Reprinted from *Information Theory*, Fourth London Symposium published by Butterworths, 88 Kingsway, London, W.C.2.

26

LEARNING IN RANDOM NETS

MARVIN MINSKY and OLIVER G. SELFRIDGE
Lincoln Laboratory, Massachusetts Institute of Technology*

INTRODUCTION

THE general nature of the problem is that an organism must learn to make the 'right', or appropriate, response to its inputs. Typically, the inputs are large amounts of data, so that the machine must learn to recognize the similarities between different inputs which call for the same response, contrasted with the distinctions that call for different responses.

The particular machines we are concerned with are *random nets*. A random net is a large set of similar and simply-acting elements whose attributes and interactive connections may be randomly established. The extent to which randomness is a part of setting up or maintaining a net varies in the literature, and more recent accounts tend to minimize the use of randomness. Some of the units are usually designated *input*, and some *output* units. The units themselves are termed *neurons* or *cells*.

The underlying reason for the interest in random nets is the belief that if 'right' responses are rewarded by some 'reinforcement', perhaps of the contributing connections, and 'wrong' ones discouraged, then the net as a whole will organize itself so as to tend to make only right responses, even when they are very complicated and abstruse.

The underlying and, to us, over optimistic hope is that the complexities of connection and function of a random net, and especially its randomness, may enable it to solve problems that are really hard.

LEARNING MODELS

THE ROLE OF EXPERIENCE

Perhaps the simplest form of 'learning' is that in which one keeps a more or less complete record of some class of past events. Thus, in the 'rote learning' scheme of Samuel[1] the machine keeps a record of every checker position that it has encountered in actual play. With this record is stored the computed estimate of the relative value or strength of that position. This memory is used later should the machine find it necessary to consider one of these positions in deciding between a number of alternative potential moves. But, from our point of view, a learning system must improve its behaviour in novel situations as well as in those it has met (a condition fulfilled by the other schemes considered by Samuel). It must learn what to do in situations *like* previous ones: we must apply what might be called this 'basic learning

* Operated with support from the U.S. Army, Navy and Air Force.

heuristic': *in a novel situation try methods like those that have worked best in similar situations.*

This principle, in appearance a truism, entails two distinct notions of undefined similarity—similarity of method and similarity of situation. Learning, without these notions, can have only limited usefulness. The simplest interpretations of the similarity idea are, perhaps, schemes in which heuristically equivalent situations are grouped into categories, and treated at least partly without making distinctions within the categories. Records of past situations are interpreted as concerning categories, and applied to new situations accordingly. The recognition of the categories may be called the pattern-recognition problem, and may be trivial or exceedingly subtle.

Thus, in the most primitive form, learning is a mere record of experience, while more advanced forms which discover and use new categories can cover the creative aspects of genius.

SIMPLE CONDITIONING

Mapping

Suppose that only a few input configurations are encountered, and that each of these must trigger some simple response. In effect we want a mapping, or switching function, of input to output. In some classical conditioning models the 'learning' is of this type*. An input S_i is presented with (or followed by) an output R_j (often produced as response to another 'unconditioned' stimulus) and the machine is to learn to respond to S_i by producing R_j.

If each of the inputs and outputs represent, say, a single active cell, then the problem is one of establishing good conduction paths between contemporaneously excited pairs. It is easy enough to invent neural nets which do this: connections between each pair S_i, R_j of cells might improve their conductivity on each occurrence of the proper sequence of activity at their terminals. One can easily imagine somewhat more complex nets in which excitation of an R_j cell causes attachment to those persisting strands of activity due to an earlier excitation of S_i.

Such learning 'switches' could be made to work quite efficiently in the form of systematic nets or direct input–output connection matrix devices such as, for example, crossbar switches. If just a few reactions are to be learned of a large possible set, then a multi-level connection scheme is indicated (cf. the systems of 'finders' and 'connectors' in telephone exchanges). But it is difficult to see what advantages would accrue from the use of random nets here, instead of reasonably regular matrices. The more complex case in which the inputs and outputs are not represented by single cell events is considered later.

