1: Computers and Symbols versus Nets and Neurons

Kevin Gurney[1]

Dept. Human Sciences, Brunel University
Uxbridge, Middx.

1 Neural net: A preliminary definition

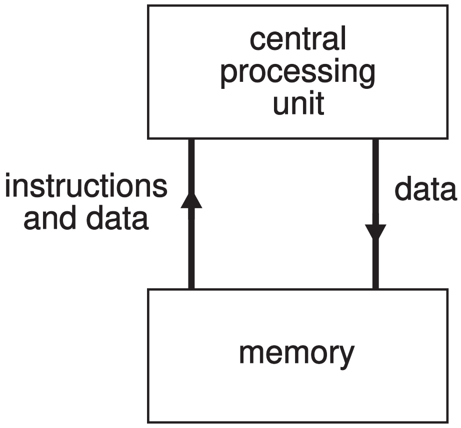
To set the scene it is useful to give a definition of what we mean by 'Neural Net'. However, it is the object of the course to make clear the terms used in this definition and to expand considerably on its content.

> A Neural Network is an interconnected assembly of simple processing elements, *units* or *nodes*, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or *weights*, obtained by a process of adaptation to, or *learning* from, a set of training patterns.

In order to see how very different this is from the processing done by conventional computers it is worth examining the underlying principles that lie at the heart of all such machines.

2 The von Neumann machine and the symbolic paradigm

The operation of all conventional computers may be modelled in the following way



von-Neumann machine

The computer repeatedly performs the following cycle of events

1. fetch an instruction from memory.
2. fetch any data required by the instruction from memory.
3. execute the instruction (process the data).
4. store results in memory.
5. go back to step 1).

What problems can this method solve easily? It is possible to formalise many problems in terms of an *algorithm*, that is as a well defined procedure or recipe which will guarantee the answer. For example, the solution to a set of equations or the way to search for an item in a database. This algorithm may then be broken down into a set of simpler statements which can, in turn, be reduced eventually, to the instructions that the CPU executes.

In particular, it is possible to process strings of symbols which obey the rules of some formal system and which are interpreted (by humans) as 'ideas' or 'concepts'. It was the hope of the AI programme that all knowledge could be formalised in such a way: that is, it could be reduced to the manipulation of symbols according to rules and this manipulation implemented on a von Neumann machine (conventional computer).

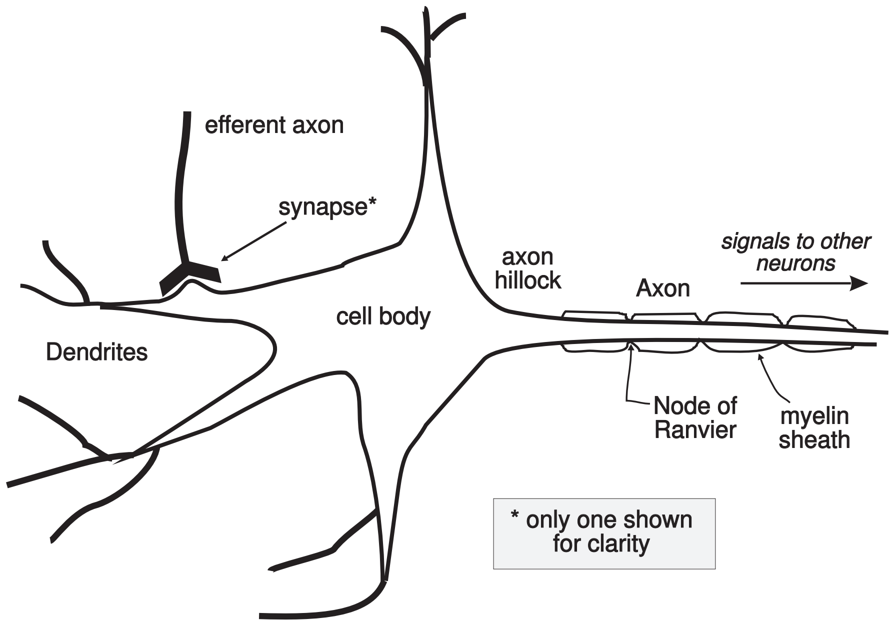
We may draw up a list of the essential characteristics of such machines for comparison with those of networks.

