Artificial neural networks, perceptrons

Relevant Readings: Sections 4.1, 4.2, 4.3, 4.4 in Mitchell

CS495 - Machine Learning, Fall 2009

Biological motivation

- Humans and other animals learn suprisingly complex functions in a rather parallel way

Biological motivation

- Humans and other animals learn suprisingly complex functions in a rather parallel way
- A human brain, for instance, has billions of neurons that somehow cooperate to give way to complex behavior

Biological motivation

- Humans and other animals learn suprisingly complex functions in a rather parallel way
- A human brain, for instance, has billions of neurons that somehow cooperate to give way to complex behavior
- Artificial Neural Networks (ANN) are motivated by this biology, but we won't aim to *closely* simulate biology's approach (although some researchers do)

Biological motivation

- Humans and other animals learn suprisingly complex functions in a rather parallel way
- A human brain, for instance, has billions of neurons that somehow cooperate to give way to complex behavior
- Artificial Neural Networks (ANN) are motivated by this biology, but we won't aim to *closely* simulate biology's approach (although some researchers do)

Applications

- Game-playing

Applications

- Game-playing
- Pattern recognition

Applications

- Game-playing
- Pattern recognition
- Robotics (just ask Jeff)

Applications

- Game-playing
- Pattern recognition
- Robotics (just ask Jeff)
- And more...

Applications

- Game-playing
- Pattern recognition
- Robotics (just ask Jeff)
- And more...

Strengths/weaknesses

- Weaknesses

Strengths/weaknesses

- Weaknesses
  - May take a long time to train

Strengths/weaknesses

- Weaknesses
  - May take a long time to train
  - The hypothesis doesn't come in a "human readable" format

Strengths/weaknesses

- Weaknesses
  - May take a long time to train
  - The hypothesis doesn't come in a "human readable" format
- Strengths

Strengths/weaknesses

- Weaknesses
  - May take a long time to train
  - The hypothesis doesn't come in a "human readable" format
- Strengths
  - Works well with noisy data

Strengths/weaknesses

- Weaknesses
  - May take a long time to train
  - The hypothesis doesn't come in a "human readable" format
- Strengths
  - Works well with noisy data
  - Classifying new instances is fast

Strengths/weaknesses

- Weaknesses
  - May take a long time to train
  - The hypothesis doesn't come in a "human readable" format
- Strengths
  - Works well with noisy data
  - Classifying new instances is fast
  - Can work with discrete and continuous inputs and outputs

Strengths/weaknesses

- Weaknesses
  - May take a long time to train
  - The hypothesis doesn't come in a "human readable" format
- Strengths
  - Works well with noisy data
  - Classifying new instances is fast
  - Can work with discrete and continuous inputs and outputs

The basic unit

- ANNs are made up of a large network of many simple units

The basic unit

- ANNs are made up of a large network of many simple units
- There are several possibilities for these simple units, such as:

The basic unit

- ANNs are made up of a large network of many simple units
- There are several possibilities for these simple units, such as:
  - Perceptrons

The basic unit

- ANNs are made up of a large network of many simple units
- There are several possibilities for these simple units, such as:
  - Perceptrons
  - Linear units

The basic unit

- ANNs are made up of a large network of many simple units
- There are several possibilities for these simple units, such as:
  - Perceptrons
  - Linear units
  - Sigmoid units

The basic unit

- ANNs are made up of a large network of many simple units
- There are several possibilities for these simple units, such as:
  - Perceptrons
  - Linear units
  - Sigmoid units
- There are also different possible approaches to training these units

The basic unit

- ANNs are made up of a large network of many simple units
- There are several possibilities for these simple units, such as:
  - Perceptrons
  - Linear units
  - Sigmoid units
- There are also different possible approaches to training these units

Perceptrons

- A perceptron uses a linear combination of inputs to produce a 1 or -1 output

Perceptrons

- A perceptron uses a linear combination of inputs to produce a 1 or -1 output
- Suppose we want a perceptron with n inputs: $x_1, x_2, \ldots, x_n$

Perceptrons

- A perceptron uses a linear combination of inputs to produce a 1 or -1 output
- Suppose we want a perceptron with n inputs: $x_1, x_2, \ldots, x_n$
- Each input x_i has a weight w_i which indicates its relative importance in determining the output

Perceptrons

- A perceptron uses a linear combination of inputs to produce a 1 or -1 output
- Suppose we want a perceptron with n inputs: $x_1, x_2, \ldots, x_n$
- Each input x_i has a weight w_i which indicates its relative importance in determining the output
  - The output is 1 if $w_0 + \sum_{i=1}^{n} w_i x_i > 0$

Perceptrons

- A perceptron uses a linear combination of inputs to produce a 1 or -1 output
- Suppose we want a perceptron with n inputs: $x_1, x_2, \ldots, x_n$
- Each input x_i has a weight w_i which indicates its relative importance in determining the output
  - The output is 1 if $w_0 + \sum_{i=1}^{n} w_i x_i > 0$
    - i.e. $w_0 + w_1 x_1 + w_2 x_2 + \ldots w_n x_n > 0$