Operant reinforcement

Suppose that we again have a variety of inputs S_i and responses R_j and that we want to learn, for each input, which output is the best in the view of some external critic or Trainer. It would be easy enough to programme a system which, for each input, tries out a variety of outputs and records for each the Trainer's rejoinder. At any time the machine could recall which trial gave

* Indeed, some authors include here nearly all of learning[2].

the most favourable result. One might further wish to make the decision dependent on many trials, perhaps averaging in some way the results in each class of experiments. One could again do this easily in a systematic input-output connection device: now each connection element has an additional input from the Trainer.

It seems to be tempting to devise random net models for this kind of task. We could make a large net, with some kind of probabilistic conduction conditions (as well as, perhaps, a random connection matrix) and excite it, reinforcing 'good' results and discouraging 'bad'. The reinforcement effect should be, in the 'good' case, to make more likely each of the microscopic decisions that occurred in the net—the hope would be that this would be reflected by a behavioural change in the same direction. Random net models along these lines were studied by MINSKY[3] and FARLEY and CLARK[4]. In such devices one encounters difficulties due to interaction of different trained reactions. The consequent cross-effects can be treated either as nuisances to be removed, if possible by tedious discriminative training or as 'generalizations' reflecting some kind of similarity among the stimuli in question as in CLARK and FARLEY[5]. The trouble with this kind of generalization is that the similarity notion involved will be imposed by the net structure: one cannot expect the particular one to be widely useful. Another difficulty is that, for difficult problems, the expectation of obtaining any 'good' reactions at all seems remote, and merely enlarging the net offers no improvement (see p. 344).

COMPLEX CONDITIONING

Pattern recognition for compound stimuli

Usually the situation in an environment is perceived to the problem-solver not as a single input stimulus event but as a complex collection of inputs obtained through different receptors, filters, etc. Indeed, each sensory datum or neural fibre can be regarded as carrying information about the outcome of a different experiment performed on the environment. And each datum, or selected combination of them, can be regarded as asserting that the environment, at that moment, has or has not some 'property' specific to that set of tests. The problem is to discern in this multiplicity of information the 'patterns' which have heuristic significance to the task.

Indeed, patterns can often be defined by listing the properties which distinguish their exemplars from those of other patterns. In the important case of patterns whose definitions are not known in advance but for which examples are available, we can use experience to gather (statistical) evidence about the distributions of properties among the patterns. Each pattern is a region in the phase-space of the properties and we can guess at the region by observing the distributions arising from study of the examples.

The property-list scheme, while not a very general form of pattern recognition, does lend itself nicely to some rather straightforward inference schemes. One can treat the separate properties as more or less independent evidence for the defined categories (accepting the risks involved in such an approximation). In this case a natural way of connecting the evidence and the categories would be the notion of conditional probability, *i.e.* Bayes's rule. This is not the only scheme that might be used, but the conditions justifying its use are fairly clear. There are several papers in the literature which treat some visual pattern recognition problems in more or less this manner: the properties may be fairly complicated geometric notions[6], random sets of points[8] or just pairs of points[7] or even single points[9].

Bayes's rule analysis

Suppose we have a space of events divided into n classes $F_1, \ldots, F_n$. For every event E one performs a number of experiments $e_1, \ldots, e_m$ each of which have two possible outcomes, called 0 and 1, and which thus provide evidence about which class E actually belongs to. We shall (very restrictively) suppose that all $\{e_i\}$ are statistically independent. We also know (*a*) the *a priori* probabilities ϕ_j that E is in class F_j and (*b*) the conditional probabilities d_{ij} that if $E \in F_j$ then $e_i(E) = 1$. For a given E we define V as the set of i's for which $e_i(E) = 1$, and $\bar{V}$ as its complement.

We want to know how to choose best which F_j an event E belongs to, with the least chance of being wrong. That is, we want to know F_j for which $\Pr(F_j/V)$ is the largest.

