Biologically Inspired Computation F21BC2

Artificial Neural Networks

Nick Taylor
Room EM 1.62
Email: N.K.Taylor@hw.ac.uk

Computational Neuroscience

Computational neuroscience is characterised by its focus on understanding the nervous system as a computational device rather than by a particular experimental technique.

Experimentation and Modelling

- Neuronal Networks
- Sensory Systems
- Motor Systems
- Cerebral Cortex

Two Disciplines

- Neurophysiology
  - Province of Biological Neuronal Network (BNN) Experimenters
- Connectionism
  - Province of Artificial Neural Network (ANN) Modellers

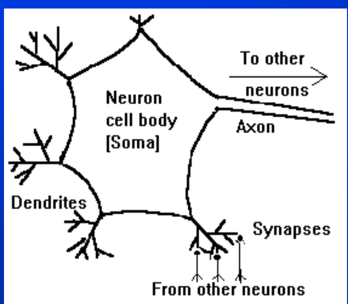
Differing Perspectives

- BNN Experimenters' agenda
  - Understanding
    - Neurogenesis; Neurotransmitters; Plasticity
  - Pathology
    - Neuronal dysfunction; Diagnosis; Treatments
- ANN Modellers' agenda
  - Performance
    - Training/execution speeds; Reliability; Flexibility
  - Applicability
    - Architectures; Complexity; Fault tolerance

Neurophysiology

- Background
- Axons, synapses & neurons
- Learning & synaptic plasticity
- Problems
- Summary

Background

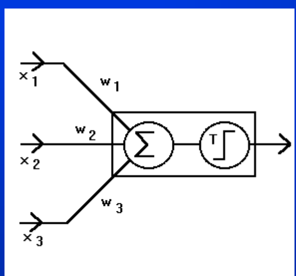


- The human brain contains about 1 billion neurons
- Each neuron is connected to thousands of others
- Neurons can be either excitatory or inhibitory
- Neurons perform very simple computations
- The computational power of the brain is derived from the complexity of the connections

Axons, Synapses and Neurons

- The primary mechanism for information transmission in the nervous system is the axon
- An axon relays all-or-nothing (binary) impulses
- Signal strength is determined from the frequency of the impulses
- An axon signal eventually arrives at a synapse
- A synapse may either attenuate or amplify the signal whilst transmitting it to a neuron
- A neuron accumulates the modified signals and produces an impulse on its own axon if the total synaptic input strength is sufficient

Model of a Neuron



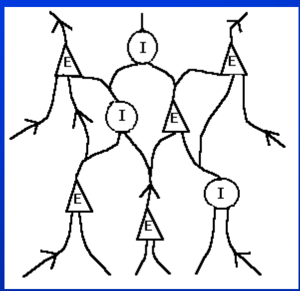
- McCulloch and Pitts model of a neuron (1943)
- Summation of weighted inputs
- Threshold, *T*, determines whether the neuron fires or not

- Firing rule:

$$\sum_i x_i w_i > T \quad \textit{then} \quad \textit{fire}$$

$$\leq T \quad \textit{then} \quad \textit{don't} \quad \textit{fire}$$

Assemblies of Neurons



- Modifications to neuron assemblies can only be achieved by adjusting the attenuation or amplification which is applied at the synapses

- Hebb Rule (1949) [*after James (1890!)*]
  - If a particular input is always active when a neuron fires then the weight on that input should be increased
- Learning is achieved through synaptic plasticity

Learning & Synaptic Plasticity I

- Long-Term Potentiation (LTP)
  - Hebbian increases in synaptic efficacy (amplifications) have been recorded on
    - Active excitatory afferents to depolarised (firing) neurons
- Long-Term Depression (LTD)
  - Decreases in synaptic efficacy (attenuations) have been recorded on
    - Inactive excitatory afferents to depolarised (firing) neurons
    - Active excitatory afferents to hyperpolarised (non-firing) neurons
    - Active inhibitory afferents to depolarised (firing) neurons

Learning & Synaptic Plasticity II

- Nitric Oxide
  - Post-synaptic messenger discovered in 1990
  - Released by depolarised (firing) neurons
  - Can affect all active afferents in a local volume
- Consequences
  - NO makes it possible for one or more firing neurons to increase the synaptic efficacy of nearby neurons even if those nearby neurons aren't firing
  - NO can boot-strap synaptic efficacies which have dropped beyond redemption back to viability

