

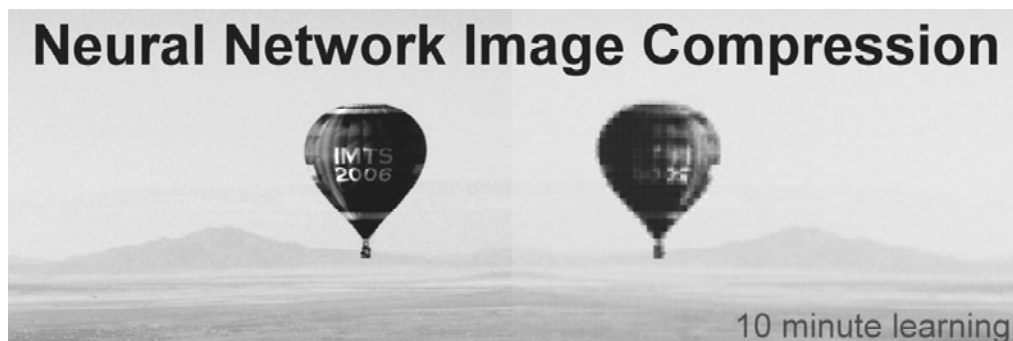
Proiect Analiza Algoritmilor

Rețele neurale

Comprimarea Imaginilor

Ureche Vlad

331CA



1. Prezentare teoretică

1.1 Modelul neuronului

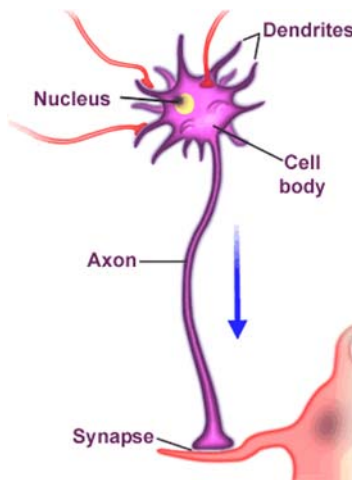
Creierul, prin intermediul neuronilor permite adaptarea la mediu. Adaptarea la mediu se face prin procesul de învățare. Astfel se încearcă mai multe comportamente până când rezultatul este cel dorit. Spre exemplu, de fiecare dată când mergem folosim reacțiile deja învățate și pe baza simțurilor încordăm anumiți mușchi pentru a rămâne drepti. Putem desprinde două procese:

- învățarea propriu-zisă – la varste fragede „ne antrenăm” să mergem pe 2 picioare
- folosirea reacțiilor și simțurilor pentru a face anumite activități



Ceea ce permite mamiferelor să învețe este numărul foarte mare de neuroni din creier. Aceștia formează o rețea foarte deasă cu ajutorul căreia se prelucrează informația venită de la senzorii. În procesul de prelucrare a informației pe baza rețelelor, neuronii folosesc două componente foarte importante:

- dendritele – „intrările” neuronului, zonele receptoare cu ajutorul cărora se poate prelua informație de la senzori sau de la alți neuroni
- axonul – „ieșirea neuronului”, zona prin care se comandă alți neuroni, mușchi sau organe



Legătura care se formează între dendrită și axon se numește sinapsă.

Procesul de învățare se poate vedea ca o buclă mare de feedback în care pe baza rezultatelor, conștient sau inconștient se modifică procesul de luare a deciziilor în

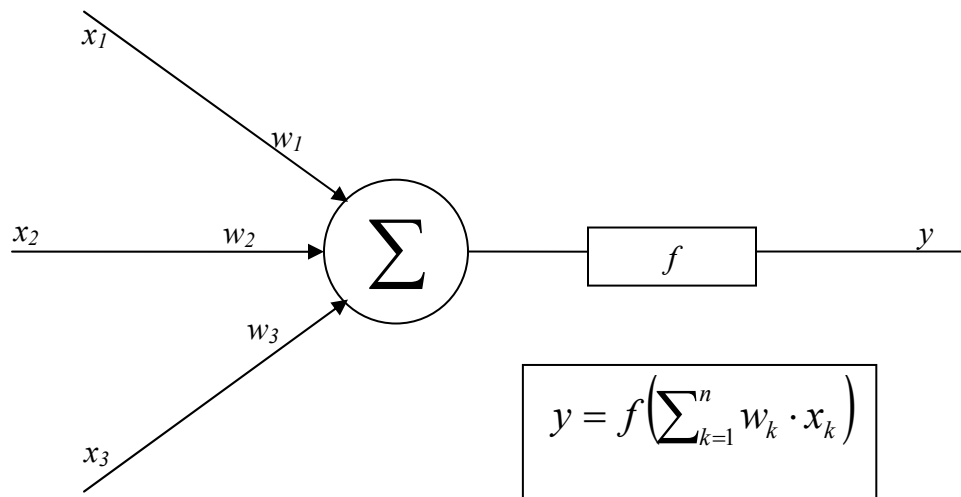
interiorul neuronului, în așa fel încât rezultatul la nivel macro sa fie mai aproape de cel dorit.

