**UNIVERSITATEA "POLITEHNICA" BUCUREŞTI**

**Facultatea de Electronică şi Telecomunicații**

**Angel Cațaron**

# REȚELE NEURALE PENTRU RECUNOAŞTEREA VORBIRII

**Rezumatul tezei de doctorat**

**Conducător ştiințific:**

**Prof. dr. ing Victor-Emil Neagoe**

**BUCUREŞTI**

**2003**

# Cuprins


Introducere ..... 3
Structura lucrării ..... 3
Recunoaşterea vorbirii cu modele clasice de rețele neurale ..... 4
Experimente ..... 5
Recunoaşterea cuvintelor ..... 5
Recunoaşterea vorbitorilor ..... 5
Concluzii ..... 6
Rețele neurale concurente ..... 6
Rețele neurale concurente ..... 7
Recunoaşterea vorbitorilor cu rețele neurale concurente. Experimente ..... 9
Recunoaşterea vocalelor cu rețele neurale concurente. Experimente ..... 9
Rezultate experimentale ..... 10
Concluzii ..... 11
Recunoaştere prin calculul relevanțelor intrărilor folosind operatori OWA ..... 12
Algoritmul Relevance LVQ (RLVQ) ..... 13
Operatorii OWA ..... 13
Algoritmul OWA - Relevance LVQ (OWA-RLVQ) ..... 14
Experimente ..... 15
Concluzii ..... 17
Recunoaştere prin calculul relevanțelor intrărilor folosind noțiuni de teoria informației ..... 17
Informația mutuală pătratică ..... 18
Algoritmul Mutual Information Relevance LVQ (MIRLVQ) ..... 19
Algoritmul Weighted LVQ ..... 19
Adaptarea factorilor de relevanță ..... 19
Energia informațională ..... 20
Algoritmul Energy Relevance LVQ (ERLVQ) ..... 20
Experimente ..... 21
Concluzii ..... 22
Contribuții originale ..... 23
Recunoaşterea vorbirii cu modele clasice de rețele neurale ..... 23
Rețele neurale concurente ..... 23
Recunoaştere prin calculul relevanțelor intrărilor folosind operatori OWA ..... 24
Recunoaştere prin calculul relevanțelor intrărilor folosind noțiuni de teoria informației ..... 24
Bibliografie ..... 26

Această lucrare investighează, pe de o parte, *cum poate fi implementat conceptul de concurență la nivelul unei colecții de rețele neurale* și, pe de altă parte, *modul în care importanța diferită a intrărilor influențează performanțele în recunoaștere ale unor tipuri de rețele neurale*. Recunoașterea formelor, domeniu în care se plasează și subiectul acestei teze, oferă un câmp foarte larg cercetării. Recunoașterea vorbirii grupează o gamă diversă de probleme precum recunoașterea cuvintelor izolate, identificarea vorbitorilor pe baza amprentei lor vocale, recunoașterea fonemelor etc. Rețelele neurale sunt un instrument care și-a demonstrat eficiența în rezolvarea unei palete largi de aplicații între care și cele de recunoaștere automată a vorbirii.

Majoritatea modelelor neurale abordează recunoașterea formelor ca o problemă unitară, globală, fără a face distincția între aportul individual diferit al intrărilor. Este cunoscut că modularitatea și principiul *divide et impera* aplicate rețelelor neurale le pot îmbunătăți performanțele. Introducem modelul *Concurrent Neural Networks (CNN)* care îmbină paradigmele învățării supervizate și nesupervizate și oferă o soluție primei probleme. Ideea concurenței este folosită la nivelul unei colecții de rețele neurale care sunt instruite independent pentru a rezolva subprobleme diferite. Recunoașterea se face prin identificarea rețelei neurale care furnizează cel mai bun răspuns. Așa cum o demonstrează și rezultatele experimentale, acuratețea recunoașterii este mai mare atunci când folosim modelul propus de noi față de cazurile în care nu folosim concurența.

A doua problemă are conotații mai largi deoarece se regăsește și în aplicații de *data mining*. Prin asocierea unei ponderi fiecărei intrări, se stabilește intensitatea cu care intrările influențează răspunsul sistemului și implicit capacitatea sa de recunoaștere corectă. Dezvoltăm algoritmul *Ordered Weighted Aggregation - Relevance LVQ (OWA-RLVQ)* care calculează ponderile sub forma unui set de coeficienți *Ordered Weighted Aggregation (OWA)*. Algoritmii *Mutual Information Relevance LVQ (MIRLVQ)* și *Energy Relevance LVQ (ERLVQ)* propuși în continuare stabilesc acești coeficienți prin maximizarea informației mutuale pe de o parte, și a unei mărimi definite cu ajutorul energiei informaționale, pe de altă parte. Avantajele algoritmilor noștri sunt confirmate experimental prin rate de recunoaștere în mod constant superioare față de alți algoritmi care rezolvă același tip de probleme.

Mulțumesc d-lui Prof. dr. ing. Victor-Emil Neagoe care în calitate de conducător științific m-a sprijinit constant prin colaborarea fructuoasă de-a lungul acestor ani, prin discuțiile pline de substanță și prin sfaturile fără de care nu aș fi putut finaliza această teză.

De asemenea, mulțumesc tuturor colegilor care au influențat într-un fel sau altul conținutul acestei teze. Mulțumesc profesorilor care mi-au transmis pasiunea și dorința de a progresa. Multe mulțumiri prietenilor care m-au încurajat să merg mai departe în timpul perioadei de pregătire a tezei.

Mulțumesc părinților pentru că prin educația pe care mi-au dat-o și prin modul în care m-au crescut m-au ajutat să ajung până aici.

Mulțumesc familiei care m-a susținut în toată această perioadă, mi-a oferit suportul moral, motivația de care am avut nevoie și care m-a învățat ca nimic nu este greu pentru că nu sunt singur.

Brașov, septembrie 2003

# Introducere

Această lucrare abordează o temă recunoaştere a formelor folosind reţele neurale, cu aplicaţii în recunoaşterea vorbirii. Sunt introduse mai multe modele originale de recunoaştere, iar dezvoltările teoretice şi rezultatele experimentale demonstrează viabilitatea acestora. Spectrul aplicaţiilor de recunoaştere a vorbirii abordate în teză este larg, începând cu recunoaşterea fonemelor, continuând cu recunoaşterea cuvintelor izolate şi terminând cu recunoaşterea vorbitorilor. Bazele de date sunt diverse, fiind vorba atât de colecţii standard pentru limba engleză cât şi de seturi alcătuite special pentru experimentele din această lucrare şi conţin cuvinte din limba română. Am testat unele modele propuse de noi şi pe seturi de date standard în recunoaşterea formelor, cu scopul de a ilustra faptul că ele pot fi folosite şi în alte contexte decât cel al recunoaşterii vorbirii.

## Structura lucrării

Teza este organizată în şapte capitole care conţin părţi teoretice şi descrieri ale experimentelor de recunoaştere a formelor, iar contribuţiile originale sunt argumentate în detaliu. Experimentele de recunoaştere vocală sunt diverse, pornind de la recunoaşterea cuvintelor izolate, continuând cu recunoaşterea vorbitorilor şi terminând cu recunoaşterea vocalelor. Bazele de date folosite acoperă o gamă largă, unele fiind publice, altele alcătuite special pentru experimentele din această lucrare. în unele experimente am utilizat şi baze de date care nu sunt destinate strict recunoaşterii vorbirii, demonstrând astfel că algoritmii propuşi pot fi folosiţi şi pentru probleme generale de recunoaştere a formelor.

Capitolul 2 face o prezentare a tipurilor fundamentale de reţele neurale şi a tehnicilor de instruire care stau la baza modelelor introduse în celelalte capitole.

În capitolul 3 este prezentat un set de experimente de recunoaştere a cuvintelor izolate şi a vorbitorilor folosind trei modele clasice de reţele neurale: perceptronul multistrat, reţeaua neurală cu întârziere în timp şi harta cu autoorganizare. Cele două baze de date, Speechdata şi Telephdata, pe care le-am folosit în aceste teste constau din cuvinte izolate pronunţate de diverşi vorbitori mai întâi în condiţii de studio, fără perturbaţii, şi apoi prin telefon. Pentru preprocesarea semnalelor vocale se foloseşte o cunoscută metodă de calcul al coeficienţilor cepstrali. Problema instruirii şi cea a recunoaşterii informaţiei din aceste două baze de date sunt deosebit de complicate şi rezultatele experimentale demonstrează acest lucru. Recunoaşterea cuvintelor izolate conduce la rezultate satisfăcătoare în majoritatea cazurilor, însă recunoaşterea vorbitorilor ridică probleme deosebite.

Reţelele neurale concurente introduse în capitolul 4 reprezintă un model conexionist de recunoaştere care porneşte de la observaţia că o reţea neurală este mai eficientă atunci când implementează o problemă mai puţin complexe sau când este specializată pe o subproblemă a problemei globale de recunoaştere. Modelul propus în acest capitol este o colecţie de reţele neurale care lucrează în paralel, iar decizia în recunoaştere se ia conform regulii *câştigătorul ia tot*. Fiecare reţea neurală componentă este instruită individual cu un set propriu de date. Fiind vorba de module care rezolvă probleme cu o complexitate intrinsecă mai scăzută, numărul de neuroni pentru fiecare modul poate fi redus semnificativ. Am instruit reţele neurale concurente având drept unităţi elementare perceptroni multistrat, reţele neurale cu întârziere în timp şi hărţi cu autoorganizare pentru aceeaşi problemă de recunoaştere a vorbitorilor din capitolul 3, iar performanţele pentru cele două baze de date au crescut semnificativ. Cu acelaşi model am implementat şi recunoaşterea unui subset de şase vocale din limba română, baza de date fiind alcătuită prin segmentarea manuală a semnalelor vocale, în scopul extragerii secvenţelor vocalizate. Preprocesarea constă din calculul coeficienţilor cepstrali care sunt vectorii de instruire. Reţeaua neurală concurentă care foloseşte hărţi cu autoorganizare specializate pe recunoaşterea câte unei vocale are performanţe mai bune decât o hartă unică, instruită cu toate vocalele.

În capitolul 5, problema recunoaşterii este abordată dintr-o perspectivă diferită pentru că ne punem problema creşterii performanţelor prin selectarea acelor caracteristici ale vectorilor de instruire şi de test care contribuie cel mai mult la o recunoaştere corectă. Fiecărei componente a vectorilor de intrare i se asociază câte un coeficient care reprezintă importanţa acestuia, iar coeficienţii sunt instruiţi în aşa fel încât să minimizeze eroarea de cuantizare a unei reţele neurale de tip LVQ. Algoritmul RLVQ calculează printr-o tehnică euristică aceşti coeficienţi. Operatorii OWA sunt o familie de operatori de agregare care se bazează pe reordonarea criteriilor care trebuie satisfăcute într-un proces de clasificare. Algoritmul OWA-RLVQ propus de noi calculează dinamic relevanţele intrărilor ca ponderi ale unui operator OWA, făcând totodată o legătură importantă între algoritmul RLVQ şi modelul matematic al operatorilor OWA. Performanţele algoritmului nostru sunt comparate cu ratele de recunoaştere obţinute cu LVQ şi RLVQ şi rezultatele înregistrate de OWA-RLVQ sunt superioare pentru toate cele trei baze de date utilizate în teste. Dovedim astfel că metoda noastră este competitivă atât pentru probleme de recunoaştere vocală cât şi pentru probleme generale de recunoaştere a formelor.

Capitolul 6 introduce două noi metode de calcul al relevanţelor pentru componentele vectorilor de intrare cu ajutorul informaţiei mutuale pe de o parte şi a energiei informaţionale pe de altă parte. Estimarea pe baza unor date de instruire a informaţiei mutuale definită cu ajutorul entropiei Shannon ridică probleme computaţionale serioase. Se pot folosi metode alternative cum ar fi cea care calculează entropia Renyi pornind de la estimarea densităţilor de probabilitate cu ferestre Parzen. Informaţia mutuală pătratică dintre două variabile aleatoare obţinută fie prin similaritatea cu entropia Renyi pătratică, fie pe baza inegalităţii lui Pinsker devine un criteriu care poate fi maximizat iterativ într-o problemă de gradient pozitiv. Algoritmul MIRLVQ propus de noi calculează iterativ relevanţele vectorilor de intrare maximizând informaţia mutuală pătratică dintre o variabilă aleatoare continuă ale cărei realizări sunt calculate pe baza datelor de instruire şi o variabilă aleatoare discretă ale cărei realizări sunt clasele din care fac parte vectorii de instruire. Energia informaţională este o mărime pe baza căreia se calculează măsura $o(X,Y)$, unde $X$ şi $Y$ sunt două variabile aleatoare, care este similară informaţiei mutuale. Al doilea algoritm propus de noi, numit ERLVQ, calculează relevanţele componentelor din vectorii de intrare prin maximizarea măsurii $o(X,Y)$ şi obţinem o formulă de actualizare similară celei din MIRLVQ. Experimentele realizate cu aceleaşi baze de date folosite în capitolul 5 arată că noua metodă este superioară din punct de vedere al ratei de recunoaştere algoritmului OWA-RLVQ.

Capitolul 7 care este o sinteză prezintă contribuţiile pe care teza le aduce la dezvoltarea domeniului recunoaşterii formelor şi posibilele direcţii viitoare de dezvoltare a rezultatelor obţinute în lucrare.

# Recunoaşterea vorbirii cu modele clasice de reţele neurale

*Acest capitol prezintă un studiu comparativ asupra performanţelor în recunoaşterea cuvintelor izolate şi a vorbitorilor a perceptronilor multistrat, a reţelelor neurale cu întârziere în timp şi a hărţilor cu autoorganizare. Au fost folosite două baze de date care conţin informaţii prelucrate prin analiză cepstrală după înregistrarea semnalelor vocale mai întâi într-un mediu fără perturbaţii, iar apoi după ce au fost transmise printr-o linie telefonică. Cu toate că pentru recunoaşterea cuvintelor izolate rezultatele au fost bune, testele de recunoaştere a vorbitorilor au demonstrat că aceste modele nu oferă întotdeauna soluţii satisfăcătoare.*

Problemele computaţionale asociate recunoaşterii vorbirii şi succesul limitat al diverselor tehnici convenţionale au condus la dezvoltarea unor abordări bazate pe reţelele neurale. Metodele descrise în literatură diferă în special prin modul în care sunt convertite semnalele vocale în diverse formate cu scopul de a fi utilizate ca intrări ale reţelelor neurale, prin categoriile de aplicaţii implementate, de exemplu recunoaşterea cuvintelor sau a fonemelor, recunoaştere dependentă sau independentă de vorbitor etc. şi prin tipul de reţea neurală folosită.

Este dificil să comparăm în mod direct rezultatele obţinute de diverşi autori pentru că în majoritatea cazurilor acestea se bazează pe seturi particulare de date. Este de dorit ca pentru toate

modelele testate să folosim aceleaşi date de instruire, obţinând astfel o imagine mult mai fidelă a performanţelor lor. Scopul acestui capitol este de a prezenta o comparaţie a rezultatelor obţinute în recunoaşterea vorbirii prin trei tehnici diferite [66]: perceptronul multistrat (MLP), reţeaua neurală cu întârziere în timp (TDNN) şi harta cu autoorganizare (SOM), pentru aceeaşi bază de date. Pentru că aceste tipuri de reţele neurale sunt foarte cunoscute, le-am asociat atributul "clasice".

