Immunology 1989 **66** 1–7

HYPOTHESIS PAPER

Immune network theory: a role for parallel distributed processing?

F. T. VERTOSICK* & R. H. KELLY† * *Department of Neurological Surgery and † Division of Clinical Immunopathology, Department of Pathology, University of Pittsburgh School of Medicine, Pittsburgh, Pennsylvania, U.S.A.*

Accepted for publication 23 July 1988

CONTENTS

Introduction

Despite their vast anatomic differences, both the nervous system and the immune system deal in the same commodity: information. Consequently, insight into their function can be gained through application of formal information theories developed in other disciplines, e.g. computer science. Jerne first used this approach in immunology by describing the immune response in terms of network theory (Jerne, 1974). This review will examine some recent developments in network theory and consider how these ideas might apply to immune networks.

Like the nervous system, the immune system must learn new information, recall previously learned information, and make decisions based upon prior experiences. Moreover, each organ system must be cognitive, in that each must perceive and respond to a specific environment. Perception, recall and decision-making represent specific facets of the more general concept known as intelligence. Research into the fundamental nature of intelligence has grown into a major industry, with the nervous system serving as the primary biological model of the intelligent machine. But is intelligence an exclusive property of neural networks? A recent advance in network theory, known as parallel distributed processing (PDP), suggests that network intelligence is primarily a function of network architecture, as opposed to the intrinsic capabilities of their individual elements (sometimes referred to as processing units in PDP). As long as the processing units satisfy certain criteria, they can be incorporated into a (potentially) intelligent network. Although the neuron is considered the biological prototype of the PDP processing unit, lymphocytes also possess the properties of processing units, as discussed below. Consequently, lymphocytes, like neurons, can be linked into an 'intelligent' network.

PDP describes networks of similar processing units operating in parallel. All units are simultaneously receiving, storing, processing and sending out data, with each unit interacting with a limited number of other units to which it is directly 'connected'. Although structurally similar to the simple networks used by Jerne to model the immune system, networks with PDP architecture, unlike Jerne networks, have great capacity to learn, recall and associate complex patterned information.

The theoretical framework of PDP was largely developed by computer scientists working in the field of artificial intelligence, but recent advances suggest direct applications in biological systems, and PDP has already been used extensively in the analysis of neural networks. PDP represents a highly publicized major refinement of network theory, which has been surprisingly ignored by theoretical immunologists. This situation may be due to the historical development of immune network theory. When Jerne first proposed the theory, in 1974, interest in parallel processing models was dampened by Minsky and Papert's book *Perceptrons* (Minsky & Papert, 1969). This treatise demonstrated the limited computational power of a simple class of

Correspondence: Dr F. T. Vertosick, Division of Clinical Immunopathology, 5725 Children's Hospital Main Tower, Pittsburgh, PA 15213-2583, U.S.A.

parallel processing networks known as perceptrons. Later analysis showed that Minsky and Papert had underestimated the power of PDP. The late 1970s saw a renaissance in parallel processing; unfortunately, this renaissance took place well after the simpler network concepts used by Jerne had become firmly entrenched in theoretical immunology.

We feel that the application of PDP to lymphocyte networks may help us to understand how the immune system processes and stores large amounts of information, and that PDP may shed some light on specific phenomena, such as antibody cycling and antigenic competition. Our present communication has a dual purpose; first, to introduce the fundamental tenets of PDP to immunologists and, second, to discuss the specific implications of PDP for immune networks.

Building a PDP network

Consider a set of identical processing units. Each unit must be equipped to send and receive some form of data, must possess a memory to store data, and must use some logic function to convert incoming data into outgoing data. The B lymphocyte, for example, fulfils these criteria, in that it receives input (from antigen-presenting macrophages or dendritic cells, other lymphocytes, and cytokines), generates output (antibody), remembers its antigenic specificity, and converts input (antigenic stimulation) into output (antibody secretion) in a quantitative fashion. Individual units in PDP networks do not have to exhibit sophisticated behaviour because, according to PDP theory, sophisticated processing is an expected capability of networks consisting of huge numbers of simple processing units.

