

Utilizarea rețelelor neuronale în compresia de date

Prof.dr. Ion IVAN, conf.dr. Constanța BODEA, prep. Daniel VERNIȘ
Catedra de Informatică Economică, A.S.E., București

Multimedia, bazele de date publice, aplicațiile client-server operează cu volume mari de date. Compresia de date este singura soluție pentru construirea aplicațiilor din aceste domenii. Software-ul de compresie de date încorporat în sistemele de operare au un caracter general. În ultimul timp algoritmi de compresie de date au evoluat spre rețele neuronale, rata lor de compresie fiind foarte bună. În lucrare se definesc concepte și se prezintă algoritmi de compresie de date bazați pe rețele neuronale, se efectuează comparații cu algoritmi clasici de compresie, este prezentat software-ul pentru implementarea algoritmilor cu rețele neuronale pentru compresie. Testările s-au efectuat pentru loturi de programe semnificative.

Cuvinte cheie : Compresie de date, rețele neuronale, algoritmi.

1. Introducere

Sistemele de operare și utilitarele moderne evoluează spre creșterea complexității. Spațiul de memorie ocupat de acestea a crescut foarte mult (tabelul 1).

Tabelul 1. Spațiul de memorie ocupat de aplicații

Produs	Necesar RAM (MB)	HDD (MB)
Windows 95	8	70
Windows NT 4.0	16	90
Borland C++ 3.1	4	70
Borland C++ 4.0	8	120
Borland C++ 5.0	20	320
Visual C++ 7.0	16	460
Corel Draw 6.0	16	90

Compresia de date este o modalitate eficientă de reducere a spațiului de memorie ocupat de un fișier pe suport. Efectele compresiei de date se reflectă în creșterea numărului aplicațiilor stocate. Acum, există în uz curent o serie de programe utilitare de compresie, destul de cunoscute, unele dintre acestea fiind prezentate în tabelul 2.

Tabelul 2. Programe pentru compresie

Denumire produs	Firma elaboratoare	Grad mediu de compresie
DriveSpace	Microsoft	50%
Stacker	Stacker	50 - 60%
Arj	Robert Jung	60%
WinZip	Nico Mak Computing	60 - 65%
WinRar95	Eugene Roshal	70%

Problema devine mai importantă pe măsură ce dimensiunile fișierelor cresc. Prelucrările

de filme, stocările de imagine TV, hyper-textul și chiar arhivele de imagini presupun fișiere de dimensiuni foarte mari. Un CD-ROM cu cărți stocate conține 120-130.000 de pagini, ceea ce reprezintă în medie 4-500 cărți. O bibliotecă de 10^6 volume poate fi astfel stocată pe 2000 de CD-ROM-uri.

2. Compresia și decompresia fișierelor

Necesitatea compresiei de date este impusă de stocarea unui volum cât mai mare de informații în memoria internă a calculatorului. Problema devine cu atât mai importantă cu cât procesul de informatizare a societății devine mai profund.

Compresia este un procedeu prin care se obține o reducere a spațiului de memorare ocupat de un set de date. Prin compresie datele inițiale sunt transformate obținându-se reprezentari echivalente numite și date compresate.

Decompresia este procedeul prin care datele compresate sunt aduse la o formă cât mai apropiată sau chiar identică cu forma pe care au avut-o înaintea compresiei.

Se consideră un alfabet A , format din simbolurile a_1, a_2, \dots, a_n . Fiecare simbol se reprezintă în memoria calculatorului pe un număr de biți. Se definește funcția $h: A \rightarrow N^*$ numită *funcția de lungime* a simbolurilor alfabetului.

Prin compresie de date, se urmărește gasirea unor procedee de trecere (transformare) a fișierului F într-un fișier F' , astfel încât I să fie mult mai mare decât I' . Decompresia este

procedeul de transformare a unui fișier F' , într-un fișier F'' sau chiar în F . Fișierul F'' nu diferă semnificativ de F . El poate fi înțeles și prelucrat cu aceleași rezultate (ca și fișierul F).

Compresia poate fi realizată în două moduri: găsirea unui alfabet eficient și restructurarea textului.

Găsirea unui alfabet eficient. Se construiește un alfabet B format din simbolurile b_1, b_2, \dots, b_n ($k \leq n$). Se urmărește definirea unei funcții $g: B \rightarrow N^*$ astfel încât $g(b_j) \leq h(a_i)$, unde perechea (b_j, a_i) reprezintă modul în care s-a făcut punerea în corespondență a simbolurilor din cele două alfabete.