În testele de recunoaştere a vorbirii cu MLP, TDNN şi SOM am folosit două baze de date numite Speechdata şi Telephdata care au fost realizate la Northern Ireland BioEngineering Center (NIBEC). Ele cuprind pronunţiile în limba engleză ale cifrelor 0 până la 9 la care au fost adăugate *nought*, *oh* şi *zero* care sunt variante ale cifrei 0. Pentru crearea fiecărei baze de date au fost alese câte 25 de persoane care au repetat cele 12 cuvinte de câte 20 de ori, rezultând un număr total de 6000 de pronunţii.

## Experimente

### Recunoaşterea cuvintelor

Cele două baze de date conţin câte 12 cuvinte pronunţate de câte 25 de vorbitori. Fiecare vorbitor a repetat fiecare cuvânt de câte 20 de ori, astfel încât atât Speechdata cât şi Telephdata conţin câte 6000 de repetiţii. De fiecare dată primele 8 repetiţii ale fiecărui cuvânt au fost folosite la instruire şi ultimele 12 la test. Datorită metodei de creare a bazelor de date, vectorii au fost organizaţi sub forma unor matrici de 15x18 componente, deci reţelele neurale folosite au avut câte 270 de intrări.

Atât pentru perceptronul multistrat cât şi pentru reţelele neurale cu întârziere în timp am folosit configuraţii cu un singur strat ascuns. Numărul de neuroni de pe stratul ascuns a fost de aproximativ 100 de neuroni, cu mici variaţii care nu au influenţat semnificativ capacitatea de învăţare. Pentru că numărul de cuvinte este 12, am stabilit ca acesta să fie şi numărul de neuroni de ieşire pentru MLP şi TDNN. Fiecare ieşire se activează pentru câte o singură clasă de vectori, generând o ieşire apropiată de 1. Ieşirile care nu se activează generează valori apropiate de 0. în faza de instruire am folosit chiar valorile 1 şi 0 pentru ieşirile dorite, iar la testare am aceptat un prag de 0,3. Aceasta înseamnă că o ieşire între 0,7 şi 1 este asimilată lui 1, iar una între 0 şi 0,3 este 0. Hărţile cu autoorganizare folosite pentru recunoaşterea cuvintelor au avut 17x15 noduri. Instruirea a avut loc în două etape. Pentru prima, cea de organizare a hărţii, am rulat 10000 de paşi ai algoritmului, iar pentru a doua, cea a accentuării zonelor, am parcurs 120000 de paşi. Prin calibrare se stabileşte pe baza distanţei euclidiene clasa căreia îi aparţine fiecare neuron. Aceasta se realizează statistic, prin prezentarea la intrările hărţii instruite a unor vectori despre care se ştie din ce clase fac parte. Nodurile rămase neetichetate nu sunt folosite în faza de test.

Rezultatele experimentelor de recunoaştere a cuvintelor pentru fiecare tip de reţea neurală şi pentru ambele baze de date sunt prezentate în Tabelul 1.

Tabelul 1. Rezultatele experimentelor de recunoaştere a cuvintelor.

| | MLP | TDNN | SOM |
|---|---|---|---|
| Speechdata | 93,3% | 95,1% | 97,1% |
| Telephdata | 76,3% | 90,5% | 96,8% |

### Recunoaşterea vorbitorilor

O problemă mai dificilă pe care am implementat-o cu modelele clasice de reţele neurale a fost recunoaşterea vorbitorilor. Ne-am propus să instruim MLP şi TDNN cu vectori grupaţi după criteriul identităţii persoanei. Astfel, cuvintele din cele două baze de date au fost separate în 25 de clase care corespund celor 25 de vorbitori. Acum nu ne mai interesează care sunt cuvintele pe care le-au pronunţat subiecţii pentru că această informaţie devine nerelevantă. Repetarea, însă, a aceloraşi cuvinte produce redundanţa necesară algoritmilor de instruire care pot extrage trăsăturile dominante ale vocii fiecărui vorbitor. Răspunsurile dorite ale vectorilor de instruire au fost ca şi la recunoaşterea cuvintelor secvenţe binare cu o componentă care are valoarea 1 şi care arată clasa din

care face parte vectorul, iar restul au valoarea 0. Fiind vorba de 25 de clase, stratul de ieşire al reţelelor neurale a fost redimensionat la această valoare. Stratul ascuns a crescut la 150 de noduri deoarece am constatat experimental că este soluţia optimă. În cazul instruirii SOM am optat pentru o reţea de 30x30 de noduri care a fost calibrată cu vectori din cele 25 de clase.

Sinteza rezultatelor obţinute pentru acest set de experimente se găseşte în Tabelul 2.

Tabelul 2. Rezultatele experimentelor de recunoaştere a vorbitorilor.

| | MLP | TDNN | SOM |
|---|---|---|---|
| Speechdata | 71,2% | 57,2% | 52,8% |
| Telephdata | 6,7% | 32,33% | 44,7% |

## Concluzii

Rezultatele experimentale prezentate în acest capitol creează o imagine a capacităţii unor modele clasice de reţele neurale de a rezolva probleme de recunoaştere a vorbirii. Recunoaşterea cuvintelor izolate din Speechdata a condus la rezultate satisfăcătoare. La recunoaşterea cuvintelor din Telephdata, perceptronul multistrat a avut o comportare foarte slabă. Modul de construire a bazei de date a permis MLP obţinerea unui rezultat apropiat de TDNN, chiar dacă structura reţelei neurale nu permite extragerea unor informaţii legate de secvenţa în timp a cadrelor ce alcătuiesc un cuvânt. Performanţele SOM arată potenţialul crescut al reţelelor neurale cu autoorganizare. Studiind ratele de recunoaştere, putem trage concluzia că modelul standard MLP nu este potrivit pentru recunoaşterea eficientă a cuvintelor izolate.

Pe de altă parte, rezultatele obţinute la testele de recunoaşterea vorbitorilor arată că modelele clasice sunt insuficiente. Ratele de succes de până la 50% dovedesc faptul că ele nu sunt capabile să extragă caracteristicile vocale ale indivizilor într-o măsură suficient de bună şi este nevoie de implementarea unor tehnici care să răspundă mai bine cerinţelor. Pentru a îmbunătăţi performanţele reţelelor neurale în recunoaşterea vorbitorilor, în capitolul următor vom introduce reţelele neurale concurente, un nou model de recunoaştere care foloseşte ideea de modularitate şi principiul *divide et impera*.

# Reţele neurale concurente

*Propunem în acest capitol un nou model neural de recunoaştere numit Concurrent Neural Networks (CNN) care constă dintr-o colecţie de reţele neurale, iar decizia la clasificare se ia prin regula câştigătorul ia tot. Fiecare reţea neurală componentă a sistemului este instruită separat pentru a răspunde corect intrărilor dintr-o singură clasă. Am aplicat acest model la recunoaşterea vorbitorilor şi a vocalelor folosind perceproni multistrat, reţele neurale cu întârziere în timp şi hărţi cu autoorganizare ca module ale sistemului concurent.*

Experimentele de recunoaştere a vorbitorilor descrise în capitolul anterior au demonstrat că modelele clasice pe care le-am testat nu oferă întotdeauna rezultate mulţumitoare. Perceptronul multistrat conduce uneori la rezultate mai slabe datorită faptului că arhitectura sa are un caracter general şi nu exploatează caracteristicile semnalului vocal. Performanţele acestor reţele scad foarte mult pentru recunoaşterea vorbitorilor. în experimentele noastre, numărul claselor folosite la recunoaşterea vorbitorilor este dublu faţă de cel folosit pentru recunoaşterea cuvintelor. Reducând numărul de vorbitori, se constată, însă, că performanţele rămân inferioare celor stabilite la recunoaşterea cuvintelor, ceea ce arată că problema este una mai complexă.

Pe de altă parte, este evident că reducerea numărului de vorbitori provoacă îmbunătăţirea comportamentului reţelelor neurale prin creşterea ratei de succes pentru că problema pe care o implementează devine mai puţin dificilă. Problema globală poate fi redusă la o sumă de subprobleme prin separarea vorbitorilor în grupuri mai mici în interiorul cărora recunoaşterea este mai uşor de realizat.

*Rețelele neurale modulare* [30] au la bază ideea că, pe lângă nivelele ierarhice de organizare a rețelelor neurale artificiale: sinapse, neuroni, straturi de neuroni şi rețeaua însăşi, se poate crea un nou nivel care combină mai multe rețele neurale. Modularitatea are o justificare neurobiologică, fiind un principiu important în arhitectura sistemelor nervoase ale vertebratelor. De altfel, sistemul vizual oferă un bun exemplu de calcul modular deoarece imaginile percepute sunt descompuse în imagini de dimensiuni reduse care sunt analizate separat, iar rezultatele individuale sunt apoi combinate.

Haykin [30] arată că o rețea neurală se numeşte modulară atunci când calculul realizat de rețea poate fi descompus în două sau mai multe module care operează pe seturi distincte de intrări, fără a comunica între ele. Ieşirile modulelor sunt mediate de o unitate de integrare care nu permite transmiterea informației înapoi către module. Unitatea de integrare trebuie să decidă cum se combină ieşirile modulelor pentru a forma ieşirea finală a sistemului şi trebuie să decidă ce seturi de vectori trebuie învățate de fiecare modul. Modularitatea poate fi privită ca o formă de implementare a principiului *divide et impera* care permite rezolvarea problemelor complexe prin descompunerea acestora în subprobleme şi combinarea soluțiilor individuale. Rețele neurale modulare au fost studiate în diverse contexte. Rueckl, Cave şi Kosslyn pun în [74] problema identificării unor obiecte cunoscute şi a găsirii locației acestora în spațiu. S-a constatat că timpul de instruire a fost mai scurt atunci când s-au folosit două rețele neurale în locul uneia singure. Tehnica numită *connectionist glue* propusă de Waibel, Sawai şi Shikano [91] presupune instruirea individuală a rețelelor neurale pentru diferite probleme de recunoaştere a vorbirii urmată de regruparea lor. Jacobs, Jordan şi Barto [34] instruiesc un sistem modular în care modulele concurează pentru a învăța diverşi vectori particulari. Anand, Mehora, Mohan şi Ranka [1] propun o metodă prin care problema clasificării vectorilor în $K$ clase este transformată în $K$ probleme de clasificare a vectorilor în două clase. Modulele decid dacă un vector aparține unei clase sau nu. Algoritmul backpropagation este modificat pentru a suplini diferența mare dintre exemplele pozitive şi cele negative pe care o presupune această soluție.

Modelul propus de noi [11] numit *Concurrent Neural Networks (CNN)* introduce o nouă tehnică neurală de recunoaştere care se bazează pe ideea competiției între mai multe rețele neurale care lucrează în paralel, decizia finală luându-se pe baza regulii *câştigătorul ia tot*. Numărul de rețele folosit este egal cu numărul de clase în care sunt grupați vectorii, iar instruirea este supervizată. Fiecare rețea este proiectată să recunoască corect vectori dintr-o singură clasă, astfel că răspunsurile cele mai bune apar doar atunci când îi sunt prezentați vectori din clasa cu care a fost instruită. Acest model este de fapt un cadru care oferă flexibilitate arhitecturii fiindcă modulele pot fi reprezentate de diverse tipuri de rețele neurale.

Pornind de la modelul CNN din [11], Neagoe [60] a dezvoltat şi experimentat modelul *Concurrent Self-Organizing Maps (CSOM)* care se detaşează ca o tehnică nouă cu performanțe excelente. A fost demonstrată importanța CSOM pentru cazul general al vectorilor cu repartiții normale multimodale şi s-a arătat că poate fi folosit cu succes pentru clase largi de probleme de recunoaştere a formelor. CSOM a fost aplicat cu succes de Neagoe [60] pentru recunoaşterea fețelor umane şi de Neagoe şi Ropot în clasificarea imaginilor satelitare [61], în aplicații de clasificare a formelor [62], pentru recunoaşterea fețelor umane şi la recunoaşterea vorbitorilor [64].

Ideea competiției dintre hărțile cu autoorganizare a fost utilizată ulterior şi de Lapidot, Guterman şi Cohen în [51] pentru recunoaşterea nesupervizată a vorbitorilor. Fiecare vorbitor este modelat printr-o hartă, în timp ce un algoritm iterativ permite ajustarea hărților.

Acest capitol prezintă noul model al rețelelor neurale concurente şi analizează performanțele sale pentru recunoaşterea vorbitorilor şi a vocalelor.

## Rețele neurale concurente

Schema generală folosită pentru instruirea rețelelor neurale concurente este cea din Figura 1. În această schemă, $n$ reprezintă numărul de rețele neurale care lucrează în paralel, dar este egal totodată şi cu numărul de clase în care sunt grupați vectorii de instruire.

Mulţimea $X$ de vectori este obţinută în urma preprocesării semnalelor vocale achiziţionate în scopul instruirii reţelelor. Din această mulţime sunt extrase seturile de vectori $X_j$, $j$=1,2 ... $n$ cu care vor fi instruite cele $n$ reţele neurale. În urma aplicării procedurii de învăţare, fiecare reţea neurală va trebui să răspundă pozitiv unei singure clase de vectori şi să dea răspunsuri negative pentru toţi celelalţi vectori. Algoritmul de instruire pentru reţeaua concurentă este următorul:

**Etapa 1.** *Se creează baza de date. Aceasta conţine vectorii de instruire obţinuţi în urma preprocesării semnalului vocal.*

**Etapa 2.** *Se extrag din baza de date seturile de vectori specifice fiecărei reţele neurale în parte. Dacă este cazul, se stabilesc ieşirile dorite.*

**Etapa 3.** *Se aplică algoritmul de instruire fiecărei reţele neurale folosind seturile de vectori create la pasul 2.*

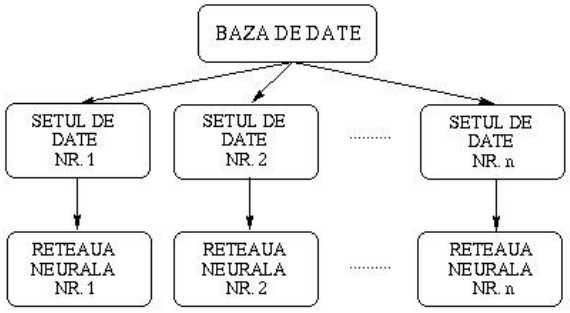


Figura 1. Modelul folosit la instruirea reţelelor neurale concurente.

Recunoaşterea se desfăşoară în paralel, conform schemei din Figura 2.

Cele $n$ reţele neurale se presupune că au fost instruite prin algoritmul descris mai înainte. La aplicarea vectorului de test, reţelele generează câte un răspuns individual, iar selecţia constă în alegerea reţelei care a generat răspunsul cel mai puternic. Reţeaua selectată prin regula *câştigătorul ia tot* este declarată câştigătoare. Indicele reţelei câştigătoare va fi indicele clasei în care este plasat vectorul de test. Această metodă de recunoaştere a formelor presupune, aşadar, că este *a priori* cunoscut numărul de clase cu care va lucra reţeaua concurentă şi că pentru fiecare clasă există suficienţi vectori de instruire. Algoritmul de recunoaştere este cel de mai jos.

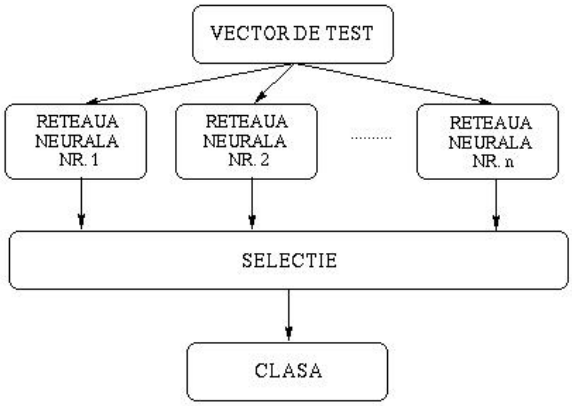


Figura 2. Modelul folosit la recunoaştere de reţelele neurale concurente.

