Computational Intelligence I: Neural Networks

Chapter 1: Introduction

Module Provider: Dr. Baoru Huang

Course structure:

COMP 575 (15 credits) consist of two parts:

- Part I: Neural Networks
- Part II: Evolutionary Computation

Typically, about 4 hours every week:

- About 20 lectures for theory.
- About 12 tutorial sessions to show examples, demos/practical and Q&As.

Assessment:

- Final examination: MCQs
- Coursework: formative/unassessed with feedback

Recommended books for this part:

- S. Haykin, Neural networks; a comprehensive foundation, New Jersey, Prentice Hall
- C. Bishop, Neural networks for pattern recognition, Oxford University Press
- C. Bishop, Pattern recognition and machine learning, Springer-Verlag
- R. Duda, P. Hart, D. Stork, Pattern classification, Wiley and Sons
- S. Theodoridis, K. Koutroumbas, Pattern recognition, Academic press

Course contents:

- **Introduction**
  - Chapter 1
- **Structural Aspects**
  - Chapter 2
- **Learning Processes**
  - Chapter 3
- **Single-Layer Perceptrons**
  - Chapter 4
- **Multi-Layer Perceptrons**
  - Chapter 5
- **Radial-basis Function Networks**
  - Chapter 6
- **Support Vector Machines**
  - Chapter 7
- **Basics of Deep Reinforcement Learning**
  - Chapter 8

Brains & (A)NNs?

- Artificial Neural Networks (or Neural Networks) have been motivated by the recognition that the human brain computes in a way vastly different from a conventional computer.

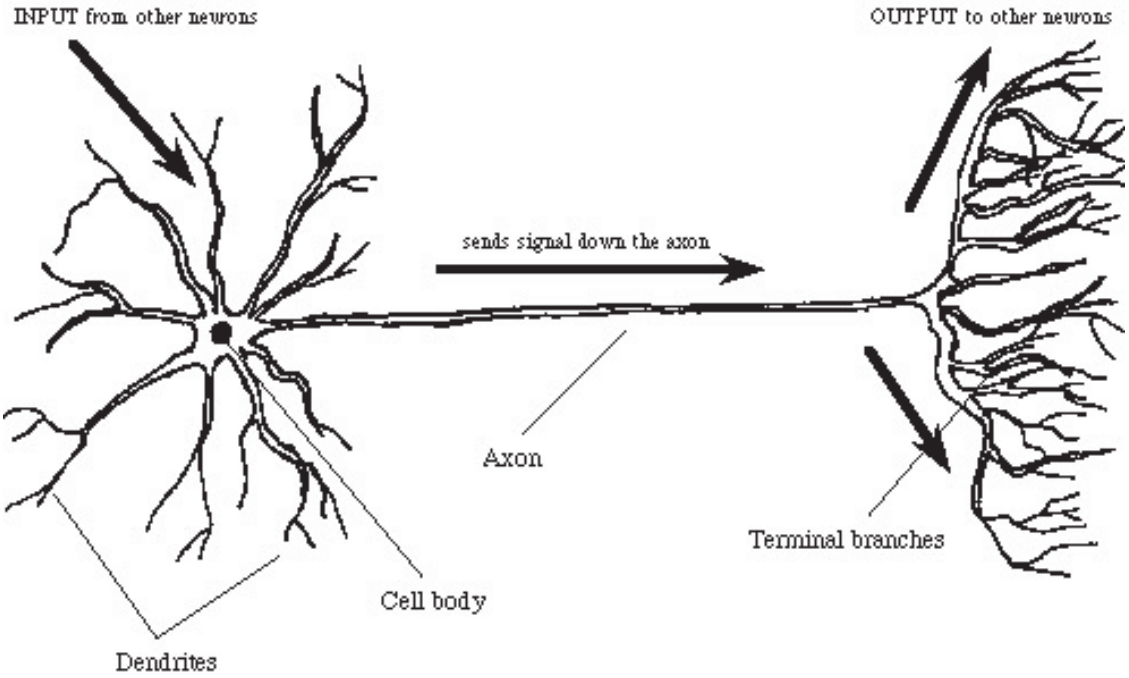
- A brain is a highly complex, nonlinear, massively parallel **computing machine** (i.e., supports information processing), which uses simple internal structural constituents, the **neurons**, to facilitate and support motor control, perception, pattern recognition, etc.