Dacă funcția $g: T \times B \rightarrow N^*$, unde T este mulțimea frecvențelor de apariție a simbolurilor se obțin algoritmi de tip Huffman. Dacă n este mult mai mare decât k , se lucrează cu forma redusă a alfabetului A , formă dată efectiv de simbolurile din alfabetul A utilizate în fișierul F .

Restructurarea textului. Se iau în considerare particularitățile fiecărui fișier: repetițiile consecutive de simboluri, deosebit de frecvente în reprezentările grafice; grupe de simboluri din alfabetul A , se pun în corespondență cu simboluri din alfabetul B , alături de simbolurile din A , utilizate în text. Un algoritm de compresie nu poate fi folosit fără existența unui algoritm de decompresie. Dacă prin compresie se construiesc perechile (a_i, b_j) , urmărindu-se minimizarea lungimii fișierului F , prin decompresie se construiesc perechile (b_j, a_i) și se obține un fișier foarte apropiat sau chiar identic cu fișierul F .

Un cod este *universal* dacă media lungimii simbolurilor din alfabetul codificat este marginită de $c_1 (H+c_2)$. Fiind dat un fișier sursă arbitrar, cu entropia diferită de zero, un cod universal atinge o lungime medie a simbolurilor, care este în cele mai multe cazuri *codul optim posibil* pentru fișierul respectiv.

Puterea de compresie oferită de un cod universal depinde direct de marimea constantelor c_1 și c_2 . Prin definiție un *cod optim asimptotic* este un cod pentru care media lungimii codurilor se apropie de entropie.

În general, un cod universal cu $c_1=1$ este optim asimptotic. Avantajul codurilor uni-

versale față de codul Huffman constă în faptul că nu este necesară cunoașterea exactă a frecvențelor de apariție a simbolurilor din fișier, fiind necesară doar cunoașterea distribuției acestora. Un alt avantaj este ca setul de simboluri este fix, deci nu este necesară parcurgerea întregului fișier sursă pentru a se determina probabilitățile de apariție a simbolurilor. În timp ce în fișierul sursă poziția simbolurilor este un parametru esențial, pentru codurile universale putem să ne gândim la compresia unui lot de fișiere sursă, ca reprezentări de întregi.

3. Algoritmi și software de compresie bazați pe rețele neuronale

Sistemul folosit este o combinație între două tipuri de modele: rețele de tip Kohonen și Grossberg. În rețea sunt folosite trei straturi, neuronii primului strat (de intrare), fiind folosiți la prelucrarea imaginii. Fiecare neuron din stratul de intrare este conectat cu toți neuronii din stratul al doilea, denumit stratul Kohonen. În final, fiecare nod Kohonen este conectat cu toate nodurile din cel de-al treilea strat, stratul Grossberg. Ponderile legăturilor dintre aceste straturi sunt determinate cu ajutorul unor reguli de instruire. Procesul de compresie/decompresie include următoarele etape:

a) Instruirea stratului de intrare al compresiei folosind un set mic de date. În general aceste date sunt încărcate înaintea compresiei fără a se mai pierde timp cu instruirea rețelei.

b) Instruirea rețelei de decompresie care este, de asemenea, o rețea neuronală de același tip cu rețeaua de compresie, folosind același set de date.

c) Împărțirea imaginii în blocuri de 4×4 sau 8×8 pixeli.

d) Compresia fiecărei subimagini în 6 sau 4 semnale prin folosirea lor ca intrări într-o rețea de compresie.

e) Transmiterea imaginii compresate la receptor, sau stocarea ei.

f) Refacerea imaginii inițiale cu o anumită toleranță, măsurată ca rată a zgomotului, introdus de semnal cu ajutorul rețelei de decompresie.

Pentru a extinde aceste rezultate la compresia de imagini în mai mult de două culori, trebuie dezvoltată o procedură de secționare a imaginilor astfel încât să se poată folosi și pe imaginile cu mai multe culori, sau cu tonuri de gri.

