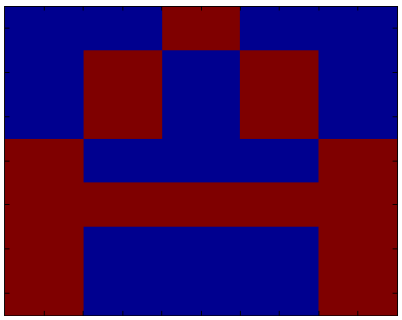
Laborator 11

Recunoașterea caracterelor folosind rețele neuronale

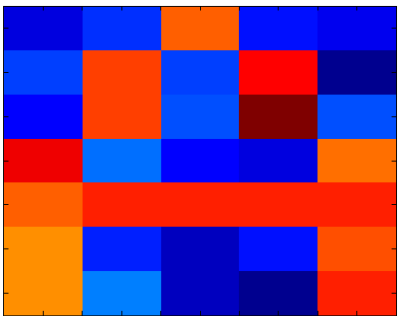
Recunoașterea caracterelor (OCR) este o problemă de clasificare cu aplicații în recunoașterea automată a numerelor de înmatriculare, extragerea automată de date din documente, căutarea după text a documentelor sub formă de imagini, etc. În cele ce urmează vom considerea problema recunoașterii caracterelor alfabetului englez, ce conține 26 de litere (A-Z), reprezentate sub forma unor imagini binare de dimensiuni 7 x 5 pixeli. Codul Matlab de mai jos generează și afișează cele 26 de litere:

```
[alphabet,targets] = prprob();
for i=1:26
    figure, imagesc(reshape(alphabet(:,i),5,7)');
    pause(1);
end
```

1. Analizați funcția *prprob* pentru a observa cum sunt generate literele (fiecare literă este stocată într-o coloană a matricei *alphabet*) și etichetele coespunzătoare (stocate în matricea *targets*).



Litera A



Litera A – exemplu perturbat

De obicei, imaginile care conțin litere sunt perturbate de zgomot (fundal complex, umbră, etc) astfel încât se depărtează de la formatul binar. În cele ce urmează vom genera exemple de litere perturbate considerând un zgomot Gaussian de medie 0 și deviație standard σ ce perturbă independent fiecare pixel al unei litere.

2. scrieți funcția ***generareazaLiterePerturbate.m*** care generează o mulțime de *n* exemple ale unei litere (primite ca input) perturbată cu zgomot

Gaussian de deviație standard σ. Valoarea fiecarui pixel trebuie sa fie intre 0 și 1 (valorile negative devin 0, valorile supraunitare devin 1).

3. scrieți funcția ***genereazaAlfabetPerturbat.m*** care generează o mulțime de *n* exemple pentru toate literele (A-Z) ale alfabetului perturbate cu zgomot Gaussian de deviație standard σ. Funcția ar trebui să întoarcă și eticheta corespunzătoare fiecărei litere generate.

Vom aborda problema de recunoștere a caracterelor prin antrenarea unei rețele neuronale care să clasifice corect literele. În acest sens vom genera o mulțime de date de antrenare, o mulțime de date de validare și o mulțime de date de test. Vom folosi mulțimea de validare pentru a selecta rețeaua cea mai promițătoare din punct de vedere al performanței.

4. generați mulțimile de date de antrenare, validare și testare care să conțină n = 50 de exemple din fiecare literă (***exempleAntrenare, exempleValidare, exempleTest***) perturbate cu zgomot de deviație standard σ = 0.3 și etichetele asociate (***eticheteAntrenare, eticheteValidare, eticheteTest***).

O posibilă arhitectură de rețea neuronală care poate fi folosită la recunoașterea caracterelor are 10 perceptroni pe stratul ascuns și 26 de perceptroni pe stratul de ieșire cu funcțiile de transfer ”logsig”. Codul Matlab care definește o astfel de rețea este următorul:

```
S1=10;
[R,Q] = size(alphabet);
[S2,Q] = size(targets);
P = alphabet;
net = newff(minmax(P),[S1 S2],{'logsig','logsig'},'traingdx');
```

5. antrenați rețeaua *net* de mai sus și simulați comportamentul ei pe mulțimea de validare (folosiți funcțiile *compet* și *sim*). Obțineți performanța rețelei antrenate pe mulțimea de validare comparând etichetele obținute cu ***eticheteValidare***.
6. încercați rețele cu alte tipuri de antrenare (spre exemplu inițializați ponderile cu valori foare mici aleatoare) sau alte arhitecturi, calculând performanța pe mulțimea de validare.
7. selectați rețeaua cea mai bună dintre cele încercate la 5 și 6 ca fiind rețeaua cu performanța cea mai bună pe mulțimea de validare. Ce performanță obțneți pe mulțimea de test? Cum arată matricea de confuzie?

8. Reluați etapele 4-7 pentru valori crecuste ale lui σ (0.4, 0.5). Cum se modifică performanța de clasificare?