CHAPTER 3

# Mathematical Models of Neurones

## 3.1. Logical Neurones

### 3.1.1. Introduction and Definition

Probably the best-known "mathematical neurone" is the logical or McCulloch-Pitts neurone (McCulloch and Pitts, 1943). It may be represented as shown in Fig. 3.1 and in its simplest possible form is a device which gives an output (to the right) if it gets an input from at least a certain number, say $\theta$, of its inputs (on the left). It thus has a threshold $\theta$ which is, again in the simplest version of the model, a constant positive integer characteristic of the "neurone". $\theta$ is often written into the diagram of the neurone as in Fig. 3.1.

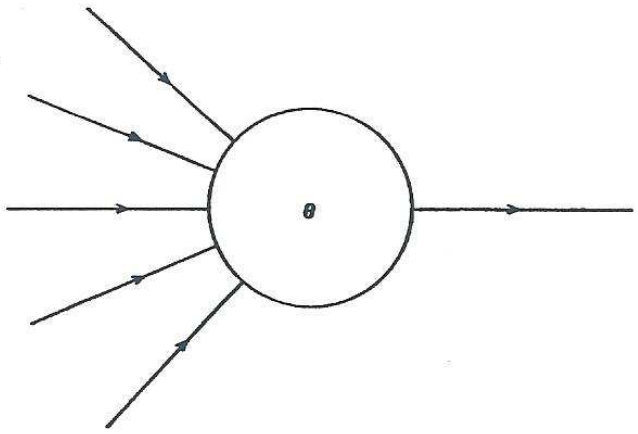



Fig. 3.1. Representation of logical or McCulloch-Pitts neurone.

The logical neurone purports to be an idealization of a real neurone and has the features of being able to be excited by its inputs (we include inhibition in a moment) and of giving an output when a threshold is exceeded. Its most peculiar feature is the way in which its behavior is a function of time. It is supposed that the neurone can only change its state at one of a discrete series of equally spaced times. Thus if one of these possible times is $t_0$ and the next is $t_0+\tau$, then the subsequent ones are $t_0+2\tau$, $t_0+3\tau$, .... Th

output from a given neurone at time $t = t_0+p\tau$ arrives as an input to all those to which it is linked, at the next time for change, i.e. at $t = t_0+(p+1)\tau$. A network of logical neurones behaves in a synchronized fashion, $t_0$ and $\tau$ being the same for every neurone in it.

Biologists often criticize the logical neurone for being too unrealistic, especially in relation to its time dependence. It is important for us to realize that this is rather unfair. The great advantage of the logical neurone is its simplicity, which often enables us easily to gain an insight into how a network of nerve cells might be expected to behave. It has many realistic features such as threshold, excitability, spatial summation and all-or-none output, but to gain simplicity we have to pay the price of lack of realism in some respects. This is a normal feature in the application of mathematics to the real world, namely that we often deliberately simplify in order to achieve mathematical tractability, whilst always recognizing the danger that such simplification can lead to spurious results.

After these introductory remarks, we now give a more formal abstract definition of a general logical neurone, under seven headings:

1. A logical neurone can exist in one of two states, which may be called "active" and "inactive".

2. It has one output, which can be connected simultaneously by one or more links to each of an arbitrary number of other logical neurones or to itself. This means it gives the same output along every link.

3. It has a total of $n_e+n_i$ inputs, $n_e$ of which are "excitatory" and $n_i$ of which are "inhibitory". $n_e$ and $n_i$ can each take any non-negative integral value.

4. It has a threshold $\theta$, which is normally taken to be a positive integer, although it could take other real values.

5. The neurone can only change its state at a discrete sequence of times $t = t_0+p\tau$, where $p$ can take any integral value, and we often take $t_0 = 0$, $\tau = 1$. Each neurone keeps its state unchanged during each time interval $t_0+p\tau \leqslant t < t_0+(p+1)\tau$, where $t_0$ and $\tau$ are constants which are the same for every neurone of a given network. This is the assumption of quantized time, which is the most unpalatable feature of the logical neurone.

6. A particular input is active at time $t_0+(p+1)\tau$ if the neurone from which it comes was active at time $t_0+p\tau$ (we shall also then say that that neurone fired at time $t_0+p\tau$). We write $N_e$ for the number of excitatory inputs which are active and $N_i$ for the number of inhibitory ones which are. Evidently $N_e \leqslant n_e$ and $N_i \leqslant n_i$. $N_e$ and $N_i$ are, of course, functions of time and this may be made explicit if necessary.