Perceptrons

- A perceptron uses a linear combination of inputs to produce a 1 or -1 output
- Suppose we want a perceptron with n inputs: $x_1, x_2, \ldots, x_n$
- Each input x_i has a weight w_i which indicates its relative importance in determining the output
  - The output is 1 if $w_0 + \sum_{i=1}^{n} w_i x_i > 0$
    - i.e. $w_0 + w_1 x_1 + w_2 x_2 + \ldots w_n x_n > 0$
  - The output is -1 otherwise

Perceptrons

- A perceptron uses a linear combination of inputs to produce a 1 or -1 output
- Suppose we want a perceptron with n inputs: $x_1, x_2, \ldots, x_n$
- Each input x_i has a weight w_i which indicates its relative importance in determining the output
  - The output is 1 if $w_0 + \sum_{i=1}^{n} w_i x_i > 0$
    - i.e. $w_0 + w_1 x_1 + w_2 x_2 + \ldots w_n x_n > 0$
  - The output is -1 otherwise
  - The w_0 weight can be thought of as controlling how "high" the perceptron has to be to count for a 1 vs. -1

Perceptrons

- A perceptron uses a linear combination of inputs to produce a 1 or -1 output
- Suppose we want a perceptron with n inputs: $x_1, x_2, \ldots, x_n$
- Each input x_i has a weight w_i which indicates its relative importance in determining the output
  - The output is 1 if $w_0 + \sum_{i=1}^{n} w_i x_i > 0$
    - i.e. $w_0 + w_1 x_1 + w_2 x_2 + \ldots w_n x_n > 0$
  - The output is -1 otherwise
  - The w_0 weight can be thought of as controlling how "high" the perceptron has to be to count for a 1 vs. -1
  - We can rewrite the test as $\sum_{i=0}^{n} w_i x_i > 0$ if we treat $x_0 = 1$ as an input that's always "on"

Perceptrons

- A perceptron uses a linear combination of inputs to produce a 1 or -1 output
- Suppose we want a perceptron with n inputs: $x_1, x_2, \ldots, x_n$
- Each input x_i has a weight w_i which indicates its relative importance in determining the output
  - The output is 1 if $w_0 + \sum_{i=1}^{n} w_i x_i > 0$
    - i.e. $w_0 + w_1 x_1 + w_2 x_2 + \ldots w_n x_n > 0$
  - The output is -1 otherwise
  - The w_0 weight can be thought of as controlling how "high" the perceptron has to be to count for a 1 vs. -1
  - We can rewrite the test as $\sum_{i=0}^{n} w_i x_i > 0$ if we treat $x_0 = 1$ as an input that's always "on"

Hypothesis space

- Question: What is the hypothesis space of a perceptron?

Hypothesis space

- Question: What is the hypothesis space of a perceptron?
- Answer: A hyperplane that separates positive and negative examples

Hypothesis space

- Question: What is the hypothesis space of a perceptron?
- Answer: A hyperplane that separates positive and negative examples
- Exercise: Can we represent a 2-input AND function with a perceptron?

Hypothesis space

- Question: What is the hypothesis space of a perceptron?
- Answer: A hyperplane that separates positive and negative examples
- Exercise: Can we represent a 2-input AND function with a perceptron?
- Exercise: Can we represent a 2-input OR function with a perceptron?

Hypothesis space

- Question: What is the hypothesis space of a perceptron?
- Answer: A hyperplane that separates positive and negative examples
- Exercise: Can we represent a 2-input AND function with a perceptron?
- Exercise: Can we represent a 2-input OR function with a perceptron?
- Exercise: Can we represent a 2-input XOR function with a perceptron?

Hypothesis space

- Question: What is the hypothesis space of a perceptron?
- Answer: A hyperplane that separates positive and negative examples
- Exercise: Can we represent a 2-input AND function with a perceptron?
- Exercise: Can we represent a 2-input OR function with a perceptron?
- Exercise: Can we represent a 2-input XOR function with a perceptron?
- What other logical functions can we represent?

Hypothesis space

- Question: What is the hypothesis space of a perceptron?
- Answer: A hyperplane that separates positive and negative examples
- Exercise: Can we represent a 2-input AND function with a perceptron?
- Exercise: Can we represent a 2-input OR function with a perceptron?
- Exercise: Can we represent a 2-input XOR function with a perceptron?
- What other logical functions can we represent?
- It turns out that with a 2-layer perceptron-based ANN, we can represent any boolean function

Hypothesis space

- Question: What is the hypothesis space of a perceptron?
- Answer: A hyperplane that separates positive and negative examples
- Exercise: Can we represent a 2-input AND function with a perceptron?
- Exercise: Can we represent a 2-input OR function with a perceptron?
- Exercise: Can we represent a 2-input XOR function with a perceptron?
- What other logical functions can we represent?
- It turns out that with a 2-layer perceptron-based ANN, we can represent any boolean function

Implementation

- How about we program a perceptron?