- The machine must be told in advance, and in great detail, the exact series of steps required to perform the algorithm. This series of steps is the *computer program.*
- The type of data it deals with has to be in a precise format - noisy data confuses the machine.
- The hardware is easily degraded - destroy a few key memory locations and the machine will stop functioning or 'crash'.
- There is a clear correspondence between the semantic objects being dealt with (numbers, words, database entries etc) and the machine hardware. Each object can be 'pointed to' in a block of computer memory.

The success of the symbolic approach in AI depends directly on the consequences of the first point above which assumes we can find an algorithm to describe the solution to the problem. It turns out that many everyday tasks we take for granted are difficult to formalise in this way. For example, our visual (or aural) recognition of things in the world; how do we recognise handwritten characters, the particular instances of which, we may never have seen before, or someone's face from an angle we have never encountered? How do we recall whole visual scenes on given some obscure verbal cue? The techniques used in conventional databases are too impoverished to account for the wide diversity of associations we can make.

The way out of these difficulties that shall be explored in this course is that, by copying more closely the physical architecture of the brain, we may emulate brain function more closely.

3 Real neurons: a review

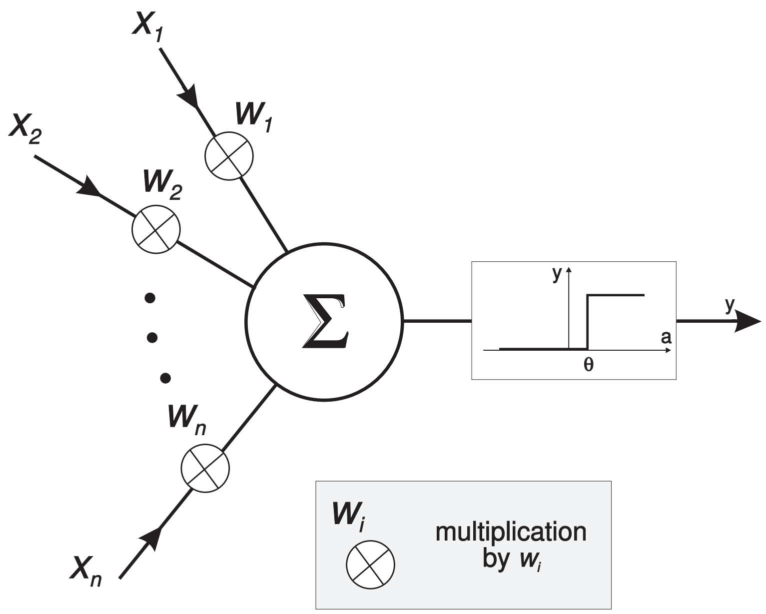


Biological neuron

Signals are transmitted between neurons by electrical pulses (*action-potentials* or 'spike' trains) travelling along the *axon*. These pulses impinge on the afferent neuron at terminals called *synapses*. These are found principally on a set of branching processes emerging from the cell body (*soma*) known as *dendrites*. Each pulse occurring at a synapse initiates the release of a small amount of chemical substance or *neurotransmitter* which travels across the synaptic cleft and which is then received at post-synaptic *receptor* sites on the dendritic side of the synapse. The neurotransmitter becomes bound to molecular sites here which, in turn, initiates a change in the dendritic membrane potential. This *post-synaptic-potential* (PSP) change may serve to increase (*hyperpolarise*) or decrease (*depolarise*) the polarisation of the post-synaptic membrane. In the former case, the PSP tends to inhibit generation of pulses in the afferent neuron, while in the latter, it tends to excite the generation of pulses. The size and type of PSP produced will depend on factors such as the geometry of the synapse and the type of neurotransmitter. Each PSP will travel along its dendrite and spread over the soma, eventually reaching the base of the axon (*axon-hillock*). The afferent neuron sums or integrates the effects of thousands of such PSPs over its dendritic tree and over time. If the integrated potential at the axon-hillock exceeds a threshold, the cell 'fires' and generates an action potential or spike which starts to travel along its axon. This then initiates the whole sequence of events again in neurons contained in the efferent pathway.

