NEURAL NETWORKS
(ELEC 5240 and ELEC 6240)
Clustering and PCNN

Bodgan M. Wilamowski

Clustering problem
Find number of clusters and its location in 4-dim. space

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Distance</th>
<th>Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8110</td>
<td>0.5850</td>
</tr>
<tr>
<td>2</td>
<td>0.8</td>
<td>0.5850</td>
</tr>
<tr>
<td>3</td>
<td>0.8</td>
<td>0.5850</td>
</tr>
<tr>
<td>4</td>
<td>0.8</td>
<td>0.5850</td>
</tr>
<tr>
<td>5</td>
<td>0.8</td>
<td>0.5850</td>
</tr>
<tr>
<td>6</td>
<td>0.8</td>
<td>0.5850</td>
</tr>
<tr>
<td>7</td>
<td>0.8</td>
<td>0.5850</td>
</tr>
<tr>
<td>8</td>
<td>0.8</td>
<td>0.5850</td>
</tr>
<tr>
<td>9</td>
<td>0.8</td>
<td>0.5850</td>
</tr>
<tr>
<td>10</td>
<td>0.8</td>
<td>0.5850</td>
</tr>
</tbody>
</table>

Forming clusters as needed using minimum distance concept
1. First pattern is applied and the cluster is formed
2. Next pattern is applied and then:
   a) If distance form all existing clusters is larger then threshold
      then a new cluster is formed
   b) Else weights of the closest cluster are updated

\[ W_i = \frac{mW_i + \alpha x}{m + 1} \]

where \( m \) is the number of previous patterns of a given set which were used to update this particular neuron and \( \alpha \) is the learning constant

Solution

Kohonen Network
The unsupervised training process example
patterns

Kohonen Network
The unsupervised training process example
normalized patterns

Kohonen Network
The unsupervised training process example
weights

Kohonen Network
The unsupervised training process example
weights
Kohonen Network
The unsupervised training process example

% Kohonen network
% call normall - vector normalization
format compact; format short g; clear all;
x(1)=12.950 1.3876; 4.1168 2.9694; 4.3185 2.3792;
6.2139 2.4288; 5.9630 0.7258; 1.9562 5.8288;
1.8184 6.0146; 2.6108 5.4870; 1.7599 4.1317;
1.0465 4.1698;[0.9459 0.3512; 0.6690 0.7433; 0.3714 0.9285];
figure(1); clf; box on; plot(x(:,1),x(:,2),'+r');
aa=axis; aa(1)=0; aa(2)=10; aa(3)=0; aa(4)=10; axis(aa);
tnormal(x), w=normal(w),
figure(2); clf; box on; hold on;
plot(x(:,1),x(:,2),'xr'); plot(w,1),w(2),w(3),'or';
aa=axis; aa(1)=0; aa(2)=11; aa(3)=0; aa(4)=11; axis(aa);
alpha=0.3; [m,n]=size(x);
for i=1:20,
    for i=m,
        netx=w'*[v]=max(netf);
        w(i)=w(i)+alpha*netx(i); w=normal(w);
    end;
    plot(w,1),w(2),w(3);
end;
plot(w,1),w(2),'+r'; w

Find a solution of a set of linear equations

\[
\begin{bmatrix}
1 & 2 & -2 \\
3 & -3 & -1 \\
-2 & 1 & 2
\end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -5 \\ 0 \\ -1 \end{bmatrix}
\]

PSpice code
*** SPICE code for solution of linear equations***
10 E1 0 value=1000*(3)^0
E2 2 0 value=1000*(3)*v(1)=3v(2)=3v(3)+9)
E3 3 0 value=1000*(-2)*v(1)=2v(2)=3v(3)+9)

Results for different neuron gains k
k=1
( 1) -3.000 ( 2) 1.9200 ( 3) 1.5800
k=10
( 1) -9.336 ( 2) 1.9803 ( 3) 2.9398
k=100
( 1) -99.33 ( 2) 1.9989 ( 3) 2.9938
k=1000
( 1) -999.3 ( 2) 1.9998 ( 3) 3.0000
k=10000
( 1) -9999.3 ( 2) 1.9999 ( 3) 3.0000

PSPice code

Dynamic feedback networks
Find minimum of a function
\[
E = 3x_1^2 + 4x_2^2 + 3x_3^2 + 2x_1x_2 - x_2x_3 - 2x_1x_3 + 2x_1 - 3x_2 + 4x_3
\]
\[
\frac{\partial E}{\partial x_1} = 6x_1 + 2x_2 - 2x_3 + 2 = 0
\]
\[
\frac{\partial E}{\partial x_2} = 2x_1 + 8x_2 - x_3 - 3 = 0
\]
\[
\frac{\partial E}{\partial x_3} = -2x_1 - x_2 + 4x_3 + 4 = 0
\]
\[
x_1 = -\frac{1}{3}x_1 + \frac{1}{3}x_3 - \frac{1}{3}
\]
\[
x_2 = -\frac{1}{4}x_1 + \frac{1}{8}x_2 + \frac{3}{8}
\]
\[
x_3 = 2x_1 - 4x_3 - 1
\]

Find a solution of a set of linear equations

\[
\begin{bmatrix}
1 & 2 & -2 \\
3 & -3 & -1 \\
-2 & 1 & 2
\end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -5 \\ 0 \\ -1 \end{bmatrix}
\]

Find a solution of a set of linear equations

\[
\begin{bmatrix}
1 & 2 & -2 \\
3 & -3 & -1 \\
-2 & 1 & 2
\end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 6 \\ 9 \end{bmatrix}
\]

Find a solution of a set of linear equations

\[
\begin{bmatrix}
1 & 2 & -2 \\
3 & -3 & -1 \\
-2 & 1 & 2
\end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 6 \\ 9 \end{bmatrix}
\]
Find a solution of a set of linear equations

\[
\begin{align*}
E_{1} & = 3.0 \times 10^{5} \\
E_{2} & = 2.0 \times 10^{5} \\
E_{3} & = 1.0 \times 10^{5}
\end{align*}
\]

*** SPICE code for solution of linear equations ***

```
subckt opamp 1 2 3
  V1 1 2 + -
  V2 2 3 + -
  V3 3 1 + -
  R11 1 2 1
  R12 2 1 0.5
  R13 1 3 -0.5
  R21 2 1 0.33333333
  R22 1 2 1
  R23 2 3 -1
  R31 2 1 13 -0.5
  R32 2 2 13 -1
  R33 2 3 13 -1
end
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image1)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image2)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image3)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image4)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image5)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image6)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image7)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image8)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image9)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image10)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image11)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image12)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image13)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image14)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image15)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image16)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image17)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image18)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image19)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image20)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image21)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image22)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image23)

```
```

Pulse Coded Neural Networks

![Pulse Coded Neural Networks](image24)