By Bayes's rule, we have

$$\Pr(F_j/V) = \frac{\Pr(V/F_j)\Pr(F_j)}{\Pr(V)}$$

$$= \Pr(V/F_j)\frac{\phi_j}{\Pr(V)}$$

Now $\Pr(V)$ is not a function of j, so that the comparison depends on computing $\Pr(V/F_j)$. Since we have assumed $\{e_i\}$ are independent,

$$\Pr(F_j/V) = \frac{\phi_j}{\Pr(V)} \prod_V P_{ij}(1 - p_{ij}).$$

Thus, equivalently, we wish to minimize

$$H_j(E) = \log \phi_j + \sum_V \log p_{ij} + \sum_{\bar{V}} \log(q_{ij} = 1 - p_{ij}). \quad \ldots.(1)$$

We may regard H_j as 'the weight of the evidence' that $E \in F_j$ in fact. Alternatively we can write

$$H_j = \sum_V \log \frac{p_{ij}}{q_{ij}} + \phi_j + \sum_i \log q_{ij} \quad \ldots.(2)$$

We can interpret (1) as adding the evidence: the first term is the *a priori* evidence; the second the positive evidence, and the third the negative. We can interpret (2) as adding merely positive evidence, since the bracketed term depends only on the *a priori* expectation.

NETS FOR BAYES'S ANALYSIS

We realize the analysis of Bayes's rule in the form of organized networks by proceeding (*Figure 1*) from the outside to the inside of (1). The final decision unit chooses that j for which H_j is largest. Thus the final stages names that of the input which is largest. We call this unit the 'Decision Demons' unit following SELFRIDGE[10]. (Indeed the entire structure of *Figure 1* closely resembles his figure 3.) Immediately below are the Cognitive Demons which evaluate the evidence. Below them are the Computational Demons acting directly on the data. Connecting the Computational Demons to the Cognitive Demons is the complete net which weights the outputs and carries them up. It is these weights that (1) calculates.

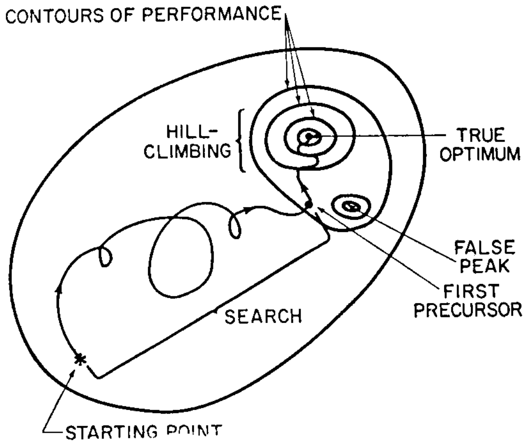

Figure 1. Machine state space showing successful search and hill-climbing

In *Figure 1* each Computational Demon has 2 outputs, in order to represent all the terms of (1). In *Figure 2* we represent (2) and the same Demons then need but one output. *Figure 2* corresponds, in fact, closely to the 'perceptron' model of ROSENBLATT[8], and our analysis applies directly to it.

Reinforcement learning

We have been assuming that we know $\{\phi_j\}$ and $\{p_{ij}\}$ for the weighted connection. Next we consider putting to the connections the task of learning a good set of weights. We use an ancient concept—pathways which contribute to 'good' responses will be reinforced. We judge the systems according as they approach, in some average sense, the weights shown by (1) and (2) to be optimum, at least for statistically independent $\{e_i\}$.

The connection weights w_{ij} are thus to be modified. In *Figure 1*, if $E \in F_k$, no change is made w_{ij} if $j \neq k$. If $j = k$ then we change w_{ij} to

$$\left.\begin{aligned} w'_{ij} &= k(w_{ij} + 1) \qquad \text{if } e_i = 1 \\ w'_{ij} &= kw_{ij} \qquad \text{if } e_i = 0 \end{aligned}\right\} \qquad \ldots.(3)$$

The 'reinforcement operator' (3) is essentially that used by BUSH and MOSTELLER[2]. The value of w_{ij} is then a simple Markov process with the expected values

$$\overline{w_{ij}} = P_{ij}k(\overline{w_{ij}} + 1) + (1 - p_{ij})k\overline{w_{ij}}$$

or

$$w_{ij} = P_{ij}\left(\frac{k}{1 - k}\right)$$

which is proportional to p_{ij} if we set k the same for all connections. The $\{\phi_j\}$ can be learnt in the same manner. Equation (3) is about as simple as one