**Etapa 1.** *Este creat vectorul de test prin preprocesarea semnalului vocal.*

**Etapa 2.** *Vectorul de test este transmis în paralel tuturor celor n reţele neurale instruite în prealabil.*

**Etapa 3.** *Blocul de selecţie stabileşte indicele reţelei cu răspunsul celmai bun. Acesta va fi indicele clasei în care este încadrat vectorul.*

Modelele din figurile Figura 1 şi Figura 2 pot fi particularizate plasând în locul celor *n* reţele neurale diferite arhitecturi. În experimentele prezentate în această lucrare am folosit perceptroni multistrat (MLP), reţele neurale cu întârziere în timp (TDNN) şi hărţi Kohonen (SOM), obţinând astfel trei tipuri diferite de reţele neurale concurente.

## Recunoaşterea vorbitorilor cu reţele neurale concurente. Experimente

Experimentele descrise în această secţiune [11], [13] compară capacitatea de recunoaştere a MLP, TDNN, SOM şi a reţelelor neurale concurente care folosesc aceste tipuri de reţele neurale drept module de bază. Rezultatele prezentate aici au fost obţinute în urma testelor de recunoaştere a vorbitorilor folosind mai întâi baza de date Speechdata, iar apoi baza de date Telephdata care conţin cuvinte pronunţate de 25 de vorbitori pe care i-am organizat în 25 de clase.

Scopul testelor este acela de a recunoaşte cei 25 de vorbitori pe baza diverselor pronunţii ale aceloraşi cuvinte, diferite de cele folosite la instruire. Am adoptat ca bază de comparaţie rezultatele obţinute cu MLP, TDNN şi SOM descrise în secţiunea 3.3.2 şi care se găsesc şi în Tabelul 3 pe coloanele marcate cu litera B (model neural de bază).

Tabelul 3. Rezultatele experimentelor de recunoaştere a vorbitorilor cu MLP, MLP-CNN, TDNN, TDNN-CNN, SOM şi SOM-CNN.

| | Speechdata | | Telephdata | |
|---|---|---|---|---|
| | B | CNN | B | CNN |
| MLP | 71,25% | 86,22% | 6,75% | 72,86% |
| TDNN | 57,28% | 89,31% | 32,33% | 72,72% |
| SOM | 52,86% | 90,42% | 44,72% | 75,25% |

Modelul concurent ales de noi grupează 25 de reţele clasice, câte una pentru fiecare vorbitor. Metoda de instruire este descrisă în continuare, fiind supervizată pentru MLP şi TDNN şi nesupervizată pentru SOM. Etapa de recunoaştere impune alegerea celei mai performante reţele care este declarată câştigătoare. Setul de experimente descrise în acest subcapitol testează modelul reţelelor neurale concurente pentru recunoaşterea vorbitorilor.

În Tabelul 3 sunt prezentate ratele de recunoaştere pentru MLP-CNN, TDNN-CNN şi SOM-CNN, în coloanele marcate cu CNN.

Rezultatele obţinute la recunoaşterea vorbitorilor cu reţele neurale concurente sunt mult îmbunătăţite faţă de modelele clasice. MLP s-a dovedit a fi sensibil mai bun decât TDNN şi SOM în testele cu modelul clasic folosind Speechdata, dar mult mai slab pentru Telephdata. Folosirea reţelelor concurente a îmbunătăţit semnificativ ratele de recunoaştere. Modelul SOM-CNN s-a dovedit a fi cel mai performant, rezultatele fiind cu mai mult de 4% mai bune decât modelul MLP-CNN şi cu aproximativ 1% mai bune decât modelul TDNN-CNN. în general, s-a păstrat o diferenţă de aproximativ 15% între testele cu Speechdata şi cele cu Telephdata. Experimentele au arătat că folosind câte o reţea specializată pentru fiecare cuvânt se obţin rezultate mai bune la recunoaştere, chiar dacă faza de instruire durează în mod evident mai mult.

## Recunoaşterea vocalelor cu reţele neurale concurente. Experimente

După ce am văzut că modelul CNN aduce îmbunătăţiri remarcabile ratelor de recunoaştere a vorbitorilor în comparaţie cu modelele clasice, în această secţiune ne vom concentra asupra recunoaşterii unor vocale din limba română. Am creat o nouă bază de date procesând prin analiza cepstrală bazată pe scala mel a pronunţiilor în limba româna ale celor 10 cifre. Fiecare dintre cei 4 subiecţi a pronunţat aceste cuvinte de câte 10 ori. De fiecare dată am extras vectorii de instruire a sistemului de recunoaştere din primele 5 repetiţii, păstrând ultimele 5 repetiţii pentru teste. Am avut astfel la dispoziţie 200 de repetiţii pentru antrenarea sistemului şi alte 200 pentru verificarea sa.

Ca şi la recunoaşterea vorbitorilor, am realizat un set de experimente folosind mai întâi o singură reţea neurală pentru recunoaşterea tuturor vocalelor, iar apoi CNN. De data aceasta am

testat doar rețeaua Kohonen, date fiind rezultatele pe care le-am obţinut la recunoaşterea vorbitorilor şi i-am comparat comportamentul cu cel al unui SOM-CNN.

Pentru varianta care foloseşte un singur SOM am instruit rețeaua cu întreaga secvență de vectori obținuți după preprocesarea semnalelor vocale selectate. Am instruit în două etape o rețea Kohonen cu 10x15 noduri prin algoritmul SOFM descris în paragraful 2.9.2. Prima etapă, cea de organizare a clusterelor, s-a desfăşurat de-a lungul a 1000 de paşi şi vecinătatea a scăzut treptat până la un singur neuron. în a doua etapă, instruirea s-a făcut în 10000 de paşi, iar vecinătatea a rămas fixată la dimensiunea minimă. în urma instruirii şi a calibrării rețelei neurale cu vectorii de instruire am obținut un set de prototipuri etichetate a cărui structură este cea din Tabelul 4.

Tabelul 4. Numărul de vectori de instruire a SOM şi SOM-CNN, numărul de prototipuri obținut în urma instruirii SOM şi numărul de prototipuri obținut după instruirea SOM-CNN.

| | A | E | I | O | U | Ă |
|---|---|---|---|---|---|---|
| Instruire | 372 | 516 | 339 | 525 | 453 | 52 |
| SOM | 42 | 37 | 16 | 21 | 21 | 2 |
| SOM-CNN | 25 | 25 | 25 | 25 | 25 | 25 |

Se observă că fonemul *ă* este cel mai slab reprezentat din cauza volumului redus al datelor de instruire disponibile. Numărul total de prototipuri este mai mic decât cel al neuronilor din rețea pentru că nu toți neuronii au fost etichetați la calibrare.

Şi aici se observă slaba reprezentare a prototipurilor fonemului *ă*. Prototipurile celorlalte foneme formează grupuri compacte şi poziția clusterelor reflectă calitățile acustico-fonetice ale fonemelor. Instruirea componentelor SOM-CNN s-a făcut după gruparea manuală a vectorilor de instruire în şase clase. Fiecare rețea neurală componentă s-a specializat în recunoaşterea unei singure clase. Am plasat în acelaşi grup atât realizările ca vocale cât şi ca semivocale ale fonemelor. Am folosit şase rețele Kohonen cu câte 5x5 neuroni instruite de asemenea în două etape. Numărul total de neuroni al SOM-CNN este egal cu numărul de neuroni al SOM folosit la experimentul cu o singură rețea neurală. Aceasta înseamnă că numărul prototipurilor ar trebui să fie egal pentru ambele tipuri de experimente: 150. Diferența este minoră şi apare datorită faptului că în primul caz nu toți neuronii au fost etichetați la calibrare, aşa cum am arătat în Tabelul 4.

Vom descrie în continuare tehnica pe care am aplicat-o la recunoaştere. Zonele vocalizate din semnalul de test sunt preprocesate cu ajutorul unei ferestre din care se calculează câte un vector de componente cepstrale. Fereastra se deplasează cu un pas de 50 de eşantioane şi obținem o colecție de vectori a căror succesiune descrie evoluția semnalului vocal. Pentru fiecare vector reținem poziția care îi corespunde în semnal şi eroarea de cuantizare minimă, respectiv eticheta pe care le calculează rețeaua neurală. Experimental am stabilit un prag maxim al erorii de cuantizare pentru a elimina fonemele care se presupune că nu aparțin nici unei clase, de exemplu consoanele. Am obținut prin acest procedeu o succesiune de etichete de clase care arată cum au fost recunoscute vocalele de către sistem. în Figura 3se poate vedea cum a fost recunoscută o repetiție a cuvântului *nouă* de SOM şi SOM-CNN.

Se remarcă faptul că SOM nu recunoaşte fonemul *ă* final. SOM a decis că în acea zonă se găseşte doar de fonemul *o*. în vorbirea curentă, trecerea de la *u* la *ă* în cuvântul *nouă* se face prin uşoara inserare între ele a semivocalei *o*, fapt sesizat atât de SOM cât şi de SOM-CNN.

**Rezultate experimentale**

Pentru a compara performanțele SOM şi SOM-CNN, în Tabelul 5 sunt descrise rezultatele următoarelor experimente [12]:

- E4.1: stabilirea procentului vocalelor şi semivocalelor recunoscute de cele două modele;
- E4.2: proporția în care dimensiunea reală a zonei vocalei sau a semivocalei este sesizată de SOM şi SOM-CNN;
- E4.3: numărul de succesiuni vocală-semivocală recunoscute corect;
- E4.4: procentul anunțurilor false în legătură cu existența unei vocale sau semivocale într-o regiune din semnal în care în realitate sunt consoane;

- E4.5: procentul confuziilor dintre două vocale.

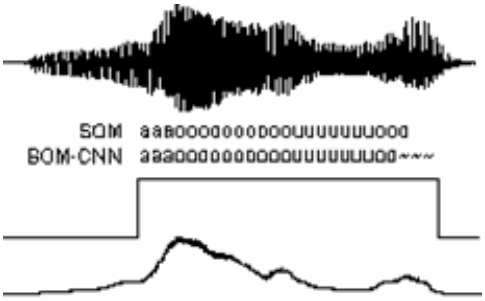


Figura 3. Interpretarea cuvântului *nouă* de către SOM şi SON-CNN. Am folosit simbolul ~ pentru fonemul *ă*.

Experimentul 4.1 ne ajută să ne formăm o imagine corectă asupra capacității de a recunoaşte noi vectori de către SOM şi SOM-CNN. Progresul obținut cu ajutorul modelului nostru este de peste 10%. Am considerat că o vocală este recunoscută corect dacă secvența de etichete este cea reală pe zona centrală a segmentului vocalic, aşa cum se poate vedea şi în figura Tabelul 5.

Fenomenul coarticulației influențează recunoaşterea în zonele de tranziție dintre două foneme. Prin experimentul 4.2 stabilim în ce măsură sistemele testate de noi determină dimensiunile corecte ale vocalelor şi semivocalelor.

Atunci când apar în diftongi, vocalele sunt pronunțate cu mai multă forță şi tind să domine semivocalele, aşa cum se întâmplă şi pentru cuvântul *doi*. Experimentul 4.3 stabileşte procentul de astfel de succesiuni pentru care sunt recunoscute atât vocala cât şi semivocala.

Experimentele 4.4 şi 4.5 determină capacitatea SOM şi a SOM-CNN de a evita confuziile. Energia semnalului vocal dă o primă informație referitoare la caracterul vocalizat sau nevocalizat al unei porțiuni din semnal. Rețeaua neurală stabileşte pentru segmentele vocalizate care este succesiunea fonemelor. Avem, astfel, două nivele de decizie a existenței vocalelor. în experimentul 4.4 am calculat rata confuziilor între vocale şi consoane.

În experimentul 4.5 am stabilit rata confuziilor între vocale, determinând care este procentul în care o vocală este înlocuită de sistemul de recunoaştere cu altă vocală.

Tabelul 5. Rezultatele experimentelor de recunoaştere a vocalelor şi semivocalelor cu SOM şi SOM-CNN.

| | SOM | SOM-CNN |
|---|---|---|
| E4.1 | 73,68% | 84,21% |
| E4.2 | 62,47% | 73,93% |
| E4.3 | 50% | 56,25% |
| E4.4 | 5,88% | 5,48% |
| E4.5 | 13,07% | 8,61% |

## Concluzii

Rezultatele obținute în urma experimentelor de recunoaştere a vorbitorilor folosind CNN au fost de fiecare dată superioare celor obținute folosind modelele clasice de rețele neurale. Pe de altă parte, rezultatele SOM-CNN au fost mai bune cu aproximativ 4% fața de cele ale MLP-CNN şi cu 1% față de TDNN-CNN.

Am testat SOM şi SOM-CNN pentru recunoaşterea unor vocale din limba română şi am constatat că procentul vocalelor şi semivocalelor recunoscute de SOM-CNN este superior celui obținut cu SOM cu mai mult de 10%. Regiunile din semnal corespunzătoare vocalelor şi semivocalelor sunt mai cuprinzătoare atunci când se foloseşte SOM-CNN decât atunci când se foloseşte SOM.

Am văzut în acest capitol că rețelele neurale concurente implementează conceptul de concurență la nivelul unei colecții de rețele neurale. Experimentele demonstrează că modelul

rețelelor neurale concurente este superior în majoritatea cazurilor folosirii unei singure rețele neurale pentru rezolvarea aceleiaşi probleme.

În continuare vom atinge a doua problemă importantă a tezei şi vom introduce o serie de algoritmi prin care se poate determina importanța intrărilor pentru rețelele neurale care folosesc algoritmi de instruire tip LVQ.

# Recunoaştere prin calculul relevanțelor intrărilor folosind operatori OWA

*Algoritmul Relevance Learning Vector Quantization (RLVQ) [10] este o variantă a algoritmului LVQ care permite determinarea printr-o metodă euristică a relevanțelor pentru componentele intrărilor. Metoda se bazează pe învățarea hebbiană şi introduce factorii de ponderare a intrărilor, adaptați automat la problema de clasificare care trebuie implementată. Pe lângă stabilirea importanței intrărilor, această metodă conduce la îmbunătățirea performanțelor algoritmului LVQ dar, în acelaşi timp, poate fi folosită pentru reducerea numărului de intrări prin renunțarea la cele mai puțin influente.*

*În acest capitol propunem o nouă metodă de calcul al relevanțelor intrărilor în RLVQ. Relevanțele sunt calculate on-line ca ponderi ale unui operator Ordered Weighted Aggregation (OWA). Operatorii OWA reprezintă o familie de operatori de agregare şi au fost definiți de Yager [93]. Principalul avantaj al acestui algoritm numit OWA-RLVQ este acela că face legătura dintre RLVQ şi consistentul model matematic OWA.*

Kohonen a introdus LVQ ca o metodă care defineşte o clasificare pe baza unui număr de vectori grupați într-un set de instruire. Aşa cum am văzut deja, cuantizarea vectorială înseamnă înlocuirea vectorilor de instruire cu un număr redus de prototipuri care formează un *codebook*. Fiecare vector din *codebook* are ataşată eticheta clasei din care face parte. Algoritmul LVQ adaptează iterativ aceşti vectori prin optimizarea unui criteriu global bazat pe distanța euclidiană.

