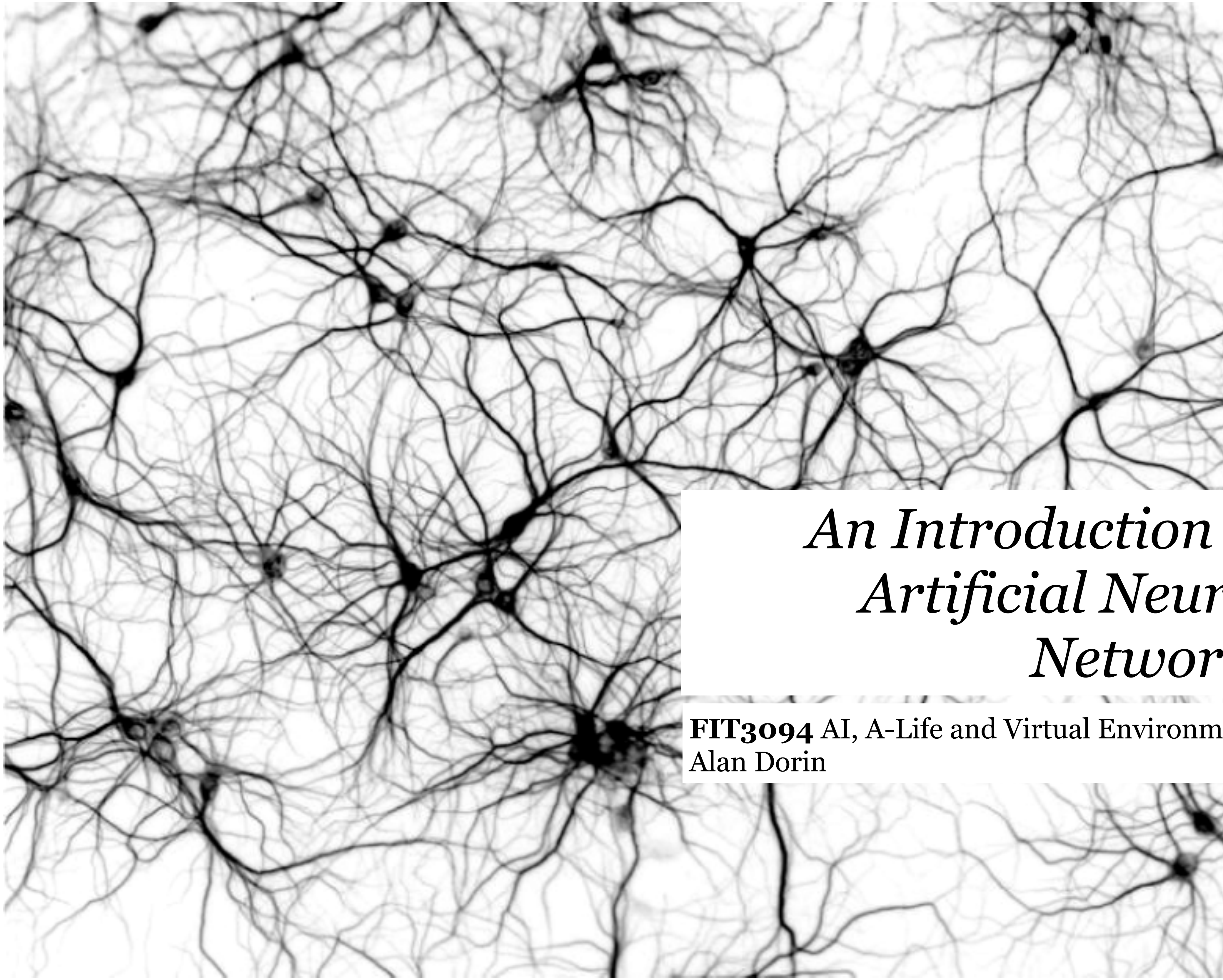


An Introduction to Artificial Neural Networks

FIT3094 AI, A-Life and Virtual Environments
Alan Dorin

MONASH University
Information Technology

Learning Objectives

To understand the basics components of biological neurons

To understand how feed-forward artificial neural networks are constructed

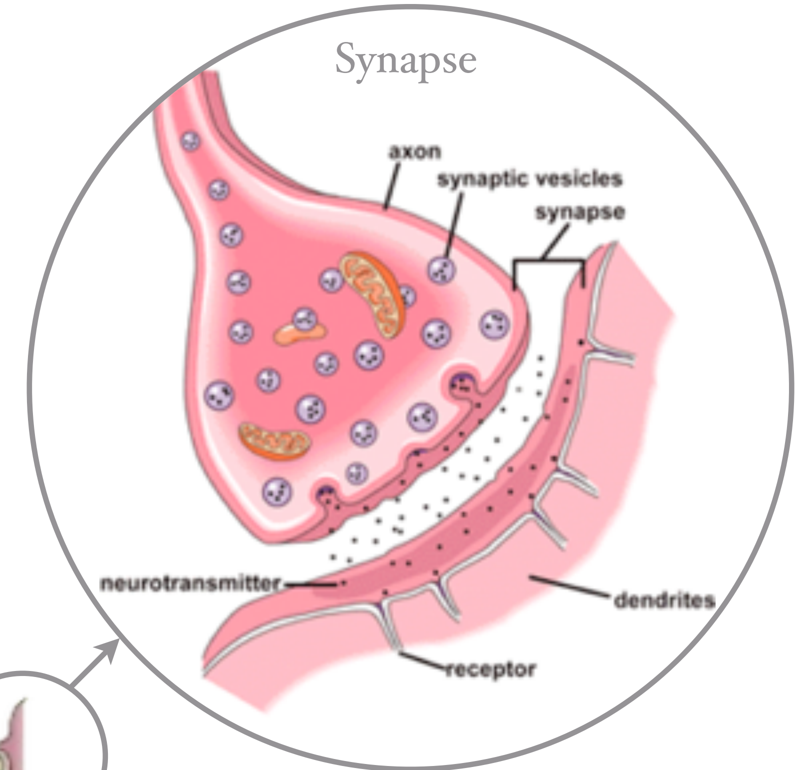
To be aware of the training methods available for artificial neural networks