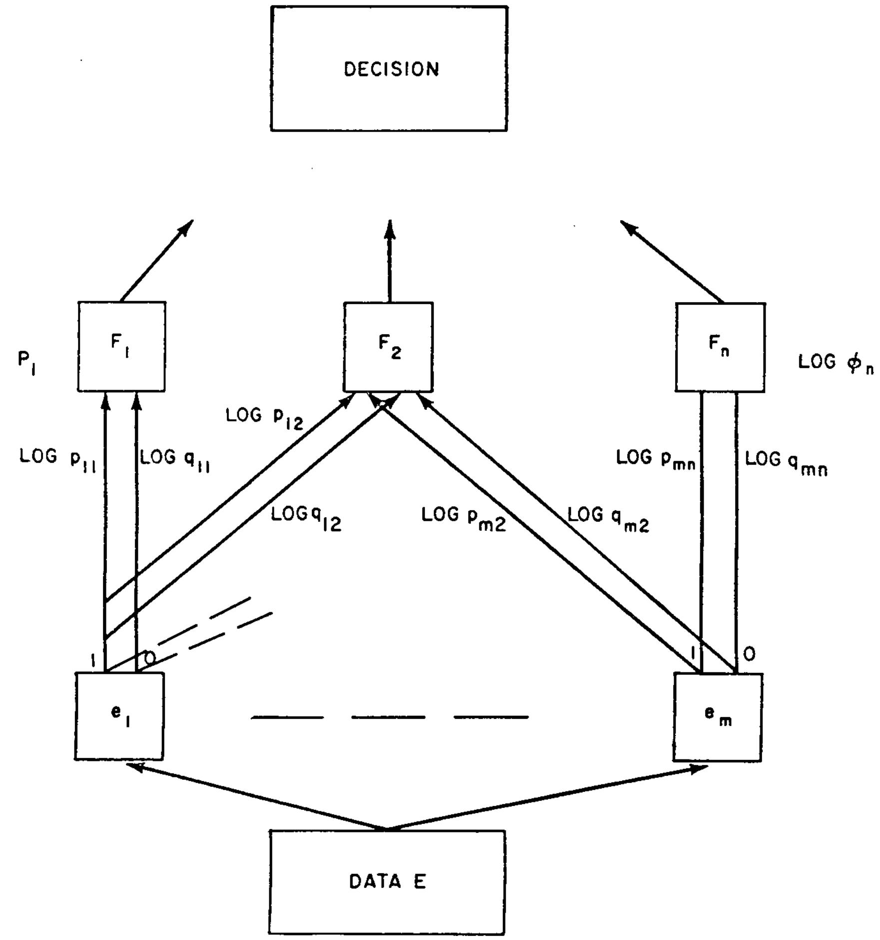

Figure 2a. Cognitive net I

could imagine—reward and punishment consist of granting or withholding an increment to the connection. The factor k is a time decay which helps to stabilize and normalize the system.

In *Figure 2*, however, the connection weights should approach $P_{ij}/(1 - P_{ij})$ rather than p_{ij}. A trivially different scheme will do so:

$$\left.\begin{aligned} w'_{ij} &= w_{ij} + 1 \qquad & e_{ij} &= 1 \\ w'_{ij} &= kw_{ij} \qquad & e_{ij} &= 0 \end{aligned}\right\} \qquad \ldots.(4)$$

Here the Markov equation

$$\overline{w_{ij}} = p_{ij}(\overline{w_{ij}} + 1) + (1 - p_{ij})k\overline{w_{ij}}$$

leads to

$$w_{ij} = \left(\frac{1}{1 - k}\right) \frac{p_{ij}}{1 - p_{ij}}$$

as is required from equation (2).