Algoritmul LVQ standard nu face distincția între componentele vectoriale mai influente şi cele mai puțin influente, importanța lor fiind considerată egală. Spre deosebire de această abordare, algoritmul *Distinction Sensitive Learning Vector Quantizer (DSLVQ)* introdus în [68] foloseşte o pondere modificabilă pentru fiecare componentă vectorială şi foloseşte o funcție de distanță ponderată pentru clasificare. Se utilizează un algoritm euristic iterativ pentru adaptarea acestor ponderi la specificul problemei: influența componentelor care în mod frecvent generează o clasificare greşită este diminuată, în timp ce influența componentelor care contribuie la clasificarea corectă este amplificată. Procesul de ponderare poate fi privit ca o transformare prin scalare a spațiului vectorilor de instruire într-un spațiu bazat pe o distanță ponderată: creşte distanța dintre valorile componentelor cele mai puternic discriminative şi o micşorează pe cea dintre valorile componentelor care aduc mai puțină informație. Aceasta facilitează distincția dintre clase şi face sistemul mai puțin sensibil la zgomot [69].

Algoritmul DSLVQ foloseşte o funcție de distanță ponderată care se bazează pe distanța euclidiană ponderată. Se pot folosi, însă, şi alte distanțe, după cum se precizează în [68]. Modificarea algoritmului LVQ este minimală, fiind înlocuită funcția distanței. Ponderile şi vectorii codebook se modifică în paralel în timpul instruirii, actualizarea având loc on-line, după procesarea fiecărui vector de intrare. Ponderile rezultate pot fi interpretate ca ranguri ale componentelor vectoriale şi folosite pentru reducerea dimensiunii spațiului de intrare.

RLVQ este o variantă modificată a LVQ, similară cu DSLVQ, şi introduce factorii de relevanță pentru pentru fiecare dimensiune a vectorilor de intrare. Metrica euclidiană este înlocuită cu una modificată prin asocierea unei ponderi fiecărei componente vectoriale. Adițional algoritmului standard, factorii de scalare sunt adaptați iterativ printr-o tehnică bazată pe învățarea hebbiană, oferind o imagine a influenței fiecărei componente vectoriale asupra procesului de clasificare.

Operatorii OWA reprezintă o clasă de operatori de agregare care realizează o agregare bazată pe reordonarea criteriilor care trebuie satisfăcute. Ponderile operatorului OWA sunt asociate poziţiilor acestor criterii obţinute în urma reordonării, şi nu unui criteriu anume.

Noul algoritm de calcul al relevanţelor pe care îl introducem în acest capitol calculează importanţele componentelor de intrare considerându-le ponderi ale unui operator OWA. Ponderile sunt adaptate on-line, în paralel cu instruirea prototipurilor. Metrica pe care o folosim este un caz particular al mediei generalizate de tip OWA [37]. Pe lângă recunoaşterea eficientă a formelor, metoda noastră face şi o conexiune interesantă între două abordări diferite: RLVQ şi OWA.

Experimentele prezentate în finalul capitolului demonstrează capacitatea tehnicii propuse de noi de a rezolva probleme dificile, iar rezultatele confirmă acest lucru. Am folosit trei baze de date care reprezintă standarde în clasificarea formelor. Cu bazele de date Iris şi Ionosphere am testat performanţele generale în recunoaştere ale algoritmului nostru. Pentru testele de recunoaştere a vorbirii am utilizat baza de date Deterding care conţine un set de vectori de referinţă în domeniu, obţinuţi din pronunţii ale vocalelor în diferite contexte.

## Algoritmul Relevance LVQ (RLVQ)

Uneori nu toate componentele vectorilor de intrare au aceeaşi importanţă în decizia pe care o ia un sistem de clasificare. Unele se pot dovedi mai influente decât altele, astfel încât putem deveni interesaţi în a calcula ponderile lor. RLVQ este un algoritm euristic, supervizat, care amplifică valoarea ponderilor acelor componente care contribuie cel mai mult la clasificarea corectă a vectorilor de instruire şi slăbeşte ponderile componentelor care au o influenţă negativă. Clustering-ul este realizat de un set de prototipuri care sunt adaptate vectorilor de instruire printr-un algoritm LVQ standard.

Algoritmul RLVQ foloseşte metrica modificată $\left\|\mathbf{x}_i - \mathbf{w}_j\right\|_\lambda = \sqrt{\sum_{k=1}^{n} \lambda_k \left(x_{ik} - w_{jk}\right)^2}$, unde $\lambda = \left[\lambda_1 \ldots \lambda_n\right]^t$ este vectorul relevanţelor, iar $\sum_{k=1}^{n} \lambda_k = 1$. Urmând o regulă similară algoritmului LVQ, factorii de ponderare sunt adaptaţi iterativ astfel [10]:

**Pasul 1.** *Pentru fiecare* $k = 1 \ldots n$

$$\lambda_k = \begin{cases} \max\left(\lambda_k - \alpha\left|x_{ik} - w_{jk}\right|, 0\right) \text{ când } \mathbf{x}_i \text{ este clasificat corect} \\ \lambda_k + \alpha\left|x_{ik} - w_{jk}\right| \text{ î n rest,} \end{cases}$$

*unde* $\alpha > 0$ *este un parametru de instruire.*

**Pasul 2.** *Ponderile* $\lambda_k$ *sunt normalizate pentru fiecare* $k = 1 \ldots n$.

Această regulă de actualizare a relevanţelor foloseşte principiul învăţării hebbiene, aşa cum se demonstrează în [10].

## Operatorii OWA

Operatorii OWA sunt o categorie specială de operatori de agregare [93]. Un operator OWA de dimensiune *n* este o funcţie $F: R^n \to R$, $F(a_1 \ldots a_n) = \sum_{i=1}^{n} w_i a_i^*$, unde $\mathbf{W} = \left[w_1 \ldots w_n\right]^t$ este un vector de ponderi asociat operatorului, cu $w_i \in [0,1]$, $\sum_{i=1}^{n} w_i = 1$, iar $\mathbf{A}^* = \left[a_1^* \ldots a_n^*\right]^t$ este o reordonare descrescătoare a argumentelor lui $\mathbf{A} = \left[a_1 \ldots a_n\right]^t$ în aşa fel încât $a_i^*$ este al *i*-lea cel mai mare dintre $a_i$, $a_1^* \geq \ldots a_n^*$.

Elementul central în utilizarea unui operator OWA este pasul de reordonare a argumentelor în urma căruia un argument particular $a_i$ nu mai este asociat unei ponderi particulare $w_i$, ci ponderea $w_i$ este asociată valorii de pe poziția $i$ obținută în urma reordonării. Se demonstrează în [93] că operatorii OWA satisfac proprietățile de comutativitate, monotonie, idempotență şi mărginire.

## Algoritmul OWA - Relevance LVQ (OWA-RLVQ)

În această secțiune introducem un nou algoritm [14] care calculează ponderile unui operator OWA ca relevanțe ale componentelor vectorilor de instruire pentru probleme de clustering supervizat, aşa cum se poate vedea în figura 4. Algoritmul LVQ1 trebuie reformulat pentru a minimiza o funcție obiectiv bazată pe această distanță modificată. Calculăm $\Delta\mathbf{w}_\mathbf{j}^* = \eta\lambda\mathbf{I}\left(\mathbf{x}_i - \mathbf{w}_j\right)^*$ dacă $\mathbf{x}_i$ este clasificat corect conform distanței $D_{ij}^*$ şi $\Delta\mathbf{w}_\mathbf{j}^* = -\eta\lambda\mathbf{I}\left(\mathbf{x}_i - \mathbf{w}_j\right)^*$ dacă $\mathbf{x}_i$ nu este clasificat corect. În relațiile de mai sus am notat cu $\left(\mathbf{x}_i - \mathbf{w}_j\right)^*$ vectorul diferență ale cărui componente sunt reordonate, iar relevanțele se vor aplica după reordonare. Din această cauză, setul de prototipuri obținut în urma acestei proceduri nu va fi acelaşi cu cel obținut în urma algoritmului LVQ standard pentru că reflectă două abordări diferite ale instruirii.

Fie $\mathbf{x}_i \in R^n$ un vector de instruire, $i = 1\ldots N$ fiind numărul de vectori de instruire. Considerăm $\mathbf{w}_k$, $k = 1\ldots P$ prototipurile care pot fi calculate printr-un algoritm LVQ. Pentru a calcula ponderile OWA, vom considera ca operator de agregare următoarea distanță modificată: $D_{ij}^* = \sqrt{\sum_{k=1}^{n} \lambda_k \left(\left|x_{ik} - w_{jk}\right|^*\right)^2}$, unde $\sum_{k=1}^{n} \lambda_k = 1$ şi $\left|x_{ik} - w_{jk}\right|^*$ este cea de-a $k$-a cea mai mare distanță dintre componentele corespondente ale vectorului de instruire şi vectorului prototip.

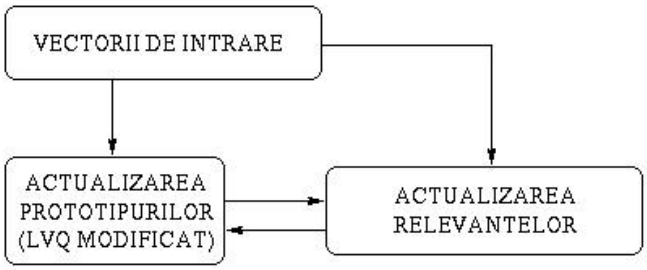


Figura 4. Reprezentarea schematică a modului în care se actualizează relevanțele şi prototipurile prin algoritmul OWA-RLVQ. Algoritmul LVQ este modificat prin înlocuirea distanței euclidiene cu distanța $D_{ij}^*$, iar relevanțele sunt calculate ca ponderi ale unui operator OWA.

Algoritmul LVQ1 trebuie reformulat pentru a minimiza o funcție obiectiv bazată pe această distanță modificată. Calculăm $\Delta\mathbf{w}_\mathbf{j}^* = \eta\lambda\mathbf{I}\left(\mathbf{x}_i - \mathbf{w}_j\right)^*$ dacă $\mathbf{x}_i$ este clasificat corect conform distanței $D_{ij}^*$ şi $\Delta\mathbf{w}_\mathbf{j}^* = -\eta\lambda\mathbf{I}\left(\mathbf{x}_i - \mathbf{w}_j\right)^*$ dacă $\mathbf{x}_i$ nu este clasificat corect. În relațiile de mai sus am notat cu $\left(\mathbf{x}_i - \mathbf{w}_j\right)^*$ vectorul diferență ale cărui componente sunt reordonate, iar relevanțele se vor aplica după reordonare. Din această cauză, setul de prototipuri obținut în urma acestei proceduri nu va fi acelaşi cu cel obținut în urma algoritmului LVQ standard pentru că reflectă două abordări diferite ale instruirii.

Un vector prototip $\mathbf{w}_k$ va eticheta cu numele clasei sale toți vectorii de intrare care îndeplinesc următoarea inegalitate: $\left|\mathbf{x}_i - \mathbf{w}_j\right|^* < \left|\mathbf{x}_i - \mathbf{w}_l\right|^*, \forall j \neq l$.

Fie $\lambda = [\lambda_1 \ldots \lambda_n]^t$, $\mathbf{x}_i = [x_{i1} \ldots x_{in}]^t$, $\mathbf{w}_j = [w_{j1} \ldots w_{jn}]^t$ şi $\mathbf{d} = [d_1 \ldots d_n]^t$, unde $d_k = x_{ik} - w_{jk}$, $k = 1 \ldots n$. Dacă $\mathbf{x}_i$ este clasificat corect conform distanței $D_{ij}^*$, o valoare mică a lui $|d_k|^*$ ar trebui să inducă o valoare mare pentru $\Delta\lambda_k$, iar o valoare mare a lui $|d_k|^*$ ar trebui să conducă la o valoare mică pentru $\Delta\lambda_k$. Astfel, pentru o clasificare corectă conform criteriului formulat anterior, $|d_k|^* \le |d_{k'}|^*$ implică $\Delta\lambda_k \ge \Delta\lambda_{k'}$, adică $-|d_k|^* \ge -|d_{k'}|^*$ înseamnă $\Delta\lambda_k \ge \Delta\lambda_{k'}$ pentru $k, k' = 1 \ldots n$. Putem, astfel, să luăm $\Delta\lambda_k = -\alpha|d_k|^*$ sau $\Delta\lambda_k = -\alpha|x_{ik} - w_{jk}|^*$, unde $\alpha > 0$ este un parametru de instruire. Atunci când clasificarea conform aceluiaşi criteriu este incorectă, $|d_k|^* \le |d_{k'}|^*$ implică $\Delta\lambda_k \le \Delta\lambda_{k'}$, pentru $k, k' = 1 \ldots n$. Putem lua $\Delta\lambda_k = \alpha|d_k|^*$ sau $\Delta\lambda_k = \alpha|x_{ik} - w_{jk}|^*$.

Folosim transformarea $\lambda_k = \dfrac{e^{\lambda_k}}{\sum_{i=1}^{n} e^{\lambda_i}}$ pentru $i = 1 \ldots n$ pentru a ne asigura că $\sum_{k=1}^{n} \lambda_k = 1$ şi $\lambda_k = [0,1]$ care sunt constrângerile impuse ponderilor operatorilor OWA.

Următoarea procedură actualizează on-line atât relevanțele cât şi rangul fiecărei componente a vectorilor de intrare.

**Pasul 1.** *Inițializează* $\eta$ *şi* $\alpha$. *Inițializează* $\lambda_k = \dfrac{1}{n}$, $k = 1 \ldots n$, *unde n este numărul de componente ale vectorului de intrare.*

**Pasul 2.** *Inițializează prototipurile.*

**Pasul 3.** *Actualizează prototipurile conform regulii LVQ modificate:*

$$\mathbf{w}_j^* = \begin{cases} \mathbf{w}_j^* + \eta\lambda\mathbf{I}\left(\mathbf{x}_i - \mathbf{w}_j\right)^* & \text{daca } \mathbf{x}_i \text{ este clasificat corect conform distantei } D_{ij}^* \\ \mathbf{w}_j^* - \eta\lambda\mathbf{I}\left(\mathbf{x}_i - \mathbf{w}_j\right)^* & \text{in rest.} \end{cases}$$

**Pasul 4.** *Actualizează relevanțele:*

$$\lambda_k = \begin{cases} \lambda_k - \alpha|x_{ik} - w_{jk}|^* & \text{daca } \mathbf{x}_i \text{ este clasificat corect conform distantei } D_{ij}^* \\ \lambda_k + \alpha|x_{ik} - w_{jk}|^* & \text{in rest.} \end{cases}$$

**Pasul 5.** *Transformă relevanțele:* $\lambda_k = \dfrac{e^{\lambda_k}}{\sum_{i=1}^{n} e^{\lambda_i}}$ pentru $k = 1 \ldots n$.

**Pasul 6.** *Calculează ponderea fiecărei componente a vectorului de intrare ca medie a indicilor de poziție calculați înaintea pasului de ordonare, pentru toți paşii parcurşi de la începutul instruirii.*

**Pasul 7.** *Repetă paşii 3-6 pentru fiecare vector de instruire.*

Acest algoritm calculează ponderile unui operator OWA ca relevanțe prin minimizarea distanței modificate $D_{ij}^*$ care este şi valoarea agregată. Trebuie să facem distincția între relevanțe şi rangul unei componente. Rangul este ataşat unei componente. Relevanța este ataşată unei poziții specifice din vectorul ordonat al distanței pe componente dintre un vector de intrare şi un prototip. Aceasta explică semnificația pasului 6.

## Experimente

În studiul nostru am testat algoritmul OWA-RLVQ pe baze de date standard pentru a vedea care este structura tipică a vectorilor de relevanțe şi care sunt performanțele în recunoaştere.