The units are then connected into a network. Like its component units, the network receives information from its environment, processes it, and generates output. There are three types of units within a PDP network; input units, output units and 'hidden' units, i.e. units which communicate with other units but which do not communicate directly with the network's external environment (Fig. 1). In a PDP immune network, for example, input units would be lymphocytes involved in antigen recognition, output units would be plasma cells secreting antigen-specific antibody, and lymphocytes producing anti-idiotypic antibodies would be examples of hidden units (in some cases, a given lymphocyte may serve as both an input unit and output unit simultaneously). For a network to function as a PDP device, four aspects of the network must be defined (Rumelhart, Hinton & McClelland, 1986):

1. A pattern of connectivity among the units, which determines how the inputs and outputs of individual units are connected.
2. A connectivity weight matrix, defining the relative strengths and signs of the connections.
3. An activation rule which defines the output of each unit as a function of the total input.
4. A learning rule which defines how experiences modify the network's connectivity weight matrix.

The complete formulation of PDP theory requires further assumptions, but these four will suffice for the purposes of this discussion. We will now consider each of these aspects in greater detail.

The pattern of connectivity

In the case of neurons or microprocessors, the connections between units will be direct physical links, i.e. axons or wires,

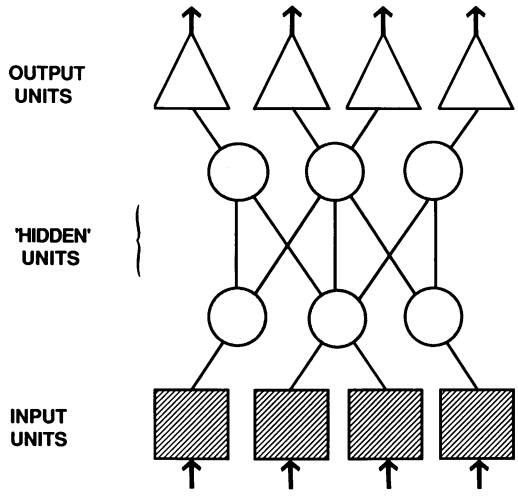


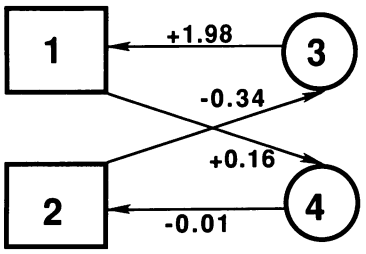
Figure 1. The basic architecture of a parallel-distributed processing (PDP) network. Three types of simple processing units are illustrated: input, output and hidden. The layers of hidden units, i.e. units that do not interact directly with the environment, allow the network to store patterns of data as internal representations, or steady states, of the network (see text). In an immune network, lymphocytes involved in immunoregulation, but not directly concerned with antigen processing or antigen-specific antibody production (e.g. lymphocytes producing anti-idiotypic antibodies) would serve as hidden units.

while in a lymphocyte network connections will be formed by chemical links (e.g. anti-idiotypic antibody, lymphokines), whose efficiency is enhanced by an effective sorting system for the units (e.g. lymph node, spleen, Peyer's patch) which employs the inherent traffic of lymphocytes to alter the proximity of responding units (Davies *et al.*, 1969; Fossum & Ford, 1985; Lanzavecchia, 1985). The connectivity pattern specifies how the individual units are 'wired' together. Will one unit be directly connected to two, three, four or more other units? Will the pattern of connectivity be uniform, i.e. with each unit connected to the same number of other units, or will some units be more connected than others? Will the network as a whole be 'highly connected' or 'poorly connected'? X-ray diffraction studies suggest that any one idiotype may be recognized by as many as forty different anti-idiotypes (Novotny, Handschumacher & Haber, 1986). Consequently, an idiotypic-anti-idiotypic network is probably highly connected.

The pattern of connectivity defines certain properties of the network, such as the speed of information propagation through the network, and the network's 'fault tolerance', or ability to perform after the destruction of processing units and/or the interruption of established connections (Hillis, 1985). The performance characteristics of a variety of connectivity patterns, also known as *topologies*, have been studied. Any detailed PDP model of an immune network will need to specify a topology, taking into consideration the theoretical properties of known topologies and matching these with available data concerning lymphocyte connectivity.