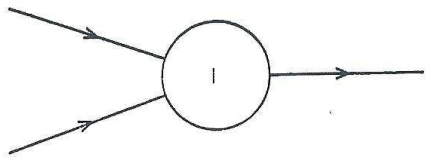
7. A neurone is active at time $t_0+(p+1)\tau$ if and only if $N_e-\phi N_i \geqslant \theta$ at that time. $\phi$ is a positive real number characteristic of the neurone and will usually be taken to be an integer. Like $\theta$, $n_e$ and $n_i$, $\phi$ may differ from one neurone to another, although it will often be interesting to investigate networks in which they have the same values for each neurone. Note that a more general rule of the type $\phi_1 N_e-\phi_2 N_i \geqslant \theta_1$ can be got into the simpler form given above by dividing through by $\phi_1$, and writing $\phi = \phi_2/\phi_1$, $\theta = \theta_1/\phi_1$.

A logical neurone is a binary device, because it has two possible states. It is often convenient to represent its state in binary arithmetic notation, saying it is in the state 0 when it is inactive and in the state 1 when active. If we have a network of $n$ neurones we can then number those neurones from 1 to $n$ and represent the state of the whole network at a given time by a binary integer. Thus if a network contains just three neurones, the binary integer 101 signifies that neurones one and three are active, while neurone two is inactive. Evidently, at any given time, a network of $n$ neurones has $2^n$ possible states.

### 3.1.2. Examples

We now consider a few simple illustrative examples, using the symbol $\rightarrow$ for an excitatory input and $\dashrightarrow$ for an inhibitory one.

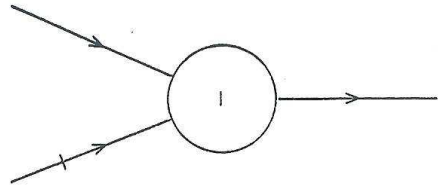
1. One neurone having $n_e = 2$, $n_i = 0$, $\theta = 1$. Because $n_i = 0$ it is unnecessary to specify $\phi$. The diagram is



and the neurone fires if $N_e \geqslant 1$, i.e. if one input or the other or both are active.

It is interesting to note that, had we set $\theta = 0$, we should have got an output even if there were no input. Such a neurone could be called "spontaneously" active, although this would differ somewhat from the more common usage of the word "spontaneous" given in Section 4.1.

2. One neurone having $n_e = n_i = 1$, $\theta = \phi = 1$. The diagram is

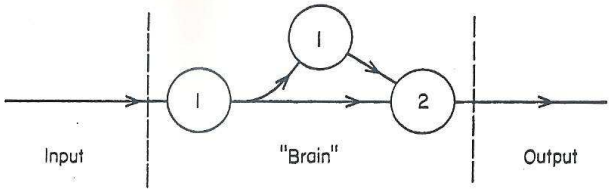


There are four possible input situations which can be tabulated most conveniently as follows:

| $N_e$ | $N_i$ | $N_e-\phi N_i$ |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | −1 |
| 0 | 0 | 0 |

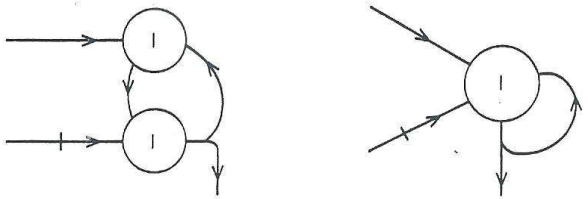
We see that the threshold condition $N_e-\phi N_i \geqslant \theta$ is only satisfied in one case, namely when the excitatory input is active but the inhibitory one is inactive. It also follows from the table that, if we set $\theta = 0$, the neurone is spontaneously active but could be turned off if $N_e = 0$ and $N_i = 1$.

3. Three neurones, all excitatory links, i.e. all $n_i = 0$.




It is easy to see that we only get an output to the right after there have been two inputs successively on the left. One may thus think of the network as a very primitive "brain" which only reacts to repeated stimuli, but not to temporally isolated ones.

4. Self-re-exciting systems.

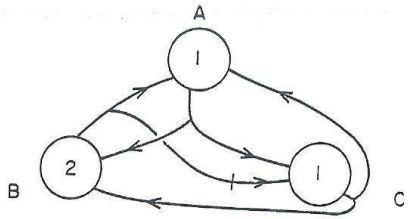



In each of these cases a single input at the top left corner continues to go round and round for ever unless it is "erased" by sending in an inhibitory



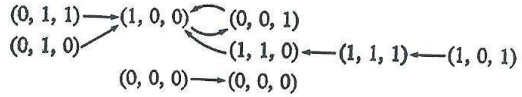
input from the bottom left-hand side. The output from such systems can be changed indefinitely by a single input at one time, which may be far in the past. They could thus serve as primitive memories. It has been suggested (see Section 6.5.1) that human and animal memory might be based on this self-re-exciting potentiality of neural networks but it is generally considered that this is unlikely for long-term memory.