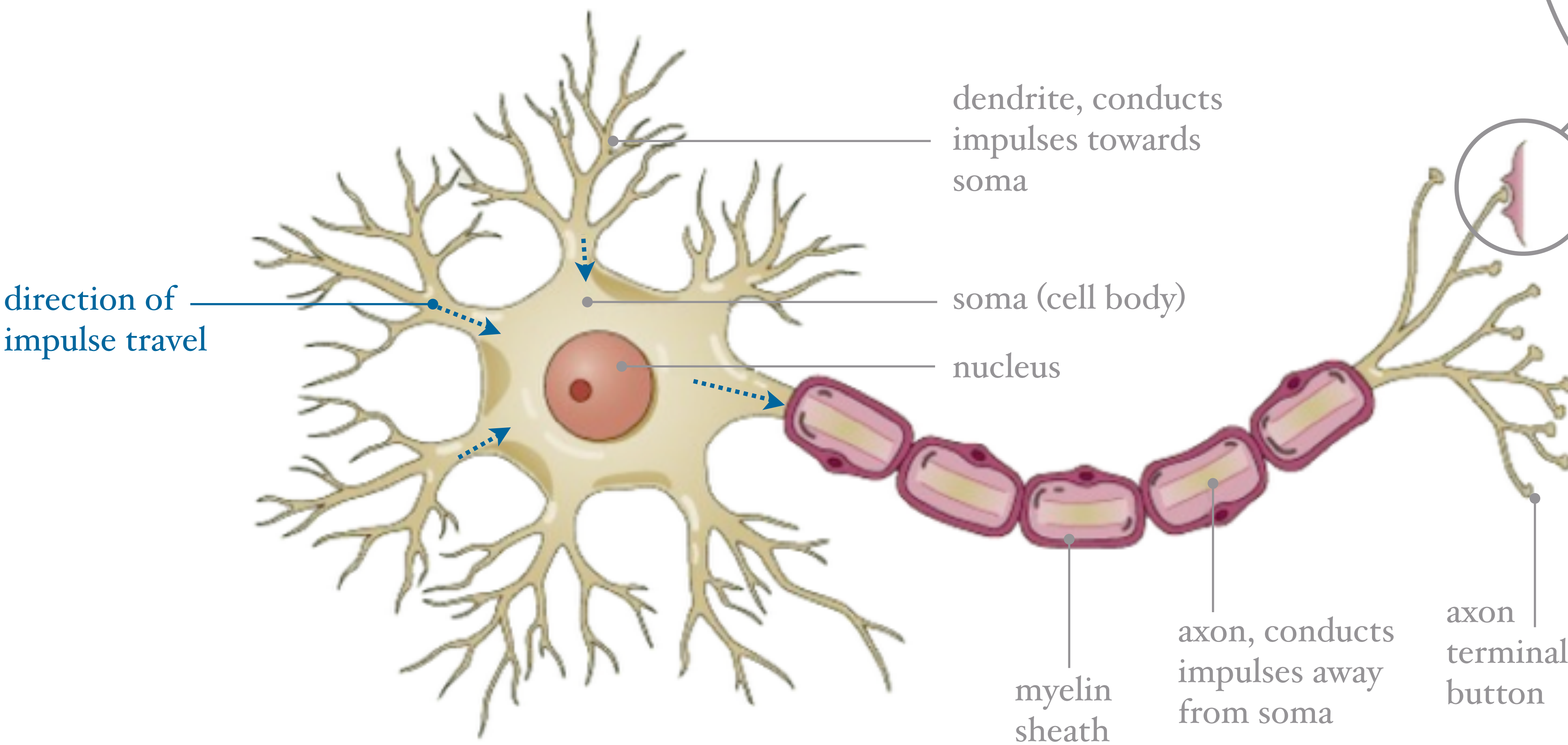
To be aware of the problems of over and under fitting in artificial neural networks.

To know suitable applications for artificial neural networks

The Biological Neuron

Animal brains are made of many cells called *neurons*. Humans have ~100 billion.
Dendrites and *axons* conduct electrical impulses to and from the cell.
Dendrites receive impulses from other cells' axons across a *synapse*.





Neuron Behaviour (in a nutshell)

As a cell receives impulses from its dendrites, its charge builds up
When a threshold is reached, the charge in the cell is reset and a signal is sent out via the axon.

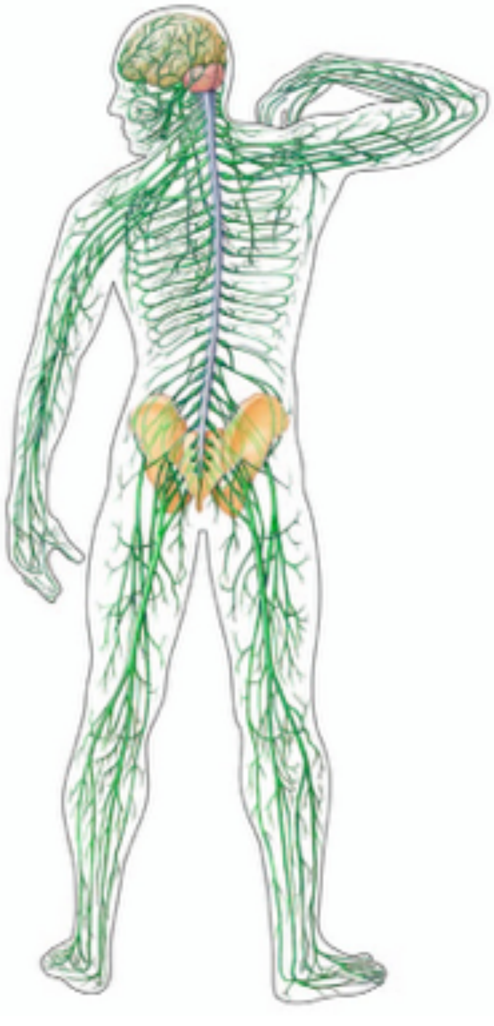
The Biological Neuron

Connections between neurons form pathways through the various parts of the brain.
Some neurons are connected to cells in the sensory organs.
Some neurons conduct signals to the motor systems and other organs of the body.

Repeated firing of neurons causes biological changes within them, allowing them to fire more easily in the future.

A lack of firing in particular neurons causes biological changes that inhibit them from firing very easily in the future.