The connectivity weight matrix

The connections in a PDP network are not homogeneous, i.e. some connections will be stimulatory and others inhibitory (depending upon whether negative or positive information is relayed between two units), and some connections will be strong



$W_{14} = +0.16$ $W_{31} = +1.98$

$W_{23} = -0.34$ $W_{42} = -0.01$

$W_{12} = W_{21} = W_{34} = W_{43} = 0$

Figure 2. A small network and its corresponding connection weights. The connection weights are given as coefficients of a connection weight matrix. The matrix is a dynamic entity which summarizes the connectivity of the network at any time. Note that the connections are unidirectional and that the connection weight of two disconnected units is zero.

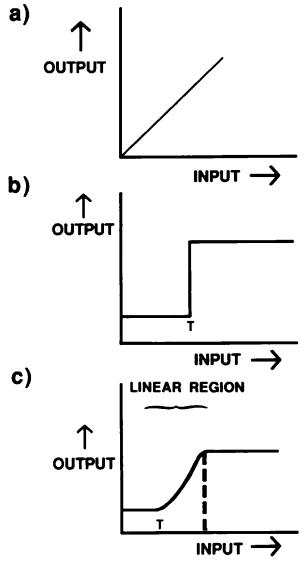


Figure 3. Simple activation rules for PDP processing units. Rule (a) gives the output of the unit as a linear function of the input. Rule (b) is a step-function, which illustrates the behaviour of a simple on–off device which is triggered when the input reaches the threshold, T. Rule (c) is a sigmoidal activation curve, which is triggered at threshold T, has a linear region where output is proportional to input, and then reaches saturation. Biological units, such as neurons and lymphocytes, follow non-linear activation rules, such as (b) and (c).

and others weak (depending upon the quantity of information relayed between units). Connections are considered polarized, i.e. information flows in only one direction through a connection. All of a network's connection strengths can be summarized in a *connection weight matrix*. For each pair of units, x and y, in the network, there is a corresponding weight matrix element, w_{xy}, such that the output from x is converted to input into y by the equation:

$$\text{input}_y = (w_{xy})\ (\text{output}_x)$$

For example, postulate that x's output is connected to y's input with a strength, or weight, of $+2$ and y's output connected to x's input with a strength of -3. If we work with dimensionless units, let x send an ouput of 1 to y. In fact, y will actually receive an input from x of $(1 \times 2) = 2$. The output of x has been 'weighted' by a factor of 2 before entering y. Likewise, if y sends an output of 0·5 to x, x will receive an input from y of $(-3 \times 0{\cdot}5)$ or $-1{\cdot}5$. In this example, $w_{xy} = +2$, while $w_{yx} = -3$. Figure 2 illustrates a small PDP network and its associated connection weights. Note that a connection weight of zero is equivalent to total disconnection of the two units. The weight matrix is a difficult concept, but it is central to PDP theory. *A PDP network learns by modifying its connection weight matrix in response to external training.* The weight matrix summarizes the entire superstructure of the network at any one time, and reflects the information stored within the network as well as the computational power of the network.

In the neurosciences, the connection weights characterize the interneuronal connections, i.e. synapses. Excitatory synapses have positive weights, while inhibitory synapses have negative weights. The magnitude of the weight represents the degree of hyperpolarization, or depolarization, of the post-synaptic membrane that occurs with synapse firing. The plasticity of the neuronal network is a function of plasticity of the connection weight matrix.

What is the analogue of a PDP connection weight in an immune network? *We postulate that the connection weight between an idiotypic lymphocyte and a corresponding anti-idiotypic lymphocyte is determined by the affinity of the idiotypic–anti-idiotypic interaction.* A positive connection weight means that the anti-idiotype 'helps' or stimulates the idiotype, while a negative weight suppresses the idiotype. These assumptions are straightforward. If the affinity is zero or near zero, then the two lymphocytes can be considered disconnected, while a high affinity predicts that the output of the anti-idiotype clone will have a strong influence on the behaviour of the idiotype. A corollary would predict affinity changes secondary to conformational alteration of the idiotope (e.g. as occurs with antigen binding).