Pentru a comprima și decomprima imagini folosind rețele neuronale, trebuie să descompunem imaginea într-un număr finit de subimagini. Se presupune că imaginea originală are o dimensiune de $N \times N$ pixeli, și poate fi descompusă în $K > 0$ subimagini de $M \times M$ pixeli. Pixelii fiecărei subimagini vor fi compresati cu o rețea neuronală în m biți de date și transmiși la recepție. Cei m biți primiți la recepție vor fi decompresati cu o altă rețea neuronală în subimagini de $M \times M$ pixeli. Se dezvoltă astfel două rețele neuronale, pentru procesele de compresie și decompresie, fiind numite rețeaua de compresie, respectiv rețeaua de decompresie.

Numărul de neuroni din stratul de intrare este de $M \times M$, care este compatibil cu numărul total de pixeli ai unei subimagini. Numărul de noduri Grossberg este m , care reprezintă numărul total de biți care se transmit. Numărul de noduri Kohonen este egal cu numărul total de modele folosite în procesul de instruire a rețelelor de compresie, respectiv decompresie.

Dacă o imagine care trebuie compresată este alb-negru, valorile pixelilor putând fi doar 0 sau 1, procesul de compresie va fi direct. Dacă imaginea are un anumit număr de nuanțe de gri, este necesară o metodă de descompunere a imaginilor într-un anumit număr de imagini alb-negru. În acest studiu, se lucrează numai cu imagini de maxim 4 nivele de gri. Din moment ce fiecare pixel are nevoie doar de doi biți pentru reprezentarea celor patru nivele de gri, adică 00, 01, 10, 11, bitul cel mai semnificativ și cel mai puțin semnificativ pot fi separați, obținându-se două imagini alb-negru.

Cele două imagini pot fi compresate, transmise și decompresate separat. Combinația dintre aceste imagini, după decompresie, va avea ca rezultat imaginea inițială în patru nuanțe de gri, sau o aproximație a acesteia. Rezultate destul de bune pentru imagini în nuanțe de gri s-au obținut și cu implementări

ale rețelelor neuronale de tip backpropagation, dar acestea prezintă un grad mare de pierdere a informației.

Modelul rețelei neuronale pentru compresia de imagine. Modelul rețelei neuronale folosită la compresia imaginilor este de tip counterpropagation și este prezentat în figura 1. Nodurile din stratul de intrare sunt considerate simple noduri de distribuție.

Un nod Kohonen K_i din stratul intermediar are următoarea proprietate:

$$K_i = \begin{cases} 1, & \text{daca } K_i = \max_q \{T_q\} \\ 0, & \text{altfel} \end{cases}$$

unde:

$$T_q = \sum_{p=1}^P w_{qp} \cdot x_p$$

Această proprietate poate fi considerată ca o regulă de genul: câștigătorul ia totul. Valoarea nodului Grossberg, G_r , putând fi determinat după următoarea ecuație:

$$G_r = \sum_{q=1}^Q w_{rq} \cdot T_q$$

Recalcularea ponderilor între intrări și stratul Kohonen se face după formula :

$$w_{qp}(n+1) = w_{qp}(n) + \alpha(x_p - w_{qp}(n)), p = 1, \dots, P$$

unde w_{qp} reprezintă ponderea legăturii conexiunii unui nod de intrare x_p cu nodul Kohonen activat K_p , α fiind rata de învățare a rețelei, și $n = 1, \dots, N$.

Refacerea ponderilor dintre straturile Kohonen și Grossberg poate fi determinată prin ecuația :

$$w_{rq}(n+1) = w_{rq}(n) + \beta(y_r - w_{rq}(n)T_q), r = 1, \dots,$$

unde w_{rq} reprezintă ponderea legăturii conexiunii unui nod Kohonen K_p cu un nod Grossberg G_r , y_r fiind valoarea dorită a ieșirii G_r .

Modelul de rețea din figura 1 este folosită în scopul compresiei. Intrarea este un bloc $M \times M$ de pixeli, adică o subimagine în două culori și ieșirea dorită este un cod de m biți de date comprimate.

Instruirea rețelei. Procesul de instruire prezintă două aspecte principale: selecția unui set de modele de instruire; instruirea rețelei de compresie și a celei de decompresie .