5. Another network of three neurones.




We now consider an isolated network of three neurones and ask what happens subsequently if the network starts in a particular state. Treating this as a purely mathematical problem, it is unnecessary to ask how the network is put into its initial state. However, it is evident that any initial state could be achieved by suitable inputs from the outside to the three cells.

It is convenient to use binary notation for states of individual neurones and a vector notation for a state $S$ of the network, thus $S$ = (state of A, state of B, state of C). There are eight ($= 2\times2\times2$) possible states $S$ and we shall have completely characterized the behavior of the network when we have tried each of these as initial states. We then get the following diagram




which the reader may easily verify. It shows that if we start the network in any state except the completely inactive one, it settles down to an oscillatory activity in which B is inactive but excitation shuttles between A and C.

This concludes our examples which were given to clarify the nature of the logical neurone. It should be emphasized that in a real brain we are concerned with so many cells (up to $10^{10}$), each having perhaps $10^4$ or more inputs, that we cannot expect to analyze its activity in the detailed way

that we have in example 5 above. The sort of approach we must adopt then will be discussed in Chapters 5 and 8.

### 3.1.3. Connection with Real Neurones

We have remarked already that many features of real neurones are well represented in the logical neurone, but that the quantization of time is not realistic. Nevertheless, we should like to have some idea of what value and significance to assign to the time interval $\tau$. In my opinion, $\tau$ should be regarded as an average estimate of the time taken between the firing of one presynaptic cell and the time when the effect of that firing first has the potential of influencing the firing of those postsynaptic cells to which the first cell is linked. Then $\tau$ is made up of four components, each of which would probably be a small multiple or submultiple of 1 msec. The first, $\tau_1$, is the time for the action potential to travel down the axon to the synaptic knobs (if the velocity of conduction is $v$ meters/sec and the axonal length is $l$ mm, then $\tau_1 = l/v$ msec). There is then a slight delay $\tau_2$, termed the synaptic delay, before the postsynaptic potential (PSP) starts to appear (Eccles, 1964, p. 42 gives some values of $\tau_2$ as 0.2–2 msec). The PSP then appears, giving $\tau_3$. Whether $\tau_3$ should be the time to the peak of the PSP or until it has, say, half decayed from its peak, is a little unclear; probably the latter. Anyway $\tau_3$ again is probably usually a few msec (see Eccles *loc. cit.*). Finally, $\tau_4$ is the time of rise of an action potential from the threshold to its peak. $\tau_4$ is probably typically less than 1 msec. Thus we should think of $\tau = \tau_1+\tau_2+\tau_3+\tau_4$ as being a few msec, probably usually less than 10 msec.

Although this gives a way of assigning a value to $\tau$, it does not make the quantization acceptable. It requires that we should only allow a cell to fire at the times $t_0+p\tau$. The arbitrariness of the choice of $t_0$ is immediately apparent: since there is normally no synchronization in the real system, there is no reason to prefer any one value of $t_0$ over another.

We shall not pursue this undoubted defect of the logical neurone any further here, but merely remind the reader that these "neurones" are useful in theoretical discussions because of their relative simplicity, and mention two other points. The first is that we have not explicitly considered the question of refractoriness (Section 2.2.4). The refractory period would normally be less than $\tau$ and so the very fact that a logical neurone cannot fire twice in less than $\tau$ seconds is sufficient to deal with it. Were that not the case, one could introduce into the definition of the logical neurone the requirement that it could not fire again until at least $x\tau$ seconds after it last did so, for some fixed integer $x > 1$. The second is that habituation can easily be incorporated into the definition by imposing some restriction on the number of times a logical neurone can fire in a given period. For example,



one could require that it cannot fire at time $t_0+p\tau$ if it has fired more than $a$ times in the time interval $t_0+(p-b)\tau \leqslant t < t_0+p\tau$, where $a$ and $b$ are fixed integers (note that the refractory period is given by the special case $a = 0, b = x-1$).