The activation rule

The activation rule defines each unit's output as a function of the total input received from other, directly connected units. In the simplest case, the output is linearly proportional to the input (Fig. 3a). Networks constructed of units with linear behaviour are, in general, uninteresting (Rumelhart, Hinton & McClelland, 1986). Useful activation rules define the output as a *non-linear function of the input*, and networks based upon these rules are said to be non-linear.

Figure 3b illustrates one of the simplest non-linear activation rules, the step function. The processing unit has a minimum output of zero and a maximum output of one, with a threshold of 'T'. For total inputs below threshold the output is zero, while for input above T the output is maximum.

A more complex activation rule is depicted in Fig. 3c. In this case, the output is a sigmoidal function of the input. The sigmoidal function is essentially a step function which contains a region wherein the output is linearly dependent upon the input. Three regions are defined by the sigmoidal activation rule: (1) below threshold, where input yields no output, (2) the linear region, where output is proportional to input and (3) saturation,

where further increases in input yield no additional output. We will postulate here that lymphocytes follow a sigmoidal activation rule. This postulate is not new, in that most quantitative models of Jerne networks assume a sigmoidal input/output curve for lymphocytes (Richter, 1978). The activation rule for a given lymphocyte may not be a rigid parameter, but rather one which is subject to local modulation by non-immunoglobulin secretory products (e.g. lymphokines, interferons, prostaglandins). The remainder of this discussion will apply only to PDP networks consisting of units employing a simple non-linear activation rule, such as the examples given in Figs 3b and c.

The learning rule

As stated above, a PDP network learns by modifying its connection weights. Three different modifications can occur: (1) new connections can be made (by changing zero connection weights into non-zero weights), (2) old connections can be removed (by changing non-zero connection weights into zero weights) and (3) or old connections can be strengthened or weakened. *The learning rule defines how the network alters its connection weights when presented with new information.*

In his book, *Organization of Behavior*, Hebb suggested the prototypical learning rule: Connections are strengthened or weakened in direct proportion to their use. Thus, if a connection is used frequently it becomes strong, but if it is used rarely it will atrophy or even vanish. Most PDP learning models are based upon some variant of this, *the Hebbian learning rule* (Hebb, 1949). The Hebbian rule has intuitive appeal, for it implies that learning depends upon repetition.

A particularly useful Hebbian variant is the delta rule, which defines changes in the connection strengths as a function of 'ideal' outputs (Rumelhart, Hinton & Williams, 1986). As an illustration of the delta rule, consider a rifle marksman whose 'ideal' output consists of hitting the centre of a target. If he misses, he will modify his next shot according to the difference between his actual output (a missed shot) and his ideal output (a centre shot). A large miss to the right will require a large correction to the left, a small miss to the left will require a small correction to the right, and so on. The marksman will proceed through a series of shots, each followed by a series of smaller and smaller corrections, until he achieves his ideal endpoint. In this manner, the marksman 'learns' what aim will produce the desired outcome. Similarly, a PDP network can learn what values of the connection weights will give a specified output by repeated applications of the delta rule; the connection weights are altered according to an error algorithm with each cycle, until the desired output state is achieved. The error algorithm defines the change in the connection weights as a function of the output error, i.e. the difference between the actual and desired output states (e.g. bacteremia vs. immune clearance). Several such algorithms have been developed, and the search for better ones continues (Rumelhart *et al.*, 1986).

Cycling of antibody production is a well-known phenomenon, which occurs in both primary and secondary antibody responses (Romble & Weigle, 1982). In a PDP model of the immune response, individual cycles would represent propagation of the changes in connection weights that occur during the learning, or remembering, of antigen patterns. The general increase in antibody affinity during the course of an immune response implies that a PDP network would obey some Hebbian learning rule, i.e. the connection weight (affinity) increases with use (antigenic stimulation).