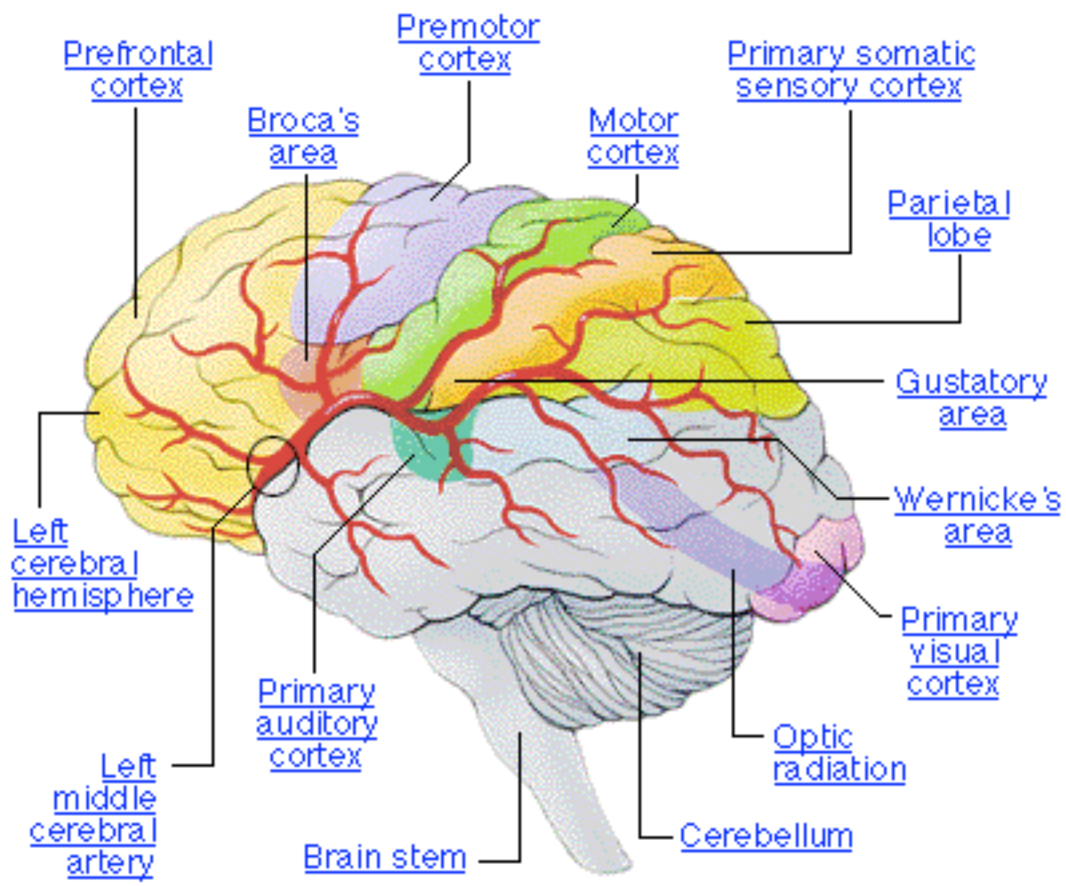
Together, these changes permit the brain/body system to learn.



The Biological Neuron

A neuron is said to *inhibit* others if it acts to reduce the likelihood of them firing.

A neuron that increases the likelihood of other neurons from firing is *excitatory*.

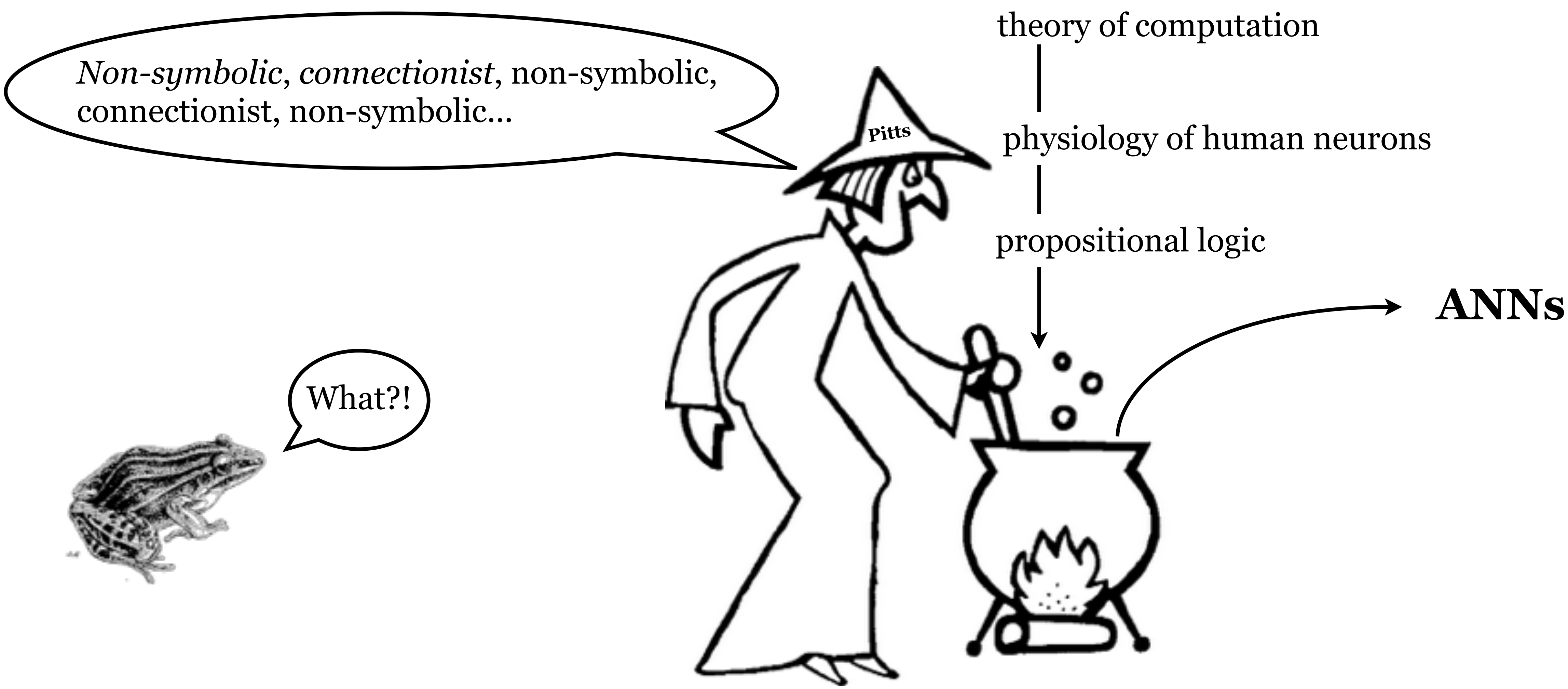


Human Brain
American Medical Association

http://www.medem.com/medlib/article/ZZZ0ZFP46JC

Although modern computer CPUs are *much* faster than biological brains, brains are *massively parallel* and they have proven quite effective!

