COGS2010
Laboratory Introduction to Models in Cognitive Science

Lecturer: Janet Wiles
j.wiles@itee.uq.edu.au
Office hours: Tues 2-4, Rm 78-333

Tutor: Ben Skellett
ben@itee.uq.edu.au
Office hours: Thur 2-4, Rm 78-430

All details are linked to the home page

http://www.itee.uq.edu.au/~cogs2010/

- Course profile
- Online Schedule – updated during semester
- Brainwave simulator
- Announcements – updated during semester
- Newsgroup uq.itee.cogs2010
- Please turn off your mobile phone

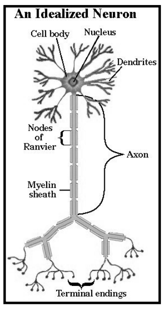
Part 1: Introduction to Neural Networks

Adapted from Lecture notes by Scott Bolland
updated by Janet Wiles

Lecture 1 Overview

- About the Brain and Neurons
- What is an artificial neural network?
- Artificial neurons in detail
- Common Network architectures
- Learning Paradigms

Brains & Neurons



The brain is composed of approximately 10 billion neurons, each of which is connected to about another 10,000.

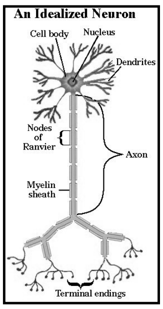
Neurons consist of

- a cell body called the soma

- a number of spine like extensions called dendrites

- a single nerve fibre called the axon which branches out from the soma and connects to other neurons

Neurons



Neurons transmit information through electrical impulses called action potentials.

Neurons combine input signals from the connections with other neurons (synapse).

When the stimulation received from incoming signals through the dendrites reaches a threshold, the neuron fires, sending a pulse along the axon to other neurons.

Neural Processing

• No new neurons are created in the brain after birth. (Current research suggests some may)

• Learning occurs by building connections between neurons and altering the strength of these connections, and deleting neurons that don't connect.

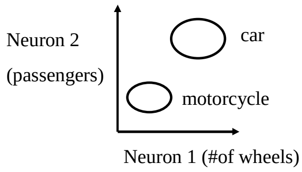
• Neurons are all-or-nothing in their firing, but the intensity of a stimulus can be encoded by the the rate of firing of neurons and the number of neurons that fire in response to the stimuli.

Neural Processing (cont.)

Neural networks degrade gracefully with a loss of neurons – they are quite robust to damage.

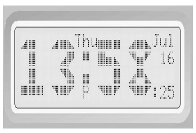
This robustness is due to the fact that information is distributed across the entire network, rather than being located in one specific place. I.e. no "grandmother neuron".

I.e. concepts are represented in terms of feature dimensions.



Local vs Distributed Representations

Local – the representation contains three lights, each representing a specific concept. Damaging the components will have serious effects.



Digits on an LCD screen can be thought of as an example of a distributed representation. Digits are represented by many points, but each point is not particular to a specific digit. Destroying points in this representation will degrade the system gracefully.

Computational Theory of Mind

- The Brain can be viewed and "understood" as a computational device.
- That is, there are inputs (e.g. from stimulus), outputs (the response), and some kind of algorithm that converts one to the other.
- However, difficult to go in one step, so usually several intermediate layers of representations and associated conversion algorithms and memory is involved.

Artificial Neural Networks

- ANNs employ biologically inspired approaches to problem solving
- Processing is different from traditional computers in that information is distributed and is processed in parallel across large numbers of simple processing units
- Simple learning algorithms are defined that alter the connections between units
- Apart from practical applications, neural net models can provide insight into many cognitive processes such as perception, learning, memory and language.

Artificial Neural Networks

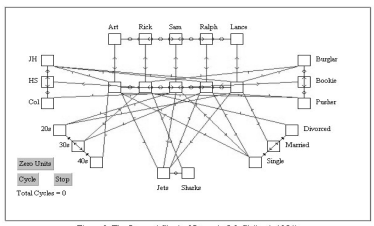


Figure 2: The Jets and Sharks Network (McClelland, 1981)

Example neural network using BrainWave. The squares are units, which represent a mathematical simplification of biological neurons. Arrows are weights (red = excitatory, blue = inhibitory)

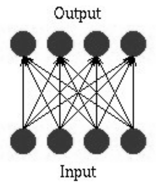
Note: unit activation is continuous rather than all or nothing.

Artificial Neural Networks

Artificial neural networks come in many different shapes and sizes. The most commonly used are the following:

- Single Layer Feedforward Networks
- Multi Layer Feedforward Networks
- Fully Recurrent Networks
- Competitive Networks
- Jordan Networks
- Simple Recurrent Networks

Single Layer Feedforward Networks

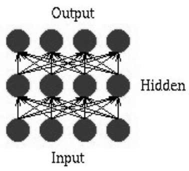


Single Layer Feedforward

The activations of the units are set and then propagated through the network until the values of the outputs units are determined.

