CS 416
Artificial Intelligence

Lecture 18
Neural Nets
Chapter 20

Model of Neurons

• Multiple inputs/dendrites (~10,000!!)
• Cell body/soma performs computation
• Single output/axon
• Computation is typically modeled as linear

Early History of Neural Nets

• Eons ago: Neurons are invented
• 1868: J. C. Maxwell studies feedback mechanisms
• 1942: Weiner et al. formulate cybernetics
• 1943: McCulloch-Pitts Neurons
• 1949: Hebb indicates biological mechanism
• 1962: Rosenblatt’s Perceptron
• 1969: Minsky and Papert decompose perceptrons

Cybernetics

• “The theoretical study of communication and control processes in biological, mechanical, and electronic systems, especially the comparison of these processes in biological and artificial systems.” (http://www.dictionary.com)

McCulloch-Pitts Neurons

• One or two inputs to neuron
• Inputs are multiplied by weights
• If product exceeds a threshold, the neuron fires
• How would we create xor?

Hebbian Modification

• “When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased” from Hebb’s 1949 The Organization of Behavior, p. 62
Perceptrons

- Each input is binary and has associated with it a weight
- "Not gates" are allowed
- The sum of the inner product of the input and weights is calculated
- If this sum exceeds a threshold, the perceptron fires

Error Correction

\[ \Delta w_i = \alpha (c - \Theta(x \cdot w)) \]

- Only updates weights for non-zero inputs
- For positive inputs
  - If the perceptron should have fired but did not, the weight is increased
  - If the perceptron fired but should not have, the weight is decreased
- For negative inputs
  - Behavior is opposite

Perceptron Example

Example modified from "The Essence of Artificial Intelligence" by Alison Cawsey
- Initialize all weights to 0.2
- Let epsilon = 0.05 and threshold = 0.5

Perceptron Example

First output is 1 since
0.2+0.2+0.2>0.5
- Should be 0, so weights with active connections are decremented by 0.05

Perceptron Example

Next output is 0 since
0.15+0.15+0.2<=0.5
- Should be 1, so weights with active connections are incremented by 0.05
- New weights work for Alison, Jeff, and Gail

Perceptron Example

Output for Simon is 1 (0.2+0.25+0.15>0.5)
- Should be 0, so weights with active connections are decremented by 0.05
- Are we finished?
Perceptron Example

- After processing all the examples again we get weights that work for all examples
- What do these weights mean?
- In general, how often should we reprocess?

2nd Class Exercise

- \( x_3 = \neg x_1, \ x_4 = \neg x_2 \)
- Find \( w_1, w_2, w_3, w_4 \), and \( \theta \) such that
  \( \Theta(x_1 w_1 + x_2 w_2) = x_1 \ XOR \ x_2 \)
- Or, prove that it can't be done

3rd Class Exercise

- Find \( w_1, w_2, w_3, w_4, w_5, \theta_1, \) and \( \theta_2 \) such that
  \( \text{output is } x_1 \ XOR \ x_2 \)
- Or, prove that it can't be done

Multi-layered Perceptrons

- Input layer, output layer, and "hidden" layers
- Eliminates some concerns of Minsky and Papert
- Modification rules are more complicated!

4th Class Exercise

- Find \( w_1, w_2, w_3, w_4, w_5, \theta_1, \) and \( \theta_2 \) such that
  \( \text{output is } x_1 \ XOR \ x_2 \)
- Or, prove that it can't be done
Recent History of Neural Nets

- 1969 Minsky & Papert “kill” neural nets
- 1974 Werbos describes back-propagation
- 1982 Hopfield reinvigorates neural nets
- 1986 Parallel Distributed Processing
  (Here’s some source code: http://www.geocities.com/CapeCanaveral/1624/)

"The report of my death is greatly exaggerated." – Mark Twain

Limitations of Perceptrons

- Minsky & Papert published “Perceptrons” stressing the limitations of perceptrons
- Single-layer perceptrons cannot solve problems that are linearly inseparable (e.g., xor)
- Most interesting problems are linearly inseparable
- Kills funding for neural nets for 12-15 years

Back-Propagation

- The concept of local error is required
- We’ll examine our simple 3-layer perceptron with xor

Back-Propagation (xor)

- Initial weights: $w_1=0.90, w_2=-0.54, w_3=0.21, w_4=-0.03, w_5 = 0.78$
- Threshold is now sigmoidal (function should have derivatives)
- $f(x\cdot w) = \frac{1}{1 + e^{-x}}$

Back-Propagation (xor)