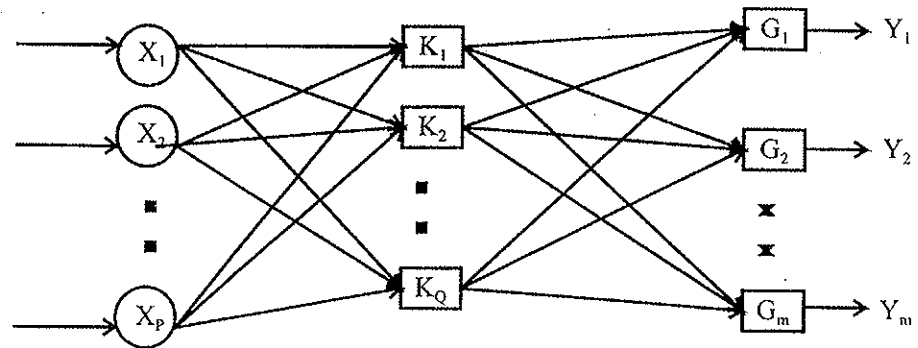


Fig.1. Rețeaua counterpropagation

Selecția modelelor de instruire este independentă de imaginea transmisă. În acest fel, se încearcă instruirea rețelei o singură dată și compresia/decompresia oricărei imagini fără reinstruirea rețelei. Reinstruirea se poate face dacă se consideră că există un set de modele mai bun decât setul curent. Instruirea rețelei de compresie este foarte ușor de înțeles. De exemplu, un model de instruire

$M \times M$ este introdus în rețea iar ținta este un cod de m biți. Dacă se utilizează modele de dimensiunea 4×4 pixeli, va rezulta un cod de 4 biți ca țintă. Figura 2 prezintă rețeaua de decompresie. Folosind coduri de 4 biți și intrări de 4×4 zerouri, ținta obținută din rețeaua de decompresie va fi setul de instruire original.

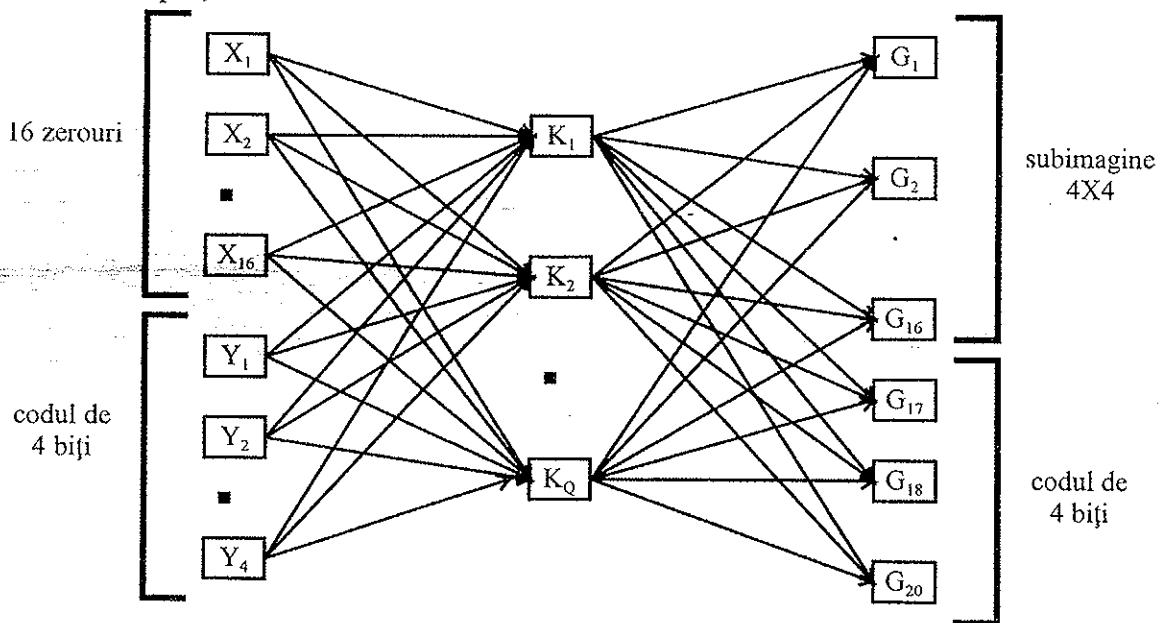
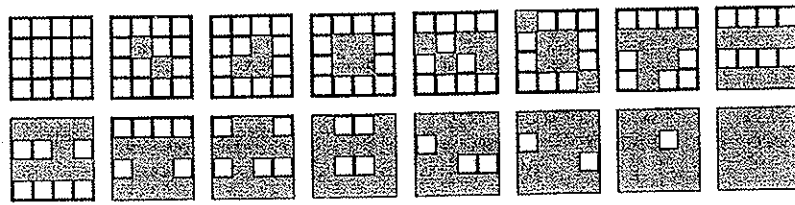


