Neural Networks: what?

- collection of simple *processing elements (PEs)* which are highly interconnected
- The elements have simple (scalar, or even binary) outputs
- The PE's are modelled on neurons
- the interconnections are modelled on synapses

Neural Networks: Why?

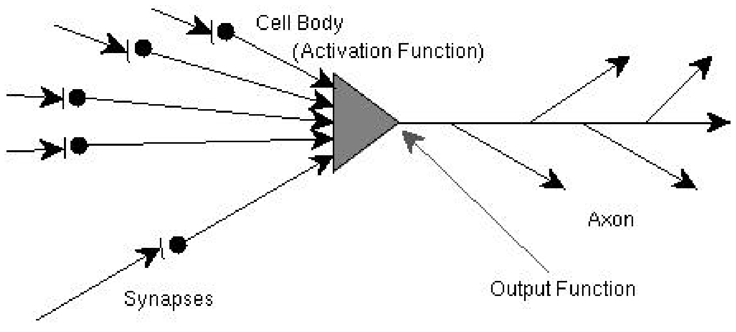
- build machines which would have brain-like capabilities is old:
- indeed, it predates ordinary computers (see, for example, *Frankenstein*, by Mary Shelley).
- Brains are adaptive: and adaptive machines may have major advantages.
- The systems *learn* in response to their *experience*.

Neural Networks: when?

- for problems where precise algorithmic solutions cannot be given,
  - or for which an algorithmic description would be unrealistically complex: e.g. sensory interpretation and many classification tasks.
- Often one can supply many examples of correct behaviour, or of correct classification
  - Controlling complex industrial chemical processes,
  - assessing complex situations automatically,
  - checking signatures,
  - searching fingerprint databases,
  - analysing radar images, interpreting speech,
  - reading handwritten characters…
- if a straightforward algorithm is available, the a programmed solution is almost certainly the right one to take.
- Brief history (handout)

Model Neurons

- Many different model varieties exist
- We look here at the version originally introduced by McCulloch and Pitts.



Input is collected through multiplicative synapses, summed, thresholded, and a binary output for this neuron produced.

Elements of the model neuron: Synapses

- I_i is the input to synapse I
- w_{ji} is the weight characterising the synapse from input i to neuron j
  - I.e. on the dendrite of neuron j (hence order of indices)
- w_{ji} is known as the weight from unit i to unit j

$w_{ji} > 0$ synapse is excitatory

$w_{ji} < 0$ synapse is inhibitory

- Note that I_i may be
  - external input
  - or the output of some other neuron I
    - when we will usually write Y_i.

Elements of the model neuron: Dendrites

- Gathers together (sums) the post-synaptic inputs
  - that is, gathers together the synaptic inputs
- Simplest form is linear summation.
- Writing Aj for the neurons's activation (depolarisation)

  $A_j = \Sigma_i X_{ji} = \Sigma_i w_{ji} Y_i$
- This represents a passive dendrite
  - linear
  - no interaction between the different inputs on the dendrite.

Model neuron output functions

- The output function corresponds to the axon hillock of the real neuron
- Many different output functions have been used
- The model neuron may be a *threshold unit*
  - (as in the original McCulloch-Pits model),
- with threshold *q*

$Y_j = 1$ if $A_j \geq \theta_j$

$Y_j = 0$ if $A_j < \theta_j$

- This is a non-linear operation
- Another simple possibility is a linear unit

$Y_j = k * A_j$

- This gives a purely linear unit

Other output functions

- Logistic (or squashing) unit

$Y_j = 1/(1+\exp(-k * A_j + B_j))$

- or

$Y_j = \tanh(k * A_j + B_j)$

- Note that *k* determines the slope, and B_j the output of the neuron for 0 input.
- Normally, a *monotonically increasing* function is used.

Model Neuron Axon

- In this model, it simply transmits the neurons output to other
  - or even the same
- neurons
- May be system output
- might, in other models, implement a delay

Note:

- Real neurons are a great deal more complex
- Yet for most of the work we will consider, very simple neural models will suffice.
- NN systems gain their power by using a large number of very simple processing elements in concert.
- There is also a great deal of interconnection
  - and this is like the brain: there are miles of axon “wire” in every cubic cm of brain

What can a threshold unit do?