4 Artificial neurons: the TLU

The information processing performed in this way may be crudely summarised as follows: signals (action-potentials) appear at the unit's inputs (synapses). The effect (PSP) each signal has may be approximated by multiplying the signal by some number or *weight* to indicate the strength of the synapse. The weighted signals are now summed to produce an overall unit *activation*. If this activation exceeds a certain threshold the unit produces a an output response. This functionality is captured in the artificial neuron known as the Threshold Logic Unit (TLU) originally proposed by McCulloch and Pitts (McCulloch and Pitts, 1943)



TLU

We suppose there are n inputs with signals $x_1, x_2, \ldots x_n$ and weights $w_1, w_2, \ldots w_n$. The signals take on the values '1' or '0' only. That is the signals are *Boolean* valued. (This allows their relation to digital logic circuits to be discussed). The activation a, is given by

$$a = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \tag{1}$$

This may be represented more compactly as

$$a = \sum_{i=1}^{n} w_i x_i \tag{2}$$

the output y is then given by thresholding the activation

$$y = \begin{cases} 1 & \text{if} \quad a \geq \theta \\ 0 & \text{if} \quad a < \theta \end{cases} \tag{3}$$

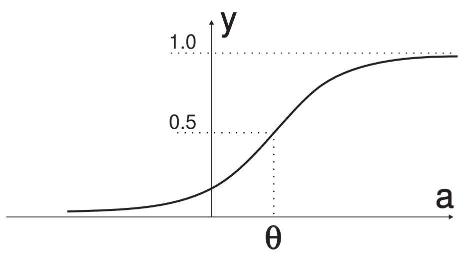
The threshold θ will often be zero. The threshold function is sometimes called a *step-function* or *hard-limiter* for obvious reasons. If we are to push the analogy with real neurons, the presence of an action-potential is denoted by binary '1' and its absence by binary '0'.

Notice that there is no mention of time so far - the unit responds instantaneously to its input whereas real neurons integrate over time as well as space; how this may be this may be remedied will be discussed later.

5 Non-binary signal communication

The signals dealt with so far (for both real and artificial neurons) take on only two values, that is they are *binary* signals. In the case of real neurons the two values are the action-potential voltage and the axon membrane resting potential. For the TLUs these were conveniently labelled '1' and '0' respectively. Now, it is generally accepted that, in real neurons, information is encoded in terms of the *frequency* of firing rather than merely the presence or absence of a pulse. (Phase information may also be important but the nature of this mechanism is less certain and will not be discussed here).

There are two ways we can represent this in our artificial neurons. First, we may extend the signal range to be positive real numbers. This works fine at the input straight away, but the use of a step function limits the output signals to be binary. This may be overcome by 'softening' the step-function to a continuous 'squashing' function



sigmoid

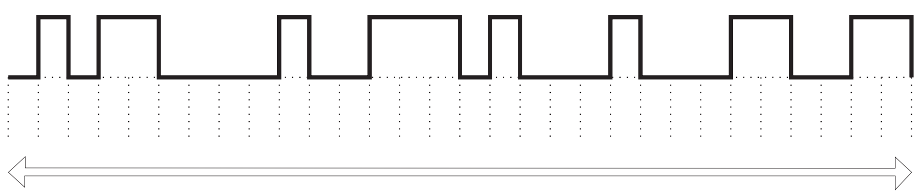
One convenient mathematical form for this is the *sigmoid*

$$y = \sigma(a) \equiv \frac{1}{1 + e^{-a/\rho}} \tag{4}$$

Here, ρ determines the shape of the sigmoid: a larger value making the curve flatter. In many texts, this parameter is omitted so that it is implicitly assigned the value 1. The activation is still given by eqn. (1) but now the output is given by (4). Units with this functionality are sometimes called *semilinear* units. The threshold now corresponds to the activation which gives $y = 0.5$. In (4) the threshold is zero and if we require a non-zero threshold then it must be included by writing

$$y = \sigma(a) \equiv \frac{1}{1 + e^{-(a-\theta)/\rho}} \tag{5}$$

As an alternative to using real (continuous or analogue) signal values, we may emulate the real neuron and encode a signal as the frequency of the occurrence of a '1' in a pulse stream.