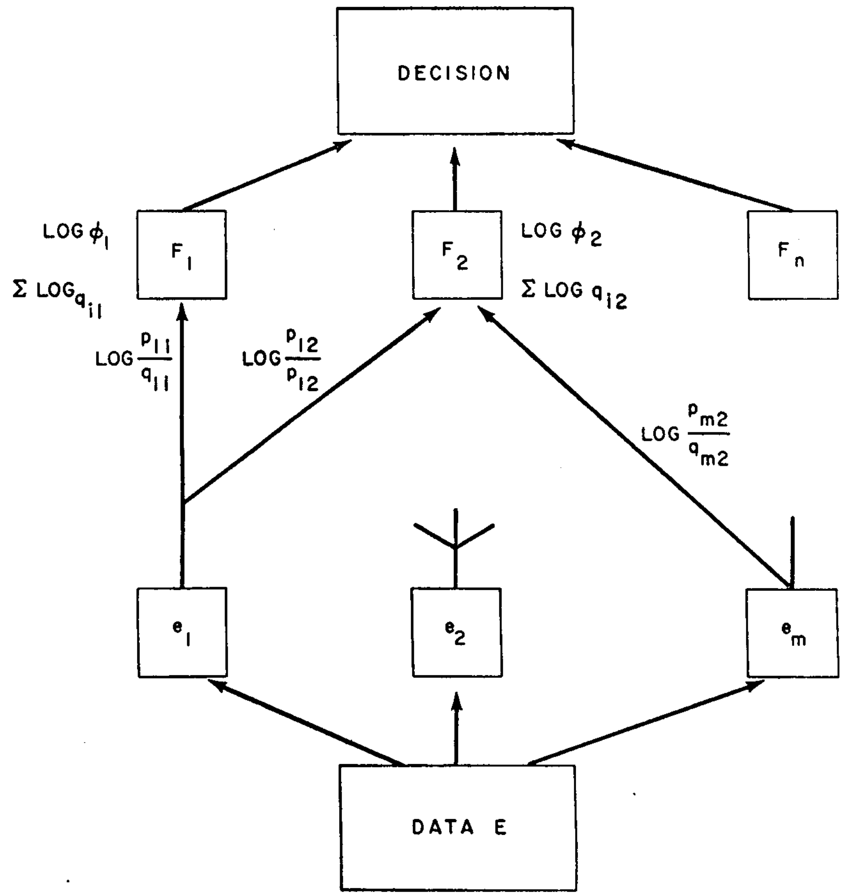

Figure 2b. Cognitive net II

The most obvious extension of our analysis must be towards interdependent e_j. An obvious case is the complete equivalence of two of the e_j; here too much weight will be given to one feature.

Interpretation as a neural net

We can imagine nets which perform these calculations, different cells (or connections) accumulating results for each coefficient. The coefficients could be represented by the 'conductivity' of 'synapses'. The above learning operators then correspond to modifiers of these conductivities or weights. Each time F_j occurs, p_{ij} is reduced by some factor k (as though some chemical reaction were allowed to occur over a fixed interval). And if with F_j also occurs e_i then p_{ij} is also given a constant increment (as though some substance were allowed to flow in at constant pressure over the interval). The conduction path corresponding to ϕ_j can have the same properties but requires different connections. There are such interpretations for both *Figure 2a* and *Figure 2b* although in the latter there are some extra complications. If desired, one could add additional decays, explicitly time-dependent, to further control the manner in which more recent experience should count more heavily, and thus compensate for possible long-term changes in the environment.

These nets are highly organized and it is not easy to see why making them more complex and random would other than harm the performance. To be sure, if there were a great many properties, not highly independent, it might do little harm should some of the connections be missing. Since the original net is totally connected, any random net (with the same unidirectional structure) would correspond to a choice of some subset of its connections. And if there were a great deal of duplication in the set of properties, there might be a substantial saving here. On the other hand, with non-independent properties the conditional probability analysis is no longer sound and one could not expect to approach optimal behaviour in any such case. To extract the information optimally one must really go on to consider the higher order joint conditional probabilities.

Net-like structures have been proposed for this sort of thing[11] but we cannot see how such proposals, which seem to require a cell or connection for *each* joint event, can avoid the fatally rapid growth of net size with only modest increases in number of properties or problem difficulty. It is our firm conviction that the solution to difficult pattern recognition problems lies, not in the increased complexity, or even efficiency, of the decision learning net, but rather in schemes which can increase the descriptive power of the inputs—the discriminative experiments themselves.

HILL-CLIMBING AND SEARCH

OPTIMIZING OR HILL-CLIMBING

The preceding paragraphs discuss the learning involved in establishing norms on a given level of description. Here we generalize this technique but thereby lose its procedural neatness.