- At birth, the brain has an existing rich structure and is able to assimilate information (experience), develop the neurons and build new rules. **Brain plasticity** (neuron development) permits adaptation of the nervous system to the environment.

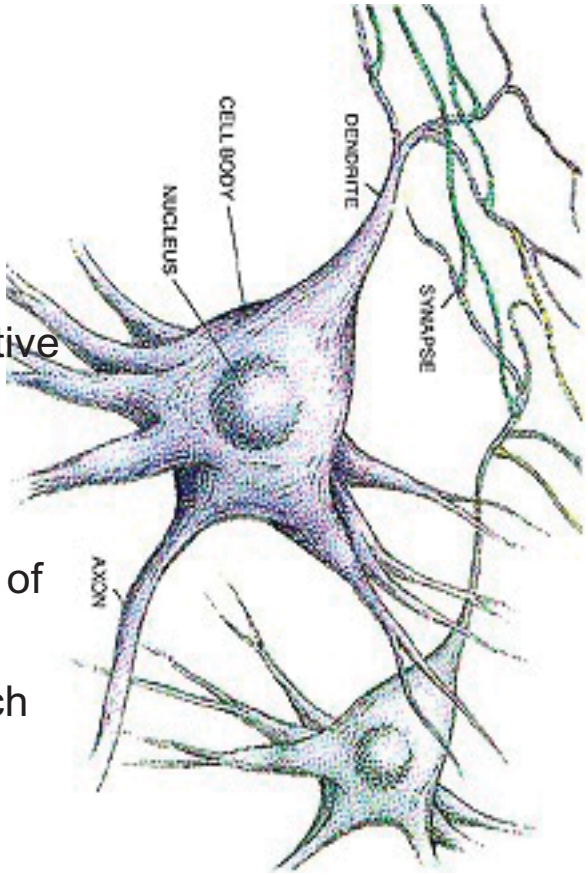
- **ANNs are biologically inspired analogues of the brain**, and are also based on primitive neuron components. Both brains and ANNs acquire and store information supplied by the environment through a **learning process**, and represent knowledge using **inter-neuron connection strengths** (synaptic weights).

- ANNs are **massively parallel distributed** processors composed of simple processing units capable of storing patterns in a **distributed** fashion.

Neuron structure



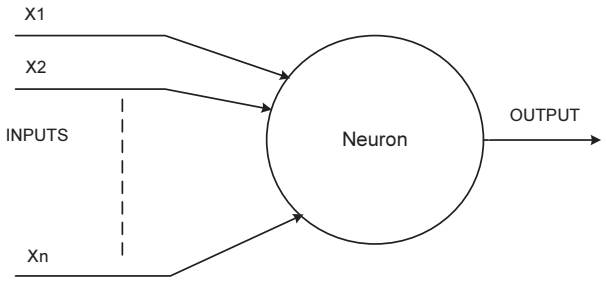
- 10 billion neurons in the human brain communicate with each other through 60 trillion **synapses** which facilitate learning via creation of new and modification of existing ones.
- **Axons** are the signal transmission structures, while **dendrites** the receptive ones; these have both morphological and functional differences.
- The electrical activity spikes are sent down the axon and split to thousands of **terminal branches**.
- These terminate in the synapses which either inhibit or excite the activity via electrochemical conversions.

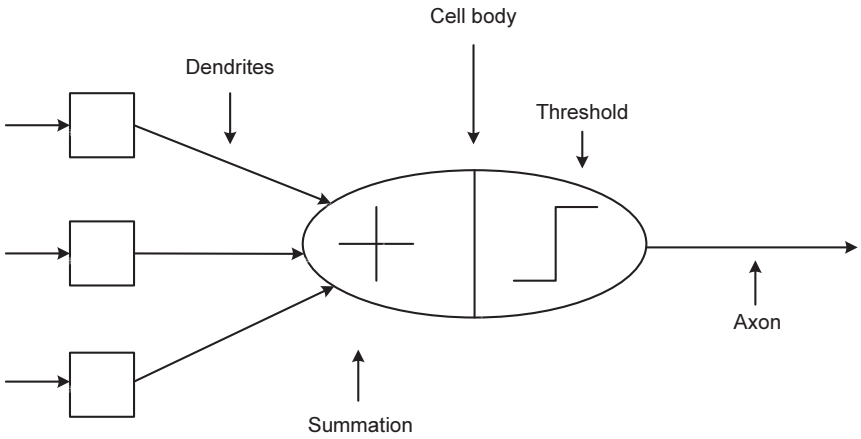


Simulating a brain!

⌘ Each brain neuron computes a combination (e.g., weighted sum) of activities/inputs in ~5ms and produces a corresponding output (this is equivalent to several hundred additions). The operations are much slower, but with much finer connections than on a microchip and occupy millions of times less volume.

⌘ Based on certain simplifications, we can implement in software or hardware the overall operation of a biological neural network.





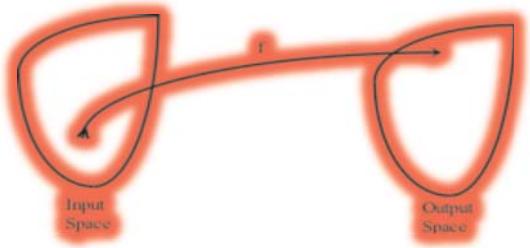
⌘ Each ANN neuron receives multiple inputs, and generates one output. Each ANN has a topology (e.g., graph representation) and parameters (e.g., connection weights).

⌘ The network has two modi operandi: the **training mode** (where weights are adjusted to learn a task) and the **online mode** (where input signals are converted to output ones).

Advantages of ANNs?

- **Nonlinearity:** neurons can be linear or nonlinear. Many signals are inherently nonlinear (e.g., speech & vision).

- **Input-to-output mapping:** sensory inputs are converted to output via a complex distributed function represented by the network.

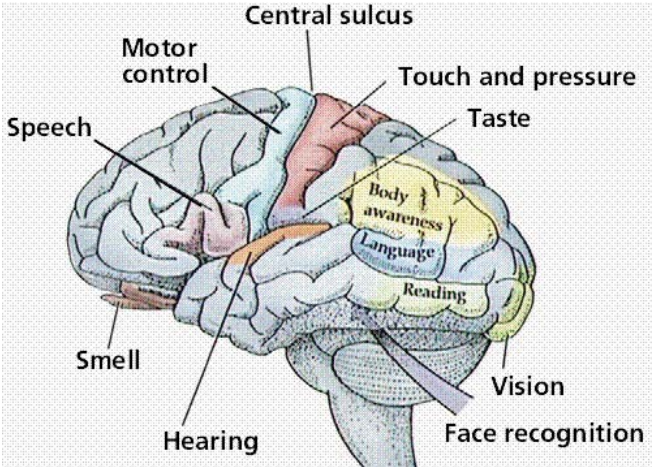


- **Adaptivity:** NNs can adapt (be retrained) so that their synaptic weights reflect the changes in the surrounding environment in real-time.

- **Evidential response:** during pattern classification, NNs can in addition to the output, provide decision confidence / probability, which is very useful for the rejection of ambiguous patterns.

- **Contextual information:** knowledge is not something separate (file in memory distinct to CPU), but it is represented by the very structure and activation state of the network. Each neuron is affected by all others; local processing and global awareness.

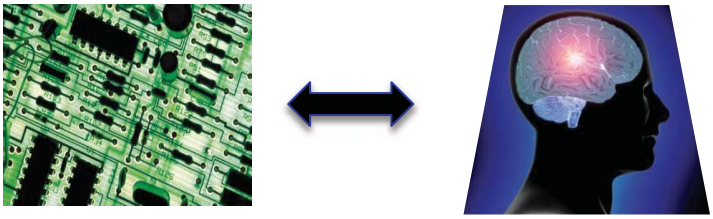
⌘ **Fault tolerance:** when a damage occurs, the network degrades gracefully, as opposed to a fatal catastrophic failure with complete information and processing loss.



⌘ **Massively parallel implementability:** their massively parallel structure is highly suitable for very-large-scale integrated electronics (VLSI, GPUs, TPUs).

⌘ **Uniformity of analysis and design:** NNs are composed of similar components (neurons), and this makes it easier to apply mathematical techniques to enable learning and network analysis. NNs also allow seamless modular integration.

⌘ **Neurobiological analogy:** since brains are the living proof that fault tolerant adaptive learning is physically feasible, ANNs can be used to interpret neurobiological behaviour (➔). Conversely, borrowing ideas from nature allows the creation of complex learning machines (➔).



NN software to use here

- Python-based, open-source software:
  - Tensorflow (Google), Keras.

  - Pytorch (Facebook/Meta AI).

- Matlab (Mathworks):

- In practice, similar ways of designing neural networks.
- Many more!

<END of Chapter 1>