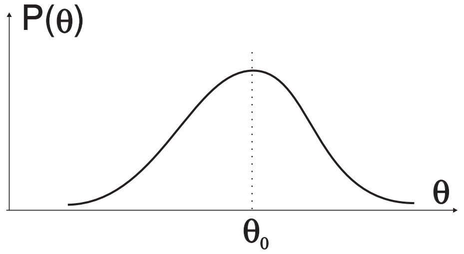
N time slots N₁ '1's or pulses

Probability of a '1' or pulse approx. $\frac{N_1}{N}$

Pulse stream

Time is divided into discrete 'slots'. If the signal level we require is p, where $0 \le p \le 1$, then the probability of a '1' appearing at each time slot will be p (If we require values in some other range then just normalise the signal first to the unit interval). The output y, is now interpreted as the probability of outputting a '1' rather than directly as an analogue signal value. Such units are sometimes known as *stochastic semilinear* units. If we don't know p an estimate may be made by counting the number of '1's, N_1, in N time slots. The probability estimate p^* is given by $p^* = N_1/N$.

In the stochastic case it is possible to reinterpret the sigmoid in a more direct way. First note that it is an approximation to the cumulative gaussian (normal distribution, cf z-scores in statistics). If we had, in fact used the latter then this is equivalent to modelling a 'noisy' threshold; that is the threshold at any time is a random variable with gaussian (normal) distribution.



Normal distribution

Thus, the probability of firing (outputting a '1') if the activation is a, is just the probability that the threshold is less than a, which is just the cumulative of the gaussian up to this value.

6 Introducing time

The artificial neurons discussed so far all evaluate their activation and output 'instantaneously' - there is no integration of signals over time. To introduce dependence on time, we define the activation implicitly by its rate of change da/dt. This requires that the weighted sum of inputs be denoted by some other quantity. Thus, put

$$s = \sum_{i=1}^{n} w_i x_i \tag{6}$$

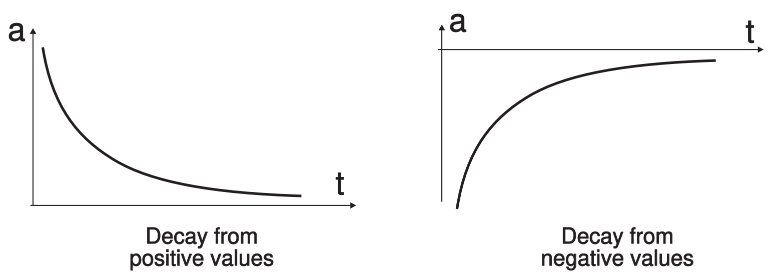
and now put

$$\frac{da}{dt} = -\alpha a + \beta s \tag{7}$$

where α and β are positive constants. The first term gives rise to activation *decay*, while the second represents input from the other neurons and may be excitatory. To see the effect of the decay term put $s = 0$. There are now two cases.

i) $a > 0$. Then $da/dt < 0$; that is, a decreases.

ii) $a < 0$. Then $da/dt > 0$; that is, a increases



Decay of activation

The neuron will reach equilibrium when $da/dt = 0$. That is when

$$a = \frac{\beta s}{a} \tag{8}$$

There are several possible modifications to (7). e.g., 'PDP' vol 3 ch 2, and (von der Malsburg, 1973; Hopfield and Tank, 1986). Stephen Grossberg has made extensive use of this type of functionality - see for example (Grossberg, 1976).

7 Network features

Having looked at the component processors of artificial neural nets, it is time to compare the features of such nets with the von Neumann paradigm. The validity of some of these properties should become clearer as the operation of specific nets is described.

- Clearly the style of processing is completely different - it is more akin to signal processing than symbol processing. The combining of signals and producing new ones is to be contrasted with the execution of instructions stored in a memory.
- Information is stored in a set of weights rather than a program. The weights are supposed to adapt when the net is shown examples from a training set.
- Nets are robust in the presence of noise: small changes in an input signal will not drastically affect a node's output.
- Nets are robust in the presence of hardware failure: a change in a weight may only affect the output for a few of the possible input patterns.
- High level concepts will be represented as a pattern of activity across many nodes rather than as the contents of a small portion of computer memory.
- The net can deal with 'unseen' patterns and generalise from the training set.
- Nets are good at 'perceptual' tasks and associative recall. These are just the tasks that the symbolic approach has difficulties with.

References

Grossberg, S. (1976). Adaptive pattern classification and universal recoding: 1. parallel development and coding of neural feature detectors. *Biological Cybernetics*, 23:121 – 134.

Hopfield, J. J. and Tank, D. W. (1986). Computing with neural circuits: A model. *Science*, 233:625 – 633.

McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 7:115 – 133.

von der Malsburg, C. (1973). Self-organisation of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85 – 100.