We suppose that the machine is essaying some task and attaining some success, albeit perhaps very slight. The task may be too intricate for analysis of any abstract kind; that is for the intellectual capacity of the machine. But the machine might be able to improve itself in spite of this. Let the machine make some small changes in one or a few of its parameters or controls or variables. If the performance improves, repeat the process: if not, return to the previous state and make a different small change. In the long run performance must improve to a local optimum where no small change in controls yields improvement. This technique is commonly referred to as 'hill-climbing'. Usually it is extraordinarily insensitive to the particular details of its handling, and may be the basis of most of the more extravagant claims about 'self-organizing systems'. It is, in fact, very powerful, but its success depends crucially on the conditions that (*a*) the machine has available some estimate of performance precise and accurate enough to discover improvements quite remote from the final goal and (*b*) that estimate can define a local non-zero gradient and (*c*) the performance function is roughly unimodal. These strong conditions are often overlooked.

If we cannot find such a sensitive performance function, it may be possible to derive one statistically, by staying in one state long enough to get a good estimate of performance over many tasks. This might obtain, for example, in teaching a machine to play chess if we refuse to evaluate its moves for it and tell it only when it has won or lost a game. Such a practice would be tedious indeed, and though ultimately it is the real judge, life is too short to teach beginners that way (cf. NEWELL[12]).

Another well-known problem is the presence of 'false peaks', that is, local maxima which are not very good—this problem will be discussed below.

VARIATIONAL OPTIMIZATION BY RANDOM NETS

Suppose again that we have a machine and a critical trainer and that we wish to optimize, in some extended sense, the performance of the machine placed in a certain environment. If the machine has a variety of parameters then it may be possible to make progress by making small changes in the phase space, retracting changes not judged gainful by the trainer.

One could use for this purpose a random net varying, say, connection weightings, thresholds, latencies, or even connections. The source of the variations could be within the net elements, operating independently of one another, but all subject to reversal upon detecting a negative trainer reinforcement signal. This would be a generalization, perhaps, of the operant reinforcement scheme discussed on p. 336.

But once more it is difficult to see why a random net should be expected to compete with more orderly devices. It will be worth much effort to ensure that the parameter variations yield performance changes of real heuristic significance, at least in regard to the class of problems the machine is to be matched against. For the random net must not be thought of as giving us a search process uncommitted to any particular approach, just because it is 'random'. On the contrary, any realization embodies a particular strategy. Similarly one must never think of an 'unweighted average' as unweighted: it is *uniformly* weighted, and obviously this is a very strong constraint and not the absence of a constraint. And it is surely difficult to think of a non-trivial search problem for which one cannot imagine a search better than random. To be sure, there may be a great convenience to the use of random search. There may be far less book-keeping, for one thing; and some unsuspected correlative pitfalls can be avoided, but one must often, we venture, pay a heavy price of inefficiency.

In any case, there is a serious deficiency of the optimization scheme, in general, which we must attend to at this point. We have in mind the often over-looked problem of finding a state from which hill-climbing can be initiated.

We distinguished first between two phases of learning a task, which we call 'search' and 'hill-climbing'. Search is an initial phase where performance does not rise essentially above its initial level (in the opinion of the trainer, or other success function). Hill-climbing is the second phase wherein the machine improves the marginal success as far as possible.

During search the machine is, as it were, casting about for tools or techniques that seem profitable for the particular task. Obviously for some tasks a machine can spend much, or little, or all, of its time engaged in search. If the machine is set to work on a task it has just mastered, it need spend no time on search; given an impossible, or at least impossibly difficult, task, it may spend all its time in fruitless search. Search ends when a technique is discovered that works observably well; at that point hill-climbing can be engaged.

We can represent these notions usefully by considering the machine at any one time as a point in a machine state phase-space (see *Figure 1*). Superimposed on this space is a set of contours, depicting the performance of the machine in that space.

As the machine changes during learning, the machine state traces a path through the space. The dimensions of the state space are the controllable parameters of the machine. Small changes in the machine should correspond to small separations in the space. At any point in the space we can define a gradient of the performance in the usual way.

The previous notions of search and hill-climbing may now be seen to correspond to machine states on a zero-level plateau and on the hillsides around the summit of optimum performance.

It has often been supposed that the chief obstacle to good machine learning is the false peak problem. A false peak is a local summit that is not the true summit or optimum and a machine can become trapped on it if it refuses to try larger changes of state.