The learning rule distinguishes a PDP network from the simpler Jerne network. In the Jerne model, the number and affinities of the idiotypic–anti-idiotypic connections are important from the viewpoint of network stability. In PDP theory, however, the characteristics of the connections are the key to the more profound aspects of the network, such as pattern memory and learning. As mentioned above, connection weights in a PDP network must be plastic. Are the idiotypic–anti-idiotypic interactions of a Jerne network plastic, i.e. do the affinities of the idiotypic–anti-idiotypic interactions change during the course of an immune response? Cycling of antigen-specific antibody affinities has been described (Doria, 1982), but no comprehensive study of how the affinities of anti-idiotypic antibodies change during an immune response has been reported, to our knowledge. Experimental verification of a PDP immune network may require detailed study of how the number and affinities of anti-idiotypic antibodies change during the acquisition of immunological memory.

The properties of PDP networks

Now that the basic tenets of PDP theory have been outlined, what are the properties of PDP networks that would make a PDP theory of the immune network desirable? Like other processing architectures, PDP networks are fundamentally computational devices, receiving input data, solving problems based upon that data, and generating an output. Before continuing, it should be noted here that the analysis of PDP networks is beyond the theoretical stage. Large scale, operational PDP machines exist, e.g. Daniel Hillis's Connection Machine, built by Thinking Machines Corporation, which consists of more than 65,000 processors linked in a Boolean n-cube topology. Interested readers are referred to Hillis's book, *The Connection Machine* (Hillis, 1985).

There are several properties which make PDP networks unique, and which make them particularly desirable as models of biological behaviour. Four of these properties will be discussed in detail; speed, content-addressable memory, fault tolerance and, pattern completion.

Speed

PDP networks process information very quickly, by virtue of their intrinsic division of labour. Processing units, or groups of units (e.g. lymphocytes), may be working on different aspects of the same problem simultaneously. Conventional computers are serial devices, in that they can only work on one aspect of a problem at a time.

Computers based upon PDP architecture, e.g. the Connection Machine, have as their chief advantage the ability to deliver approximate answers, within minutes, to complex problems which would require hours for a conventional Cray supercomputer to solve exactly. Of course, this is a disadvantage if exact solutions are required. For biological systems, however, the very rapid output of approximate solutions is probably preferable to the laborious output of exact solutions, and so PDP architecture would be an advantage. Speed, of course, is a relative term. Connections of the immune system operate on a much slower scale than the nanosecond relays in a computer or the millisecond scale of neuronal action potentials, owing to the time

required for cell sorting and chemical communication. Nevertheless, rapid computation is probably equally important for both immune and neural networks.

Content-addressable memory (CAM)

Serial computers contain a main processing unit and a separate depot of memory storage which is intermittently accessed by the main processing unit. In contrast, PDP devices have many processing units, each with its own limited memory storage. One might reasonably ask the question, 'Where is the main memory of a PDP device?' Hopfield pointed out that the PDP network's steady states can store information, with each steady state mapped to a pattern of data (Hopfield, 1982, 1984). The network 'remembers' these patterns by jumping between steady states. The information stored in the network's steady states is said to be in the *content-addressable memory*, or the CAM. The CAM is the main memory of a PDP device. *In PDP networks, therefore, memory is not allocated to any one centralized location, but can be considered a collective property of the entire network.* In a PDP model of the immune system, it follows that some portion of immunological memory will be stored at the network level.

Fault tolerance

Large PDP networks are reasonably buffered against random failure of processing units and/or the interruption of individual connections. This is in marked contrast to serial devices which can be crippled by the failure of a single critical connection or logic circuit. From the biological viewpoint, high fault tolerance may be one of the most attractive features of PDP devices. Jerne networks, for example, have a very low fault tolerance. The loss of a single anti-idiotype may predispose to the neoplastic proliferation of the corresponding idiotype clone during an immune response. Real biological systems must be fault tolerant, given their natural senescence.

