

# Welcome to the computational cognitive modelling workshop!

## **Part 2: Artificial neural networks**

# Part 2: **Artificial neural networks**

*Olivia Guest*

*Chris Brand*

*Nick Sexton*

*Nicole Cruz De Echeverria Loebell*

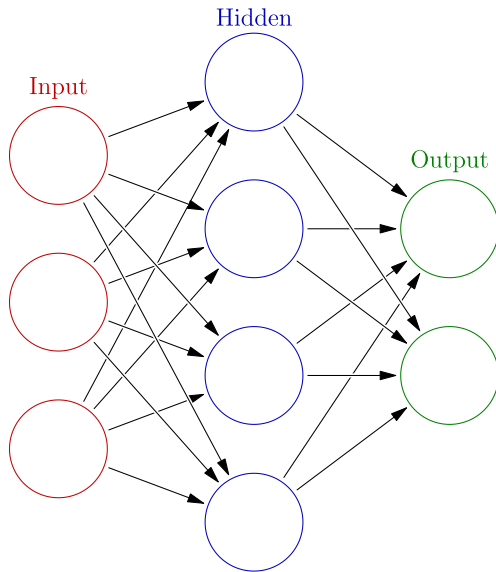
# What is a neural network?

A mathematical model

- ▶ Inspired by the nervous system
- ▶ A set of *units*, connected by *weights*
- ▶ The network *runs* by passing *activations* from the *input* (to the *hidden*) to the *output* units

# What is a neural network?

A mathematical model



# Why use artificial neural networks for modelling?

Some aspects of their behaviour are like their namesake!

- ▶ Learn pretty much any input-output data
- ▶ Uncover rules on their own about data
- ▶ Generalise what from what they have learnt
- ▶ Cope well with noise and damage

# How does an artificial neural network run?

By using maths, predictably!

Input units are set to a *pattern*

Calculate hidden units' states:

$$\begin{array}{rcl} 1 \times 0.5 & = & 0.5 \\ 1 \times 0.0 & = & 0.0 \\ 0 \times 0.8 & = & 0.0 \quad + \\ \hline & & 0.5 \end{array}$$

Same for output units:

$$\begin{array}{rcl} 0.5 \times 0.25 & = & 0.125 \\ 0.3 \times 1.5 & = & 0.45 \\ 1.6 \times -0.3 & = & -0.48 \\ -0.4 \times 1.1 & = & -0.44 \quad + \\ \hline & & -0.345 \end{array}$$

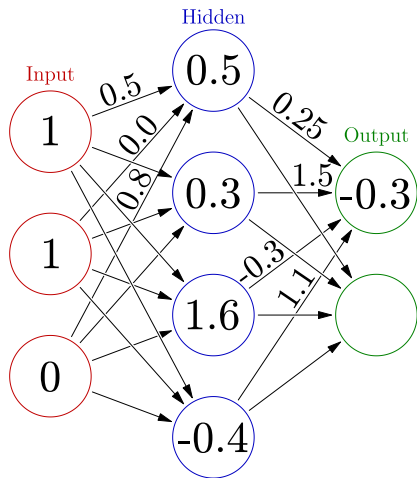


Figure: Glosser.ca / CC-BY-SA-3.0

# How does an artificial neural network run?

By using maths, predictably!

But we/programmers are lazy:

$$a_i = \sum_1^N x_j \times w_{ji}$$

where  $a_i$  is the unit whose state we want to calculate,  $N$  are the units on the previous layer, and  $w_{ji}$  is the weight on the connection between  $i$  and  $j$ .

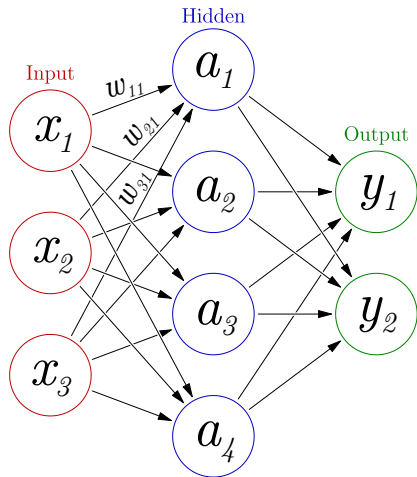


Figure: Glosser.ca / CC-BY-SA-3.0

# How do networks learn?

Cunning!

- ▶ Many options: Hebbian learning, back-propagation of error, Boltzmann machine learning, self-organising map algorithm, etc.
- ▶ All learning algorithms work by changing the connection weights
- ▶ Learning can be divided into *supervised*, *unsupervised*, and *reinforcement*



# Hebbian Learning

## A simple training rule

*Cells that fire together, wire together*

Carla Shatz

*The general idea is an old one, that any two cells or systems of cells that are repeatedly active at the same time will tend to become 'associated', so that activity in one facilitates activity in the other.*

(Hebb 1949, p. 70)

# Hebbian Learning

Cells that fire together, wire together

$$w_{ij} = \sum_i^N \eta x_i x_j$$

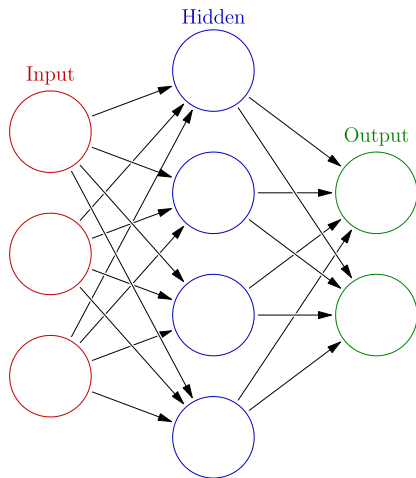


Figure: Glosser.ca / CC-BY-SA-3.0