But it seems to us that by far the most general problem, in really difficult tasks, is search, or getting off the zero-level plateau. We call any point above that plateau a 'precursor', so that search is culminated by attaining a precursor.

SEARCH FOR PRECURSORS

In general, a task and the machine given it are not isolated things; rather, the machine is often given a task because its previous behaviour has shown it ready. Perhaps one may expect that the machine by only small changes from its initial state itself be one. That is, something may perhaps be guessed *a priori* about the nature of the space and placing of the summit.

But, supposing this not to be the case, we may estimate how long the machine may have to suffer in search. If the number of machine states is N, and n of them are precursors which form the foothills around the summit, one must expect the machine to spend about N/n trials before finding a precursor. N/n may be a very large number.

A good example of the distinction between search and hill-climbing is a machine programmed to learn a task where the performance is binary; that is, the machine either does or does not succeed. Here, of course, there is no hill-climbing phase at all.

We note here the applicability of a general rule: during search, machine changes should be large. For to say otherwise is to say that success will often be adjacent to failure; in fact, the opposite will be true. Let the machine then, take giant strides around its space till it happens on a precursor and enters hill-climbing! Thus search and not hill-climbing is needed against the Mesa phenomenon. (See next section.)

Chief among the large strides that a machine must make when it finds itself spending inordinately much time in search is the generation of new tools or demons. The significance of a new demon is that it alters, perhaps drastically and favourably, the entire geometry of the machine state space, so that there may be accessible precursors within a stone's throw. In the long run the successful machine must also find good demons that generate good demons (cf. references 13, 14), and the point is that changes must be tried and evaluated on different levels of processing and discourse. To jump ahead a little, a really potent random net must be able to generalize on the *kinds* of changes of connections that prove profitable; if local differences prove to be a good function for a retina to compute at some place, it should try them everywhere.

It is our conviction that all of the techniques above are sharply limited; that each will tend to fail disastrously at some stage of increase in problem difficulty. The elementary learning models and optimizers are tools capable only of managing well-abstracted data collections of modest size. 'Thinking' can be realized only in a process which can manage these tools and generate new ones when necessary.

THE MESA PHENOMENON

The work of FRIEDBERG *et al.*[15,16] illustrates some of the pitfalls that can be encountered in attempting to explore a space by hill-climbing without a preliminary search phase. The problem domain in this word was the attempt to discover *programmes* to make certain simple computations on a simple digital computer. The method was to explore the space of programmes, with reinforcement' of *instructions* appearing in partly-successful programmes. The first experiments failed dramatically[15], with the average before-success effort several orders of magnitude greater than would be obtained by completely random search. The cause of this appears to lie in the structure of the phase space of these programmes; they either work, or they don't (in the particular case in point there are no more than 3 degrees of performance) so that hill-climbing techniques are effectively precluded. (It is true that for more complex problems one could define a larger set of degrees of performance, and better implement the notion of gradient). Furthermore, a 'small' change in this kind of programme leads, in general, to no change at all in performance, or to a 'large' change; that is, to a different degree of performance. The space apparently is composed of large numbers of flat regions. The flat elevated regions might be thought of as 'table-lands' or 'mesas'. Any tendency of the search generator to make 'small' steps would then result in excessive time spent before reaching a success. While this difficulty is noted in the second paper[16], our analysis of their further experiments yields conclusions opposite to theirs. In that paper are described a series of modifications which are alleged to improve the learning part of the system, and indeed with these modifications the 'time till success' is greatly reduced (but not down to the chance level!) The several modifications seem to reflect, not improvements in the learning mechanism, but rather increases in average step-size—making effective larger changes in the programme. Each 'improvement' seems to amount to introducing an additional form of 'noise', contributing to escaping the 'mesa phenomenon'. Ultimately, it seems, the 'learning numbers' (which summarize the reinforcement history of individual instructions) come to be used, not to preserve and emphasize information concerning success, but instead to detect and remove any tendency of the machine to persist in a region.