The network acts as a vector-valued function, taking one vector as input and returning another vector as output.

Multi Layer Feedforward Networks

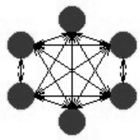


Multi Layer Feedforward

The activations of the units are set and then propagated through the network until the values of the outputs units are determined.

Differs from Single Layer Nets, in that hidden layers are included. However, more advanced learning algorithms are required as the correct representations at the hidden layers are unknown and must be derived.

Fully Recurrent Networks


Fully Recurrent Network

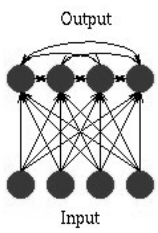
Perhaps the simplest of neural network architectures.

All units are connected and considered to be both an input and an output.

Typically, a pattern is given to the whole network, during learning when the weights are modified.

When a degraded version is given to the network, the network attempts to reconstruct the pattern.

Competitive Networks



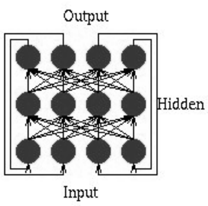
Competitive Network

Similar to a single-layer feedforward network except that there are connections, usually negative between output nodes.

These output nodes compete to represent the input pattern.

If connections are local, the network can be made to organize itself topologically. That is, close output nodes represent similar inputs.

Jordan Networks



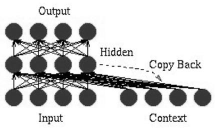
Jordan Network

Recurrent networks are useful in that they allow networks to process sequential information.

That is, the context of the input is stored so that it can influence processing. E.g. pronouncing letters is dependant on context (CHair versus Cat).

Jordan Networks use the previous output as input to the network.

Simple Recurrent Networks



Simple Recurrent Network

Simple recurrent networks use the hidden representation from the previous time step as extra input.

Creates a context that is not dependant on current output as in Jordan networks.

How a Neural Network Computes

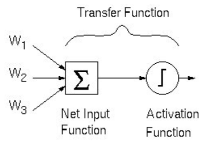


Figure 4: The Transfer Function

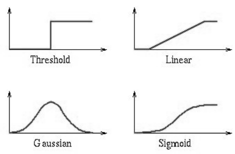


Figure 5: Some common activation functions.

The transfer function multiplies each weight projecting to the unit by the activation of the (input) units which the weight projects from. A bias value is added to the weighted sum of the inputs to calculate the net input. Then a very simple activation function is applied to this input to determine the output of the unit.

Computing AND

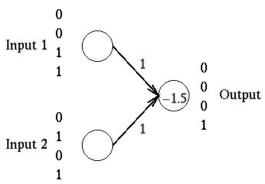


Figure 6: AND Network.

The AND network has 2 input units and a single output unit.

Inputs and outputs are either 0 or 1.

The network uses a threshold activation function which outputs a 0 if the net input is less than 0 and 1 otherwise. The weights are both 1 and the bias is –1.5.

Input 1	Input 2	Net input	Output
0	1	0*1 + 1*1 –1.5 = -0.5	0
1	1	1*1 + 1*1 – 1.5 = 0.5	1

Letter Recognition Example

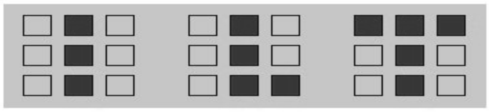


Figure 7: ILT Input Representations.

Suppose we want to develop a network that recognizes the letters “I”, “L” and “T”.

Figure 7 above shows the input representations. Red = a value of 1, and grey a unit value of 0.

Letter Recognition Example

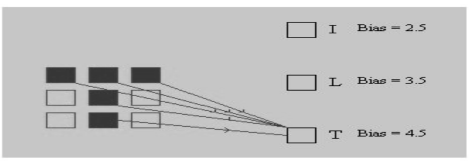


Figure 8: T Weights.

To recognize “t” we can connect the units that will be active in the “T” to the “T” output unit with a weight of 1 and set the bias to 4.5.

So when “T” is presented, the net input is 5, which exceeds the bias, thus activating the desired output node.

Letter Recognition Example

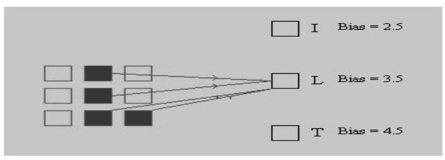


Figure 9: L Weights.

Similarly, to recognize “L” we can connect the active units to the “L” output and set the bias to 3.5

When “L” is presented, the net input will be 4, which will exceed the bias, thus activating the L node.