Artificial Neural Networks (ANN)



McCulloch and **Pitts** threw some ideas together and came up with the idea of an *artificial neural network* (1943)

Hebb later demonstrated an updating rule for the weights between neurons that allowed the neural network to *learn* (1949)

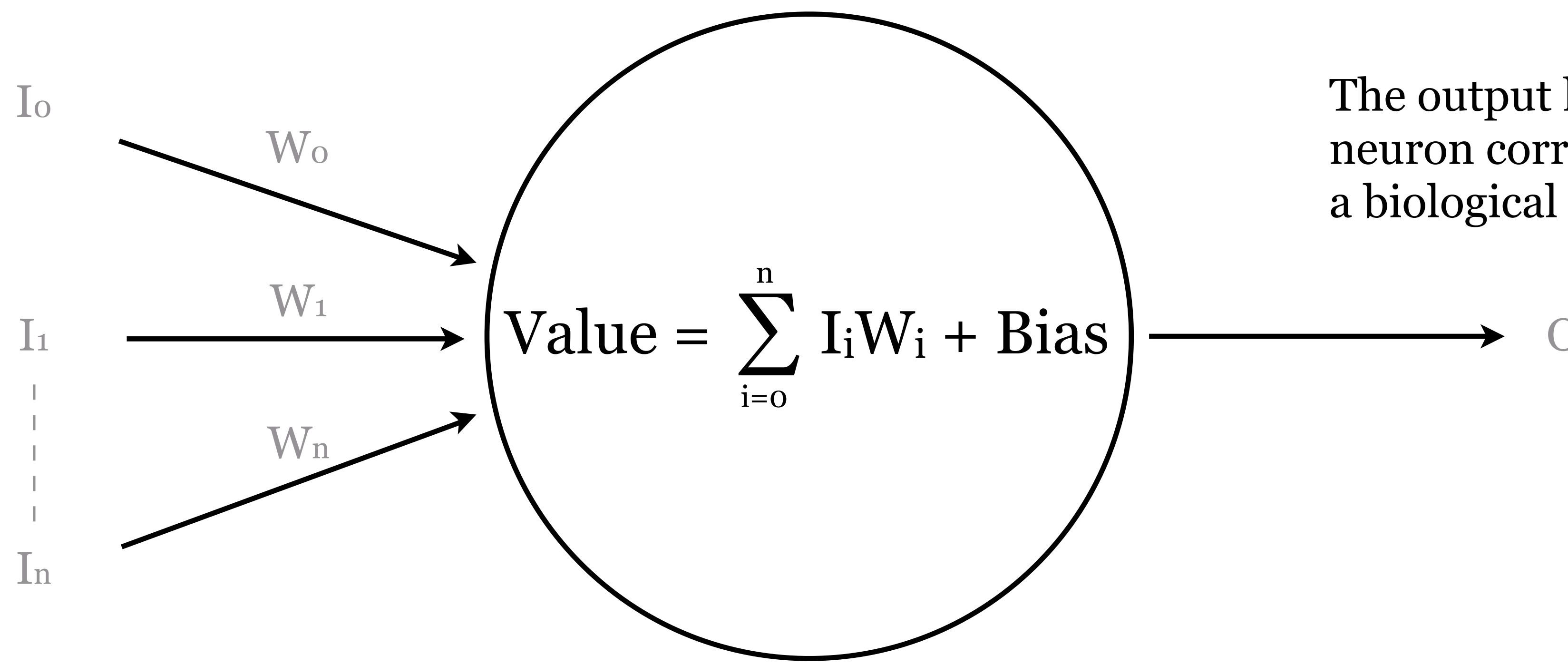
Minsky and **Edmonds** built the first neural network computer (1951)

Frog drawing by Dolores Avendaño,
www.studynow.com/dolores/

Artificial neuron

The bias determines the artificial neuron's inhibitory or excitatory effect.

The input lines I to the neuron correspond to biological dendrites.

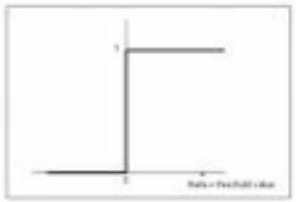


The output line of the neuron corresponds to a biological axon.

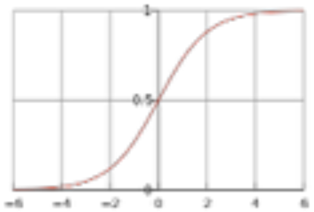
Artificial neuron activation function

Output neurons can emit continuous values or binary signals.

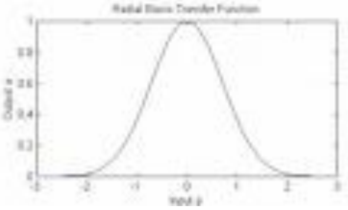
To obtain smooth changes in output, an *activation function* is applied to the output of a neuron. Here are a few examples...



A *step function* can be used to emit one of two values depending on whether or not a threshold is reached.



A *logistic function* is similar, however the transition across the threshold is smooth. This function is very commonly used.



A *Gaussian function* allows a gradual ramping up and down around a central input.

An activation function may also be applied to the input of a neuron. Typically inputs are assumed to have linear activation functions.