## 3.2. Real Time Neurones

### 3.2.1. Basic Definition

From a functional point of view we know a very great deal about the activity of a neurone when we have a record of its internal electrostatic potential and of how this has been altered by EPSP's and IPSP's due to activity in attached cells. This suggests that we may define a much more realistic mathematical neurone by concentrating attention on this internal potential as a measure of the state of a neurone, which is thus characterized by a parameter $V$ (Gluss, 1967; Griffith, 1967a). It is most convenient to choose $V$ so that its physical significance would be the deviation of the internal potential from its resting value of around $-70$ mV. So the resting potential corresponds to $V = 0$.

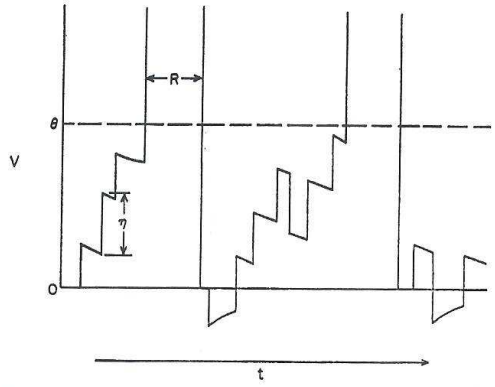



Fig. 3.2. Time course of $V$ for a "real time" neurone with reset to zero assumption. (Compare with Fig. 2.3)

Three things can happen to $V$ (see Fig. 3.2 for illustration).

1. If a presynaptically attached, i.e. input, cell fires at time $t$, then at time $t+\delta$ we alter $V$ to $V+\eta$. $\delta$ represents the delay between the time the input cell reaches its threshold to firing and the peak of the resulting EPSP or IPSP. $\delta$ is the same as $\tau$ as defined in Section 3.1.3, providing $\tau_3$ is taken as some-

where between zero and the time to the peak of the PSP. $\eta$ corresponds to the height of the EPSP, in which case it is positive, or the IPSP, in which case it is negative.

2. At any time $t'$ at which $V$ changes from being $< \theta$ to being $\geqslant \theta$, we say the cell fires. $\theta$ is called the threshold. For $t' < t < t'+R$, we set $V(t) = \infty$ (or some large value $> \theta$). We then put $V(t'+R) = 0$. $R$ is called the refractory period (it corresponds to the absolute refractory period of a real neurone). Both $R$ and $\theta$ are fixed constants characteristic of the cell.

3. At all times not covered by rules 1 and 2, $V$ satisfies the differential equation

$$\frac{dV}{dt} = -\varepsilon V. \tag{1}$$

$\varepsilon$ is a fixed positive constant, characteristic of the cell, and corresponds to an average estimate of the time constant of decay of isolated EPSP's or IPSP's. $\varepsilon^{-1}$ is probably of the order of a few msec.

#### 3.2.2. Remarks

There is no reason why $\theta$, $R$ and $\varepsilon$ should not differ from one neurone to another nor why $\delta$ and $\eta$ should not alter from one synapse to another.

Although it is natural to interpret $V$ as being related to the internal potential, there is another interpretation which might be more correct for some neurones. We remarked earlier (Section 2.2.4) that the time course of a PSP is a combination of the effects of the discharge of the membrane capacity and of the enzymatic destruction of the transmitter. In the limiting case of a very short electrical time constant and long life of transmitter, the PSP would strictly follow the transmitter concentration on the post-synaptic membrane. $V$ could then be interpreted as "total transmitter level" at all input synapses to the cell, counting transmitter at excitatory synapses as "positive" and at inhibitory ones as "negative". Since there is no reason to expect the internal potential to follow the transmitter level during an action potential, this would mean that we could consider replacing rule 2 above with the following extreme alternative: at any time $t'$ at which $V(t') \geqslant \theta$ we say the cell fires, providing it has not fired previously in the period $t'-R < t < t'$. In that case $V(t)$ would satisfy the equation (1) at all times not covered by rule 1 alone.

We thus have two possible limiting versions of the model, one in which $V$ is reset to zero at time $R$ after the cell fires and the other in which $V$ is unaffected by the actual firing of the cell. It is probable that the first is usually nearer to the truth (see Eccles, 1964, Section 4B) and in its strict form implies that the value of $V$ after a firing is totally unaffected by the extent to which $V$ exceeds the threshold $\theta$ at the time of firing. This is obviously not so with the second version, with which it is obviously possible in principle for $V$ to become so large that, in the absence of further input, it would cause the cell to fire twice or more successively. Using equation (1), this would happen if $V$ were suddenly raised to a value $V_0$ satisfying $V_0 \geqslant \theta e^{\varepsilon R}$. It is probable that the repetitive response of certain cells in the spinal cord (Renshaw cells, see Eccles, 1964, pp. 119–121) is largely due to this kind of mechanism.