Letter Recognition Example

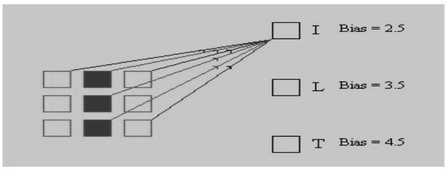


Figure 10: I Weights.

The "I" pattern is more tricky. Simply connecting the active units in a similar way will give a false positive when presented with "L" or "T". However, giving these other nodes an weight of –1 will decrease the net input to below the bias if these extra units are active, thus only activating the "I" node during the correct pattern.

Letter Recognition Example

- While it is possible in a simple example such as this to select the weights by hand, it quickly becomes infeasible as the size of the network and the number of patterns increases.
- Thus we need learning algorithms to automatically select the appropriate weights.

How a Neural Network Learns

The output of a network given a particular input depends on the connection structure and the associated weights.

During learning, generally the connection structure is held fixed and the weights are modified to improve performance.

How the weights are modified depends on the objective of the network and the information available to the learning rule.

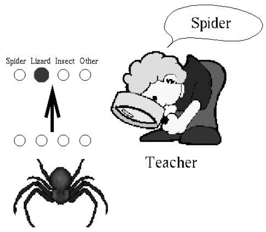
There are three main learning paradigms:

1. Supervised Learning with a Teacher
2. Supervised Learning with Reinforcement
3. Unsupervised Learning

Supervised Learning with a Teacher

In this learning paradigm the learning algorithm is given a set of input/output pairs.

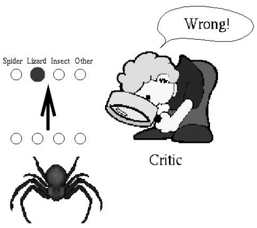
Given a specific input, the weights are modified so that the network will give the required output in future.



In this example, the network is given a set of images to classify. If the network is shown a spider, but classifies it a lizard, the weights will be adjusted to make the network respond "spider" in future.

Supervised Learning with Reinforcement

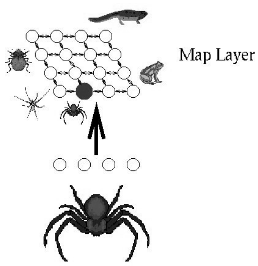
Similar to learning with a teacher, except the network is given a measure of performance rather than explicitly the correct answer. For example, using number of goals scored to train a network to play soccer.



If the network is shown a spider, but classifies it a lizard, the network is given information that it is incorrect rather than what the output should have been. For this reason, learning with a critic is often more difficult and takes longer.

Unsupervised Learning

Unsupervised learning does not receive information for either a teacher or critic. Instead it relies on an internal criterion to guide learning.



For example, the objective may be to create an output representation so that similar inputs activate output units that are close.

The network is shown a series of animals and gradually changes the weights so that similar animals are mapped to adjacent units.

References

- http://www.itee.uq.edu.au/~cogs2010/cmc/ chapter 1

Lab Next Week

- Chapter 1 and 2 from CMC above
- In lab 78-208

Discussion:

What is the role of models in cognitive science?

Why model?

"The aim of modelling is just that – modelling, not realism. A model's purpose is to capture the essence of a problem and allow its exploration to facilitate greater understanding, not as a black box whose emergent behaviour provides all the answers."

James Watson, 2002

What is a (good) model?

- Types of models include
  - Computational models
    - Agents and behaviours specified by programs
  - Scale models (eg dolls house)
    - Same relationships between parts, different size & materials
  - Wind tunnels
    - Same function, different materials
  - Mathematical models
    - Equations that describe
  - Boxes and arrows diagrams
  - Maps
    - Same relationships, different scale
  - Biological models (eg rats that develop Alzheimer's Disease)
    - Same functional outcomes, different animals
- A model abstracts relevant properties and omits irrelevant ones

Models contribute to understanding in many ways, including

- New explanations for cognitive phenomena
- Converging evidence for empirical studies
- Testing the internal consistency and completeness of theories
- Testing the conditions under which phenomena occur & when they don't
- Exploring emergent properties of parameterised systems intractable to mathematical analysis
- Exploring emergent properties of systems of agents with complex dynamics that are not inherent in the behaviour of a single agent
- Deepening our understanding of what is, against a background of what might have been

Why simulate a model?

"For those, like me, who are not mathematicians, the computer can be a powerful friend to the imagination. Like mathematics, it doesn't only stretch the imagination. It also disciplines and controls it."

Dawkins, 1986, p. 74.

Reference:

Dawkins, R. (1986). *The Blind Watchmaker*. Penguin Books.