Laborator 11

Aplicaţie:

Recunoaşterea caracterelor folosind algoritmul backpropagation

Vom proiecta o reţea 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 şi cu o precizie rezonabilă a literelor afectate de zgomot (bruiate).

Literele se pot reţine în vectori de dimensiune 35. Este implementată în Matlab funcţia **prprob** care returnează o matrice formată din vectorii de intrare numită **alphabet**; de asemenea vectorii target sunt definiţi în această funcţie printr-o variabilă denumită **targets.** Fiecare ieşire target corespunzătoare unei litere este un vector cu 26 de elemente care reţine 1 pe poziţia corespunzătoare literei pe care o reprezintă şi 0 în rest.

Arhitectura reţelei

Reţeaua primeşte ca intrare 35 de valori booleene printr-un vector de dimensiune 35. Se doreşte identificarea literei prin returnarea unui răspuns reprezentat printr-un vector de ieşire de dimensiune 26. Spunem că reţeaua recunoaşte corect caracterele dacă răspunde cu 1 pe poziţia care reprezintă numărul de ordine al literei în momentul introducerii acesteia în reţea, iar celelalte valori din vector sunt 0. De asemenea, reţeaua trebuie să poată identifica zgomotele (literele bruiate). În practică, reţeaua nu primeşte ca intrare un vector de valori booleene perfect. Reţeaua trebuie definită astfel încât să facă cât de puţine greşeli posibile de clasificare a vectorilor care au un zgomot de medie 0 şi deviaţie standard cel mult 0.2.

Deci, reţeaua necesită vectori de intrare de dimensiune 35 şi 26 de neuroni pe nivelul de ieşire pentru a identifica literele. Vom defini o reţea cu două nivele având ca funcţii de transfer funcţia log-sigmoid. Vom considera această funcţie întrucât ieşirile

sale aparţin intervalului (0, 1), fapt care indică o bună aproximare a unor valori de ieşire booleene. Să presupunem că nivelul ascuns conţine 10 neuroni, număr ales pe baza experienţei; dacă reţeaua are probleme legate de capacitatea de învăţare atunci vom mai adauga neuroni acestui nivel.

Reţeaua este antrenată astfel încât să returneze 1 pe poziţia corectă în vectorul de ieşire şi 0 în rest, însă pot apărea în reţea şi vectori de intrare cu zgomot pentru care nu vor fi returnate valori perfecte de 1 şi 0. De aceea, după antrenarea reţelei ieşirea este trecută printr-o funcţie 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 ieşire formată din valoarea 1 pe poziţia corespunzătoare literei şi 0 în rest. Rezultatul acestei post-procesări este de fapt cel care va fi utilizat.

Iniţializarea reţelei

Vom crea reţeaua folosind funcţia 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 recunoaşte literele care sunt afectate de zgomot este indicată antrenarea folosind atât vectori ideali, cât şi folosind vectori de intrare cu zgomot. Mai întâi reţeaua va fi antrenată doar pe baza mulţimii de antrenare formate din vectori ideali până când se obţine o eroare de estimare scăzută.

După aceea reţeaua se va antrena pe baza a 10 mulţimi formate din vectori ideali şi vectori cu zgomot. Reţeaua este antrenată pe baza mulţimii de antrenare formate din două copii ale alfabetului având litere neafectate de zgomot şi două copii ale alfabetului format din litere bruiate, în acelaşi timp. Sunt utilizate două copii ale alfabetului ideal pentru a se menţine abilitatea acesteia de a clasifica vectori de intrare ideali.

Din păcate, după antrenarea descrisă mai sus reţeaua poate învăţa să clasifice anumiţi vectori bruiaţi dificili cu preţul unei clasificări corecte a unui vector nebruiat. De aceea, reţeaua este antrenată din nou doar pe baza unor vector ideali. Acest lucru va asigura corectitudinea recunoaşterii unei litere ideale.
Antrenarea va fi realizată folosind algoritmul de backpropagation care utilizează rata de învăţare adaptivă şi momentum cu funcţia **traingdx**.

Antrenarea fără zgomot

Reţeaua este iniţial antrenată fără zgomot pentru un număr maxim de 5000 de epoci sau până când eroarea reţelei (suma pătratelor erorilor) este mai mică decât 0.1. Se consideră ca funcţie de performanţă funcţia 'sse', iar parametrul **mc** va fi setat 0.95.

Antrenarea cu zgomot

Pentru a obţine o reţea care să recunoască şi literele bruiate, o vom antrena folosind două copii ale literelor din alphabet şi două copii ale alfabetului având litere bruiate. Vectorul valorilor target conţine patru copii ale vectorului target iniţial. Vectorii cu zgomot vor avea un zgomot de medie 0 şi deviaţii standard 0.1 şi 0.2. Aceasta va întări abilitatea neuronului de a învăţa să identifice corect litere bruiate şi în acelaşi timp să recunoască litere ideale.
Antrenarea folosind şi vectori afectaţi de zgomot se va face în 300 de epoci, iar valoarea erorii admise va creşte la 0.6, deoarece poate fi intuită apariţia unei erori mai mari datorate introducerii unui număr mai mare de vectori.
Se va repeta acest demers pentru 10 iteraţii.

Antrenarea fără zgomot din nou

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

Performanţa sistemului

Coeficientul de încredere al reţelei de recunoaştere a caracterelor se măsoară testând sistemul cu sute de vectori de intrare cu diferite valori ale zgomotului.
Vom determina procentul de eroare obţinut dacă vectorilor de intrare li se adaugă pe rând un nivel de zgomot repartizat normal de medie 0 şi deviaţie standard cuprinsă între 0 şi 0.5 cu pasul de 0.05. Pentru fiecare nivel de zgomot se introduc în reţea 100 de observaţii pentru fiecare literă, se calculează ieşirile asociate, care sunt atribuite funcţiei de transfer competitive care returnează litera cea mai probabilă. Se adună erorile de clasificare şi se calculează procentul obţinut.
Să se ploteze nivelul de zgomot versus procentul de eroare corespunzător calculate atât pentru reţeaua antrenată fără litere bruiate, cât şi pentru reţeaua antrenată considerând în mulţimea de antrenare şi litere bruiate

Pentru a obţine o precizie mai mare de clasificare se poate considera o reţea având mai mulţi neuroni în nivelul ascuns şi de asemenea rezoluţia vectorilor de intrare poate fi mărită la o reţea de 10 × 14, iar reţeaua poate fi antrenată folosind vectori cu un nivel mai ridicat de zgomot.

Testare reţelei

Pentru a testa reţeaua vom crea o literă cu zgomot pe care o vom introduce în reţea 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: Definiţi şi antrenaţi o reţea care să recunoască cifre. Studiaţi şi comparaţi curbele de eroare obţinute în cazul antrenării unei reţele fără zgomot, respectiv în cazul cu zgomot.