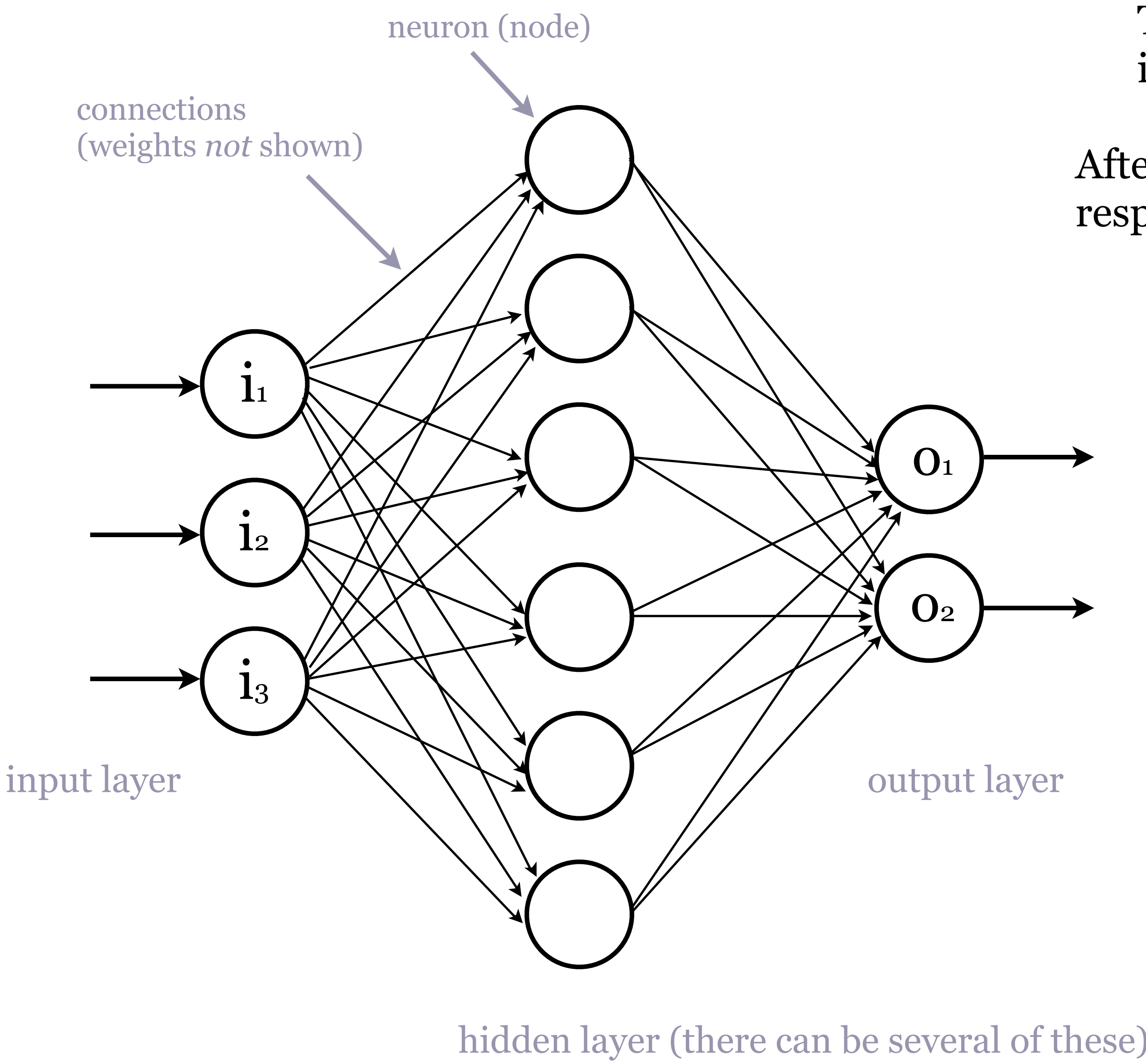
Artificial Neural Network

A set of neurons is connected into a neural network.

The network must be *trained*:

> Test data is fed into the network via its inputs.
>
> The network's responses are read from its outputs.
>
> The connection weights are adjusted after each test to improve the response of the network as desired.

After training, real data is fed into the network and its responses are used to control the behaviour of some system.

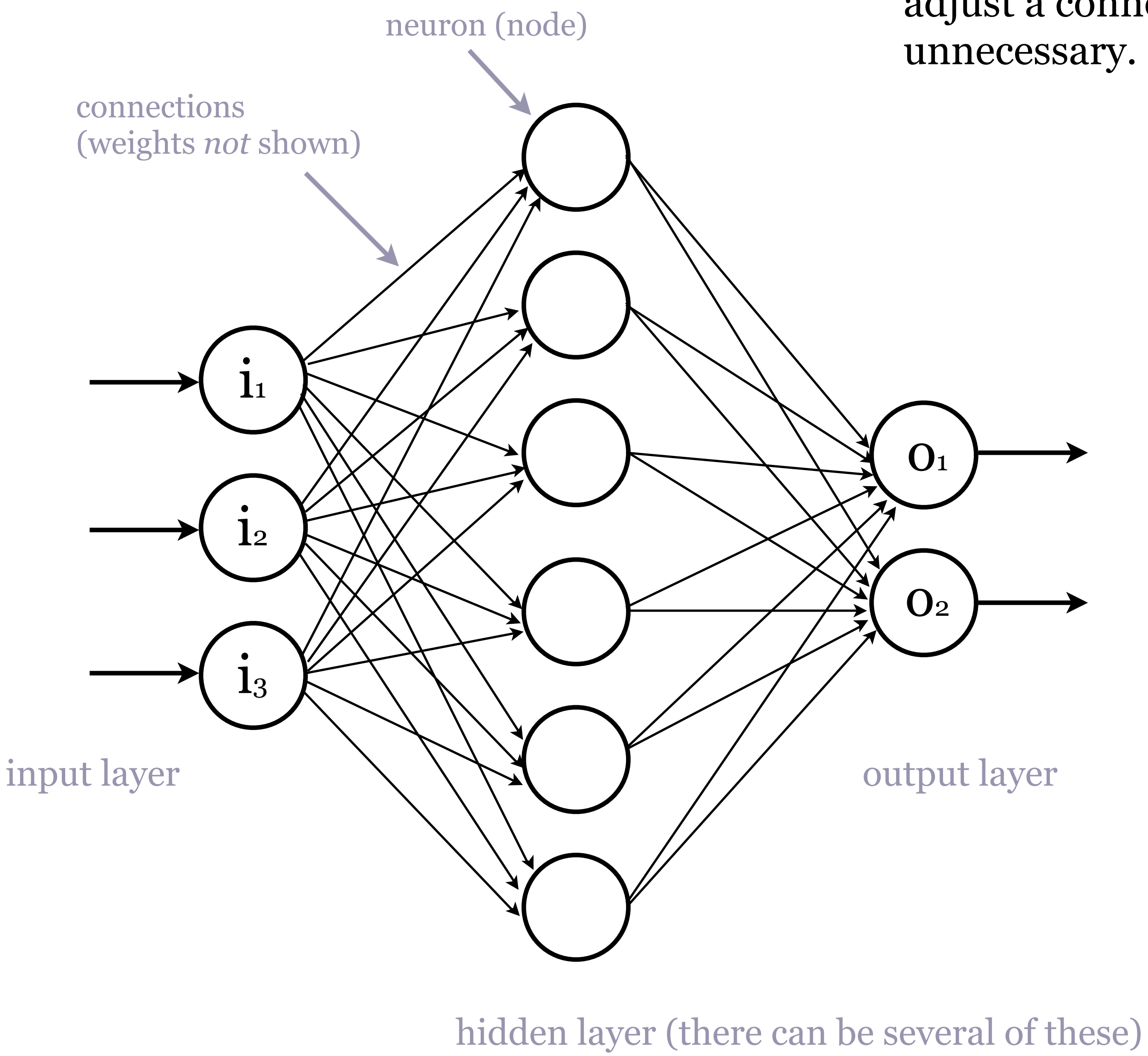


Artificial Neural Network

The illustrated network is *fully connected*.

If some connections are left out, the network is *sparsely* connected.

If a fully connected network is trained properly, it can potentially adjust a connection weight to zero, making sparse networks unnecessary.



Each connection between nodes has a direction.
The illustrated network is *feed-forward*.

