

Laborator 11

Aplicație:

Recunoașterea caracterelor folosind algoritmul backpropagation

Vom proiecta o rețea pe care o vom antrena astfel încât să recunoască 26 de litere ale alfabetului. Literele vor fi reprezentate printr-un grid 5×7 de valori booleene. Sistemul poate să nu fie perfect, deci literele pot fi afectate de zgomot. Ne propunem să construim un mecanism care să determine o clasificare corectă a literelor ideale și cu o precizie rezonabilă a literelor afectate de zgomot (bruiate).

Literele se pot reține în vectori de dimensiune 35. Este implementată în Matlab funcția **prprob** care returnează o matrice formată din vectorii de intrare numită **alphabet**; de asemenea vectorii target sunt definiți în această funcție printr-o variabilă denumită **targets**. Fiecare ieșire target corespunzătoare unei litere este un vector cu 26 de elemente care reține 1 pe poziția corespunzătoare literei pe care o reprezintă și 0 în rest.

Arhitectura rețelei

Rețeaua primește ca intrare 35 de valori booleene printr-un vector de dimensiune 35. Se dorește identificarea literei prin returnarea unui răspuns reprezentat printr-un vector de ieșire de dimensiune 26. Spunem că rețeaua recunoaște corect caracterele dacă răspunde cu 1 pe poziția care reprezintă numărul de ordine al literei în momentul introducerii acesteia în rețea, iar celelalte valori din vector sunt 0. De asemenea, rețeaua trebuie să poată identifica zgomotele (literele bruiate). În practică, rețeaua nu primește ca intrare un vector de valori booleene perfect. Rețeaua trebuie definită astfel încât să facă cât de puține greșeli posibile de clasificare a vectorilor care au un zgomot de medie 0 și deviație standard cel mult 0.2.

Deci, rețeaua necesită vectori de intrare de dimensiune 35 și 26 de neuroni pe nivelul de ieșire pentru a identifica literele. Vom defini o rețea cu două nivele

având ca funcții de transfer funcția log-sigmoid. Vom considera această funcție întrucât ieșirile sale aparțin intervalului (0, 1), fapt care indică o bună aproximare a unor valori de ieșire booleene. Să presupunem că nivelul ascuns conține 10 neuroni, număr ales pe baza experienței; dacă rețeaua are probleme legate de capacitatea de învățare atunci vom mai adauga neuroni acestui nivel.

Rețeaua este antrenată astfel încât să returneze 1 pe poziția corectă în vectorul de ieșire și 0 în rest, însă pot apărea în rețea și vectori de intrare cu zgomot pentru care nu vor fi returnate valori perfecte de 1 și 0. De aceea, după antrenarea rețelei ieșirea este trecută printr-o funcție de transfer competitivă **compet** care va returna output-ul corespunzător literei celei mai probabile, deci vectorului de intrare afectat de zgomot i se va asocia o ieșire formată din valoarea 1 pe poziția corespunzătoare literei și 0 în rest. Rezultatul acestei post-procesări este de fapt cel care va fi utilizat.

Inițializarea rețelei

Vom crea rețeaua folosind funcția newff.

S1 = 10;

[R,Q] = size(alphabet);

[S2,Q] = size(targets);

P = alphabet;

net = newff(minmax(P),[S1 S2],{'logsig' 'logsig'},'traingdx');

*net.LW{2,1} = net.LW{2,1}*0.01;*

*net.b{2} = net.b{2}*0.01;*

Pentru a recunoaște literele care sunt afectate de zgomot este indicată antrenarea folosind atât vectori ideali, cât și folosind vectori de intrare cu zgomot. Mai întâi rețeaua va fi antrenată doar pe baza mulțimii de antrenare formate din vectori ideali până când se obține o eroare de estimare scăzută.

După aceea rețeaua se va antrena pe baza a 10 mulțimi formate din vectori ideali și vectori cu zgomot. Rețeaua este antrenată pe baza mulțimii de antrenare formate din două copii ale alfabetului având litere neafectate de zgomot și două copii ale alfabetului format din litere bruiate, în același timp. Sunt utilizate două copii ale

alfabetului ideal pentru a se menține abilitatea acesteia de a clasifica vectori de intrare ideali.

