CS 4824/ECE 4424: Perceptron I

Acknowledgement:
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.
Human Intelligence

- Brian is responsible for human intelligence by performing
  - Learning
  - Memorization
  - Cognition and recognition
  - Decision making

- Brian consists of nerve cells called neurons
  - Neurons can propagate nervous signal
  - Neurons form giant network of signal propagation
Neuron

- Dendrite
- Synapse
- Axon
- Axon from another cell
- Axonal arborization
- Nucleus
- Cell body or Soma
- Synapses
Comparison

- **Brain**
  - Network of neurons
  - Nerve signals propagate via neural network
  - Parallel computation
  - Robust (neurons die everyday without any impact)

- **Computer**
  - Bunch of gates
  - Electrical signals directed by gates
  - Sequential and parallel computation
  - Fragile (if a gate stops working, computer crashes)
Artificial Neural Networks

- **Key idea:** emulate biological neurons for computation

- **Artificial neural network (ANN)**
  - Units are called "nodes" and correspond to neurons
  - Connections between nodes correspond to synapses

- **Correspondence between ANN and biological neural network**
  - Numerical signal transmitted between nodes corresponds to chemical signals between neurons
  - Nodes modifying numerical signal correspond to neurons firing rate
ANN

• **Node:** $i$

• **Weights:** $W$
  • Strength of the connection from node $i$ to node $j$
  • Input signals $x_i$ weighted by $W_{ji}$ and linearly combined:
    • $a_i = \sum_i W_{ji} x_i + w_0 = W_{ji} x$

• **Activation function:** $h$
  • Numerical signal produced: $y_i = h(a_j)$
ANN: Node

- Schematic

\[ a_j = \sum_{i} w_{ji} x_i + w_{jo} \]

activation function \( h(a_j) \)

output links
ANN: Activation Function

- Generally non-linear
  - Else, the network is just a linear function

- Mimics the firing in neurons
  - Nodes should be “active” (output close to 1) when fed with the “right” inputs
  - Nodes should be “inactive” (output close to 0) when fed with the “wrong” inputs
Common Activation Functions

Identity

\[ h(a) = a \]

Threshold

\[ h(a) = \begin{cases} 
1 & \text{if } a \geq 0 \\
-1 & \text{if } a < 0 
\end{cases} \]

Sigmoid

\[ h(a) = \sigma(a) = \frac{1}{1 + e^{-a}} \]
Representing Boolean Functions

- Design ANN for logic gates

- What should be the weights of the following units to represent AND, OR, NOT gates?

**AND**

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_1 \land x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$w_0 = -1$

$w_1 = 0.75$

$w_2 = 0.75x_1 x_2$

**OR**

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_1 \lor x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$w_0 = -0.5$

$w_1 = 1$

$w_2 = 1$

**NOT**

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$\neg x_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$w_0 = 0.5$

$w_1 = -1$

$w_2 = 7x_1$
Representing Boolean Functions

- ANN can be used to design various logic gates
- So ANN can be used to approximate any boolean functions