Baza de date Iris [9] conține 3 clase alcătuite din câte 50 de vectori. Două dintre ele nu sunt liniar separabile. Problema este detectarea claselor pe baza celor 4 componente vectoriale. Am

instruit OWA-RLVQ cu 6 prototipuri şi am obţinut o rată medie de recunoaştere de 96,6%. Un vector tipic al relevanţelor este $[0{,}15\ 0{,}21\ 0{,}23\ 0{,}38]^t$. Am luat $\eta = 0{,}3$ şi $\alpha = 0{,}2$. Experimentele au arătat că a patra componentă vectorială are influenţa cea mai mare, iar prima componentă vectorială este cea mai puţin semnificativă, aceste rezultate fiind confirmate şi de cele din [10]. Rangurile obţinute pentru fiecare componentă după instruirea cu RLVQ şi OWA-RLVQ folosind acelaşi set de prototipuri iniţiale sunt prezentate în Tabelul 6.

Tabelul 6. Rangul componentelor vectoriale pentru baza de date Iris.

| Rangul | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Componentele RLVQ | 4 | 2 | 3 | 1 |
| Componentele OWA-RLVQ | 4 | 3 | 2 | 1 |
| Ponderile componentelor OWA-RLVQ | 1,86 | 1,44 | 1,24 | 1,17 |

Baza de date Deterding (Vowel Recognition) [9] conţine vectori extraşi din pronunţiile vocalelor în 11 contexte. Fiecare pronunţie este repetată de 6 ori de cei 15 vorbitori. Problema cere să se folosească pronunţiile primilor 8 vorbitori pentru instruire şi pronunţiile ultimilor 7 vorbitori pentru recunoaştere. Un vector tipic al relevanţelor obţinut prin algoritmul OWA-RLVQ este $[0{,}032\ 0{,}039\ 0{,}043\ 0{,}044\ 0{,}077\ 0{,}136\ 0{,}137\ 0{,}15\ 0{,}163\ 0{,}173]^t$. Rata de recunoaştere pe care am obţinut-o cu acest algoritm a fost de 46,75%. Rata medie de recunoaştere pentru LVQ1 la acest experiment a fost de 44,8%, iar pentru RLVQ a fost de 46,32%. Problema este cunoscută ca fiind una dificilă, de aceea am folosit 59 de prototipuri. În experimentele OWA-RLVQ am luat $\eta = 1{,}7$ şi $\alpha = 1{,}9$. Rangurile obţinute pentru această bază de date sunt prezentate în Tabelul 7. Componenta 2 este considerată cea mai importantă de RLVQ şi pe poziţia a doua de OWA-RLVQ, după cum se poate remarca studiind aceste rezultate. Componenta 10 este cea mai puţin importantă pentru ambele modele.

Tabelul 7. Rangul componentelor vectoriale pentru baza de date Deterding.

| Rangul | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Componentele RLVQ | 2 | 5 | 1 | 9 | 6 | 3 | 4 | 8 | 7 | 10 |
| Componentele OWA-RLVQ | 8 | 2 | 4 | 5 | 6 | 9 | 3 | 7 | 1 | 10 |
| Ponderile componentelor OWA-RLVQ | 5,21 | 5,21 | 5,20 | 5,17 | 5,16 | 5,15 | 5,15 | 5,14 | 5,14 | 5,13 |

Baza de date Ionosphere [9] constă din 351 de vectori de date cu câte 34 de caracteristici. Vectorilor le sunt asociate etichetele "bad" şi "good", fiind vorba de o clasificare binară. Problema constă în folosirea primilor 200 de vectori, echilibraţi numeric între exemple pozitive şi negative, pentru instruire şi ultimii 151 pentru test. Am folosit 8 prototipuri, iar valorile parametrilor pentru instruirea OWA-RLVQ au fost $\eta = 3{,}3$ şi $\alpha = 3{,}5$. Rata medie de recunoaştere a fost de 93,37%, în timp ce pentru LVQ1 am obţinut 90,06%, iar pentru RLVQ am obţinut 92,71%. Ratele uzuale de recunoaştere corectă pentru această bază de date încep, aşa cum se arată în [9], de la 90%. în acest context, rezultatele obţinute de noi sunt foarte bune. Rangurile celor mai importante 5 componente vectoriale obţinute cu RLVQ şi OWA-RLVQ sunt prezentate în Tabelul 8.

Tabelul 9 compară ratele de recunoaştere obţinute cu LVQ1, RLVQ şi OWA-RLVQ folosind de fiecare dată acelaşi set iniţial de prototipuri. în general am obţinut rezultate mai bune cu algoritmul OWA-RLVQ faţă de cele obţinute cu RLVQ. Ambele metode sunt superioare algoritmului LVQ1, fapt care ilustrează utilitatea relevanţelor în clasificare. Trebuie accentuată observaţia că valorile relevanţelor obţinute cu RLVQ şi OWA-RLVQ nu se suprapun în mod obligatoriu pentru că reflectă două abordări diferite. în cazul metodei propuse de noi, relevanţele se referă la distanţele reordonate, în timp ce algoritmul RLVQ generează un set de valori asociate în mod precis anumitor componente ale vectorilor de intrare.

Tabelul 8. Rangul componentelor vectoriale pentru baza de date Ionosphere. Sunt reprezentate cele mai importante 5 componente.

| Rangul | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Componentele RLVQ | 20 | 28 | 26 | 12 | 6 |
| Componentele OWA-RLVQ | 14 | 12 | 1 | 3 | 28 |
| Ponderile componentelor OWA-RLVQ | 19,20 | 19,16 | 19,10 | 19,07 | 19,06 |

Tabelul 9. Comparație între ratele de recunoaştere obținute cu LVQ1, RLVQ şi OWA-RLVQ.

| Baza de date | LVQ1 | RLVQ | OWA-RLVQ |
|---|---|---|---|
| Iris | 91,33% | 95,33% | 96,60% |
| Deterding | 44,80% | 46,32% | 46,75% |
| Ionosphere | 90,06% | 92,71% | 93,37% |

## Concluzii

În acest capitol am introdus o nouă metodă de calcul al relevanțelor dimensiunilor vectorilor de intrare ca ponderi OWA. Algoritmul propus calculează on-line relevanțele, oferind posibilitatea unei permanente adaptări la datele de intrare. Rezultatele experimentale au arătat că algoritmul propus de noi oferă o bună rată de recunoaştere. în acelaşi timp, factorii de relevanță pot fi utilizați pentru stabilirea on-line a rangului fiecărei dimensiuni a vectorilor de intrare.

Este interesant de observat că abordarea noastră este o nouă metodă de calcul al ponderilor OWA. în comparație cu metoda descrisă în [21] şi [94] care foloseşte tehnica gradientului descrescător, noi realizăm acest lucru în mod dinamic.

În capitolul următor vom dezvolta două noi metode care calculează factorii de relevanță prin maximizarea unor mărimi bazate pe informația mutuală şi pe energia informațională.

# Recunoaştere prin calculul relevanțelor intrărilor folosind noțiuni de teoria informației

*Selecția componentelor vectorilor de intrare într-un sistem de clasificare este o etapă de preprocesare absolut necesară, în special atunci când avem de a face cu cantități foarte mari de date. Introducem în acest capitol algoritmul Energy Relevance LVQ (ERLVQ) bazat pe energia informațională Onicescu [65] ca o metodă incrementală de instruire pentru clasificare supervizată şi calcul al relevanțelor. Algoritmul Mutual Information Relevance LVQ (MIRLVQ) propune o tehnică alternativă de dezvoltare a acestei metode pornind de la informația mutuală.*

Informația mutuală este o mărime care a fost utilizată pentru a implementa diverse metode de selecție a intrărilor [5], [49]. În acest caz, informația mutuală evaluează "conținutul informațional" al fiecărei componente în mod individual, raportat la clasa de ieşire. Metoda de selecție a componentelor constă în căutarea unui subset de componente relevante din setul initial de componente disponibile. Acest subset se alege în aşa fel încât să maximizeze informația mutuală. O importanță însemnată în acest algoritm o are estimarea informației mutuale. Estimarea informației mutuale este o problemă dificilă datorită funcțiilor de densitate de probabilitate condiționată şi a complexității computaționale ridicate. Mulți algoritmi de selecție a componentelor vectoriale care se bazează pe informația mutuală folosesc histograma ca estimator al densității de probabilitate [5], [49] . În spații cu multe dimensiuni, însă, histograma nu este practică şi nici foarte precisă [32].

Informația mutuală bazată pe entropia Renyi pătratică [70] a fost utilizată recent într-o metodă eficientă de selecție a caracteristicilor [32]. Acest algoritm numit *toward-optimal feature selection methodology - mutual information (OFS-MI)* constă din două criterii bazate pe informația mutuală şi un algoritm de căutare. Primul criteriu numit *feature relevancy criterion (FRC)* are scopul de a

selecta componentele relevante, iar al doilea, *feature similarity criterion (FSC)*, este folosit pentru a reduce redundanța între componentele selectate.

Torkkola [80], [81], [82], [83], [84] maximizează informația mutuală pătratică obținând o metodă de instruire a unui sistem pentru extragerea caracteristicilor. Un astfel de sistem transformă vectorii de intrare în vectori cu un număr mai mic de dimensiuni prin recalcularea componentelor, fără a afecta performanțele clasificatorului. Diferența dintre un sistem de selecție a caracteristicilor şi unul de extragere (transformare, construcție, generare) a caracteristicilor este aceea că primul foloseşte mai departe un subset al componentelor vectoriale originale, în timp ce al doilea construieşte vectori cu un număr redus de dimensiuni prin recalcularea acestora, fără a păstra în mod necesar valorile inițiale. Metoda lui Torkkola adaptează informația mutuală pătratică din [70] pentru variabile aleatoare discrete şi o maximizează printr-o metodă de gradient ascendent pentru a obține o transformare convenabilă.

Algoritmul *Mutual Information Relevance LVQ (MIRLVQ)* propus de noi în acest capitol maximizează informația mutuală pătratică pentru a obține relevanțele care conduc la îmbunătățirea performanțelor unui clasificator bazat pe un algoritmul LVQ. Algoritmul *Energy Relevance LVQ (ERLVQ)* este similar lui MIRLVQ, însă maximizează o mărime bazată pe energia informațională. Experimental vom arăta că aceşti algoritmi conduc la rate de recunoaştere mai bune decât algoritmul OWA-RLVQ introdus tot de noi.

## Informația mutuală pătratică

Informația mutuală este o măsură a dependenței dintre două variabile aleatoare şi se defineşte prin diferența de entropii Shannon $I(X,Y)=H(Y)-H(Y|X)$, unde $X$ şi $Y$ sunt două variabile aleatoare.

Deoarece entropia Shannon nu poate fi estimată din datele de instruire cu un efort computațional rezonabil, folosim o soluție alternativă de estimare a informației mutuale propusă de Torkkola în [84]. Acesta a arătat că informaţtia mutuală dintre o variabilă aleatoare discretă $C$ şi o variabilă aleatoare continuă $Y$ se poate estima prin relația:

$$I_{ED}(C,Y)=\sum_c \int_{\mathbf{y}} p^2(c,\mathbf{y})d\mathbf{y}+\sum_c \int_{\mathbf{y}} p^2(c)p^2(\mathbf{y})d\mathbf{y}-2\sum_c \int_{\mathbf{y}} p(c,\mathbf{y})p(c)p(\mathbf{y})d\mathbf{y}.$$

Notăm cu $M_p$ numărul de eşantioane din clasa $p$, cu $\mathbf{y}_k$ eşantionul numărul $k$ din întreg setul de eşantioane disponibile fără a ne interesa clasa căreia îi aparține şi cu $\mathbf{y}_{pj}$ acelaşi eşantion arătând că el aparține clasei $p$ şi că are indicele $j$ în clasă.

Rescriem relația de mai sus folosind aproximările densităților de probabilitate cu ferestre Parzen şi rezultă:

$$\begin{aligned} I_{ED}(C,Y) &= \frac{1}{N^2}\sum_{p=1}^{M}\sum_{k=1}^{M_p}\sum_{l=1}^{M_p} G\left(\mathbf{y}_{pk}-\mathbf{y}_{pl},2\sigma^2\mathbf{I}\right)+\frac{1}{N^2}\left(\sum_{p=1}^{M}\left(\frac{M_p}{N}\right)^2\right)\sum_{k=1}^{N}\sum_{l=1}^{N} G\left(\mathbf{y}_k-\mathbf{y}_l,2\sigma^2\mathbf{I}\right)- \\ &- 2\frac{1}{N^2}\sum_{p=1}^{M}\frac{M_p}{N}\sum_{j=1}^{M_p}\sum_{k=1}^{N} G\left(\mathbf{y}_{pj}-\mathbf{y}_k,2\sigma^2\mathbf{I}\right). \end{aligned}$$

Am folosit metoda ferestrelor Parzen care estimează densitățile de probabilitate prin medii ale valorilor calculate prin diverse nuclee. Atunci când se foloseşte un nucleu gaussian, relația de aproximare se poate scrie astfel: $p(\mathbf{y})\cong\frac{1}{N}\sum_{n=1}^{N} G\left(\mathbf{y}-\mathbf{y}_n,\sigma^2\mathbf{I}\right)$.

Am folosit, de asemenea, următoarea proprietate a nucleului gaussian:

$$\int_{-\infty}^{\infty} G\left(\mathbf{y}-\mathbf{y}_i,\sigma^2\mathbf{I}\right)G\left(\mathbf{y}-\mathbf{y}_j,\sigma^2\mathbf{I}\right)d\mathbf{y}=G\left(\mathbf{y}_i-\mathbf{y}_j,2\sigma^2\mathbf{I}\right).$$

## Algoritmul Mutual Information Relevance LVQ (MIRLVQ)

În această secțiune introducem o nouă metodă de instruire a importanței componentelor vectorilor de intrare, obținută prin maximizarea informației mutuale dintre etichetele claselor şi o transformare a vectorilor de intrare. Aşa cum am arătat în secțiunea precedentă, informația mutuală pătratică este o sumă de termeni de gradul doi, fiind dezvoltată pornind de la similaritatea cu entropia pătratică Renyi. Informația mutuală pătratică estimată cu ferestre Parzen oferă avantajul că se calculează ca sumă de gaussiene, fiecare dintre ele putând fi evaluată folosind perechi de eşantioane din setul de instruire.

Considerăm că avem o variabilă aleatoare $X$, iar $\mathbf{x}_i,\ i = 1 \ldots N$ este un set de realizări ale acestei variabile aleatoare. Realizările vor fi chiar vectorii de intrare. Considerăm o variabilă aleatoare discretă $C$ şi $c_j,\ j = 1 \ldots M$ sunt realizările sale care reprezintă etichetele claselor din care fac parte vectorii $\mathbf{x}_i$. Pentru ca instruirea să fie eficientă, trebuie ca fiecărei clase să i se asocieze mai mulți vectori de intrare.

### Algoritmul Weighted LVQ

Am arătat că actualizarea factorilor de relevanță prin algoritmii DSLVQ şi RLVQ este motivată euristic. Din acest motiv, am introdus algoritmul OWA-RLVQ care calculează relevanțele ca ponderi ale unui operator OWA prin minimizarea distanței modificate $D_{ij}^{*}$. Algoritmul MIRLVQ va folosi metrica ponderată $D_{ij}$ care a fost introdusă în algoritmul RLVQ.

Includem aceste variații ale lui LVQ într-o clasă generală de algoritmi pe care o numim *Weighted LVQ*. Un algoritm *weighted LVQ* are următoarele caracteristici:

a) Foloseşte o distanță ponderată;
b) Relevanțele şi prototipurile sunt actualizate simultan în timpul fazei de instruire. Actualizarea poate fi realizată incremental după prelucrarea fiecărui vector de instruire;
c) Relevanțele care se obțin după încheierea algoritmului de instruire pot fi privite ca ranguri ale componentelor vectorilor de intrare şi se pot utiliza pentru reducerea spațiului intrărilor (*feature selection*).