1.2 Modelul neuronului în informatică

Neuronul biologic este o entitate foarte complexă și de aceea modelarea acestuia pe calculator ar fi foarte complicată. Există însă un model de neuron accesibil calculatorului, cu ajutorul căruia se pot face operații specifice.

Modelul se bazează pe o serie de ponderi atașate intrărilor neuronului. Pe baza semnalelor de intrare de la alți neuroni și a ponderilor se calculează o valoare care reprezintă ieșirea neuronului.

În biologie se consideră că toate semnalele între neuroni sunt de tip +1, 0 sau -1. Echivalentul ar fi un semnal cuantificat. Însă, într-o masă de neuroni aceștia nu sunt sincronizați și chiar mai mult, emit semnale cu frecvențe diferite. Astfel, după o mediere putem observa că ceea ce se transmite între neuroni nu este o informație cuantificată ci mai mult una continuă. Deci în modelul neuronului din informatică toate intrările, ieșirile și ponderile vor fi de tip real.

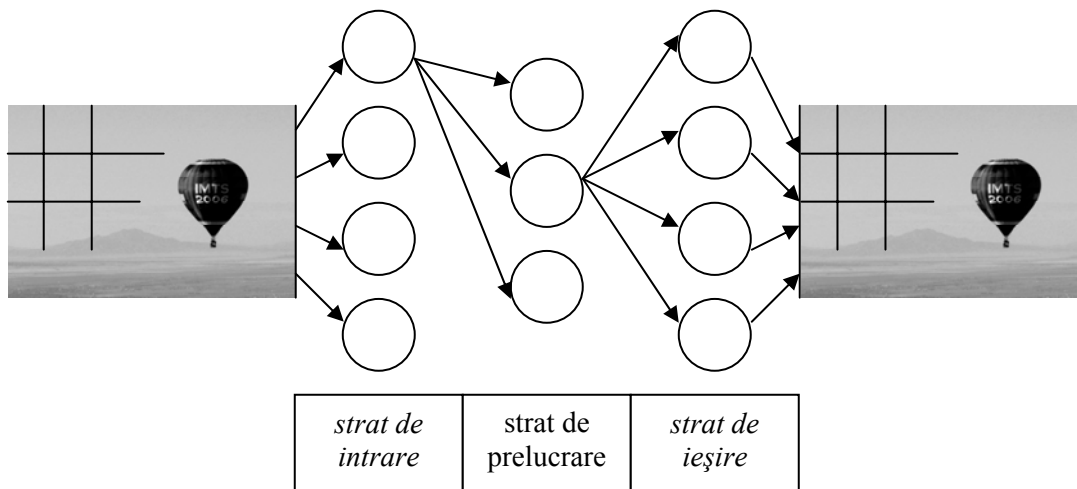


1.3 Rețeaua neurală. Model, Observații

Am spus deja că neuronii nu există de unii singuri ci grupați în rețele. Avantajul de a avea o rețea de neuroni este acela că la o eroare apărută într-o parte a rețelei răspunsul final nu va varia atât de mult. Acest fapt apare datorita posibilității neuronilor de a se substitui reciproc.

Rețelele neurale informatice, odată create nu au nici o informație stocată. Aceste pot fi asemănate cu un copil care nu știe sa mearga încă. Pentru a le „învăța” există două tipuri de procese posibile:

- buclă de feedback, în care pe baza ieșirilor se recalculază ponderile de la intrările fiecărui neuron. Bucla se repetă până când rezultatele obținute la ieșirea rețelei sunt suficient de apropiate de rezultatele dorite
- învățare cognitivă, în care legăturilor des folosite li se atribuie o pondere mai mare, într-un proces de învățare „dinamic”



2. Comprimarea Imaginilor

2.1 Procesul de comprimare

2.1.1 Prima parte: Învățarea imaginii

Pentru a permite rețelei neurale să „învețe” imaginea, trebuie să îi dăm la intrare piese de 8*8 pixeli pe care să-i cerem să-i reproducă la ieșire. Pașii care trebuie făcuți:

- împărțirea imaginii în piese de 8*8 pixeli și scalarea de la 256 de niveluri de culoare la număr real $0 < x < 1$
- aplicarea pieselor de imagine la intrarea rețelei neurale și adunarea feedback-ului care trebuie aplicat pentru fiecare bucată. Ne așteptăm ca fiecare bucată de imagine să fie cât mai bine reprodusă de rețeaua neurală la ieșire
- medierea feedbackului și aplicarea acestuia la ponderile rețelei
- repetarea pașilor de mai sus până când rezultatul este suficient de apropiat de imaginea inițială

2.1.2 Partea a doua: Comprimarea efectivă

După cum am zis mai sus, stratul de prelucrare conține mai puțini neuroni decât straturile de intrare/ieșire. Vom folosi acest fapt pentru a lua ieșirile stratului de prelucrare pentru a fi stocate ca imagine comprimată. Pașii sunt:

- aplicarea pieselor de imagine la intrarea rețelei
- scrierea în imaginea comprimată a ieșirilor neuronilor din stratul de prelucrare, scalate la 256 de niveluri sau la 256*256 de niveluri, după caz
- notarea ponderilor neuronilor din stratul de ieșire, pentru a putea reface imaginea la decomprimare

Observație: Deși rețelele neurale nu se pretează la comprimarea imaginilor color, aceasta se poate face prin împărțirea imaginii în 3 imagini separate, pe fiecare dintre canalele R,

G și B. Aceste canale vor fi folosite pentru a învăța trei rețele neurale separate. Evident și rezultatele vor fi trecute de trei ori în fișier, odată pentru fiecare rețea

2.2 Procesul de decomprimare

Putem observa că neuronii de prelucrare, fiind mai puțini, au fost obligați să rețină în ieșirile lor doar „esențialul” din imagine. Rolul stratului de ieșire este acela de a reface imaginea pe baza indiciilor date de neuronii de intrare. Pașii pentru decomprimarea imaginii sunt:

- Încărcarea ponderilor stratului de ieșire din fișier
- Punerea la intrarea stratului de ieșire a semnalelor luate din fișier (semanle de la stratul de prelucrare) și generarea pieselor de imagine. La acest pas trebuie făcută o translatăre a valorilor din fișier la real, $0 < x < 1$ și o discretizare a ieșirii rețelei pe 256 de niveluri, ca la o imagine pe 8 biți
- formarea imaginii întregi pe baza pieselor

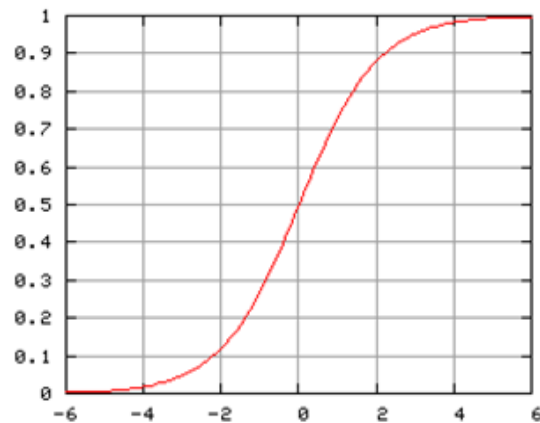
2.3 Tip de neuroni simulați

Neuronii simulați pentru această temă sunt neuroni cu ponderi la intrare. Aceștia fac suma înrărilor ponderate și o prelucrează. Astfel:

$$y = \frac{1}{1 + \exp\left(\sum_{k=1}^n w_k \cdot x_k\right)}$$

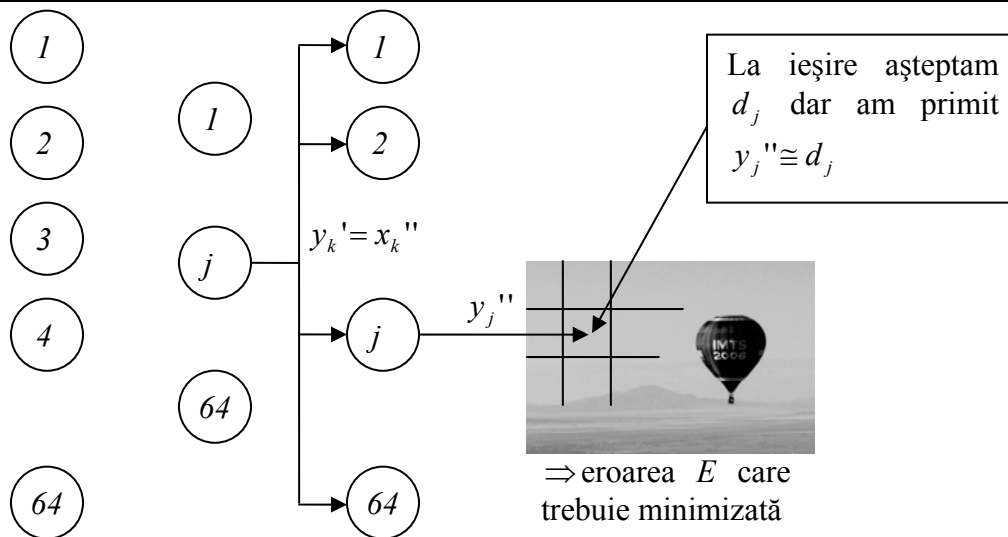
Conform funcției aplicate, neuronii se numesc sigmoizi. Mai trebuie precizat și faptul că neuronii simulați au și un prag static de excitare, ca o pondere aplicată unei intrări care are totdeauna valoarea 1.