The fault tolerance of a PDP device is due, in part, to the 'global' distribution of information storage which they employ. Conventional computers, for example, store information regionally in their memories. The difference between regional and global storage of information can be illustrated by comparing the image stored in a conventional photograph to the image stored in a holographic plate. If one cuts away one-half of a conventional photograph, then one-half of the information contained in the original image will be irretrievably lost (unless both halves are identical, i.e. information is redundantly stored in the image). A holographic image can be divided in half, however, and the original image can be completely reconstructed from the information stored in either half. Data stored in a PDP network are similar to holographic images, in that they can be recovered from some subset of the original network. While this property of holograms and PDP networks seems counterintuitive, it is, nevertheless, real. For the interested reader with a limited background in network theory, Tank and Hopfield recently have provided a detailed and lucid discussion of the steady state concept of network memory storage (Tank & Hopfield, 1987).

Pattern completion

PDP networks deal best with *patterned information*, i.e. information arrayed in a spatial format. This derives from the spatial character of the networks themselves. *The steady states which comprise the CAM consist of a set of spatially oriented patterns stored by the network.*

Moreover, PDP networks are capable of completing patterns, or images, from incomplete information. If a network has stored a photographic image, for example, it can recognize that image again when provided with only a small subset of picture elements (Hinton & Sejnowski, 1986). In effect, the network will automatically 'fill the gaps' present in incomplete data.

Because the external environment rarely provides perfect input, pattern completion is essential to the process of cognition in the nervous system. Fortunately, the brain is an excellent pattern completion device; recall, for example, how easily one can identify a popular song after hearing only the first few notes. The ability to recognize a friend after he has grown a new beard is another example of pattern completion. Even with half of the facial image obscured, there are enough recognizable features remaining to allow the brain to associate the bearded face with the unbearded one stored in memory.

Not surprisingly, PDP models are gaining favour with cognitive theorists, but will they gain favour with theorists in immunology? The answer to this question depends, to some extent, upon how one defines cognition as it applies to the immune system. As portrayed by conventional immunology, immune cognition consists simply of individual lymphocytes recognizing individual epitopes. If the immune system has a PDP architecture, then this may be a gross underestimate of the immune system's cognitive abilities. A PDP network could, in theory, store and recall complex antigen patterns containing multiple epitopes, as will be discussed below.

Cognition and the immune system

The brain's perception of the external environment can be broken down into two fundamental processes; sensation and cognition. Sensation occurs at the level of the sensory organ and is characterized by individual units responding to specific stimuli, e.g. the response of retinal rods to photons of light. Cognition, on the other hand, requires higher level integration of the information provided by the sensory organs, and comparison of this information with patterns previously stored in memory. In the case of the visual system, integration of retinal information occurs within the visual associative areas of the cerebral cortex. Thus the retina responds to light reflected from an apple, while the cortex reconstructs the image of an apple and, by comparing this image with previously stored visual images, perceives an apple. By virtue of their pattern completion and storage capabilities, PDP devices provide a paradigm for understanding how sensation leads to cognition.

Like visual cognition, cognition within the immune system would require higher level integration, at the network level, of information received simultaneously from many responding lymphocyte clones. In a PDP network this information would be stored in CAM as an *antigen image*, which can be recalled later.

For example, consider the immune response to a strain of bacteria. Overall, hundreds of different lymphocyte clones may respond to the multiple epitopes present on the bacteria. According to the PDP paradigm, the responding lymphocytes, in turn, generate new anti-idiotypes or alter the affinities of existing anti-idiotypes, which, in turn, affect the number and affinities of their corresponding anti-idiotypes, and so on.