Un algoritm *weighted LVQ* constă din următorii paşi:

**Pasul 1.** *Găseşte cel mai apropiat prototip* $\mathbf{w}_j$*, câştigătorul, pentru care valoarea lui* $\left\|\mathbf{x}_i - \mathbf{w}_j\right\|_\lambda$ *este minimă pentru întreagă colecția de vectori prototip.*

**Pasul 2.** *Actualizează prototipul* $\mathbf{w}_j$ *câştigător după un algoritm LVQ.*

**Pasul 3.** *Actualizează vectorul relevanțelor* $\lambda$.

**Pasul 4.** *Actualizează rangul fiecărei componente a vectorilor de intrare ca o medie a paşilor anteriori.*

Ne propunem să transpunem algoritmii de tip *weighted LVQ* într-un cadru teoretic diferit. Introducem algoritmul MIRLVQ ca un algoritm din această clasă bazat pe maximizarea informației mutuale. O problemă majoră pe care o ridică acestă abordare este marea complexitate computațională pe care o implică estimarea informației mutuale. Pentru a evita acest dezavantaj, vom folosi metoda de estimare a informației mutuale cu ferestre Parzen.

### Adaptarea factorilor de relevanță

Notăm $\mathbf{y}_i = \lambda \mathbf{I}\left(\mathbf{x}_i - \mathbf{w}_j\right)$, unde $\mathbf{x}_i$ este vectorul de intrare $i$, $\mathbf{w}_j$ este prototipul $j$ obținut cu un algoritm din categoria weighted LVQ, $\lambda$ este vectorul relevanțelor şi $\mathbf{y}_i$ este vectorul de ieşire. Aceasta este transformarea pe care dorim să o implementăm, $\lambda$ fiind un vector global care caracterizează toate perechile intrare-prototip. Considerăm informația mutuală $I\left(c_j; \mathbf{y_i}\right) = I\left(c_j; \mathbf{x}_i, \mathbf{w}_j, \lambda\right)$ unde $\mathbf{y}_i$ sunt realizări ale variabilei aleatoare continue $Y$. În relația anterioară, am folosit două notații care au aceeaşi semnificație cu $I(C,Y)$. Această egalitate înglobează ipoteza formulată în relația prin care l-am definit pe $\mathbf{y}_i$, conform căreia ieşirea sistemului

nostru se calculează folosind un vector de intrare $\mathbf{x}_i$, un prototip $\mathbf{w}_j$ asociat lui calculat printr-un algoritm tip weighted LVQ şi vectorul relevanţelor $\lambda$ care este comun tuturor vectorilor din setul de instruire. Atunci când vectorul de intrare $\mathbf{x}_i$ este asemănător prototipului $\mathbf{w}_j$, componentele vectorului $\lambda$ trebuie întărite. Putem interpreta această similaritate prin faptul că informaţia mutuală dintre clasa $c_j$ şi valoarea agregată $\mathbf{y}_i$ este mare. Pornind de la această observaţie, putem calcula relevanţele prin maximizarea informaţiei mutuale şi folosim tehnica gradientului crescător. La fiecare pas de instruire $t$+1 vom avea $\lambda^{(t+1)} = \lambda^{(t)} + \alpha \frac{\partial I}{\partial \lambda} \Leftrightarrow \lambda^{(t+1)} = \lambda^{(t)} + \alpha \sum_{i=1}^{n} \frac{\partial I}{\partial \mathbf{y}_i} \mathbf{I}\left(\mathbf{x}_i - \mathbf{w}_j\right)$.

Trebuie să estimăm acum $\frac{\partial I}{\partial \mathbf{y}_i}$ şi pentru aceasta folosim măsura $I_{ED}(C,Y)$ a informaţiei mutuale dintre o variabilă aleatoare continuă şi una discretă. Implementăm un algoritm de gradient stohastic folosind două eşantioane aleatoare $\mathbf{x}_1$ şi $\mathbf{x}_2$ reprezentative pentru întreaga bază de date [83]. Prin estimarea informaţiei mutuale cu ferestre Parzen, se pot face ajustări ale lui $\lambda$ în direcţia gradientului pozitiv. Tratând separat cele două cazuri, în care eşantioanele sunt dinaceeaşi clasă şi în care eşantioanele sunt din clase diferite, se obţine formula de ajustare a factorilor de relevanţă:

$$\lambda^{(t+1)} = \lambda^{(t)} - \frac{\alpha}{8\sigma^2} G\left(\mathbf{y}_1 - \mathbf{y}_2, 2\sigma^2 \mathbf{I}\right)\left(\mathbf{y}_2 - \mathbf{y}_1\right)\mathbf{I}\left[\left(\mathbf{x}_1 - \mathbf{w}_{j(1)}\right) - \left(\mathbf{x}_2 - \mathbf{w}_{j(2)}\right)\right],$$

unde prin $\mathbf{w}_{j(1)}$ şi $\mathbf{w}_{j(2)}$ am notat prototipurile câştigătoare pentru vectorii $\mathbf{x}_1$ şi $\mathbf{x}_2$.

Am stabilit această regulă de modificare iterativă a relevanţelor prin maximizarea informaţiei mutuale $I(C,Y)$. Vom dezvolta în continuare o metodă similară de calcul al factorilor de relevanţă prin maximizarea unei mărimi bazate pe energia informaţională a variabilelor aleatoare. În final, vom detalia algoritmul care calculează în paralel atât prototipurile $\mathbf{w}_j$ cât şi relevanţele $\lambda$.

## Energia informaţională

Pentru variabila aleatoare continuă $X$, energia informaţională introdusă de O. Onicescu în [65] se defineşte astfel: $E(X) = \int_{-\infty}^{\infty} p^2(\mathbf{x}) d\mathbf{x}$, unde $p(\mathbf{x})$ este funcţia de densitate de probabilitate a variabilei aleatoare. În cazul în care avem o variabilă aleatoare discretă $C$, energia informaţională se defineşte prin: $E(C) = \sum_{i=1}^{n} p^2(c_i)$, unde distribuţia aleatoare este definită prin $p(c_k) \geq 0$ şi $\sum_{k=1}^{n} p(c_k) = 1$.

Energia informaţională medie a variabilei $C$ condiţionată de variabila aleatoare $T$ se defineşte astfel: $E(C|T) = \sum_{k=1}^{m} p(t_k) E(C|t_k)$, unde $E(C|t_k) = \sum_{k=1}^{n} p^2(c_i|t_k)$.

În [2] se defineşte o măsură de dependenţă unilaterală între două variabile aleatoare: $o(C,T) = E(C|T) - E(C)$. Valoarea $o(C,T)$ corespunde cantităţii de informaţie despre $C$ pe care o înglobează $T$, putând fi privită ca un indicator al dependenţei unilaterale ce caracterizează variabila aleatoare $C$ în raport cu variabila aleatoare $T$. Se poate constata, aşadar, că $o(C,T)$ şi $I(C,T)$ măsoară practic acelaşi fenomen [2].

## Algoritmul Energy Relevance LVQ (ERLVQ)

Iniţial, $o(X,Y)$ a fost definită pentru două variabile aleatoare discrete. Ea poate fi însă extinsă pentru variabile aleatoare continue sau variabile aleatoare continue împreună cu variabile aleatoare discrete. Maximizăm $o(Y,C)$ pentru a obţine valorile relevanţelor:

$$\lambda^{(t+1)} = \lambda^{(t)} + \alpha \sum_{i=1}^{N} \frac{\partial o(Y,C)}{\partial \mathbf{y}_i} \mathbf{I}\left(\mathbf{x}_i - \mathbf{w}_j\right).$$

Folosind aproximările cu ferestre Parzen obţinem:

$$o(Y,C) = \frac{1}{N}\left(\sum_{p=1}^{M} \frac{1}{M_p}\right) \sum_{k=1}^{M_p} \sum_{l=1}^{M_p} G\left(\mathbf{y}_{pk} - \mathbf{y}_{pl}, 2\sigma^2 \mathbf{I}\right) - \frac{1}{N^2} \sum_{k=1}^{N} \sum_{l=1}^{N} G\left(\mathbf{y}_k - \mathbf{y}_l, 2\sigma^2 \mathbf{I}\right),$$

unde am folosit următoarele notaţii: $N$ - numărul total de eşantioane, $M$ - numărul de clase, $M_p$ - numărul de eşantioane din clasa $c_p$. Pentru a dezvolta formula de actualizare a relevanţelor, folosim două eşantioane $\mathbf{y}_1$ şi $\mathbf{y}_2$ consecutive ca reprezentante ale claselor. Şi avem două cazuri, cel în care eşantioanele aparţin aceleiaşi clase şi cel în care eşantioanele sunt în clase diferite. Obţinem în final formula de actualizare a relevanţelor:

$$\lambda^{(t+1)} = \lambda^{(t)} - \frac{\alpha}{4\sigma^2} G\left(\mathbf{y}_1 - \mathbf{y}_2, 2\sigma^2 \mathbf{I}\right)\left(\mathbf{y}_2 - \mathbf{y}_1\right) \mathbf{I}\left[\mathbf{x}_1 - \mathbf{w}_{j(1)} - \mathbf{x}_2 + \mathbf{w}_{j(2)}\right].$$

Următoarea procedură actualizează simultan prototipurile şi factorii de relevanţă:

**Pasul 1.** *Iniţializează* $\eta$ *şi* $\alpha$. *Iniţializează vectorul relevanţelor:* $\lambda_k = \frac{1}{n},\ k = 1 \ldots n$.

**Pasul 2.** *Iniţializează prototipurile.*

**Pasul 3.** *Actualizează vectorii prototip conform regulii LVQ modificate folosind distanţa* $D_{ij}$*:*

$$\mathbf{w}_j = \begin{cases} \mathbf{w}_j + \eta\lambda \mathbf{I}\left(\mathbf{x}_i - \mathbf{w}_j\right) \text{ daca } \mathbf{x}_i \text{ este clasificat corect} \\ \mathbf{w}_j - \eta\lambda \mathbf{I}\left(\mathbf{x}_i - \mathbf{w}_j\right) \text{ in rest.} \end{cases}$$

**Pasul 4.** *Actualizează relevanţele:*

$$\lambda^{(t+1)} = \lambda^{(t)} - \frac{\alpha}{4\sigma^2} G\left(\mathbf{y}_1 - \mathbf{y}_2, 2\sigma^2 \mathbf{I}\right)\left(\mathbf{y}_2 - \mathbf{y}_1\right) \mathbf{I}\left[\mathbf{x}_1 - \mathbf{w}_{j(1)} - \mathbf{x}_2 + \mathbf{w}_{j(2)}\right].$$

**Pasul 5.** *Transformă relevanţele* $\lambda_k = \frac{e^{\lambda_k}}{\sum_{i=1}^{n} e^{\lambda_i}}$ *pentru* $k = 1 \ldots n$.

**Pasul 6.** *Calculează rangul fiecărei componente a vectorului de intrare ca medie a rangului calculat la paşii anteriori.*

**Pasul 7.** *Repetă paşii 3-6 pentru fiecare vector de intrare.*

Remarcăm asemănarea algoritmilor MIRLVQ şi ERLVQ care, cu toate că sunt dezvoltaţi diferit, conduc la rezultate similare. Diferenţa este minoră şi apare la pasul 4. Rezultatele experimentale sunt identice deoarece constanta care face diferentă între cele două variante introduce doar o scalare a ratei de instruire $\alpha$.

## Experimente

Am testat algoritmii MIRLVQ şi ERLVQ în aceleaşi condiţii ca şi OWA-RLVQ pentru a putea face o comparaţie obiectivă a rezultatelor şi a performanţelor lor. Deoarece MIRLVQ şi ERLVQ sunt similari, rezultatele înregistrate în tabelele Tabelul 10,

Tabelul 11, Tabelul 12 şi Tabelul 13 sunt cele pentru ERLVQ. Ele sunt identice cu cele obţinute cu MIRLVQ, cu observaţia că în primul caz valoarea constantei $\alpha$ trebuie înjumătăţită.

Importanţa componentelor poate fi diferită de cea obţinută pentru OWA-RLVQ din următorul motiv: OWA-RLVQ este focalizat pe o separare cât mai bună a centrelor clusterelor. ERLVQ încearcă, prin definiţie, o distincţie cât mai bună la limita dintre clustere deoarece foloseşte întotdeauna câte două eşantioane consecutive din clase diferite. Aşadar, OWA-RLVQ găseşte acele componente care sunt importante pentru accentuarea centrelor clusterelor, iar ERLVQ pe acelea care fac o discriminare cât mai bună între două clase adiacente.

Comparaţia ratelor de recunoaştere pentru algoritmii LVQ1, RLVQ, OWA-RLVQ şi ERLVQ din Tabelul 13 arată că ultimul dintre ei oferă cele mai bune rezultate pentru toate cele trei baze de

date testate. Pentru baza de date Deterding care reprezintă o cunoscută problemă de recunoaşterea vorbirii, creşterea de performanţă obţinută cu ERLVQ este de aproape 1%. Această metodă oferă rezultate satisfăcătoare şi pentru probleme generale de recunoaştere a formelor cum sunt Iris şi Ionosphere pentru care performanţele rămân superioare.

Tabelul 10. Rangul componentelor vectoriale pentru baza de date Iris.

| Rangul | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Componentele RLVQ | 4 | 2 | 3 | 1 |
| Componentele OWA-RLVQ | 4 | 3 | 2 | 1 |
| Componentele ERLVQ | 1 | 2 | 3 | 4 |

Tabelul 11. Rangul componentelor vectoriale pentru baza de date Deterding.

| Rangul | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Componentele RLVQ | 2 | 5 | 1 | 9 | 6 | 3 | 4 | 8 | 7 | 10 |
| Componentele OWA-RLVQ | 8 | 2 | 4 | 5 | 6 | 9 | 3 | 7 | 1 | 10 |
| Componentele ERLVQ | 2 | 1 | 3 | 4 | 6 | 8 | 9 | 5 | 10 | 7 |

Tabelul 12. Rangul componentelor vectoriale pentru baza de date Ionosphere. Sunt reprezentate cele mai importante 5 componente.

| Rangul | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Componentele RLVQ | 20 | 28 | 26 | 12 | 6 |
| Componentele OWA-RLVQ | 14 | 12 | 1 | 3 | 28 |
| Componentele ERLVQ | 8 | 24 | 16 | 12 | 6 |

Tabelul 13. Comparaţie între ratele de recunoaştere obţinute cu LVQ1, RLVQ, OWA-RLVQ şi ERLVQ.

| Baza de date | LVQ1 | RLVQ | OWA-RLVQ | ERLVQ |
|---|---|---|---|---|
| Iris | 91,33% | 95,33% | 96,60% | 97,33% |
| Deterding | 44,80% | 46,32% | 46,75% | 47,18% |
| Ionosphere | 90,06% | 92,71% | 93,37% | 94,03% |

## Concluzii

Am introdus în acest capitol o nouă metodă de calcul al relevanţelor pentru componentele vectorilor de intrare într-un un sistem de recunoaştere a formelor. Cei doi algoritmi, MIRLVQ şi ERLVQ, sunt similari dar primul este dezvoltat prin maximizarea entropiei pătratice Renyi, iar al doilea maximizează o măsură bazată pe energia informaţională a lui Onicescu. MIRLVQ şi ERLVQ reprezintă o tehnică alternativă algoritmului OWA-RLVQ pentru calculul relevanţelor.

