Neural network history:
- McCulloch-Pitts neurons – 1943
- Widrow’s ADALINE and MADALINE – 1960/62
- Nilsson’s “Learning Machines” (perceptrons) – 1965
- Kohonen’s Self Organized Feature Maps - 1982
- Hopfield’s recurrent networks – 1982/84
- Rumelhart’s Error Backpropagation – 1986

Artificial neuron
\[
\text{net} = \sum_{i=1}^{n} w_i x_i
\]

McCulloch-Pitts neurons
\[
\text{net} = \sum_{i=1}^{n} w_i x_i + w_{+1}
\]
\[
o = \begin{cases} 
1 & \text{if net} \geq 0 \\
0 & \text{if net} < 0
\end{cases}
\]

McCulloch-Pitts neurons with memory
\[
\text{net} = \sum_{i=1}^{n} w_i x_i + w_{+1}
\]
\[
o = \begin{cases} 
1 & \text{if net} \geq 0 \\
0 & \text{if net} < 0
\end{cases}
\]

Examples of logical operations using McCulloch-Pitts neurons:
- AND
- OR
- NOT

McCulloch-Pitts neurons
\[
\text{net} = \sum_{i=1}^{n} w_i x_i
\]
\[
o = \begin{cases} 
1 & \text{if net} \geq 0 \\
0 & \text{if net} < 0
\end{cases}
\]

How to design larger neural networks with McCulloch-Pitts neurons?
- Boltzmann Machines
- Stimulated Annealing

Widrow’s ADALINE and MADALINE
Lineralization and LMS training
- ADALINE
- MADALINE

Minster and Papert book?
HW 1 Design McCulloch-Pitts neuron, which implement the following truth table:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[-x - 3y - z + 1.5 \geq 0\]

Selecting rectangular area with 4 partitions

neuron equations:

\[x - 1 > 0\]
\[-x + 2 > 0\]
\[-y + 2.5 > 0\]
\[-y - 0.5 > 0\]

Results for the first layer \(k=10\)

MATLAB code for processing

```matlab
%% net=51; k=10;
W=[1 0 -1; 1 0 2; 0 -1.25; 0 1 -0.5];
for i=1:n
    x(i)=si*(x(i)-y(i));
end
for j=1:n
    y(j)=-si*(y(j)-x(i));
end
for m=1:4
    net(m,n)=w1(m,1)*y(i)+w1(m,2)*x(i)+w1(m,3); % net(m,n)=1/((1+exp(-k*(net(i)-net(n))));
end
out(n)=1/(1+exp(-k*(net(i)-net(n))));
end;
end;
```

```matlab
figure(1); clf; mesh(x,y,net); view(20,50);
```
Design neural network with unipolar McCulloch-Pitts neurons, which has two input and three outputs. Each output respond to the patterns located in three areas as shown on figure below. Draw neural network and indicate value of each weight.

Another area with 4 partitions
neuron equations:

\[
\frac{x}{2} + \frac{y}{2} > 1 \\
\frac{x}{3} + \frac{y}{1} > 1 \\
\frac{x}{2} + \frac{y}{\infty} > 1 \\
\frac{x}{-2} + \frac{y}{1} < 1
\]

Weights in the first layer:
1 1 -2;   1 3 -3;   1 0 -2;   1 -2 2;
Weights in the second layer:
1 1 1 1 -2;