Neuronii din stratul de intrare se pot considera identități, deoarece au o singura intrare. Ieșirea este totdeauna egală cu intrarea, fără a face prelucrare prin funcția sigma.



2.4 Tip de rețea simulată

Rețeaua de neuroni simulată pentru comprimarea imaginilor este una simplă, cu 3 straturi, unul de intrare, unul de prelucrare și unul de ieșire. Straturile de intrare și de ieșire au totdeauna același număr de neuroni, pătrat perfect, de obicei 64. (8*8) Comprimarea imaginii se face cu ajutorul stratului de prelucrare, care este întotdeauna format dintr-un număr mai mic de neuroni decât straturile de intrare și de ieșire.



2.5 Modelul matematic

2.5.1 Estimarea erorilor

La fiecare iterație a procesului de învățare, rezultatul este puțin diferit de ceea ce se dorea. Pe baza acestei diferențe se poate calcula eroarea absolută:

$$E = \frac{1}{2} \sum_{\text{toate-piese}} \sum_j (y_j'' - d_j)^2$$

Se poate calcula și Signal to Noise Ratio:

$$SNR = 10 \log_{10} \left(\frac{E}{\sum_{\text{toate-piese}} \sum_j (d_j)^2} \right)$$

2.5.2 Feedback rețea

Scopul fiecărei runde de feedback este să permită minimizarea E prin manipularea ponderilor. Astfel vom avea:

$$\delta w_{ji}'' = -\eta \cdot \frac{\partial E}{\partial w_{ji}''} \text{ pentru neuronii de ieșire respectiv } w_{ji}' \text{ pentru prelucrare}$$

unde η este o constantă numită *constanta de învățare*. Dacă η e prea mic, schimbările sunt nesemnificative și nu se observă o scădere importantă a lui E . Dacă, în schimb η e prea mare apar oscilații la valoarea lui E .

Se pot calcula matematic valorile derivatelor parțiale.

Schimbările asupra rețelei trebuie să conțină feedback-urile pentru toate piesele care compun imaginea mediate, pentru a nu da posibilitatea recunoașterii unei piese mai bine. (Dacă s-ar fi aplicat succesiv, ultima piesă ar deveni cel mai bine recunoscută)

3. Rezultate și probleme întâmpinate

1. Codificare de 256×256 niveluri – pentru a putea comprima imaginea am folosit 16 neuroni în stratul de prelucrare. Pentru a scrie în fișier semnalele nivelului de prelucrare a fost necesară o codificare pe octeți. Am încercat să folosesc un singur octet dar din cauza cuantificării semnalului am obținut erori mai mari decât din cauza procesului limitat de învățare. Am decis apoi să modific cuantificarea astfel încât să ocupe doi octeți, ceea ce a permis valori ale SNR mult mai mari, dependente de timpul alocat învățării rețelei.

Valorile pentru rata de compresie se încadrează între 1:1.3 și 1:2. Teoretic aceasta este 1:2 dar în fișier trebuie incluse și ponderile stratului de ieșire care au dimensiune fixă.