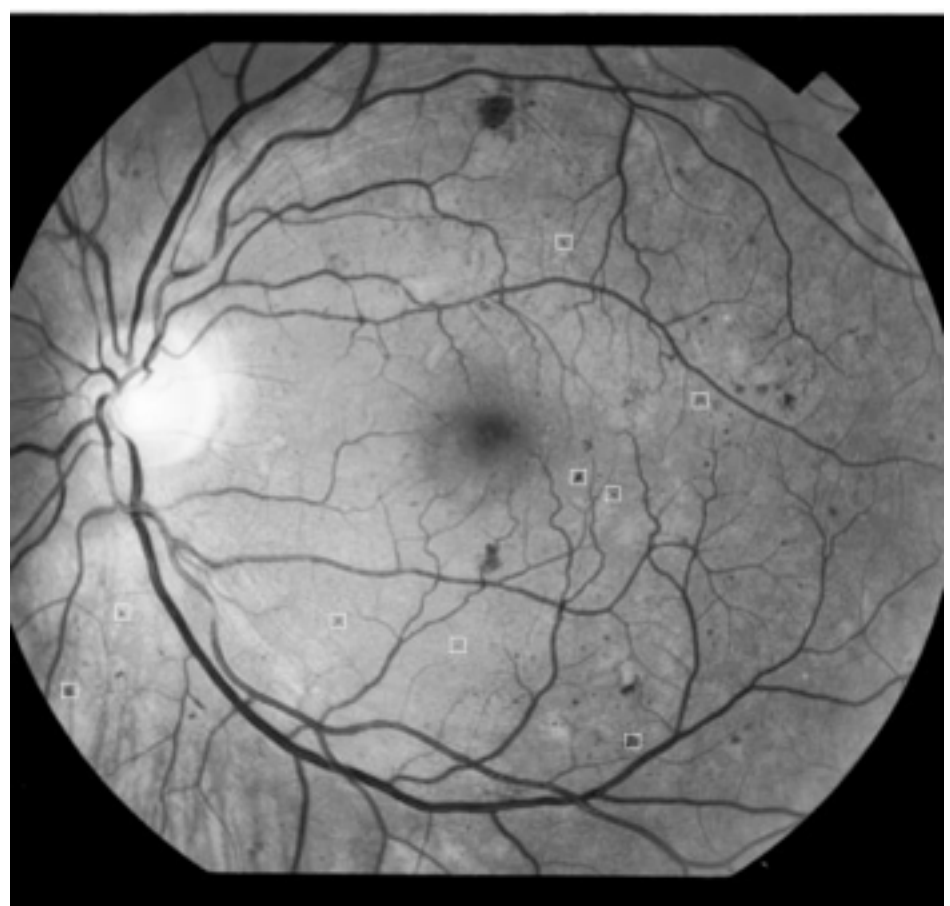
Networks with feedback connections are *recurrent*.

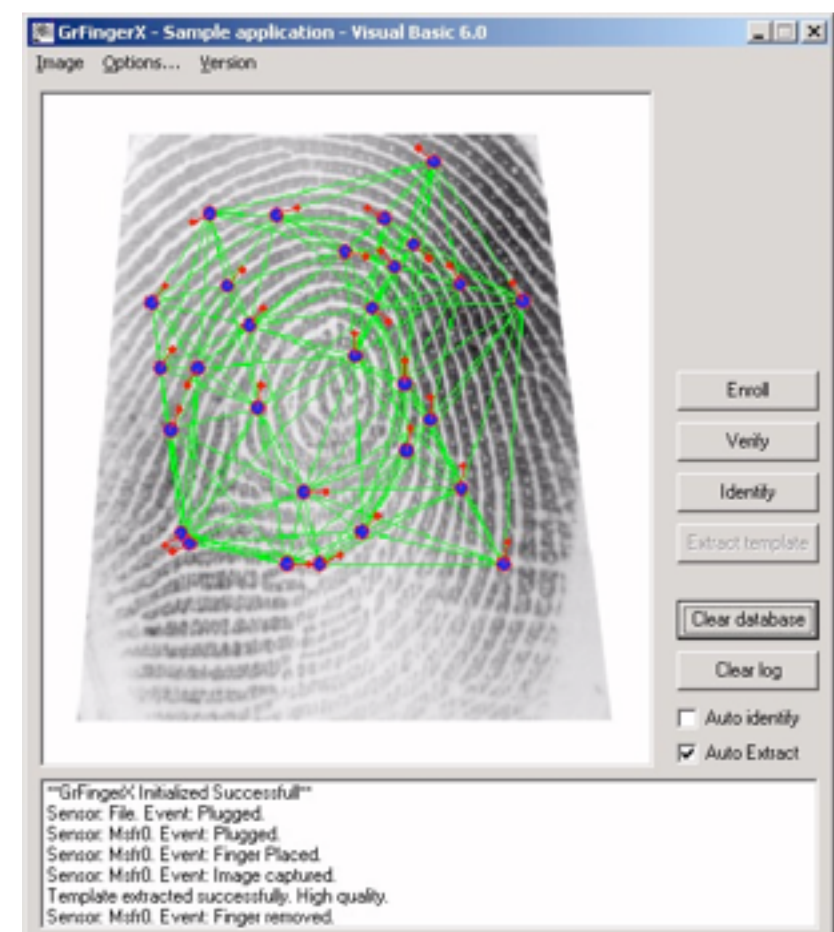
Artificial Neural Network sample applications

ANNs are useful for many kinds of pattern recognition, classification, optimisation, control, function approximation etc.

handwriting e.g. postcodes on envelopes, text entry on touch screens and tablets

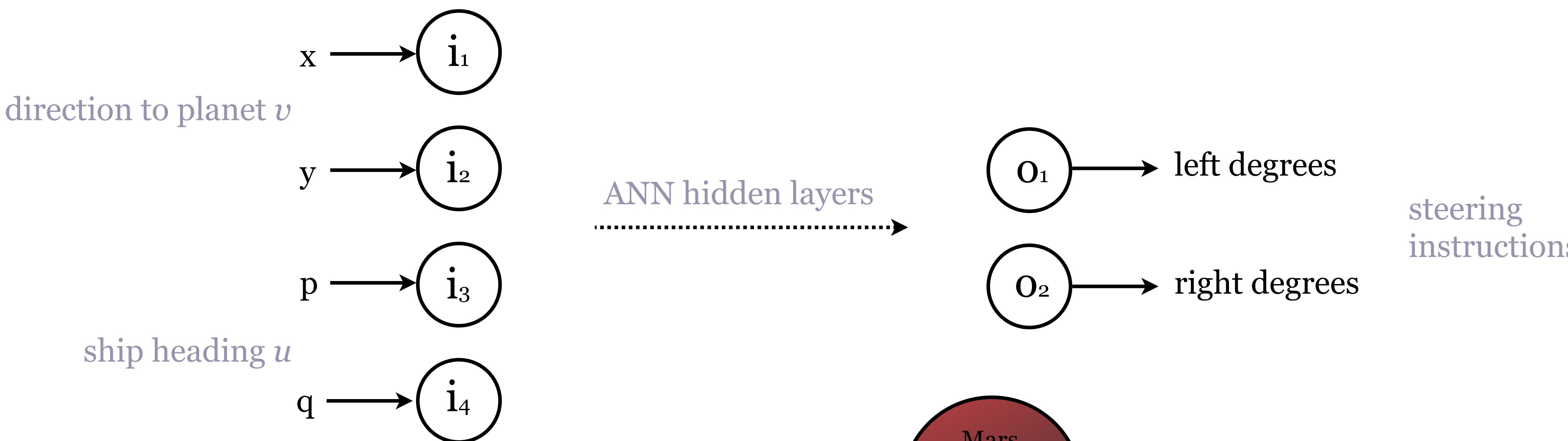
voice recognition, retinal scan, facial and fingerprint recognition