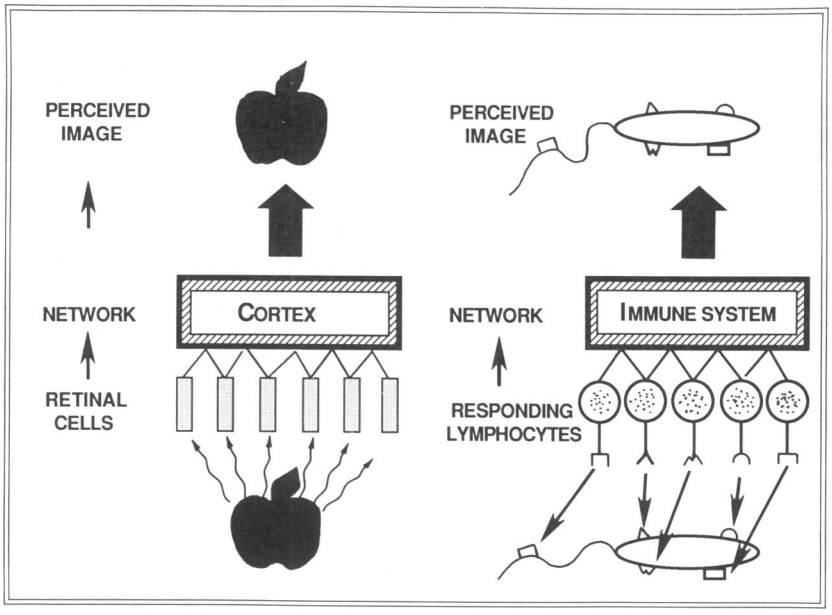


Figure 4. Perception as the integration of multiple response units at the network level.

Changes in the connection weights (idiotype–anti-idiotype affinities) propagate throughout the weight matrix, according to some Hebbian learning rule, as the immune response matures. The altered weight matrix ultimately acquires a new steady state configuration which the immune network will, in future, identify with that bacterial strain. The responding clones form an 'antigen retina', with the immune network acting as a visual cortex, integrating the pattern of individual clonal responses into an antigen image. The pattern completion property of a PDP immune network would allow recognition of the bacterial strain even when the system is presented with fragments of the whole bacterial cell. While lymphocytes perceive and recall individual epitopes, the PDP network perceives and recalls complex epitope patterns. As in the brain, sensation occurs at the cellular level, while cognition, association, and learning of complex patterned information would occur at the organ level (Fig. 4).

In the PDP model, the response to any one epitope will be influenced by other epitopes introduced simultaneously. Thus PDP theory may shed light on the phenomenon of antigenic competition (Taussig, 1977).

The PDP immune network: a summary

Our proposed PDP immune network model may be summarized as follows:

1. The immune network exhibits a PDP architecture with individual lymphocytes serving as processing units and idiotypic–anti-idiotypic interactions serving as connections. The topology of such a network is conjectural at present.

2. The strength of an idiotypic–anti-idiotypic connection is equivalent to the affinity of the anti-idiotype for the idiotype. The sign of the connection will be positive if the anti-idiotype stimulates the idiotype, and negative if it suppresses.

3. Lymphocyte units exhibit a sigmoidal activation rule. Consequently, the immune network is non-linear.

4. The network employs a Hebbian learning rule, in that the strength of individual connections is proportional to their usage. We conjecture that the changes in antibody affinities that occur during an immune response are a manifestation of that learning rule. Moreover, we conjecture that the cycling which occurs during a typical immune response represents the propagation of connection weight changes throughout the network, with each cycle representing one iteration in the network's attempt to approximate the 'ideal' response to a given antigen pattern.

5. Complex antigen patterns consisting of multiple epitopes can be learned and stored at the network level, with each pattern mapped into a steady state of the network.

6. We speculate that lymphokines could participate in a PDP network in at least three ways: (a) Trophic lymphokines, e.g. interleukin-2, would have an obvious impact upon the number of processing units. (b) Lymphokines may also influence the activation rule of specific lymphocyte clones (Fig. 3c) by changing the triggering threshold, for example, or by altering the slope of the linear region. Thus, the activation rule is not fixed, but would be influenced by the prevailing lymphokine milieu which exists within an individual lymphoid tissue microenvironment at any time point (Kelly & Wolstencroft, 1972). Given fluctuations in type and amount of lymphokines produced during an immune response, their local chemical modulation of a lymphocyte's output in response to a fixed input would be expected to greatly increase network complexity. In the case of a PDP network, however, complexity is an advantage, in that it increases the number of network steady state configurations and the computational power of the network. (c) The cycling of antibody affinities requires the cooperation of several subsets of lymphocytes, presumably in concert with lymphokines (Romble & Weigle, 1982). Our proposed learning rule, i.e. the Hebbian increase of idiotypic–anti-idiotypic affinities during acquisition of immunologic memory, is likely to involve similar processes. Exactly how the learning rule is implemented, and how the development of

memory for a given antigen is translated into changes of idiotype–anti-idiotype affinities within all of the 'hidden' units of the lymphocyte network, remain topics for future investigation.