CELL ASSEMBLIES

The proponents of random nets have been much stimulated by the suggestions of D. O. HEBB[17] that in the brain large random nets of neurons tend to form clusters of co-acting neurons called cell-assemblies; these are aroused in the presence of members of the groups of associated stimuli which caused them to form, and they may inhibit or arouse one another. The centre of activity in such a net is supposed to move around more or less jerkily from one cell assembly to another depending on the particular input stimuli and inter-assembly 'association' connections.

The reasons for enthusiasm about cell assemblies is not entirely clear to us. Each assembly, once formed, acts as a single unit; then the behaviour of the net as a whole depends on the structure and organization among the cell assemblies. One would be tempted at any stage to replace each assembly by a single simulated neuron, if one were exploring the matter with a computer (cf. ROCHESTER[18] who explores the formation of 'cell assemblies' but not their mutual relations). Now, unless one has a reason to expect good to come of this it would seem wasteful; one might even fear that the assemblies themselves would collect into larger self-exciting, cross-inhibiting groups. On the other hand, if there were some assurance that the assemblies formed did represent valuable *properties* not already accessible on the single cell level, this might be a start in the direction of net models which could work non-trivial problems. Unfortunately, there seems to be no published description of experiments which give any encouragement to this hope.

CONCLUSION

A random net may be a useful technique for small local jobs like (*a*) performing correlations or averages among inputs (*b*) classifying inputs by assigning connections or (*c*) optimizing categorizing by improving connections and weights of connections.

One should not expect more complex behaviour from a random net than this without some adequate theory to suggest and propose it.

REFERENCES

1 SAMUEL, A. 'Some Studies in Machine Learning Using the Game of Checkers', I.B.M. Journal of Research and Development, Vol. 3, No. 3, July 1959

BUSH, R. R. and MOSTELLER, F. *Stochastic Models for Learning*, New York, Wiley 1955

3 MINSKY, M. 'Neural Analog Nets', Ph.D. Thesis, Princeton University, available University Microfilms, Ann Arbor, Michigan

4 FARLEY, B. G. and CLARK, W. A. 'Simulation of Self-Organizing Systems by Digital Computer', Trans. of I.R.E. P.G.I.T., U.S.A. Sept. 1954

5 CLARK, W. A. and FARLEY, B. G. 'Generalization of Pattern Recognition in Self-Organizing Systems', Proc. of Western Joint Computer Conference, U.S.A. 1955

6 DOYLE, W. 'Recognition of Sloppy, Hand-Printed Characters', Proc. of Western Joint Computer Conference, U.S.A. 1959

7 BLEDSOE, W. W. and BROWNING, I. 'Pattern Recognition and Reading by Machine', Proc. of Eastern Joint Computer Conference, I.R.E., U.S.A. 1959

[8] Rosenblatt, F. 'Two Theorems of Statistical Separability, . . . ', Proc. Symp. Mech. Thought Processes, National Physical Laboratory, Britain, Nov. 1958
[9] Roberts, L. 'Pattern Recognition with an Adaptive Network', Proc. of Western Joint Computer Conference, I.R.E., U.S.A. 1960
[10] Selfridge, O. G. 'Pandemonium: A Paradigm for Learning', Proc. Symp. Mech. Thought Processes, National Physical Laboratory, Britain, Nov. 1958
[11] Uttley, A. M. 'Conditional Probability Machines', *Automata Studies*, Princeton, 1956
[12] Newell, A. 'The Chess Machine', Proc. of Western Joint Computer Conference, I.R.E., U.S.A. 1955
[13] Fisher, R. A. *Genetical Theory of Natural Selection*, New York, Dover Publications, 1959
[14] Darlington, C. D. *The Evolution of Genetic Systems*, New York, Basic Books, 1958
[15] Friedberg, R. M. 'A Learning Machine, Part I', *I.B.M. Journal of Research and Development*, Vol. 2, No. 2, 1958
[16] Friedberg, R. M. *et al.* 'A Learning Machine, Part II', *I.B.M. Journal of Research and Development*, Vol. 3, No. 3, July 1959
[17] Hebb, D. O. *Organization of Behaviour*, New York, Wiley, 1949
[18] Rochester, N. *et al.* 'Tests on a Cell Assembly Theory, etc.', I.R.E., IT-2 (U.S.A.) No. 3, Sept. 1956

Printed in Northern Ireland at The Universities Press, Belfast.