua SOFM

Tipuri de vecinatati
- discrete - continue

Vecinatati discrete

Se foloseste direct positia in grila suport.



Retele Kohonen - reteaua SOFM

Tipuri de vecinatati
- discrete - continue

Vecinatati continue

Se foloseşte o funcţie de vecinătate, având de regulă o formă gaussiană:

$$V_c = e^{-\rho_c^2 / 2\sigma^2}$$

unde σ^2 - *dispersie*, ρ_c -distanţa dintre neuronul-clasă c şi neuronul câştigător c^*:

$$\rho_c = |P_c - P_{c^*}|$$

Retele Kohonen - reteaua SOFM

Adaptarea ponderilor:
- unilaterala - bilaterala

Adaptarea unilaterala

Se face adaptarea ponderilor numai pentru neuronii din vecinătatea V_{c^*}, prin apropierea prototipurilor acestora de modelul curent $\mathbf{x}^{(m)}$.

$$\mathbf{w}_c = \mathbf{w}_c + \eta \cdot (\mathbf{x}^{(m)} - \mathbf{w}_c) \quad c \in V_{c^*}$$

$$\mathbf{w}_c = \mathbf{w}_c \quad c \notin V_{c^*}$$

Retele Kohonen - reteaua SOFM

Adaptarea ponderilor:
- unilaterala - bilaterala

Adaptarea unilaterala

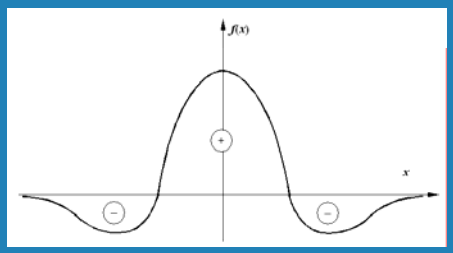
Neuronii-clasă din vecinătatea neuronului câştigător sunt *stimulaţi*, iar neuronii-clasă care rămân în afara vecinătăţii neuronului câştigător sunt *penalizaţi:*

$$\mathbf{w}_c = \mathbf{w}_c + \eta^+ \cdot (\mathbf{x}^{(n)} - \mathbf{w}_c) \quad c \in V_{c*}$$

$$\mathbf{w}_c = \mathbf{w}_c - \eta^- \cdot (\mathbf{x}^{(n)} - \mathbf{w}_c) \quad c \notin V_{c*}$$

Retele Kohonen - reteaua SOFM

Inhibare laterala



Retele Kohonen - reteaua SOFM

Adaptarea vecinatatilor – faza initiala

Iniţial, vecinătatea fiecărui neuron este foarte largă, cuprinzând deseori practic toţi neuronii-clasă din reţea. În aceasta fază, vecinătăţile sunt relativ largi, astfel incat prototipurile unui număr mare de neuroni-clasă sunt apropiate, iar neuronii-clasă vecini acţionează asemănător.

Retele Kohonen - reteaua SOFM

Adaptarea vecinatatilor – faza intermediara

Dupa depasirea fazei initiale, aceste vecinătăți sunt diminuate treptat.

Pe măsură ce vecinătățile se micşorează, prototipurile încep să se diferențieze şi se formează un model de organizare a informației de intrare.

Retele Kohonen - reteaua SOFM

Adaptarea vecinatatilor – faza finala

Spre finalul procesului, când vecinătățile scad la 0 (fiecare vecinătate conține numai neuronul-clasă central), adaptarea prototipurilor se face la un număr din ce în ce mai mic de neuroni-clasă, iar procesul de diversificare se accentuează.

Retele Kohonen - reteaua SOFM

Adaptarea ratei de invatare

Inițial, în faza de ordonare, rata de învățare se menține la valori ridicate, apropiate de unitate, pentru a permite orientarea de principiu a tuturor prototipurilor din rețea către diferitele modele din setul de antrenare. Ulterior, în etapa de convergență, când modificarea prototipurilor trebuie să se facă în zone din ce în ce mai apropiate de centrul grupării de vectori asociați unui neuron-clasă, rata de învățare se reduce.

Retele Kohonen - reteaua SOFM

Algoritm

1. Date de intrare: modelele de antrenare $\mathbf{x}^{(m)}$, $m = 1,\ldots,M$; numărul de neuroni-clasă C; rata de învățare η și dispersia σ^2; factorii de descreștere pentru rata de învățare (δ_η) și dispersie (δ_σ).
2. Inițializarea prototipurilor rețelei cu valori aleatorii, în intervalul (0 , 1):
   for $c = 1$ to C do $\mathbf{w}_c$ = random( )
3. Adaptarea prototipurilor neuronilor-clasă:
   for $m = 1$ to M do
   // Calculul distanțelor de la modelul $\mathbf{x}^{(m)}$ la prototipurile $\mathbf{w}_c$
   for $c = 1$ to C do
   $D_{m,c} = ||\mathbf{x}^{(m)} - \mathbf{w}_c||$

Retele Kohonen - reteaua SOFM

Algoritm

// Selectarea neuronului-clasă câștigător
$D_{min} = D_{m,1}$; $c^* = 1$;
for $c = 2$ to C do
if $D_{m,c} < D_{min}$ then
$D_{min} = D_{m,c}$
$c^* = c$
// Calculul funcției de vecinătate pentru neuronii-clasă
for $c = 2$ to C do
if $c \neq c^*$ then $V_c = e^{-\rho_c^2 / 2\sigma^2}$
// Adaptarea prototipurilor
for $c = 1$ to C do
$\mathbf{w}_c = \mathbf{w}_c + \eta \cdot V_c \cdot (\mathbf{x}^{(m)} - \mathbf{w}_c)$

Retele Kohonen - reteaua SOFM

Algoritm

4. Criteriul de oprire: dacă condiția de oprire este satisfăcută, algoritmul se încheie. În caz contrar, se trece la pasul 5.
5. Îngustarea vecinătăților în etapa de ordonare:
   $\sigma^2 = \sigma^2 \cdot \delta_\sigma$
6. Reducerea ratei de învățare în etapa de convergență:
   $\eta = \eta \cdot \delta_\eta$
7. Se revine la pasul 3.