- Error at last layer (hidden $\rightarrow$ output) is defined as: $\delta_i = (F(\hat{y}, x) - c)\cdot \hat{y}_i$
- Error at previous layer (input $\rightarrow$ hidden) is defined as: $\delta_j = \sum_{i \in \text{out}} \frac{\partial F}{\partial \hat{y}_j} \cdot \delta_i$
- Change in weight: $\Delta w_j = -\beta \sum_{i \in \text{out}} \frac{\partial F}{\partial \hat{y}_j} \cdot \delta_i$
- Where: $\frac{\partial F}{\partial \hat{y}_j} = \alpha \hat{y}_i(1-\hat{y}_i)$

"I hate math... so little room to make small errors." – Caleb Schaefer, UGA student

Cypher: It means, buckle your seatbelt, Dorothy, because Kansas is going bye-bye.
Back-Propagation (xor)

- (0,0) Æ 0 – 1st example
  - Input to hidden unit is 0, sigmoid(0)=0.5
  - Input to output unit is (0.5)(-0.03)=-0.015
  - Sigmoid(-0.015)=0.4963 Æ error=-0.4963
  - So, $\delta_w = -0.0062$

Why are we ignoring the other weight changes?

Back-Propagation (xor)

- (0,1) Æ 1 – 2nd example
  - $i_h = -0.54$ Æ $o_h = 0.3862$
  - $i_o = (0.3862)(-0.03)+0.78 = 0.769$ Æ $o_o = 0.6683$

  $\frac{\partial E}{\partial w_{h}} = (0.5)(0.4963)(1-0.4963)(-0.4963) = -0.0620$
  - Example’s contribution to $\Delta w_{h}$ is –0.0062

Back-Propagation (xor)

- Initial performance = -0.2696
- After 100 iterations we have:
  - $w = (0.913, -0.521, 0.038, -0.232, 0.288)$
  - Performance = -0.2515
- After 100K iterations we have:
  - $w = (15.75, -7.671, 7.146, -7.149, 0.0022)$
  - Performance = -0.1880
- After 1M iterations we have:
  - $w = (21.38, -10.49, 9.798, -9.798, 0.0002)$
  - Performance = -0.1875

Hopfield Nets

- Created neural nets that have content-addressable memory
- Can reconstruct a learned signal from a fraction of it as an input
- Provided a biological interpretation

What is the Purpose of NN?

- To create an Artificial Intelligence, or
  - Although not an invalid purpose, many people in the AI community think neural networks do not provide anything that cannot be obtained through other techniques
- To study how the human brain works?
  - Ironically, those studying neural networks with this in mind are more likely to contribute to the previous purpose

Quick List of Terms

- Presynaptic Modification: Synapse weights are only modified when incoming (afferent) neuron fires
- Postsynaptic Modification: Synapse weights are only modified when outgoing (efferent) neuron fires
- Error Correction: Synapse weights are modified relative to an error – can be pre- or postsynaptic; requires some form of feedback
- Self-supervised: Synapse weights are modified relative to internal excitation of neuron – can be pre- or postsynaptic
Self-supervised Neurons

- One example is a neuron that has the following synaptic modification rule:

\[ \Delta w_{ij} = \alpha y_j (x_i - w_{ij}) \]

\[ y_j = x \cdot w = x^T w \]  \( \leftrightarrow \) Internal excitation

\[ 0 = E[\Delta x, y_j] - E[\Delta y_j, w_{ij}] \]  \( \leftrightarrow \) Convergence of weights

\[ E[x, x^T] w_{ij} = E[y_j, w_{ij}] \]

\[ E[x, x^T] y_j = E[y_j, w_{ij}] \]  \( \leftrightarrow \) Eigenvalue equation!

More Self-Supervision

- Previous rule could not learn to distinguish between different classes of data
- However, if the rule is modified to:

\[ \Delta w_{ij} = \alpha \Theta (y_j, x_i - w_{ij}) \]

- The neuron will learn to only respond to a certain class of inputs
- Different neurons respond to different classes

Some Brain Facts

- Contains \( \sim 100,000,000,000 \) neurons
- Hippocampus CA3 region contains \( \sim 3,000,000 \) neurons
- Each neuron is connected to \( \sim 10,000 \) other neurons
- \( \sim 1,000,000,000,000,000 \) (10\(^{15}\)) connections!
  - Contrary to a BrainPlace.com, this is considerably less than number of stars in the universe – 10\(^{20}\) to 10\(^{22}\)
- Consumes \( \sim 20-30\% \) of the body’s energy
- Contains about 2\% of the body’s mass