Fig. 2. Decompresia datelor unei imagini

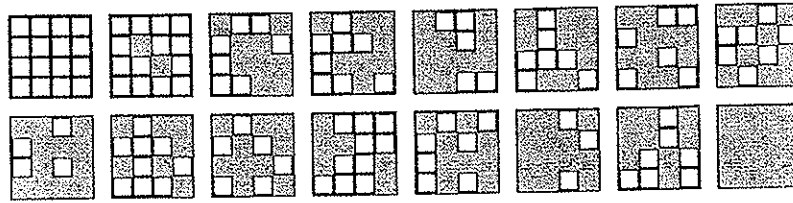
Pentru o acoperire cât mai bună a diferitelor tipuri de imagini se folosesc 3 seturi de modele de instruire. Prin folosirea unor seturi de instruire diferite, se obțin ponderi diferite pentru rețelele de compresie și decompresie. Primul set de modele are drept caracteristică principală, numărul diferit de pixeli dintre modelele de instruire. Cel de-al doilea set de modele este generat aleator prin folosirea unei funcții de generare a numerelor aleatoare în limbajul C. Ultimul

set de modele este generat prin contorizarea celor mai frecvente subimagini care apar în binecunoscutul exemplu de test 'Lena'. Cele trei modele sunt prezentate în figura 3. Modelele sunt folosite pentru instruirea rețelelor de compresie și decompresie.

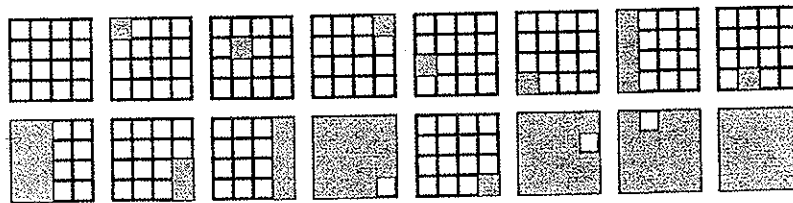
Pentru subimagini de dimensiuni mai mari, ca de exemplu 8×8 sau 12×12 , rezultatele sunt destul de slabe, cantitatea de informație pierdută fiind cu mult mai mare decât la modelele 4×4 .



a. Primul set de modele de instruire



b. Al doilea set de modele de instruire



c. Al treilea set de modele de instruire

Fig. 3. Trei seturi de modele de instruire

Studierea compresiei de imagine folosind rețelele neuronale prezintă o nouă abordare în domeniul compresiei/decompresiei de date. Calitatea procesului, în momentul de față, este destul de slabă. Trebuie găsite noi metode de compresie folosind rețelele neuronale. Îmbunătățirea calității procesului poate fi făcută micșorând rata compresiei. Dacă în locul folosirii șirurilor de 4 biți, s-ar asigura o redundanță de încă 4 biți, aceasta ar putea fi suficientă pentru o calitate considerabilă a imaginii. Principala dificultate în procesele de compresie/decompresie folosind rețelele neuronale este selecția unui număr cât mai mic de modele de instruire, pentru reducerea timpului compresiei și al decompresiei. Selecția modelelor de instruire este unul din viitoarele domenii de cercetare în compresia de date.

4. Rezultate experimentale

Experimental, s-a realizat cu ajutorul aplicației **Danar Multimedia Compressor v2.0**, prin rularea cu aceleași seturi de date a următorilor algoritmi: codificare Huffman; compresie aritmetică; codificare LZW cu lungime variabilă și dicționar de 15 biți;

codificare LZW folosind un model de rețea neuronală de tipul celui prezentat anterior. Caracteristicile fișierelor de intrare utilizate sunt date în tabelul 3.

Tabelul 3

Nume	Conținut	Lungimea
1.bmp	fișier imagine	64000 baiți
2.exe	fișier executabil	42000 baiți
3.txt	fișier text	13500 baiți
4.txt	fișier cu numere aleatoare	20000 baiți

Se constituie tabelul cu timpii de compresie înregistrați.