Habituation as an effect on the threshold of a cell may be conveniently introduced by allowing $\theta$ to vary with time according to the following rules:

1. When the cell fires alter $\theta$ to $\theta+h$.
2. At all other times $\theta$ satisfies

$$\frac{d\theta}{dt} = -\beta(\theta-\theta_0). \tag{2}$$

$h$, $\beta$ and $\theta_0$ are positive constants characteristic of the cell.

### 3.3. Computer Simulations

#### 3.3.1. Logical Neurones

Many workers have simulated neural networks on a digital computer (see, e.g. Farley and Clark, 1961, and Harmon and Lewis, 1966). We shall not discuss detailed programming problems here but draw attention to a few numerical points relating to the space and time requirements for such simulations.

With a network of $n$ neurones we have the following space requirements:

1. *Values of $\theta$ and $\phi$ for each neurone.* For a general network we need to store $2n$ numbers but, if we assume all neurones have the same values for these parameters, we need only store 2.

2. *Connectivity of the network.* For each neurone we must say how many links it has to each other neurone and whether they are excitatory or inhibitory (by putting + or − in front of the number). This requires up to $n^2$ numbers. Alternatively we could give a rule to determine whether neurone $x$ is linked to neurone $y$ or not. For example, we could say that if the remainder after dividing $xy$ by $n$ lies between 1 and 10 we have one excitatory link from neurone $x$ to neurone $y$ but otherwise there are no links. Such rules are easy to program, require a space in the store which is essentially independent of $n$, and can be useful in setting up pseudo-random networks to simulate the random networks discussed in Chapter 5.

3. *Present and immediately preceding state of network.* The present state is always calculated from the preceding state, and the latter must generally be stored until the present one has been completed. This means storing $2n$ numbers, each being 0 or 1.

Thus as far as space requirements are concerned, a computer can simulate a network containing about as many neurones as it can store numbers. In other words, even $10^{10}$ is not out of range providing the connectivity is largely specified by giving rules rather than actually enumerating all the links (which for the human brain would still only need $10^{14}$ or so numbers, rather than $n^2 = 10^{20}$).

In considering time requirements, the important time is the time $T$ required to calculate the new state of the network from the preceding one. If each neurone is linked to $q$ others then this is approximately given by $T \approx n(qt_1 + t_2)$, where $t_1$ is the time required to calculate the contribution of a given link to the sum $N_e - \phi N_i$ and $t_2$ is the time to see whether the resulting sum is $\geqslant \theta$. As a consequence, the time requirements impose a much more serious restriction on the size of network which can be examined than do the space requirements. For example, if $qt_1 + t_2 = 100$ μsec and $n = 10^4$, $10^6$ or $10^{10}$, $T = 1$ sec, 1 min 40 sec or $11\frac{1}{2}$ days respectively.

Thus, even if we knew all the necessary parameters, putting a McCulloch-Pitts version of the human brain on a computer would pose more of a problem of speed than of storage space. In view of the great speed of a modern computer, this must appear something of a paradox. In fact it is resolved by realizing that the normal version of digital computer is ill-designed for handling this sort of simulation because its central arithmetic unit can only operate at one time, albeit very fast, on a very small amount of the data in its store. If we knew enough to simulate the human brain, we would use separate electronic circuits for each neurone (quite realistic analog circuits even for neurones operating continuously in time have been described, see Harmon, 1959, 1961) and would have these circuits working simultaneously in parallel. Thus the time problem would be removed and, in fact, the artificial brain could be built to operate faster than the real one because the electronic time constants could certainly be in the microsecond and probably the nanosecond range (if memory could be suitably incorporated). If the latter were achieved, such an artifact could get through 100 years ($= 3 \times 10^9$ sec) of human thought in about 50 minutes.

### 3.3.2. Real Time Neurones

With a digital computer simulation, under each neurone at time $t$ we need the current value of $V$ (and of $\theta$ if it is allowed to vary) and also the times of firing of all attached presynaptic neurones back to the times $t - \delta$. By storing and continually updating this information we calculate the evolution of activity in the network as a function of time. Two points may be made about programming such a calculation. First, because a cell can only fire when an impulse arrives from another cell (i.e. on an upward jump of $\eta$, see Fig. 3.2), we need only calculate $V$ at such times (which are known because it is known



when the input cell has fired). Hence to advance the computer simulation of the network by a given finite time $\delta t$ we need only perform a finite number of calculations, even though $V$ is a function of the continuous variable $t$. Secondly, although it may seem necessary to calculate continually which cell fires next in the whole network, this is not so because no cell can influence another until at least $\delta_{\min}$ (the smallest of the delays $\delta$) after it has fired. Hence we can most conveniently perform the calculation by going through the neurones in order and calculating for each if and when it fires in the next time interval $\delta_{\min}$. This simplifies the programming problem considerably, and was used in my previous work (Griffith, 1967a, Chapters 4 and 5).