Conclusion

The above discussion draws more heavily on computer science than immunology, and some will question whether highly technical theories derived by artificial intelligence specialists will prove useful in the analysis of immune function. Nevertheless, it seems pointless for theoretical immunologists to retain relatively simple network concepts while ignoring the major revolution that has taken place in network theory. Serendipity has been a kindly scientific ally in the past but, in the case of immune networks, we may be forced to wrestle with the increasingly complex abstractions of network theory, or give up our attempts to understand immune function in terms of network theory altogether.

ACKNOWLEDGMENTS

We would like to thank our colleagues, Drs A. B. Carpenter, B. S. Rabin and H. J. Seltman, for their critical review and helpful comments during the preparation of this manuscript.

REFERENCES

DAVIES A., CARTER R., LEUCHARS E., WALLIS V. & KOLLER P. (1969) The morphology of immune reactions in normal, thymectomized and reconstituted mice. I. The response to sheep erythrocytes. *Immunol.* **16,** 57.

DORIA G. (1982) Immunoregulatory implications of changes in antibody affinity. In: *Regulation of Immune Response Dynamics* (eds C. DeLisi and J. Hiernaux), Vol. 2, 1st edn, p. 1. CRC Press, Boca Raton.

FOSSUM S. & FORD W. (1985) The organization of cell populations within lymph nodes: their origin, life history and functional relationships. *Histopathology*, **9,** 469.

HEBB D. (1949) *The Organization of Behavior*, 1st edn. John Wiley, New York.

HILLIS D. (1985) *The Connection Machine*, 1st edn. MIT Press, London.

HINTON G. & SEJNOWSKI T. (1986) Learning and relearning in Boltzman machines. In: *Parallel Distributed Processing* (eds D. Rumelhart and J. McClelland), Vol. 1, 1st edn, p. 289. MIT Press, London.

HOPFIELD J. (1982) Neural networks and physical systems with emergent computational abilities. *Proc. natl. Acad. Sci. U.S.A.* **79,** 2554.

JERNE N. (1974) Towards a network theory of the immune system. *Ann. immunol. (Inst. Pasteur)*, **125C,** 373.

KELLY R. & WOLSTENCROFT R. (1972) Germinal centre proliferation in response to mitogenic lymphokines. *Clin. exp. Immunol.* **10,** 49.

LANZAVECCHIA A. (1985) Antigen-specific interaction between T and B cells. *Nature (Lond.)*, **314,** 537.

MINSKY M. & PAPERT S. (1969) *Perceptrons*, 1st edn. MIT Press, London.

NOVOTNY J., HANDSCHUMACHER M. & HABER E. (1986) Location of antigenic epitopes on antibody molecules. *J. mol. Biol.* **189,** 715.

RICHTER P. (1978) The network idea and the immune response. In: *Theoretical Immunology* (eds G. Bell, A. Perelson and G. Pimbley), 1st edn, p. 539. Marcel Dekker, New York.

ROMBLE C. & WEIGLE W. (1982) Cyclic antibody production in immune regulation. In: *Regulation of Immune Response Dynamics* (eds C. DeLisi and J. Hiernaux), vol. 1, 1st edn, p. 9. CRC Press, Boca Raton.

RUMELHART D., HINTON G. & MCCLELLAND J. (1986) A general framework for parallel distributed processing. In: *Parallel Distributed Processing* (eds D. Rumelhart and J. McClelland), Vol. 1, 1st edn, p. 45. MIT Press, London.

RUMELHART D., HINTON G. & WILLIAMS R. (1986) Learning internal representations by error propagation. In: *Parallel Distributed Processing* (eds D. Rumelhart and J. McClelland), Vol. 1, 1st edn, p. 318. MIT Press, London.

TANK D. & HOPFIELD J. (1987) Collective computation in neuronlike circuits. *Sci. Amer.* **257,** 104.

TAUSSIG M. (1977) Antigenic competition. In: *The Antigens* (ed. M. Sela), Vol. 4, 1st edn, p. 333. Academic Press, New York.