- *Simple threshold units* have output 1 if

  $\Sigma\ w_i I_i > \theta$

  And output 0 otherwise

- N+1 parameters
  - N weights, θ
- Known as a *decision unit*

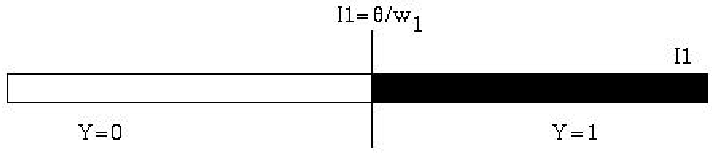
Decision Surfaces

- The decision surface is the surface at which the output of the unit is precisely equal to the threshold.

$$\Sigma w_i I_i = \theta$$

On one side of this surface, the output will be 0,
and on the other side it will be 1.
In 1 dimension: the surface (just a point) is

$I_1 = \theta / w_1$



Decision Surfaces in 2 dimensions

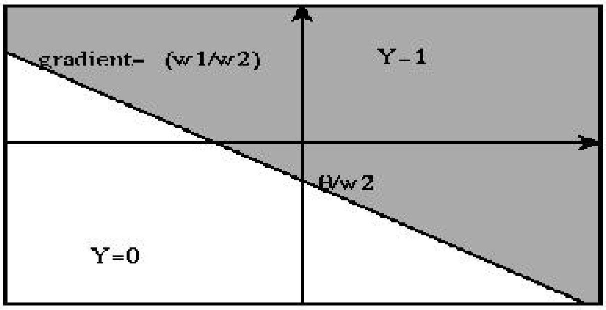
In 2 dimensions, the surface is

$$w_1 I_1 + w_2 I_2 = \theta$$

which we can write

$$I_2 = -\frac{w_1}{w_2} I_1 + \frac{\theta}{w_2}$$

which is the equation of a line of gradient -(w1/w2) with intercept θ/w2



General decision surfaces

- In general, the decision surface is a *hyperplane* of dimension one less than the dimension of the input space.

Input Space	Hyperplane
line	point
surface	line
3-d Volume	surface

- the decision surface cuts the input space into two halves
- If the threshold is 0, then the decision surface passes through the origin
- We can use a bias unit, with a bias weight w_b instead of a threshold

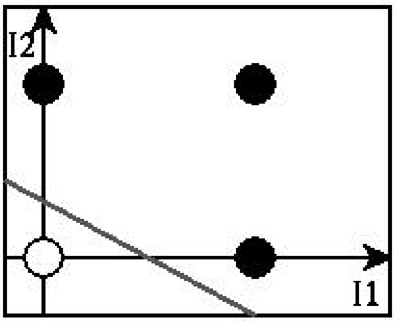
$w_1 I1 + w_2 I2 - w_b.1 = 0$

- Precisely equivalent to a threshold of $-w_b$
  - but makes all the parameters into weights

Decision surfaces for simple logical predicates: AND

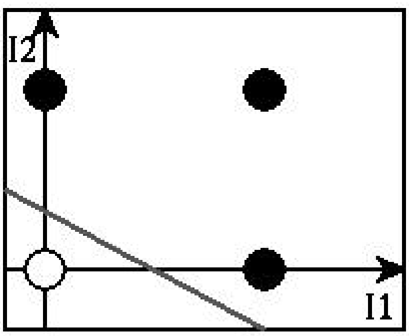
- Consider a 2-neuron system, restricting I1 and I2 to be 0 or 1.
- identify 0 with FALSE and 1 with TRUE, we can consider the logical predicates such as AND and OR

I1	I2	I1 AND I2
0	0	0
0	1	0
1	0	0
1	1	1



Simple logical predicates: OR

I1	I2	I1 OR I2
0	0	0
0	1	1
1	0	1
1	1	1

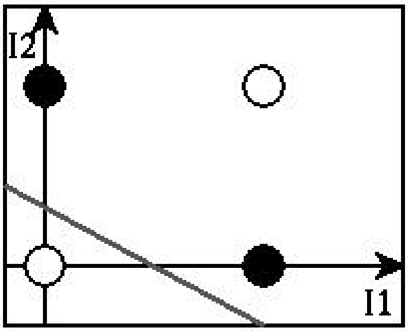


AND and OR are easy to implement: we can see that the gradient of the decision surface line should be negative, and the intercept greater than 1 for AND, and between 0 and 1 for OR.

Simple logical predicates: EOR

- Can we implement Exclusive-Or this way?

I1	I2	I1 EOR I2
0	0	0
0	1	1
1	0	1
1	1	0



EOR presents impossible problems:
no single straight-line decision surface can separate the filled circles from the open circles
EOR cannot be implemented in this way.

 31Y7 - Biologically-inspired Systems Slide 26