Finally, the time and space requirements are considerably increased, depending on circumstances, by a factor of ten at the very least or probably much more.

## 3.4. Symbolic Logic and Switching Circuits

### 3.4.1. Symbolic Logic

McCulloch and Pitts (1943) pointed out an interesting isomorphism between the input–output relations of their idealized neurones and the truth functions of symbolic logic. This has attracted a lot of attention and many people have thought that it casts great light upon brain function and the neural basis of the logic of human thought. Personally I do not believe that this is so, at least to date, nor do I think that the logical notation which thus becomes available to describe neural activity has much real use. I think the latter because the logical expressions required to describe a neurone seem to me much more cumbersome and difficult to manipulate than other more usual ones, especially when any large number of neurones or interconnections are being considered, and because of the difficulty (Kleene, 1956) of dealing with networks which, like examples 4 and 5 of Section 3.1.2, have any re-entrant paths (which probably includes all networks of any biological interest). Not everyone would agree with this verdict and so we give here a brief introduction to the matter to help the reader form his own opinion (for a clear elementary account of symbolic logic, see Basson and O'Connor, 1965).

The relevant part of symbolic logic is concerned with the question of the truth of composite statements, given the truth or falsity of the constituent simpler ones. In elementary logic there are only two alternatives considered: everything is either true or false. Composite statements are formed from simpler ones by combining them, using certain logical symbols. We now introduce some of these, using symbols like $x$ or $y$ to stand for the simple statements.

1. Or. The symbol is $V$ and from $x$ and $y$ we can construct $xVy$ (also $yVx$ which is identically the same). The definition of a logical symbol is exhibited by a so-called truth table which shows whether $xVy$ is true or false when you know whether each of $x$ and $y$ separately are. The defining truth table in this case is:

| $x$ | $y$ | $xVy$ |
|---|---|---|
| True | True | True |
| True | False | True |
| False | True | True |
| False | False | False |

Thus, in ordinary English, the logical "or" means "either one or the other or both".

2. And. The symbol is a dot "." and from $x$ and $y$ we construct $x.y$. Again we can write a truth table. This time we shall note the binary character of the alternative "true" or "false" and write a "1' to indicate "true" and a "0" to indicate false, obtaining the table:

| $x$ | $y$ | $x.y$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

3. "Not" is written $\sim$, and logical implication (i.e. if this, then that) is written $\supset$. The truth tables are:

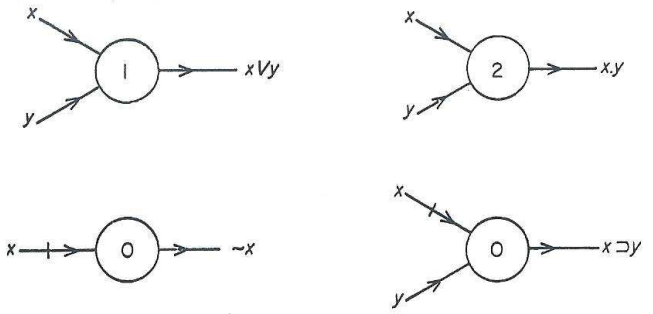
| $x$ | $\sim x$ |
|---|---|
| 1 | 0 |
| 0 | 1 |

| $x$ | $y$ | $x \supset y$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

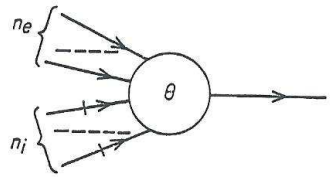
$x \supset y$ is identically the same as $(\sim x)Vy$, i.e. if the first of these statements is true so is the second and vice-versa, and as the reader will readily see on examining its truth table does not have quite the same meaning as implication does in ordinary English (we do not normally say that a false proposition implies the truth of a true proposition). However, this is the way in which logicians have found it useful to define it.