2. Rețeaua învață foarte lent

Imaginea originală:



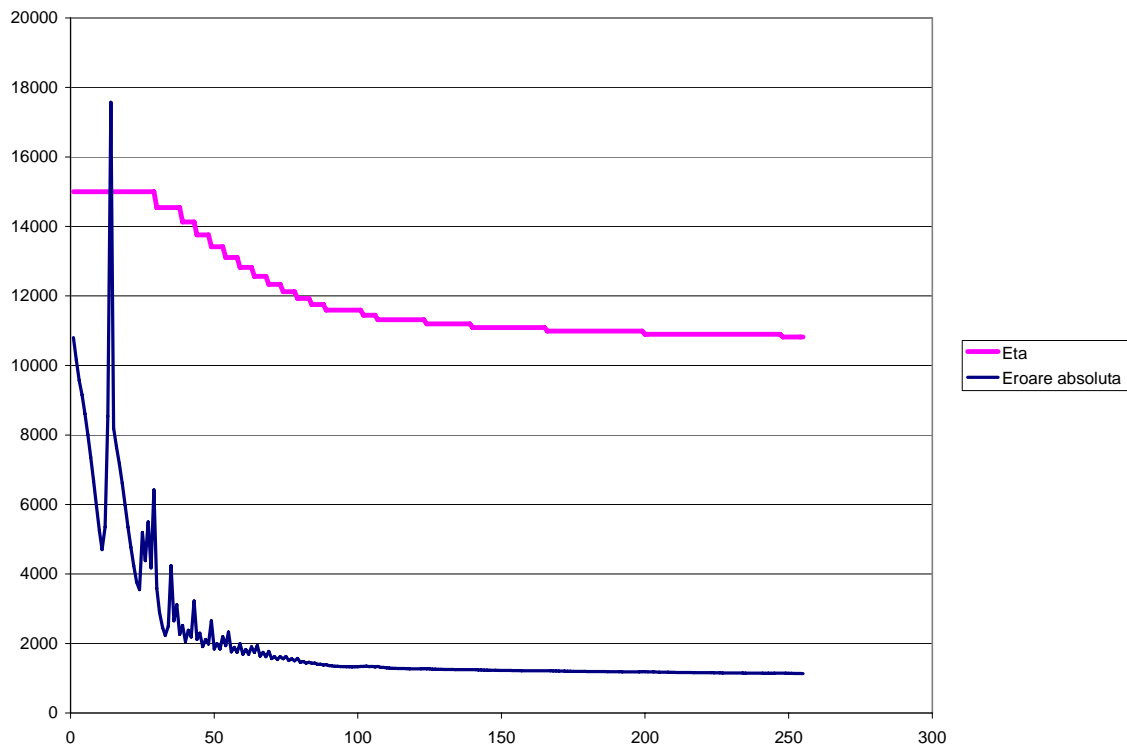
Imaginea după o oră de învățare:



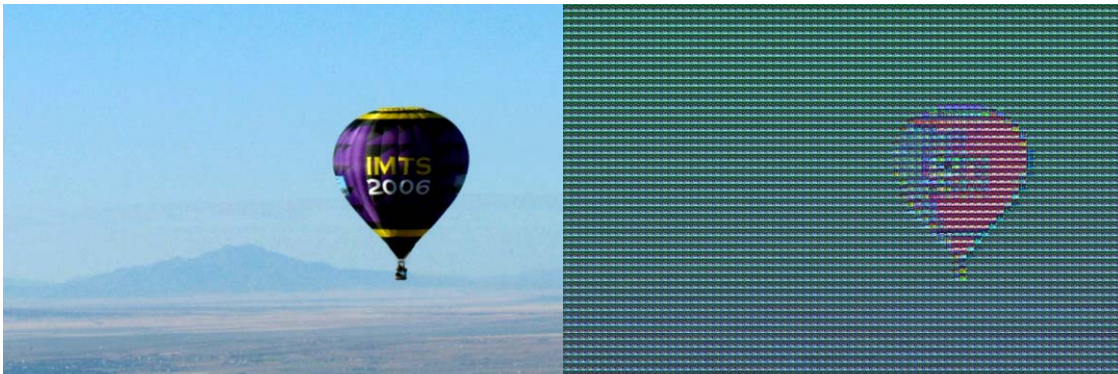
Imaginea după 4 ore de învățare:



3. Eta – Pentru a face cele mai semnificative schimbări fără a aduce oscilații, când am detectat oscilații am micșorat eta. Mai jos este un grafic al valorilor Erorii și Eta (de fapt $\eta' = 10000 + 1000 \cdot \eta$ ca să apară frumos pe grafic)



4. Nu funcționează corect în cazul color: (asta dacă nu cumva e vina mea...)



4. Concluzie

Bine că există JPEG....

5. Bibliografie

1. www.florinpop.ro/articole
2. <http://www.ee.duke.edu/~cec/research/neuralcompression/model.html>
3. <http://easybmp.sourceforge.net/>

6. Alte rezultate



Interfața grafică pe care nu am apucat să o mai termin:

