Connectionist Computing
COMP 30230/41390

Gianluca Pollastri

office: E0.95, Science East.

email: gianluca.pollastri@ucd.ie

Credits

- **Geoffrey Hinton, University of Toronto.**
  - **borrowed some of his slides for "Neural Networks" and "Computation in Neural Networks" courses.**

- **Ronan Reilly, NUI Maynooth.**
  - **slides from his CS4018.**
- **Paolo Frasconi, University of Florence.**
  - **slides from tutorial on Machine Learning for structured domains.**

Lecture notes on Brightspace

- **Strictly confidential...**
- **Slim PDF version will be uploaded later, typically the same day as the lecture.**
- **If there is demand, I can upload onto Brightspace last year's narrated slides.. (should be very similar to this year's material)**

Books

- No book covers large fractions of this course.

- Parts of chapters 4, 6, (7), 13 of Tom Mitchell's "Machine Learning"
- Parts of chapter V of Mackay's "Information Theory, Inference, and Learning Algorithms", available online at:

http://www.inference.phy.cam.ac.uk/mackay/itprnn/book.html

- Chapter 20 of Russell and Norvig's "Artificial Intelligence: A Modern Approach", also available at:

http://aima.cs.berkeley.edu/newchap20.pdf

- More materials later..

Marking

- **3 landmark papers to read, and submit a 10-line summary on Brightspace about: each worth <u>6-7%</u>**
- **a connectionist model to build and play with on some sets, write a report: <u>30%</u>**
- **Final Exam in the RDS (<u>50%</u>)**

Connectionism

- A computational approach to modelling *the brain* which relies on the interconnection of *many simple units* to produce complex behavior
- Not the usual paradigm where there is a powerful central processor that executes serially a static program:
  - simple elements,
  - parallel processing,
  - learning..

What is it good for?

- **To understand how the brain works:**
  – **by creating and training/testing connectionist models we can try to guess how the brain stores memories, processes language, recognises faces, etc.**
- **To understand and develop a different type of computation:**
  – **we can't recognise faces, language, etc. using traditional computational paradigms, while the brain can. Can't computers perform these tasks by emulating the brain?**
- **Who cares about the brain after all?**
  – **learning computers are useful even if they do things that have nothing to do with the brain.**

Connectionism, AI and Deep Learning

- Connectionism starts from Cognitive Science and the study of the brain.
- Between the mid 80s and circa 2005 it produced the vast majority of what we now call Deep Learning.
- While between 2006 and now Deep Learning has come up with new algorithms, much of what differentiates it from Connectionism is about emphasis and packaging rather than genuinely new science.

It started as a split field

Very roughly:

- Cognitive Scientists: interested in the brain. Connectionist computation as means to understand how we think.
- Machine Learning connectionists: interested in the algorithms.

A split field: attitudes

- **Let's assume that we design and train a connectionist model that plays backgammon (we will talk about this later).. The model is better than humans at playing backgammon**
- **Attitude 1:** ***it certainly does not work like the brain,*** **hence we failed.**
- **Attitude 1½:** ***it does well what the brain does well,*** **great!**
- **Attitude 2:** ***the algorithm must be a powerful one,*** **why don't we try the same algorithm on drug design (**which incidentally humans are a total failure at**)?**

Connectionism again

- **A computational approach which relies on the interconnection of *many simple units* to produce complex behaviour**

- **Not the usual paradigm where there is a powerful central processor that executes serially a static program:**
  - **simple elements,**
  - **parallel processing,**
  - **learning..**

Note

- **Parallel processing: the theoretical model entails this - in practice, most times, one is running connectionist code on (somewhat) serial machines.**
- **This isn’t a problem. Many very simple operations which could in principle run in parallel (but we run serially because of the hardware we have available).**

Note 2

- Parallel processing: there is a profound difference in scale between the theoretical connectionist parallelism (brain-like) and what we can get even with an array of GPUs.

- Nevertheless, GPUs are probably the single most important contributor to the repackaging/upgrade of Connectionist Machine Learning to Deep Learning.

What this course is about

- General (quick) overview of connectionism, history of connectionism
  - Models of the neuron
  - Perceptron, Hopfield networks, Boltzmann machine
- Learning, algorithms
  - Supervised Learning: PAC learning, VC dimension.
  - Feed-forward Neural Networks: Gradient Descent, Backpropagation.
  - Reinforcement learning, Unsupervised learning.
  - Deep Networks: Recurrent Neural Networks; Neural Networks for graphs.
- Applications: speech, images, other stuff..

The brain

- **“It’s big and very complicated and made of yukky stuff that dies when you poke it around” (G.Hinton*)**

*** one of the fathers of connectionism and arguably the inventor of "Deep Learning"**

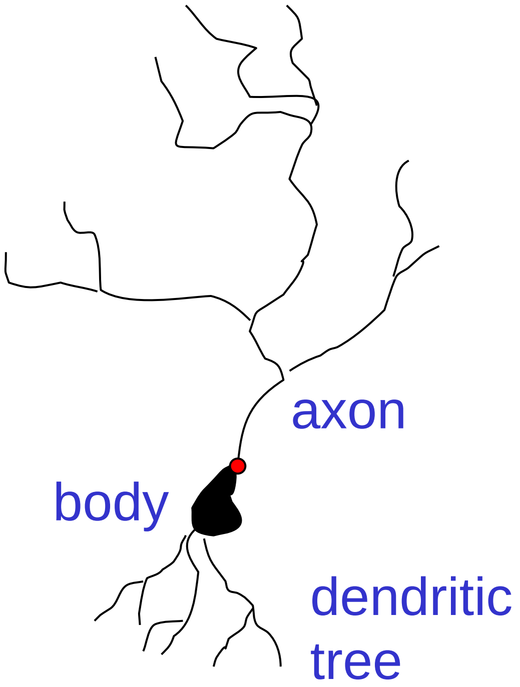
Facts about the brain

- The brain consists of around 10^{11} neurons.
- Neurons are connected: each neuron receives between 10^3 and 10^4 connections. Hence there are 10^{14} to 10^{15} connections in the brain (100-1000 Tbytes to store 1 number for each of them).

- The "currency" of the brain is the action potential or voltage spike.
- There appears to be considerable localisation of function in the brain.

A typical cortical neuron

- **Gross physical structure:**
  - **One axon that branches**
  - **A dendritic tree that collects input from other neurons**
- **Axons typically contact dendritic trees at synapses**
  - **A spike of activity in the axon causes charge to be injected into the post-synaptic neuron**
- **Spike generation:**
  - **Outgoing spikes whenever enough charge has flowed in at synapses to depolarise the cell membrane**

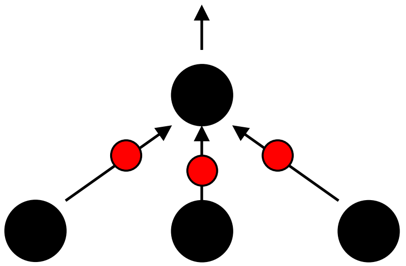


Synapses

- **When a spike travels along an axon and arrives at a synapse it causes transmitter chemical to be released**
  - **There are several kinds of transmitter**
- **The transmitter molecules diffuse across the synaptic cleft and bind to receptor molecules in the membrane of the post-synaptic neuron thus changing their shape.**
  - **This opens up holes that allow specific ions in or out.**
- **The effectiveness of the synapse can be changed**
  - **vary the amount of transmitter**
  - **vary the number of receptor molecules.**
- **Synapses are slow, but they have advantages over RAM**
  - **Very small**
  - **They adapt using locally available signals (but how?)**

How the brain works (simplifying)

- Each neuron receives inputs from other neurons
  - Cortical neurons use spikes to communicate
  - The timing of spikes is important
- The effect of each input line on the neuron is controlled by a synaptic weight
  - The weights can be positive or negative



- The synaptic weights adapt so that the whole network learns to perform useful computations
  - Recognising objects, understanding language, making plans, controlling the body
- A huge number of weights can affect the computation in a very short time. Much better bandwidth than computers.

Modularity and the brain

- Different bits of the cortex do different things.
  - Local damage to the brain has specific effects
  - Specific tasks increase the blood flow to specific regions.
- But cortex looks pretty much the same all over.
  - Early brain damage makes functions relocate
- Cortex is made of general purpose stuff that has the ability to turn into special purpose hardware in response to experience.

- Looks like we should be able to do clever computations using simple, general-purpose processing units.

Enough of the brain

- But how can we model it using a computer?
- Let’s start with a single neuron.

Idealised neurons

- We want to model neurons in an idealised fashion
  - Removing complicated details that are not essential for understanding the main principles.
  - Allowing us to apply mathematics.
  - Complexity can always be added
- It is often worth understanding models that are known to be wrong (but we mustn't forget that they are wrong!)
  - E.g. neurons that communicate real values rather than discrete spikes of activity.
  - E.g. neurons that do stuff in predetermined moments (with a clock) instead of whenever they want.

Linear neurons

- **Simple but limited**
  - **If we can make them learn we may get insight into more complicated neurons**

bias

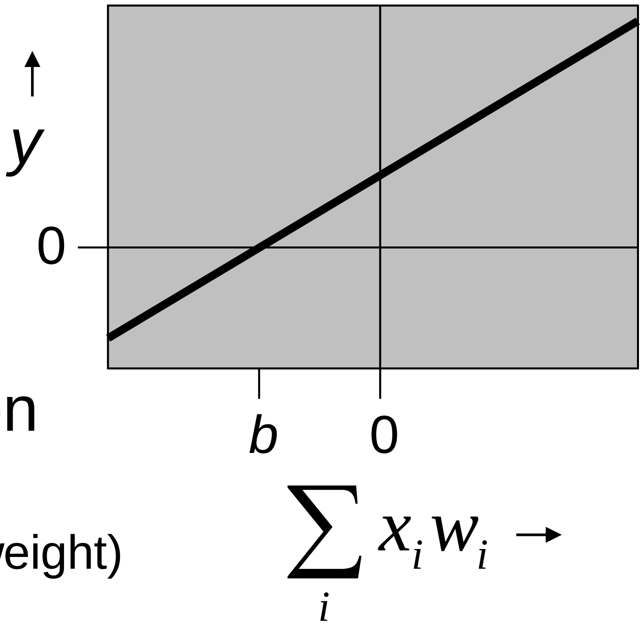
i^th input

$$y = b + \sum_i x_i w_i$$

output

sum over inputs

weight on i^th input (synaptic weight)

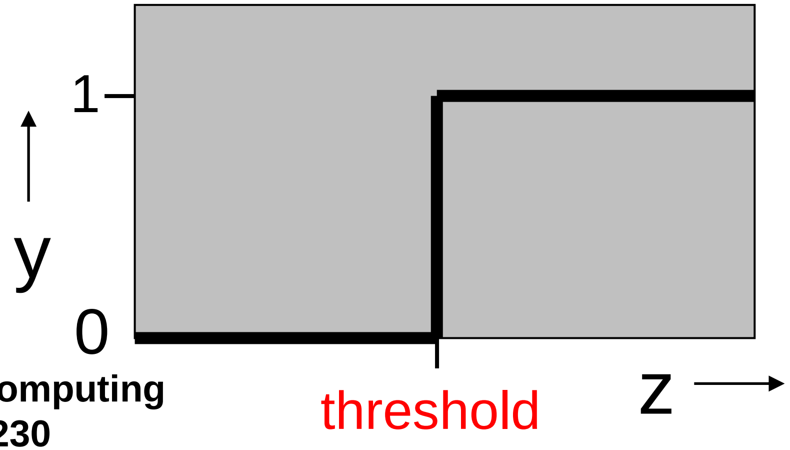


Binary threshold neurons

- **McCulloch-Pitts (1943):**
  - **First compute a weighted sum of the inputs from other neurons**
  - **Then send out a fixed size spike of activity *if the weighted sum exceeds a threshold*.**
  - **Maybe each spike is like the truth value of a proposition and each neuron combines truth values to compute the truth value of another proposition.**

$$z = \sum_i x_i w_i$$

$$y = \begin{cases} 1 \text{ if } z>\theta \\ 0 \text{ otherwise} \end{cases}$$



Sigmoid neurons

- **These give a real-valued output that is a smooth and bounded function of their total input.**
  - **Typically they use the logistic function or tanh function**
  - **They have nice derivatives which is why we like them.**
- **If we treat y as a probability of producing a spike: stochastic binary neurons**
- **Otherwise, simply a neuron that can be spiking a bit..**

$$z = b + \sum_i x_i w_i$$

$$y = \frac{1}{1 + e^{-z}}$$