Algoritmii propuşi de noi oferă o altă perspectivă pentru recunoaşterea formelor, calculând relevanţele prin folosirea ca suport matematic al puternicului aparat reprezentat de teoria informaţiei. Metoda noastră stabileşte on-line valorile relevanţelor prin continua adaptare la fluxul de date de instruire. Relevanţele pot fi folosite la ordonarea în mod dinamic a componentelor vectorilor de intrare în funcţie de importanţa lor, dar şi la reducerea dimensiunii vectorilor prin renunţarea la componentele mai puţin importante.

# Contribuţii originale

Lucrarea abordează următoarele două probleme:

- Cum se poate transfera conceptul de concurenţă la nivelul unei colecţii de reţele neurale pornind de la ideea de modularitate?
- Care este impactul contribuţiei individuale a componentelor intrărilor asupra performanţelor unor sisteme de recunoaştere a formelor bazate pe reţele neurale?

Modelul reţelelor neurale concurente (CNN) din capitolul 4 este o soluţie consistentă pentru prima problemă, iar algoritmii OWA-RLVQ, MIRLVQ şi ERLVQ din capitolele 5 şi 6 reprezintă răspunsul nostru la a doua întrebare.

Contribuţiile pe care le aduce această lucrare se regăsesc în capitolele 3, 4, 5 şi 6.

**Recunoaşterea vorbirii cu modele clasice de reţele neurale**

Este dificil să comparăm performanţele mai multor tipuri de reţele neurale atâta timp cât testele sunt realizate pe date diferite. Din acest motiv, nu putem trage o concluzie relevantă prin simpla alăturare a unor rezultate disponibile în diverse lucrări. Am preferat să facem un studiu comparativ al calităţii recunoaşterii cu perceptroni multistrat, reţele neurale cu întârziere în timp şi cu hărţi cu autoorganizare a unui set de cuvinte izolate, pe de o parte, şi a vorbitorilor dintr-un grup de subiecţi, pe de altă parte. Capitolul 3 detaliază metoda de lucru şi rezultatele pe care le-am obţinut. Am folosit bazele de date Speechdata şi Telephdata care conţin pronunţii în limba engleză ale tuturor cifrelor, prima dată înregistrate într-un studio special amenajat şi a doua oară înregistrate prin telefon. Datorită limitărilor pe care le produce transmisia pe liniile telefonice, a doua bază de date ridică dificultăţi suplimentare în recunoaştere. Metoda de preprocesare a semnalelor vocale descrisă în secţiunea 3.2 este binecunoscută. Aceasta constă din calculul coeficienţilor cepstrali pe ferestre de semnal care sunt suprapuse pentru a surprinde evoluţia temporală a semnalelor vocale. Experimental am stabilit că aceste modele neurale nu ne conduc la rezultate satisfăcătoare chiar dacă pentru testele de recunoaştere a cuvintelor izolate avem rate de succes bune. Am putea argumenta că problemele de recunoaştere a vorbitorilor din Speechdata şi Telephdata pe care ne-am propus să le rezolvăm sunt dificile însă, cu toate acestea, rezultatele pe care le-am grupat în Tabelul 2 nu sunt suficiente.

**Reţele neurale concurente**

Este evident că pe măsură ce o reţea neurală are de rezolvat o problemă din ce în ce mai puţin dificilă ea conduce la performanţe mai bune. Pentru a reduce complexitatea recunoaşterii printr-o singură reţea neurală a întregii colecţii de cuvinte izolate sau a tuturor vorbitorilor, am ales soluţia unei reţele neurale modulare formată din reţele neurale specializate pe subprobleme ale problemei iniţiale. în capitolul 4 am introdus noul model de clasificare al Reţelelor Neurale Concurente (*Concurrent Neural Networks* - CNN) ca o colecţie de reţele neurale de dimensiuni reduse care lucrează în paralel, iar clasificarea se face conform regulii *câştigătorul ia tot*.

Instruirea reţelelor neurale concurente porneşte de la presupunerea că fiecare modul este instruit cu propriul set de date. Sistemul este alcătuit din reţele neurale cu diverse arhitecturi. Noi am folosit module de tip perceptron multistrat, de tip reţea neurală cu întârziere în timp şi de tip hartă cu autoorganizare, însă se pot folosi şi alte variante. Schema de recunoaştere constă dintr-o colecţie de module instruite pe câte o subproblemă şi un modul care selectează cel mai bun răspuns. Algoritmii de instruire şi de recunoaştere din secţiunea 4.2 implementează aceste două tehnici care sunt particularizate în paragraful 4.3 pentru perceptroni multistrat, reţele neurale cu întârziere în timp şi hărţi cu autoorganizare. MLP-CNN şi TDNN-CNN folosesc module instruite supervizat şi seturile de vectori de instruire conţin atât exemple pozitive cât şi negative. Spre deosebire de acestea, SOM-CNN constă din module care sunt instruite printr-un algoritm nesupervizat şi datele constau doar din exemple pozitive. Secţiunea 4.3 oferă detaliile legate de testele de recunoaştere a vorbitorilor din bazele de date Speechdata şi Telephdata cu reţele neurale concurente. în comparaţie

cu rezultatele din capitolul 3, de data aceasta progresele sunt substanțiale iar performanțele sunt, în medie, cu 30%-35% mai bune.

Remarcăm comportamentul foarte bun al SOM-CNN pe care îl folosim mai departe în recunoaşterea unor vocale din limba română. în acest scop, am realizat o bază de date care constă din secvențe vocalizate extrase din pronunții în limba română ale celor zece cifre. Am preprocesat aceste semnale vocale prin aceeaşi metodă folosită pentru Speechdata şi Telephdata pe care am descris-o în paragraful 3.2. Energia semnalului calculată pe ferestre oferă o primă informație legată de prezența sau absența unei vocale şi ne permite să selectăm doar aceste zone. Identitatea vocalelor este stabilită apoi de o rețea neurală concurentă SOM-CNN.

Am realizat cinci tipuri de experimente diferite pentru a ne forma o imagine corectă asupra modelului propus de noi în acest capitol şi pentru o analiză detaliată a sa. Primul experiment a fost cel de recunoaştere a celor şase vocale şi semivocale cu care am instruit sistemul. Rezultatele experimentale au arătat în mod clar performanțele foarte bune ale SOM-CNN pentru care am obținut un progres de peste 10% față de SOM, până la 84,21%.

Celelalte experimente au analizat elemente foarte importante care contribuie la calitatea recunoaşterii: proporția în care dimensiunile reale ale zonelor vocalizate şi nevocalizate sunt sesizate de SOM şi SOM-CNN, numărul de succesiuni vocală-semivocală recunoscute corect, proporția în care sistemul sesizează prezența unei consoane în locul unei vocale sau a unei semivocale şi procentul în care sunt confundate vocalele. De fiecare dată când am folosit rețelele neurale concurente rezultatele au fost superioare. Putem considera că modelul propus de noi în capitolul 4 este o bună alternativă pentru sistemele clasice de recunoaştere a formelor.

**Recunoaştere prin calculul relevanțelor intrărilor folosind operatori OWA**

În capitolele 5 şi 6 analizăm problema impactului contribuției individuale a componentelor intrărilor asupra performanțelor unor sisteme de recunoaştere a formelor care folosesc rețele neurale.

Importanțele componentelor vectoriale ale intrărilor într-un sistem de recunoaştere a formelor, pe care le-am numit relevanțe sau factori de relevanță, sunt calculate în capitolul 5 ca ponderi ale unor operatori speciali de agregare numiți operatori OWA. Am prezentat această clasă de operatori în secțiunea 5.3, am subliniat proprietățile lor matematice şi am prezentat o tehnică de instruire supervizată a lor. Metoda introdusă de noi prin algoritmul OWA-RLVQ foloseşte relevanțele pentru a defini un nou algoritm LVQ în care prototipurile sunt modificate după o regulă ce înlocuieşte distanța euclidiană cu una care înglobează ponderile OWA. în paralel cu actualizarea prototipurilor sunt calculați şi factorii de relevanță după o regulă descrisă în detaliu în paragraful 5.4. Această regulă se bazează pe ideea că un vector clasificat corect induce o mărire a relevanțelor, în timp ce unul clasificat incorect provoacă diminuarea acestor factori.

Am experimentat algoritmul OWA-RLVQ pe baze de date standard, disponibile în rețeaua Internet, pentru recunoaşterea formelor, după cum se poate vedea in paragraful 5.5. Am obținut rezultate superioare atât față de algoritmul LVQ standard cât şi față de algoritmul RLVQ care calculează, de asemenea, relevanțele, însă după o metodă motivată euristic.

Algoritmul OWA-RLVQ este important deoarece face legătura între algoritmul RLVQ şi consistentul model matematic al operatorilor OWA. Metoda noastră are rezultate bune atât pentru probleme de recunoaştere vocală cât şi pentru probleme generale de recunoaştere a formelor. Relevanțele pot fi folosite şi pentru stabilirea dinamică a rangului fiecărei dimensiuni a vectorilor de intrare.

**Recunoaştere prin calculul relevanțelor intrărilor folosind noțiuni de teoria informației**

În capitolul 6 introducem o nouă metodă de recunoaştere a formelor care foloseşte factori de relevanță calculați prin maximizarea fie a informației mutuale dintre intrări şi clase, fie a unei mărimi bazate pe energia informației. Algoritmul ERLVQ, care se dovedeşte a fi similar cu MIRLVQ, este o alternativă la OWA-RLVQ deoarece calculează relevanțele care conduc la o bună

delimitare a claselor, în timp ce al doilea algoritm calculează relevanţele care reorientează prototipurile claselor către centroizii acestora.

Informaţia mutuală se calculează ca o diferenţă a două entropii Shannon. Este cunoscută, însă, complexitatea de calcul pe care o implică estimarea entropiei Shannon prin eşantioane. Pentru a evita acest dezavantaj, am folosit o metodă recentă de aproximare a informaţiei mutuale prin estimarea densităţilor de probabilitate cu ferestre Parzen. Această metodă pe care am tratat-o în detaliu în secţiunea 6.3 foloseşte nuclee gaussiene multidimensionale. Una dintre proprietăţile lor remarcabile este că integrala produsului a două nuclee gaussiene este tot o gaussiană. în anexa B am demonstrat această proprietate. Estimarea densităţii de probabilitate cu ferestre Parzen se foloseşte pentru estimarea entropiei Renyi care include şi entropia Shannon ca un caz particular. Printr-o similaritate cu estimarea entropiei Renyi se poate estima informaţia mutuală sub forma informaţiei mutuale pătratice, după cum arătăm în paragraful 6.4.

Odată stabilit acest cadru formal, am dezvoltat algoritmul MIRLVQ printr-o metodă de gradient crescător pentru maximizarea informaţiei mutuale dintre variabila aleatoare continuă a intrărilor şi variabila aleatoare discretă a claselor. Pentru aceasta, mai întâi am introdus în secţiunea 6.5 algoritmul general *Weighted LVQ* care sintetizează etapele parcurse de un algoritm ce foloseşte relevanţele pentru clasificarea prin LVQ. OWA-RLVQ este o particularizare a acestui algoritm, iar MIRLVQ se bazează tot pe acest şablon.

Elementul cheie al lui MIRLVQ este faptul că el actualizează relevanţele prin compararea apartenenţei la aceeaşi clasă a doi vectori de instruire consecutivi. Vectorii nu mai sunt apoi folosiţi, lucru care îi conferă acestui algoritm atributul de on-line. El poate fi folosit pentru fluxuri de date de instruire şi relevanţele se adaptează continuu, reflectând în mod dinamic schimbările care se petrec asupra intrărilor.

Algoritmul ERLVQ introdus în paragraful 6.7 este similar lui MIRLVQ, deşi este dezvoltat prin maximizarea unei mărimi bazate pe energia informaţională. Definiţia energiei informaţionale ne permite să folosim şi de această dată estimarea densităţilor de probabilitate cu ferestre Parzen şi obţinem o formulă de actualizare a relevanţelor pe baza unor vectori consecutivi din setul de instruire. Regulile de modificare iterativă a relevanţelor pe care le-am dedus prin cele două metode au fost integrate în algoritmul *weighted LVQ* rezultând procedura din secţiunea 6.7.

Am experimentat algoritmul ERLVQ în condiţii similare cu OWA-RLVQ pentru a putea face o comparaţie corectă a rezultatelor. Am constatat că abordarea prin prisma teoriei informaţiei conduce la rate de recunoaştere mai bune, inclusiv pentru baza de date Deterding care este un cunoscut benchmark în recunoaşterea vorbirii.

Cele două modele de recunoaştere a formelor sintetizate prin algoritmii OWA-RLVQ şi ERLVQ reflectă două abordări diferite ale aceleiaşi probleme. Factorii de relevanţă calculaţi fie prin prisma teoriei operatorilor OWA, fie prin cea a teoriei informaţiei conduc la rezultate care îmbunătăţesc ratele de recunoaştere în comparaţie cu algoritmi puternici care implementează aceeaşi gamă de aplicaţii. în plus, relevanţele sunt utile la ordonarea dinamică a componentelor vectorilor de intrare în funcţie de importanţa lor (*feature ranking*) sau la eliminarea componentelor puţin importante (*feature selection*).

Pentru orice tehnică de recunoaştere sau clasificare automată a formelor se poate face presupunerea că rezultatele obţinute ar putea fi dependente de datele de instruire şi de test. Acest lucru este adevărat în majoritatea cazurilor, motiv pentru care nu putem susţine superioritatea absolută a unei metode. Putem, însă, să folosim date cât mai diverse pentru a testa propriile tehnici sau ne putem verifica metodele în diferite contexte şi atunci spectrul cazurilor pentru care abordarea noastră este superioară altora se lărgeşte. Din această cauză am testat modelul CNN atât pentru recunoaşterea cuvintelor izolate cât şi pentru recunoaşterea vorbitorilor şi pentru recunoaşterea vocalelor. Dacă pentru primele două categorii de teste am folosit cuvinte în limba engleză, la recunoaşterea vocalelor am utilizat cuvinte pronunţate în limba română. Atunci când am testat algoritmii OWA-RLVQ şi ERLVQ, pe lângă pronunţii vocale am folosit şi eşantioane care, în mod uzual, sunt folosite pentru sisteme generale de recunoaştere a formelor.

# Bibliografie


[1] R. Anand, K. Mehrotra, C.K. Mohan, S. Ranka "Efficient classification for multiclass problems using modular neural networks". IEEE Transactions on Neural Networks 6, 117-124, 1995.

[2] R. Andonie, F. Petrescu "Interacting systems and informational energy". Foundations of control and engineering 11, no. 2, 53-59, 1986.

[3] M. Andrei, I. Ghiţă "Limba română". Editura Didactică şi Pedagogică, Bucureşti, 1983.

[4] J. Basak, R.K. De, S.K. Pal "Unsupervised feature selection using a neuro-fuzzy approach". Pattern Recognition Letters 19, 997-1006, 1998.

[5] R. Battiti "Using mutual information for selecting features in supervised neural net learning". IEEE Transactions on Neural Networks 5, 537-550, 1994.

[6] G. Beliakov "How to build aggregation operators from data". International Journal of Intelligent Systems 18, 903-923, 2003.

[7] Y. Bengio, R. Cardin, R. de Mori, E. Merlo "Programable execution of multilayered networks for automatic speech recognition". Communications of the ACM 32, 195-199, 1989.

[8] C.M. Bishop "Neural networks for pattern recognition". Oxford University Press, New York, 1995.

[9] C.L. Blake, C.J. Merz "UCI repository of machine learning databases". University of California, Irvine, Department of Information and Computer Sciences, http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

[10] T. Bojer, B. Hammer, D. Schunk, K.T. von Toschanowitz "Relevance determination in learning vector quantization". In M. Verleysen (Ed.), Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2001), D-Side publications, 271-276, 2001.