Problems

- Hebbian learning paradigm inadequate
- Scant information on plasticity of inhibitory synapses
- Little known about the implications of the NO discovery for more global forms of plasticity
- Frequency-based models and analyses practically non-existent
- Behaviour of populations of neurons very complex and difficult to investigate

Neurophysiology Summary

- Much is already known
  - Enough to build models
- Neurophysiological correlates for many computational requirements have been found
  - LTP, LTD, NO
- Much is still unknown
  - Enough to severely restrict the models
- NO research is still in its infancy
  - Wider implications yet to be investigated

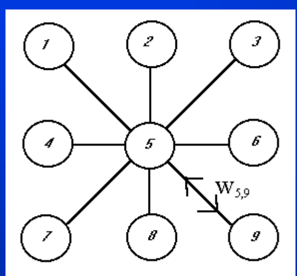
Connectionism

- Background
- Architectures
- Applications
- Problems
- Summary

Background

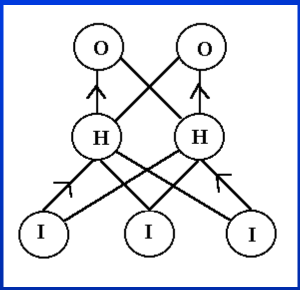
- Artificial Neural Networks (ANNs) are inspired, but not constrained, by biological neuronal networks
- Two very commonly used architectures
  - The Hopfield Network
    - Single layer, total connectivity within layer, auto-associative
  - The Multi-Layer Perceptron
    - Multiple layer, total connectivity between adjacent layers, no connectivity within layers, hetero-associative

The Hopfield Network



- Each node connected to every other node in the network
- Symmetric weights on connections ($w_{5,9} = w_{9,5}$)
- Node activations either -1 or +1
- Training performed in one pass: $\mathbf{w_{i,j} = \frac{1}{N} \Sigma\, p_i\, p_j}$
- Execution performed iteratively: $\mathbf{s_i = sign\, \{\Sigma\, w_{i,j}\, s_j\}}$

The Multi-Layer Perceptron



- Each node connected to every node in adjacent layers
- Connections feed forward from input nodes (I), through hidden nodes (H) to output nodes (O)

- Training performed iteratively: $\Delta w_{j,i} = \eta\, \delta_j\, s_i$
- Execution performed in one pass: $s_i = f\,(\Sigma\, w_{i,j}\, s_j)$

Hopfield Applications

- Content Addressable Memory
  - Partial patterns can be completed to reproduce previously learnt patterns in their entirety
    - Partially incorrect patterns are simply partial patterns
- Optimisation
  - Learnt patterns are simply attractors - minima of some energy function defined in terms of the $w_{i,j}$ and s_i variables
    - Using the objective function in an optimisation problem as the energy function, with suitably defined weights and activation equations, a Hopfield network can find minima of the objective function

MLP Applications

- Classification/Mapping
  - Kolmogorov's Mapping Neural Network Existence Theorem (Hecht-Nielsen)

    Any continuous function, $f : [0,1]^n \rightarrow \Re^m$, can be implemented exactly by a three-layer MLP having n input units, $(2n+1)$ hidden units and m outputs.

  - Applications are legion
    - Classification into categories by attribute values
    - Character recognition
    - Speech synthesis (NETtalk)
    - Vehicle navigation (ALVINN)

Problems

- Local minima
  - Hopfield: Linear combinations of learnt patterns or optimal solutions become attractors
  - MLP: Gradient descent training is the inverse of Hill-climbing search and is just as susceptible to local minima as the latter is to local maxima
- Limited storage capacity (Hopfield)
  - Less than $N/ln(N)$ patterns can be memorised safely
- Over-training (MLP)
  - Too many free variables ($w_{i,j}$) thwart generalisation

Connectionism Summary

- Neurologically inspired
  - Biological neurons and assemblies of neurons
- Broad applicability
  - Various architectures and training paradigms
- Readily implemented
  - Simple algorithms and data structures
- Reliability problems
  - Sub-optimality, capacity limitations, over-training, *Black Box* naivety