

<p style="text-align: center;">ZACHODNIOPOMORSKI UNIWERSYTET TECHNOLOGICZNY W SZCZECINIE</p> <p style="text-align: center;">Wydział Elektryczny</p> <p style="text-align: center;">KATEDRA AUTOMATYKI PRZEMYSŁOWEJ I ROBOTYKI,</p> <p style="text-align: center;">LABORATORIUM Z METOD SZTUCZNEJ INTELIGENCJI</p>				
<p style="text-align: center;">SPRAWOZDANIE Z WYKONANIA ĆWICZENIA</p>				
<p>NR ĆWICZENIA</p> <p style="text-align: center;">3</p>	<p>TEMAT: SIECI TYPU FEED-FORWARD ORAZ SIECI HOPEFIELD'A W ZADANIACH ROZPOZNAWANIA TEKSTU.</p>			
<p>Kierunek studiów: AR</p>		<p>STUDIA: S1</p>	<p>ROK: 2016</p>	<p>Podpis:</p>
<p>Skład zespołu:</p> <p>1. <u>Damian Józwiak</u> Nr albumu: 33790</p>	<p>Prowadzący:</p> <p>DR INZ. KRZYSZTOF JAROSZWSKI</p>	<p>DATA ODDANIA:</p>	<p>OCENA:</p>	

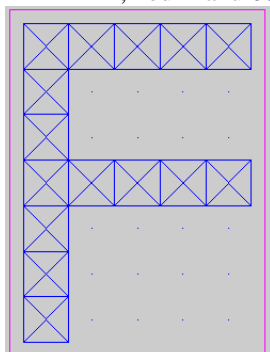
Opis problemu:

1. Zadaniem była prosta implementacja algorytmu OCR w sieciach neuralnych. Przy zadaniu założono iż znaki docierają do systemu już w formie przetworzonej jako wektory 35 elementowe, stworzone z macierzy 5x7. W tym zadaniu nie zajmowano się binaryzacją obrazów ani przycinaniem całości na pojedyncze znaki. System zakłada że znaki są reprezentowane jako: 0 – tło, 1 – treść. Dodatkowo jako wyjście sieci przyjęto 3-bitowy kod znaku, będącą imitacją kodów ASCII:

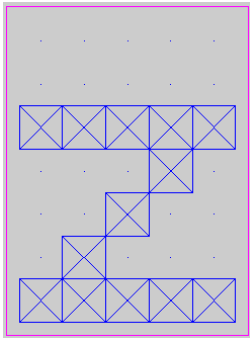
```
%corresponding to T members codes in 'made up' 3-bit coding(ASCII wanna be)
T_c=[0 0 0;0 0 1; 0 1 0]';
```

Trzy Znaki wybrane przez autora to odpowiednio:

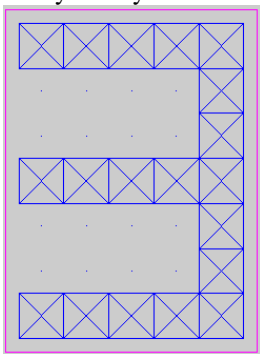
F – duże F, kod znaku 000;



z – małe z, kod znaku 001;



3 – cyfra trzy kod znaku 010;



*wyświetlone za pomocą funkcji: `plotchar([7x5 matrix]);`

Ich reprezentacja w matlabie:

```
T=[
    1 1 1 1 1 ...
    1 0 0 0 0 ...
    1 0 0 0 0 ...
    1 1 1 1 1 ...
    1 0 0 0 0 ...
    1 0 0 0 0 ...
    1 0 0 0 0; ...
    0 0 0 0 0 ...
    0 0 0 0 0 ...
    1 1 1 1 1 ...
    0 0 0 1 0 ...
    0 0 1 0 0 ...
    0 1 0 0 0 ...
    1 1 1 1 1; ...
    1 1 1 1 1 ...
    0 0 0 0 1 ...
    0 0 0 0 1 ...
    1 1 1 1 1 ...
    0 0 0 0 1 ...
    0 0 0 0 1 ...
    1 1 1 1 1; ...

    ]';
```

Sieć typu feed-forward:

1. Na wstępie autor pragnie zaznaczyć iż do nauki sieci typu feed-forward zostały stworzone dodatkowe wektory, zawierające 2 odwrócone piksele, po dwa dla każdego znaku:

```
input=[T(:,1)'; ...
      %%----- F_damaged1
      1 1 1 1 1 ...
      1 0 0 0 0 ...
      1 0 0 0 0 ...
      1 1 1 0 0 ...
      1 0 0 0 0 ...
      1 0 0 0 0 ...
      1 0 0 0 0; ...
      %%----- F_damaged2
      1 1 1 1 1 ...
      0 0 0 0 0 ...
      1 0 0 0 0 ...
      1 1 1 1 1 ...
      1 0 0 1 0 ...
      1 0 0 0 0 ...
      1 0 0 0 0;
      %%----- z
      %%----- z_damaged1
      T(:,2)'; ...
      %%----- z_damaged1
      0 0 0 0 0 ...
      0 0 0 0 0 ...
      1 1 1 1 1 ...
      0 0 0 1 0 ...
      0 0 0 0 0 ...
      0 0 0 0 0 ...
      0 1 0 0 0 ...
      1 1 1 1 1;
      %%----- z_damaged2
      0 0 0 0 0 ...
      0 0 0 0 0 ...
      1 1 1 1 1 ...
      0 0 0 1 0 ...
      0 0 1 0 0 ...
      0 1 0 0 0 ...
      1 1 0 0 1
      %%----- z
      %%----- 3_damaged1
      T(:,3)'; ...
      %%----- 3_damaged1
      1 1 1 1 1 ...
      0 0 0 0 1 ...
      0 0 0 0 1 ...
      1 1 1 0 1 ...
      0 0 0 0 1 ...
      0 0 0 0 0 ...
      0 1 1 1 1;
      %%----- 3_damaged2
      1 1 0 0 1 ...
      0 0 0 0 1 ...
      0 0 0 0 1 ...
      1 1 1 1 1 ...
      0 0 0 0 1 ...
      0 0 0 0 1 ...
      1 1 1 1 1];
```

Oraz odpowiadające elementom macierzy *input* kody:

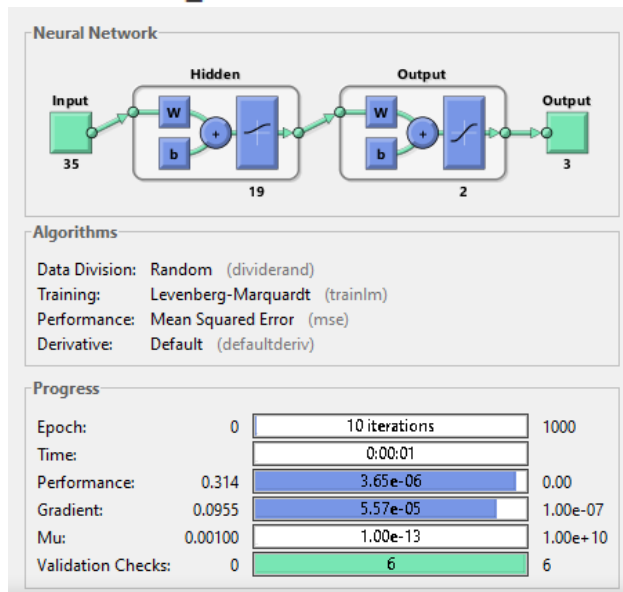
```
target=[0 0 0;0 0 0;0 0 0;0 0 1;0 0 1;0 0 1;0 1 0;0 1 0;0 1 0];
```

2. Uczenie sieci neuronowej:
Metodą prób i błędów dobrano ilość warstw ukrytych i funkcje aktywacji:

```
netff = feedforwardnet(19); %% mean value of input and output neurons
netff.layers{1}.transferFcn = 'logsig';
netff.layers{2}.transferFcn = 'tansig';
netff = train(netff,input,target);
view(netff)
```