[11] A. Cațaron, V.-E. Neagoe "Concurrent neural networks for speaker recognition". Proceedings of the International Conference on Telecommunications (ICT 2001), Bucureşti, 2001.

[12] A. Cațaron "Vowel recognition using concurrent neural networks". Proceedings of the 8th International Conference on Optimization of Electrical and Electronic Equipments (OPTIM 2002), Braşov, 779-782, 2002.

[13] A. Cațaron, S. Mătase "Speaker recognition using multi-layer perceptron, time delay neural network, self-organizing map and concurrent neural networks". Proceedings of the 8th International Conference on Optimization of Electrical and Electronic Equipments (OPTIM 2002), Braşov, 775-778, 2002.

[14] A. Cațaron, R. Andonie "RLVQ determination using OWA operators". In M.H. Hamza (Ed.), Proceedings of the 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA 2003), September 8-10, Benalmadena, Spain, ACTA Press, 434-438, 2003.

[15] R.K. De, N.R. Pal, S.K. Pal "Feature analysis: neural network and fuzzy set theoretic approaches". Patterns Recognition 10(30), 1579-1590, 1997.

[16] J.R. Deller, J.G. Proakis, J.H.L. Hansen "Discrete-time processing of speech signals". Prentice Hall, Upper Saddle River, New Jersey, 1987.

[17] R.O. Duda, P.E. Hart, D.G. Stork "Pattern classification". John Wiley & Sons Inc., New York, 2001.

[18] J.L. Elman "Finding structure in time". Cognitive Science 14, 179-211, 1990.

[19] D. Erdogmus, J.C. Principe "Generalized information potential criterion for adaptive system training". IEEE Transactions on Neural Networks 13, 1035-1044, 2002.

[20] D. Filev, R.R. Yager "Learning OWA operator weights from data". Proceedings of the Third IEEE International Conference on Fuzzy Systems, Orlando, IEEE Press, 468-473, 1994.

[21] D. Filev, R.R. Yager "On the issue of obtaining OWA operator weights". Fuzzy Sets and Systems 94, 157-169, 1998.
[22] J.W. Fisher III, J.C. Principe “A methodology for information theoretic feature extraction". Proceedings of the IEEE World Congress on Computational Intelligence, Anchorage, Alaska, 1712-1716, 1998.
[23] B. Freisleben, C.A. Bohn "Speaker-independent word recognition with backpropagation networks". In R.F. Albrecht, C.R. Reeves, N.C. Steele (Eds.), Artificial neural nets and genetic algorithms, Proceedings of the International Conference in Innsbruck, Springer, Wien, 243-248, 1993.
[24] L.M. Fu "Neural networks in computer intelligence". McGraw-Hill, New York, 1994.
[25] R.M. Golden "Mathematical methods for neural network analysis and design". The MIT Press, Cambridge, Massachusetts, 1996.
[26] S. Guiasu "Information theory with applications". McGraw-Hill, New York, 1977.
[27] B. Hammer, T. Villmann "Input pruning for neural gas architectures". In M. Verleysen (Ed.), Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2001), D-Facto publications, 283-288, 2001.
[28] B. Hammer, T. Villmann "Batch-RLVQ". In M. Verleysen (Ed.), Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2002), D-Side publications, 295-300, 2002.
[29] B. Hammer, T. Villmann "Generalized relevance learning vector quantization". Neural Networks 15, 1059-1068, 2002.
[30] S. Haykin "Neural networks - a comprehensive foundation". Macmillan College Publishing Company, New York, 1994.
[31] D.O. Hebb "The organization of behaviour: a neuropsychological theory". Wiley, New York, 1949.
[32] D. Huang, T. Chow "Searching optimal feature subset using mutual information". In M. Verleysen (Ed.), Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2003), D-Side publications, 161-166, 2003.
[33] I. Iordan, V. Robu "Limba română contemporană". Editura Didactică şi Pedagogică, Bucureşti, 1978.
[34] R.A. Jacobs, M.I. Jordan, A.G. Barto "Task decomposition through competition in a modular conectionist architecture: the what and where vision tasks". Cognitive Science 15, 219-250, 1991.
[35] G. Jumarie "Relative information". Springer-Verlag, Berlin, 1990.
[36] N.B. Karayiannis “An axiomatic approach to soft learning vector quantization and clustering". IEEE Transactions on Neural Networks 10, 1153-1165, 1999.
[37] N.B. Karayiannis "Soft learning vector quantization and clustering algorithms based on ordered weighted aggregation operators". IEEE Transactions on Neural Networks 11, 1093-1105, 2000.
[38] N.B. Karayiannis, M.M. Randolph-Gips "Soft learning vector quantization and clustering algorithms based on non-euclidian norms: multinorm algorithms". IEEE Transactions on Neural Networks 14, 89-102, 2003.
[39] S.V. Kartalopoulos "Understanding neural networks and fuzzy logic". IEEE Press, Piscataway, New Jersey, 1996.
[40] T. Kohonen "The "neural" phonetic typewriter". IEEE Computer Magazine, March, 11-22, 1988.
[41] T. Kohonen "Self-organization and associative memory". Series in Information Sciences 61, Springer-Verlag, Berlin, 1989.
[42] T. Kohonen "The self-organizing map". Proceedings of the IEEE 78, 1464-1480, 1990.

[43] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen "SOM_PAK: The self-organizing map program package". Laboratory of Computer and Information Science, Helsinki University of Technology, Finland, http://www.cis.hut.fi/research/som_research/nnrc_programs.html, 1995.
[44] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, K. Torkkola "LVQ_PAK: The learning vector quantization program package". Laboratory of Computer and Information Science, Helsinki University of Technology, Finland, http://www.cis.hut.fi/research/som_research/nnrc_programs.html, 1995.
[45] T. Kohonen "Self-organizing maps". Springer-Verlag, 1997.
[46] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, A. Saarela "Self organization of a massive document collection". IEEE Transactions on Neural Networks 11, 574-585, 2000.
[47] M.A. Kraaijveld, J. Mao, A.K. Jain “A nonlinear projection method based on Kohonenş topology preserving maps". IEEE Transactions on Neural Networks 6, 548-559, 1995.
[48] N. Kwak, C.-H. Choi "Improved mutual information feature selector for neural networks in supervised learning". Proceedings of the International Joint Conference on Neural Networks (IJCNN 1999), vol. 2, 1313-1381, 1999.
[49] N. Kwak, C.-H. Choi "Input feature selection for classification problems". IEEE Transactions on Neural Networks 13, 143-159, 2002.
[50] N. Kwak, C.-H. Choi "Input feature selection by mutual information based on Parzen windows". IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 1667-1671, 2002.
[51] I. Lapidot (Voitovetsky), H. Guterman, A. Cohen "Unsupervised speaker recognition based on competition between self-organizing maps". IEEE Transactions on Neural Networks 13, 877-887, 2002.
[52] M. Last, A. Kandel, O. Maimom "Informatic-theoretic algorithm for feature selection". Pattern Recognition Letters 22, 799-811, 2001.
[53] W. Li "Mutual information functions versus correlation functions". Journal of Statistical Physics 60, 823-837, 1990.
[54] D.J.C. MacKay "Information theory, inference, and learning algorithms". Cambridge University Press, 2003.
[55] W.S. McCulloch, W Pitts “A logical calculus of the ideas immanent in nervous activity". Bulletin of Mathematical Byophisics 5, 115-133, 1943.
[56] W.S. McCulloch, W. Pitts "How we know universals: the perception of auditory and visual forms". Bulletin of Mathematical Byophisics 9, 127-147, 1947.
[57] P. Mitra, C.A. Murthy, S.K. Pal "Unsupervised feature selection using feature similarity". IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 301-312, 2002.
[58] V.-E. Neagoe, O. Cula “A fuzzy connectionist approach to vowel recognition". In B. Reusch and D. Dascalu (Eds.), Real World Applications of Intelligent Technologies (part II), Institutul Național de Cercetare şi Dezvoltate în Microtehnologii, Bucureşti, 1998.
[59] V.-E Neagoe, O. Stănăşilă "Recunoaşterea formelor şi rețele neurale - algoritmi fundamentali"". Editura Matrix Rom, Bucureşti, 1999.
[60] V. Neagoe "Concurrent self-organizing maps for automatic face recognition". Proceedings of the 29th International Conference of the Romanian Technical Military Academy, Bucharest, Romania, 35-40, 2001.
[61] V. Neagoe, A. Ropot "Concurrent self-organizing maps for classification of multispectral sattelite imagery". Proceedings of the 6th World Multiconference on Systemics, Cybernetics, and Informatics (SCI 2002), Orlando, Florida, 126-131, 2002.

[62] V. Neagoe, A. Ropot "Concurrent self-organizing maps for pattern classification". Proceedings of the First IEEE International Conference on Cognitive Informatics (ICCI 2002), Calgary, Alberta, Canada, 304-312, 2002.
[63] V. Neagoe, I. Iatan "Face recognition using a fuzzy-gaussian neural network". Proceedings of the First IEEE International Conference on Cognitive Informatics (ICCI 2002), Calgary, Alberta, Canada, 361-368, 2002.
[64] V. Neagoe, A. Ropot "Concurrent self-organizing maps – a powerful tool for face and speaker recognition". Proceedings of the International Conference COMMUNICATIONS 2002, Edited by the Technical Military Academy, Bucharest, Romania, 276-281, 2002.
[65] O. Onicescu "Energia informațională". Studii şi Cercetări Matematice 18, 1419-1420, 1966.
[66] F.J. Owens, R. Andonie, G.H. Zheng, A. Cataron, S. Manciulea "A comparative study of the multy-layer perceptron, the multi-output layer perceptron, the time-delay neural network and the Kohonen self-organizing map in an automatic speech recognition task". Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems (EIS 1998), ICSC Academic Press, 624-629, Tenerife, Spain, February 11-13, 1998.
[67] S.K. Pal, R.K. De, J. Basak "Unsupervised feature evaluation: a neuro-fuzzy approach". IEEE Transactions on Neural Networks 11, 366-376, 2000.
[68] M. Pregenzer, D. Flotzinger, G. Pfurtscheller "Distinction sensitive learning vector quantization - a noise-insensitive classification method". Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 1994), Orlando, Florida, 2890-2894, 1994.
[69] M. Pregenzer, G. Pfurtscheller, D. Flotzinger "Automated feature selection with a distinction sensitive learning vector quantizer". Neurocomputing 11, 19-29, 1996.
[70] J.C. Principe, D. Xu, J.W. Fisher III "Information-theoretic learning". In Simon Haykin (Ed.), Unsupervised adaptive filtering, Wiley, New York, 2000.
[71] I. Purcaru "Informație şi corelație". Editura Ştiințifică şi enciclopedică, Bucureşti, 1988.
[72] L. Rabiner, B.-H. Juang "Fundamentals of speech recognition". Prentice Hall, Englewood Cliffs, New Jersey, 1993
[73] A. Rosetti, A. Lăzăroiu "Introducere în fonetică". Editura Ştiințifică şi Enciclopedică, Bucureşti, 1982.
[74] J.G. Rueckl, K.R. Cave, S.M. Kosslyn "Why are "what" and "where" processed by separate cortical visual systems? A computational investigation". Journal of Cognitive Neuroscience 1, 171-186, 1989.
[75] D.E. Rumelhart, J.L. McClelland (Eds.) "Parallel distributed processing: explorations in the microstructure of cognition". MIT Press, Cambridge, MA, 1986.
[76] A. Sato, K. Yamada "Generalized learning vector quantization". In âdvances in neural information processing systems", G. Tesauro, D. Touretzky, T. Leen (Eds.), vol. 7, MIT Press, 423-429, 1995.
[77] R. Setiono, H. Liu "Neural-network feature selector". IEEE Transactions on Neural Networks 8, 654-662, 1997.
[78] C. Shannon "A mathematical theory of communication". The Bell System Technical Journal 27, 379-423, 623-656, 1948.
[79] V. Ştefănescu "Aplicații ale energiei şi corelației informaționale". Editura Academiei, Bucureşti, 1979.
[80] K. Torkkola, W.M. Campbell "Mutual information in learning feature transformations". Proceedings of the International Conference of Machine Learning, Stanford, CA, USA, 1015-1022, 2000.
[81] K. Torkkola "Nonlinear feature transforms using maximum mutual information". Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2001), Washington DC, USA, 2756-2761, 2001.

[82] K. Torkkola "Learning discriminative feature transforms to low dimensions in low dimensions". In “Advances in neural information processing systems" 14 (NIPS 2001), Vancouver, BC, Canada, MIT Press, 2001.
[83] K. Torkkola "On feature extraction by mutual information maximization". Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2002), Orlando, Florida, 2002.
[84] K. Torkkola "Feature extraction by non-parametric mutual information maximization". Journal of Machine Learning Research 3, 1415-1438, 2003.
[85] V. Torra "Learning weights for weighted OWA operators". Proceedings of the IEEE International Conference on Industrial Electronics, Control, and Instrumentation (IECON 2000), Nagoya, Japan, 2000.
[86] V. Torra "Weighted OWA operators for synthesis of information". Proceedings of the 5th IEEE International Conference on Fuzzy Systems, New Orleans, USA, 996-971, 2000.
[87] L.P.J. Veelenturf “Analysis and applications of artificial neural networks". Prentice Hall, London, 1995.
[88] J. Vesanto, E. Alhoniemi “Clustering of the self-organizing map". IEEE Transactions on Neural Networks 11, 586-600, 2000.
[89] X.L. Xie, G. Beni “A validity measure for fuzzy clustering". IEEE Transactions on Pattern Analysis and Machine Intelligence 13, 841-847, 1991.
[90] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K.J. Lang "Phoneme recognition using time-delay neural networks". IEEE Transactions on Acoustics, Speech, and Signal Processing 37, 328-339, 1989.
[91] A. Waibel, H. Sawai, K. Shikano "Modularity and scaling in large phoneme neural networks". IEEE Transactions on Acoustics, Speech, and Signal Processing 37, 1989.
[92] H. Watanabe “A new view of the formal entropy as a measure of interdependence and its application to pattern recognition". Proceedings of the IEEE International Conference on System, Man, and Cybernetics, Nashville, USA, 2827-2832, 2000.
[93] R.R. Yager "On ordered weighted averaging aggregation operators in multicriteria decisionmaking". IEEE Transactions on System, Man, and Cybernetics 18, 188-190, 1988.
[94] R.R. Yager, D. Filev "Induced ordered weighted averaging operators". IEEE Transactions on System, Man, and Cybernetics 29, 141-150, 1999.
[95] R.R. Yager "Induced aggregation operators". Fuzzy Sets and Systems 137, 59-69, 2003.
[96] D.S. Yeung, X.Z. Wang "Improving performance of similarity-based clustering by feature weight learning". IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 556-561, 2002.
[97] J.M. Zurada "Introduction to artificial neural systems". West Publishing Company, St. Paul, Minessotta, 1992.
[98] *** "Stuttgart Neural Network Simulator". SNNS Group, Institute for Parallel and Distributed High-Performance Systems (IPVR), University of Stuttgart, Germany, http://www-ra.informatik.uni-tuebingen.de/SNNS/, 1995.

**Apariții în anul 2004**

[99] R. Andonie, A. Cațaron „An informational energy LVQ approach for feature ranking”. Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2004), Bruges, Belgium, 2004.
[100] A. Cațaron, R. Andonie “Computing OWA weights as relevance factors”. Proceedings of the 9th International Conference on Optimization of Electrical and Electronic Equipments (OPTIM 2004), Braşov, 2004.
[101] A. Cațaron, R. Andonie “Energy generalized LVQ with relevance factors”. Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2004), Budapest, Hungary, 2004.