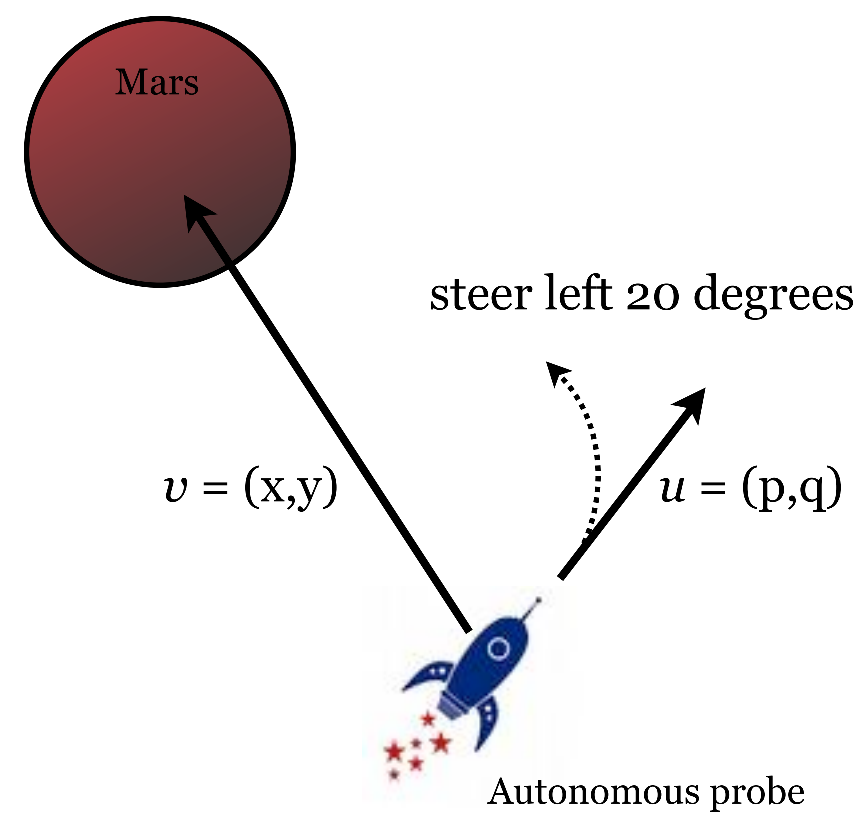


Artificial Neural Network sample game application

The neural network acts as a *mapping* between a set of inputs and a set of outputs.



E.g., Landing a probe on a planet



Artificial Neural Network sample game application

human current state

human player sword position → i_1

human player shield position → i_2

human player crouch level → i_3

human player movement direction → i_4

ANN hidden layers

NPC response

o_1 → defend (shield height)

o_2 → attack (swing sword)

from *Soul Caliber II*, Namco

Artificial Neural Network game applications

In games, ANNs can be used to control NPCs by:

1. pre-training
2. switching off the learning mechanism
3. using the fixed controller during the game

1. pre-training
2. allowing them to learn the player's behaviour during the game
3. using the learning controller during the game

ANN, more game applications

Steering Control

In: obstacles, road or path direction, available exits, location and velocity of neighbours
Out: tank track control; car steering wheel, accelerator and brake; rocket thrusters.

Behaviour determination

In: health, strength, hunger, stimuli (visual, aural…), resource availability
Out: eat, run, turn, swing sword, jump, smile, swim, shoot, collect treasure, build wall etc.

In: opponent ground forces, aerial forces, force activity, positions and movements
Out: move own forces, reinforce defensive lines, mount pre-emptive strike

Human prediction

In: human player position, velocity (*perceptible* current state variables), state history (last few).
Out: human player predicted next state or behaviour.

Using Artificial Neural Networks

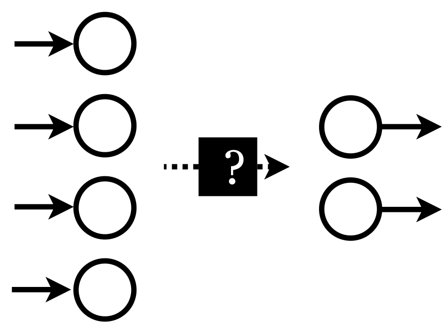
Determine:

- structure
- learning mechanism
- training data

Determining Structure

Question	Frequent answer for game app's
Feed-forward or recurrent network?	Feed-forward.
How many inputs?	As few as possible, only relevant input.
How many hidden layers?	Usually one is enough.
How many nodes in the hidden layer?	It depends! Try 2 x no. inputs? Experiment!
How many outputs?	How many controls do you need?

Determining the Learning Mechanism



Learning involves adjusting the weights of the neural network so that the desired mapping between inputs and outputs is achieved.

Supervised learning

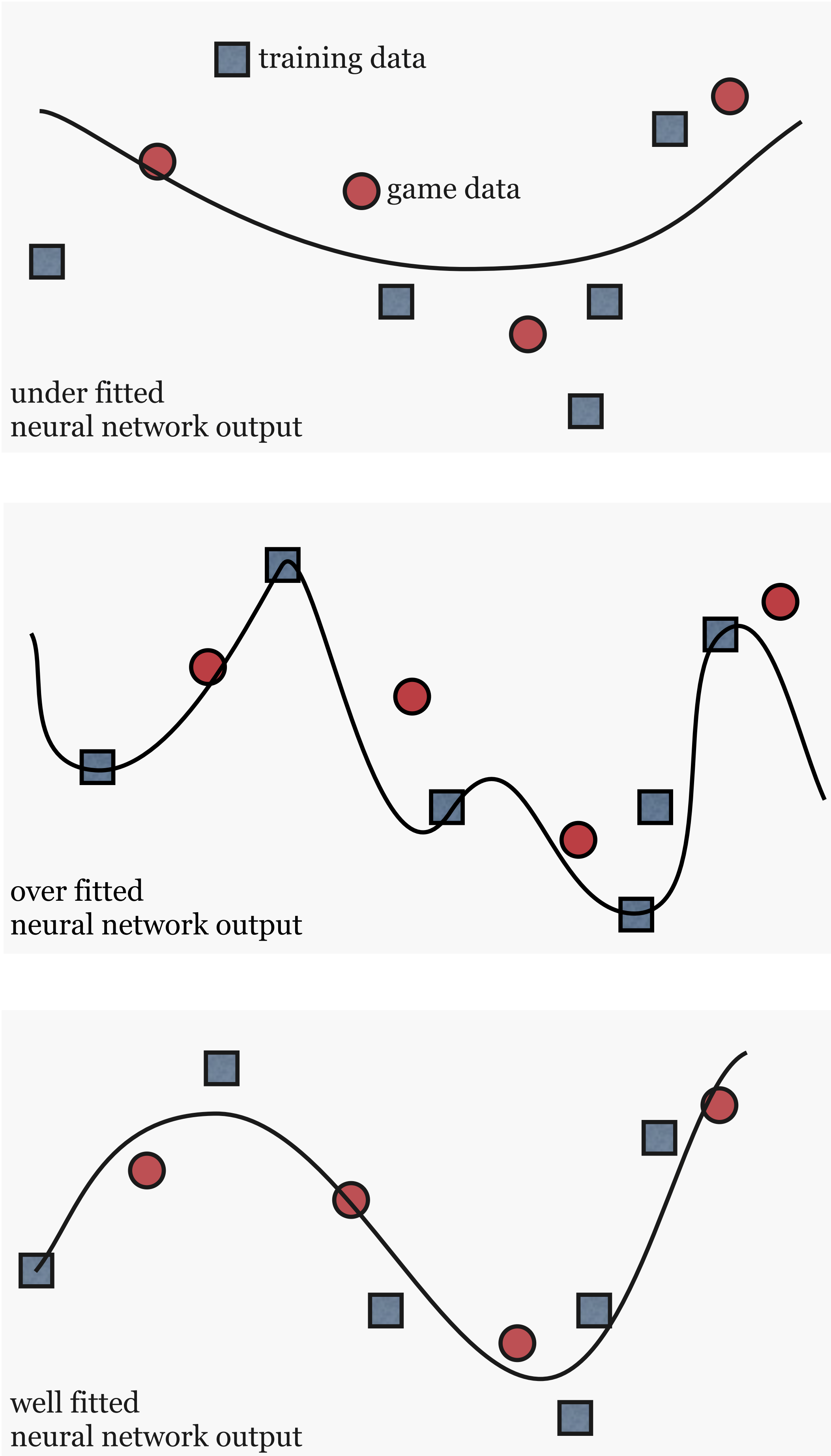
A set of training data is required: a series of pre-constructed input/output pairs.
After each data pair is presented, the error the network produces guides the adjustment of its weights.
Training continues until the error falls within an acceptable tolerance.