McCulloch and Pitts now draw attention to the fact that if we let "0" in the truth function correspond to "not firing" and "1" to "firing", then the truth functions are very like certain simple McCulloch-Pitts neurones. The corresponding neurones are:




The truth tables we have given are also the tables showing the input–output relations of these neurones. Conversely, any neurone can be represented by a truth function. Suppose the neurone is




having $n_e$ excitatory inputs which we represent by $x_1, x_2, \ldots, x_{n_e}$, and $n_i$ inhibitory inputs represented by $x_{n_e+1}, \ldots, x_{n_e+n_i}$. Any total specification of input is given uniquely by specifying the value of each $x_j = 0$ or 1. There are thus $2^{n_e+n_i}$ possible different total inputs which we can number using a parameter $\alpha = 1, 2, \ldots, 2^{n_e+n_i}$. For example, the total input in which an impulse arrives along every individual input might be that with $\alpha = 1$. It occurs if and only if the truth value of

$$X_1 = x_1.x_2.x_3.\ldots.x_{n_e+n_i}$$

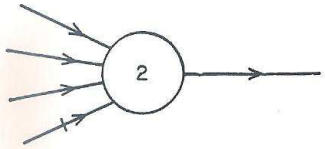
is 1. For every $\alpha$ there will be a corresponding $X_\alpha$. For some $\alpha$ the neurone

will fire (because $N_e - \phi N_i \geqslant \theta$). Let the set of all such $\alpha$ be called $S$. Then the expression

$$X = \bigvee_{\text{all } \alpha \text{ in } S} X_\alpha$$

represents the neurone as a truth function.

We now give a worked example in order to clarify the derivation. We consider the following neurone




and let $\phi = 1$. Evidently the neurone fires if either two of $x_1$, $x_2$ and $x_3$ fire but not $x_4$ or if all of $x_1$, $x_2$ and $x_3$ fire, irrespective of whether $x_4$ does or not. This gives five total inputs which cause the neurone to fire, which we shall number from 1 to 5, with corresponding $X_\alpha$:

$$X_1 = x_1 . x_2 . x_3 . x_4$$
$$X_2 = x_1 . x_2 . x_3 . \sim x_4$$
$$X_3 = x_1 . x_2 . \sim x_3 . \sim x_4$$
$$X_4 = x_1 . \sim x_2 . x_3 . \sim x_4$$
$$X_5 = \sim x_1 . x_2 . x_3 . \sim x_4$$
$$S = (1, 2, 3, 4, 5)$$
$$X = X_1 \vee X_2 \vee X_3 \vee X_4 \vee X_5 .$$

We can simplify X slightly into the form

$$X = (x_1 . x_2 . x_3) \vee \{((x_1 . x_2) \vee (x_1 . x_3) \vee (x_2 . x_3)) . \sim x_4\}$$

but we see that it is in any case quite complicated even for this neurone which has only 4 inputs.

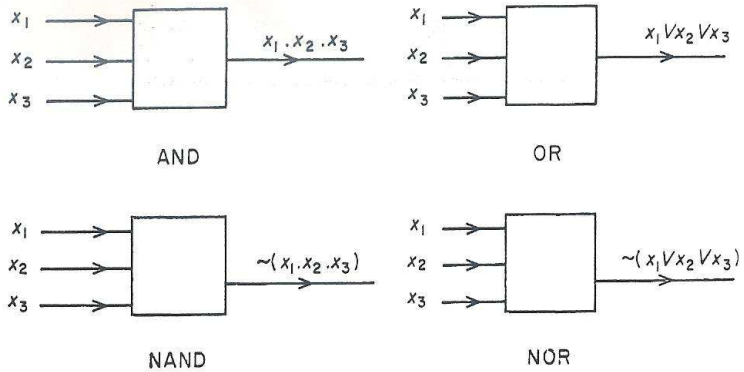
We have only established the isomorphism for single neurones, but it is easy to extend it by induction to networks without re-entrant paths and, in the other direction, to more complicated logical expressions (which then generally correspond to networks rather than to single neurones). For this and discussion, see McCulloch and Pitts (1943), Kleene (1956).

### 3.4.2. Switching Circuits

These are used extensively in digital computers and digital equipment generally (for introductory account, see Oppenheimer, 1966). Nowadays they are usually bought as standard modules (complete circuits) each of which gives as its output a definite logical function of its input. Both input



and output are standardized electric voltage pulses of height typically a few volts and duration typically from a few nanoseconds to a few microseconds. Probably the most extensively used basic units are the AND, OR, NAND and NOR gates. Examples of these follow:




and show the relation between input and output. NAND and NOR gates are more common in practice than one might perhaps expect because of the relative ease of making circuits to perform their functions. It is clear that if we have just two inputs $x_1$ and $x_2$ then AND and OR gates perform the basic logical functions $x_1 . x_2$ and $x_1 \vee x_2$, while with only one input $x_1$ NAND or NOR perform the function $\sim x_1$. We have seen already that we can build truth functions corresponding to an arbitrary logical neurone and therefore we can also construct a switching circuit to represent it, using these gates, although not necessarily very economically. Incidentally, the AND and OR gates actually shown above represent logical neurones having $n_e = 3$, $n_i = 0$ and $\theta = 3$ and 1 respectively. Finally, note that arbitrary networks of logical neurones, even with re-entrant paths, can be built out of these switching modules. If one was actually doing this, it would be natural to keep the output of all the neurones in the network synchronized using gates controlled by a master clock multivibrator.