3. Symulacja dla zestawu uczącego:

```
output_ff = netff(input); % same as simulate
disp(output_ff');
```



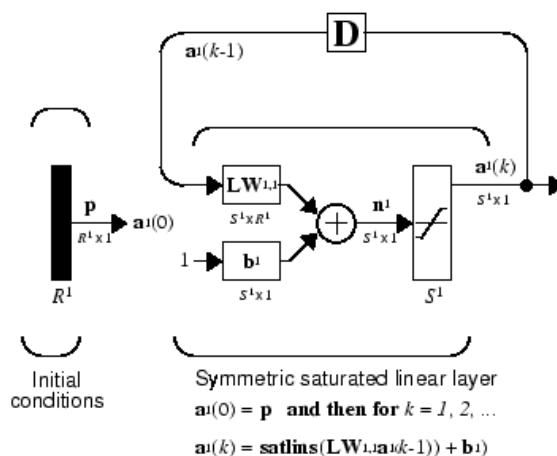
Wyniki:

0	1.0000	0.0002
0	1.0000	0.0007
0	1.0000	0.0001
0	0.0000	0.9999
0	0.0000	1.0000
0	0.0000	0.9998
0	1.0000	0.0000
0	1.0000	0.0000
0	0.9995	0.0015

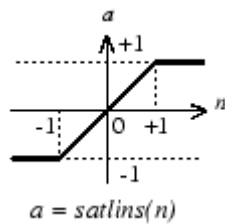
Wyniki dla tego typu sieci są więcej niż zadowalające, dając mse na poziomie $3.65 \cdot 10^{-6}$

Sieć hopfield'a:

1. Do trangu tego typu sieci wystarczy sam zestaw poprawnych wzorców zamieszczonych w macierzy T , oznaczonych jako wartości początkowe (Initial conditions). Wyjście tej sieci jest podawane na wejście dopóki wyjście się nie ustabilizuje i nie zrówna się z wektorem wejściowym.



Jako funkcję aktywacji sieć ta wykorzystuje funkcję *satlins*, która w naszym przypadku zwraca 0 dla 0 oraz 1 dla 1:



2. Implementacja:

```
neth= newhop(T);
for j=1:9
test=input(:,j);
[Y,a,b]=sim(neth,{1 1},{},test);
closest=(sum(abs(Y{1}-T(:,1)))/35)*100;% just setting initial closest to 0 vec
weGot=T_c(:,1);% and corresponding 7x5 model matrix
for i=1:3
foo=(sum(abs(Y{1}-T(:,i)))/35)*100;
if foo<closest
weGot=T_c(:,i)';
closest=foo;
end
end
fprintf('Błąd = %.2f%%\nKod:',closest);
disp(weGot);
end
```

Implementacja jest dość prosta:

- 1) Definiujemy nową sieć hopfield'a *newhop()* z wektorem T jako wartości początkowe,
 - 2) Dla każdego elementu macierzy input zawierającej znaki testujące z poprzedniej części dotyczącej sieci feed-forward(1 cała + 2 uszkodzone macierze na każdy znak) przeprowadzamy symulację i znajdujemy kod odtworzonego znaku(na podstawie najmniejszego błędu)
3. Przykładowe wartości błędów na znaków:

a) F:

```
Błąd = 1.33%
Kod:    0    0    0

Błąd = 1.64%
Kod:    0    0    0

Błąd = 2.75%
Kod:    0    0    0
```

b) z:

```
Błąd = 1.33%
Kod:    0    0    1

Błąd = 1.50%
Kod:    0    0    1

Błąd = 4.22%
Kod:    0    0    1
```

c) 3:

Błąd = 1.27%
Kod: 0 1 0

Błąd = 1.69%
Kod: 0 1 0

Błąd = 2.85%
Kod: 0 1 0

4. Wnioski:

- Bez testów z większym zasumieniem trudno definitywnie znaleźć sieć z mniejszym błędem detekcji znaku, z drugiej strony nie ma możliwości porównania wyjść tych dwóch typów sieci ze względu na różne wyjścia – sieć hopfield'a zwraca wektor 35 elementowy (obraz rozpoznanego znaku), który pośrednio możemy przekształcić na 3 bitowy kod znaku, sieć typu feedforward zwraca bezpośrednio kod znaku ze względu na wybrany zestaw docelowy.
- Ostatecznie obie sieci uzyskały zadowalające wyniki dla zestawu testującego.