Unsupervised learning

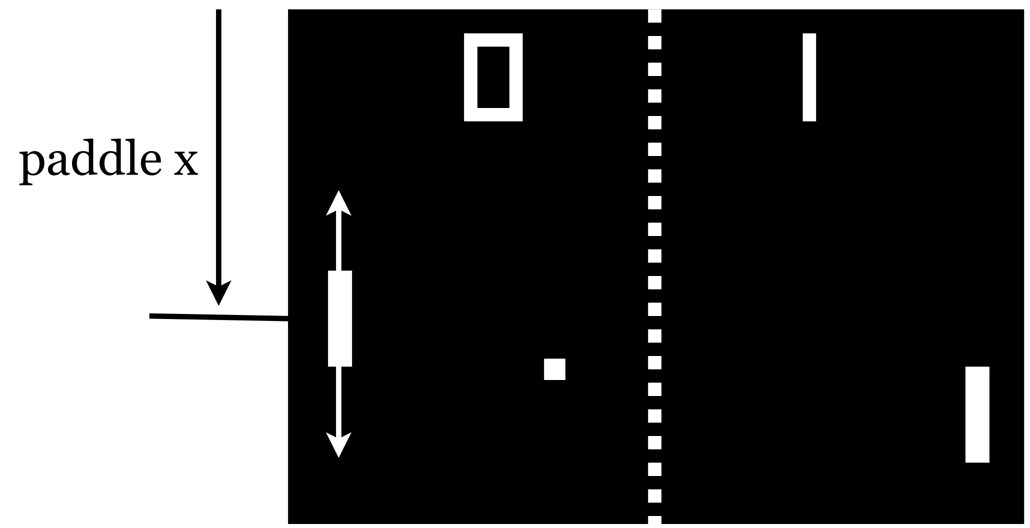
The network attempts the automatic discovery of patterns or features in a set of training data, it may try to optimise some function e.g. its success is gauged by how well it classifies or clusters the input data.

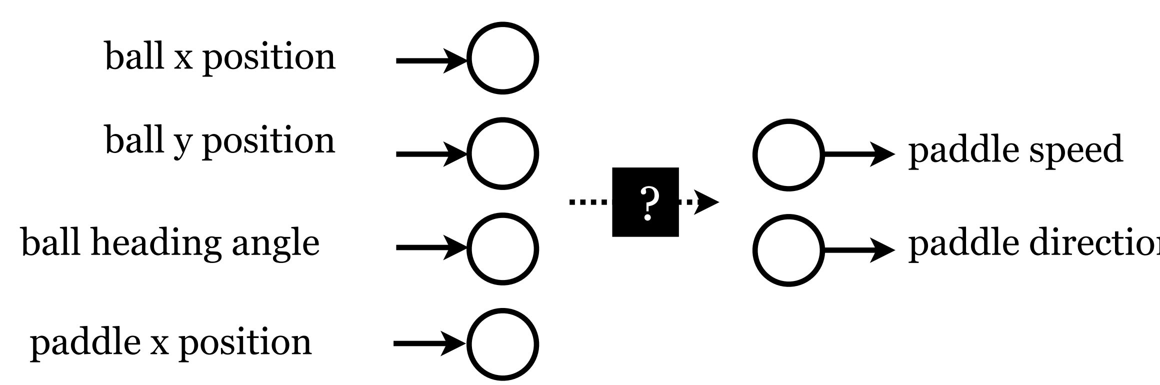
Reinforcement learning

The network is presented with a set of training data typically generated from its interactions in a "real world" / environment. Neurons that respond appropriately (allowing the neural network to approach some long-term goal for instance) are rewarded, whilst neurons that do not are penalised.



ANN Pong controller training





NPC ANN controller training

Where does the training input-data set come from?

- Trial and error automated game play by software
- Prior game play by developers
- Live game play by (an individual or many) human players

Where does the training output-data set come from (if it is required)?

- An objective, analytic measure of success (e.g. distance from target)
- Developers' hand classification of the success of the output

Summary

Biological neurons are the inspiration for artificial neural networks.

ANNs are good at pattern matching and classification.

They may be used in games as NPC controllers for behaviour and response to human player actions.

ANNs require training and can suffer from under or over fitting, a problem which must be avoided, especially if they are trained “live” during real game play.

Extra

Hebbian learning: *Cells that fire together, wire together.*

Increase the weights between cells that fire simultaneously.
Decrease the weights between cells that fire out of synchronisation.

Back-propagation (supervised learning):

```
Initialize the weights in the network (often randomly)

  Do for each example e in the training set
  {
        O = neural-net-output(network, e) ; forward pass
        T = teacher output for e
        Calculate error (T - O) at the output units
        Compute delta_wh for all weights from hidden layer to output layer ; backward pass
        Compute delta_wi for all weights from input layer to hidden layer ; backward pass continued
        Update the weights in the network by delta_wh and delta_wi
  }
  Until all examples classified correctly or stopping criterion satisfied

Return the network
```