## 3.5. Further Mathematical Features of McCulloch-Pitts Networks

### 3.5.1. Matrix Formulation of Activity of a Network

We saw in example 5 of Section 3.1.2 that the present state of a network could be conveniently represented as a vector having 0's and 1's for its elements. The threshold condition for a logical neurone involves the

TABLE 1.1

Some data for average human brains. (From Blinkov and Glezer, 1968)

| | | | % of brain weight |
|---|---|---|---|
| Brain weight, male | 1400 gm | Cerebral hemispheres | 88 |
| Brain weight, female | 1300 gm | Cerebellum | 10 |
| Brain volume | 1200 ml | Brain stem | 2 |
| Spinal cord weight | 27–38 gm | | |
| Spinal cord length | 42 cm | | |

The development of the brain as a whole in relation to the spinal cord is also rather extreme in man, as is illustrated in Table 1.2, and this has also

TABLE 1.2

Spinal cord weight as percentage of brain weight. (From Blinkov and Glezer, 1968)

| Animal | % |
|---|---|
| Man | 2 |
| Macaque monkey | 12 |
| Domestic cat | 24 |
| Tortoise | 120 |

been considered to be a reason for man's intellectual predominance. Another thing which has been noted in this connection is the high degree of folding (called convolution) of the surface part (cerebral cortex) of the cerebral hemispheres, thus giving a relatively large surface to volume ratio, although man is not the most extreme animal in this respect (see Table 1.3). All these

TABLE 1.3

Areas of outer surface of cerebrum and of the cerebral cortex in $cm^2$. (From Elias and Schwartz, 1969)

| Animal | Outer surface | Cerebral cortex | Ratio |
|---|---|---|---|
| Kangaroo | 53·0 | 74·9 | 1.41 |
| Fox | 68 | 135 | 2.01 |
| Man | 795 | 2275 | 2.86 |
| Bottlenose dolphin (Atlantic) | 567 | 2700 | 4.47 |
| Bottlenose dolphin (Pacific) | 693 | 3343 | 4.75 |
| False killer whale | 1488 | 7392 | 4.97 |



arguments are extremely superficial however and, until we do understand in considerable detail how the human brain works, we cannot possibly tell whether similar things could be done by brains lacking particular gross features which happen to be always present in the brains of mentally normal humans.

With a digital computer one might well expect that, other things being equal, the larger it is or the more circuits or store that it has, the more things will it be able to do. Therefore it is natural to ask whether larger brains are generally better brains, and especially whether man's brain is the largest of the lot. The answer to the latter question is that man does have one of the largest brains but not the largest. A selection of average or typical brain weights is given in Table 1.4 and illustrates this point (also see the Frontispiece). There is, however, considerable variation in weight from one specimen

TABLE 1.4

Brain weights in grams for various animals. (Crile and Quiring, 1940; Tower, 1954; Blinkov and Glezer, 1968)

| Animal | Weight | Animal | Weight |
|---|---|---|---|
| Fin whale | 6785 | Domestic cat | 25 |
| Indian elephant | 4400 | Alligator | 8·4 |
| Porpoise (dolphin) | 1735 | Tortoise | 0·3 |
| Man | 1400 | Field mouse | 0·2 |
| Walrus | 1126 | Common toad | 0·07 |
| Orangutan | 372 | Cockroach | 0·0002 |

to another, as is shown in Table 1.5. Over a large range of size there does not seem to be any clear-cut relation between intelligence and brain size, which is rather surprising.

TABLE 1.5

Weight of certain normal human brains in grams. (From Cobb, 1965)

| | | | |
|---|---|---|---|
| Australian bushwoman | 794 | European man (average) | 1400 |
| Anatole France (at 80) | 1017 | Thackeray | 1658 |
| Japanese woman (average) | 1250 | Bismarck | 1807 |
| Walt Whitman | 1282 | Cuvier | 1830 |
| European woman (average) | 1300 | Daniel Webster | 1895 |

It has also been suggested that the percentage of brain relative to the body is especially relevant to intelligence. There would seem to be much less reason to expect this and Table 1.6 shows some obstacles that this view faces.