Tabelul 4

	1.bmp	2.exe	3.txt	4.txt
Huf.exe	15273	13306	13353	17237
Ca.exe	4616	4667	4474	4809
Lzw.exe	14913	11744	13332	14501
LzwNN.exe	15620	12770	13988	16026

S-au calculat gradele de compresie pentru fiecare caz.

Tabelul 5 (în procente)

	1.bmp	2.exe	3.txt	4.txt
Huf.exe	27.22	24.79	40.38	1.92
Ca.exe	27.78	25.25	40.81	1.99
Lzw.exe	48.44	36.15	58.28	3.12
LzwNN.exe	51.25	38.60	60.42	3.12

Din rezultatele prezentate rezultă că cei mai favorabili timpi de compresie, pentru toate

cele trei variante se înregistrează în cazul programului **Ca.exe**. Referitor la gradul de compresie programul cu rezultatele cele mai bune s-a dovedit a fi **LzwNN.exe**, în toate variantele luate în calcul. Pentru fișierele eterogene, cum ar fi **4.txt** se observă un grad de compresie foarte mic, precum și timpi de compresie ridicați. Aceste valori pot fi privite ca grade minime de compresie pentru algoritmi prezentați.

Concluzii

Instrumente puternice dezvoltate în ultimul timp în informatică și-au găsit aplicații în compresia de date (fractali, rețele neuronale, algoritmi genetici). Se observă caracterul deschis al domeniului. Se impun însă analize cantitative și calitative ale intrărilor, ieșirilor, transformărilor specifice proceselor de compresie/decompresie și găsirea unei zone de optimalitate. Aceasta trebuie să ia în considerare minimizarea pierderii de informație, maximizarea ratei de compresie, minimizarea duratelor de compresie și minimizarea spațiului ocupat. Fiecarei tipologii de algoritm i se asociază limite ce trebuie estimate. Pe baza acestora are loc o ierarhizare a algoritmilor și crearea unor procese echilibrate de compresie/decompresie. Software-ul dedicat acestor operații nu creează baze de date privind comportamentul, ceea ce impune perfecționarea lui. În acest sens, componentele adăugate trebuie să masoare caracteristicile fișierelor de intrare și consumul de resurse, precum și caracteristicile fișierelor de ieșire. Bazele de date trebuie să contribuie la ajustarea comportamentului programelor de compresie prin modificarea criteriilor de selecție a algoritmilor.

Asigurarea unei largi circulații a textelor sursă ce corespund implementărilor de algoritmi este o condiție pentru dezvoltarea de aplicații pe compresie de date.

Software-ul aplicativ devine complet dacă operează cu fișiere compresate proprii. De exemplu, un mediu de programare pentru limbajul C++ este complet dacă la ieșire oferă un program sursă compresat, dacă la compilare se face decompresia fișierului text sursă și compresia fișierului obiect, dacă la link-editare se face decompresia fișierului obiect și compresia fișierului executabil, iar la execuție se face decompresia și lansarea în execuție a fișierului executabil. Acele funcții de compresie sau decompresie la dispoziția programatorului au rolul de a încorpora utilitate în aplicații și de a elimina transparența compresiei/decompresiei prin comenzi impuse utilizatorului.

Bibliografie

- [Be90] Bell T.C., Cleary J.G., Witten I.N., Text Compression, Prentice Hall Englewood Cliffs, N.J. 1990
- [Bo97] Constanța Bodea, Paralel Algorithms for Neural Networks, A.S.E., Proceedings of the 3rd International Symposium of Economic Informatics, Bucharest 1997.
- [Dc95] ***, Data Compression Conference, Snowbird, UT, 28-30 martie 1995, ISBN 0-8186-7012-6, Editura IEEE Compu Press
- [Iv95] I.Ivan, D. Verniș, Analiza comparată a algoritmilor de compresie date, PC World, nr.12, Decembrie 1995
- [Iv95] I.Ivan, D. Verniș, Comprimarea datelor, PC Report, nr. 9, Septembrie 1996
- [Iv96] I.Ivan, D. Verniș, Evaluarea seturilor de date destinate compresării, Revista de statistică, nr.1, Ianuarie 1996
- [Ve96] Daniel Verniș, Comunicarea "Algoritmi de compresie pentru șiruri de date numerice", Cercul de Structuri de Date, A.S.E. Bucuresti, 1996.
- [Ve97] Daniel Verniș, Implementarea algoritmilor de compresie folosind rețele neuronale, Studiu ASE, iunie 1997.