Din păcate, după antrenarea descrisă mai sus rețeaua poate învăța să clasifice anumiți vectori bruiți dificili cu prețul unei clasificări corecte a unui vector nebruiat. De aceea, rețeaua este antrenată din nou doar pe baza unor vectori ideali. Acest lucru va asigura corectitudinea recunoașterii unei litere ideale.

Antrenarea va fi realizată folosind algoritmul de backpropagation care utilizează rata de învățare adaptivă și momentum cu funcția **traingdx**.

Antrenarea fără zgomot

Rețeaua este inițial antrenată fără zgomot pentru un număr maxim de 5000 de epoci sau până când eroarea rețelei (suma pătratelor erorilor) este mai mică decât 0.1. Se consideră ca funcție de performanță funcția 'sse', iar parametrul **mc** va fi setat 0.95.

Antrenarea cu zgomot

Pentru a obține o rețea care să recunoască și literele bruiate, o vom antrena folosind două copii ale literelor din alfabet și două copii ale alfabetului având litere bruiate. Vectorul valorilor target conține patru copii ale vectorului target inițial. Vectorii cu zgomot vor avea un zgomot de medie 0 și deviații standard 0.1 și 0.2. Aceasta va întări abilitatea neuronului de a învăța să identifice corect litere bruiate și în același timp să recunoască litere ideale.

Antrenarea folosind și vectori afectați de zgomot se va face în 300 de epoci, iar valoarea erorii admise va crește la 0.6, deoarece poate fi intuită apariția unei erori mai mari datorate introducerii unui număr mai mare de vectori.

Se va repeta acest demers pentru 10 iterații.

Antrenarea fără zgomot din nou

Odată antrenată rețeaua cu zgomot este necesară reantrenarea acesteia fără zgomot pentru a asigura clasificarea corectă a literelor ideale. De aceea rețeaua va fi în acest moment reantrenată pe baza mulțimii de antrenare formate din vectori neafectați de zgomot pentru un număr de 500 de epoci.

Performanța sistemului

Coeficientul de încredere al rețelei de recunoaștere a caracterelor se măsoară testând sistemul cu sute de vectori de intrare cu diferite valori ale zgomotului.

Vom determina procentul de eroare obținut dacă vectorilor de intrare li se adaugă pe rând un nivel de zgomot repartizat normal de medie 0 și deviație standard cuprinsă între 0 și 0.5 cu pasul de 0.05. Pentru fiecare nivel de zgomot se introduc în rețea 100 de observații pentru fiecare literă, se calculează ieșirile asociate, care sunt atribuite funcției de transfer competitive care returnează litera cea mai probabilă. Se adună erorile de clasificare și se calculează procentul obținut.

Să se ploteze nivelul de zgomot versus procentul de eroare corespunzător calculate atât pentru rețeaua antrenată fără litere bruiate, cât și pentru rețeaua antrenată considerând în mulțimea de antrenare și litere bruiate

Pentru a obține o precizie mai mare de clasificare se poate considera o rețea având mai mulți neuroni în nivelul ascuns și de asemenea rezoluția vectorilor de intrare poate fi mărită la o rețea de 10×14 , iar rețeaua poate fi antrenată folosind vectori cu un nivel mai ridicat de zgomot.

Testare rețelei

Pentru a testa rețeaua vom crea o literă cu zgomot pe care o vom introduce în rețea folosind codul de mai jos:

```
noisyJ = alphabet(:,10)+randn(35,1) * 0.2;  
plotchar(noisyJ);  
A2 = sim(net,noisyJ);  
A2 = compet(A2);  
answer = find(compet(A2) == 1);  
plotchar(alphabet(:,answer));
```

Temă: Definiți și antrenați o rețea care să recunoască cifre. Studiați și comparați curbele de eroare obținute în cazul antrenării unei rețele fără zgomot, respectiv în cazul cu zgomot.