COMPUTER SIMULATION STUDIES OF NORMAL AND ABNORMAL NEURAL NETS

Basim A. Aziz Assaf

A Thesis

in

The Department

of

Physics

Presented in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy at
Concordia University
Montreal, Canada

September, 1974

ABSTRACT

Computer Simulation Studies of Normal and Abnormal Neural Nets

By

Basim A. Aziz Assaf

Computer simulation studies of the behaviour of randomly-connected artificial neural nets is presented. These studies show that nets connected according to the same distribution law are characterized by different behavioural patterns. The studies also indicate that these differences in behaviour can be traced to the different microscopic connectivities of these nets. Statistical analysis of the frequency distribution of the connections of the cells in these nets seem to support the view that the differences in the microscopic connectivity patterns of these nets lead to different behavioural characteristics. In this study the EEG histogram of the nets is used to classify them into normal and abnormal nets. The study also shows that some other tests can be used successfully to distinguish between the normal and the abnormal behaviour of the nets.

ACKNOWLEDGEMENTS

The author would like to express his deepest thanks and appreciation to his advisor, Dr. P. Anninos, whose guidance, encouragement and active participation made this work possible. He would also like to thank the members of his examining committee for reading this thesis and offering their comments. In particular, he would like to thank Dr. S. Raman for his suggestions regarding the addition of the theoretical section of the EEG activity. He would also like to thank his graduate programme advisor, Dr. D. Charlton, for his patience and understanding and his previous research advisor Dr. A. Smith. The author also wishes to thank Mr. V. McLeod of the Computer Science Department for his help with the programming and with the use of the overlays.

The author also wishes to express his thanks and appreciation to his wife for her patience and encouragement. he would also like to thank his friends Mr. and Mrs. T. Al-Shaikhly for their help in drawing some of the figures.

TABLE OF CONTENTS

ILLUSTRATIONS

Figure

LIST OF SYMBOLS

A	The number of neurons in the net.
r	The refractory period.
τ	The synaptic delay.
h	The fraction of inhibitory neurons in the net.
μ^+	Average number of axon branches emanating from an excitatory neuron.
μ^-	Average number of axon branches emanating from an inhibitory neuron.
K^+	Average EPsP produced by an excitatory neuron in arbitrary units of amplitude.
K^-	Average IPsP produced by an inhibitory neuron in arbitrary units of amplitude.
θ	Firing threshold of the neurons.
n	The minimum number of EPsP's necessary to trigger a neuron
α_n	The fraction of active neurons in the net at $\tau=n\tau$.
S	Standard deviation.
$\bar{X}$	The weighted mean.
σ	The fraction of external incoming connections.

CHAPTER I

INTRODUCTION

This thesis is concerned with the normal and abnormal operation of artificial neural nets and the relationship between the structure and function of these nets using computer simulation.

Nerve net modelling has been a subject of extensive research in recent years. In 1943, McCullock and Pitts used Boolean algebra and they applied symbolic logic to neural functioning by exploring the logical capabilities of a deterministically connected network of formal neurons. Rochester et al (1956) used computer simulation techniques to study a network of q u a s i-randomly connected neurons consisting of axons with excitatory and inhibitory inputs and demonstrated that the system exhibited reverberations in response to inputs. Farley and Clark (1961) carried out a similar study in which they simulated a net of 36 x 36 neurons in which the interconnections were specified by a two-dimensional probability distribution. They included temporal and spatial summation but did not include inhibitory connections. The result of their work showed that a randomly connected network can exhibit sustained oscillation under certain conditions. They also confirmed Burele's mathematical analysis (1956) that the activity in the net could lead to either saturation or quiescence.

Minsky (1956) studied the properties of randomly connected nets and Von Neumann (1958) observed the analogy between neurons and the logical elements in a computer, while Lewis (1964) designed an electronic model consisting of a set of active non-linear electronic circuits connected in parallel to reproduce the physiological data published by Hodgkin and Huxley (1952) about the changes in the axon membrane current of the squid. Harth and Edgar (1967) investigated the cognitive functions of highly damped randomly connected neural nets and found close analogies between the function of the net and the associated functions of the cerebral cortex.

Other theoretical works on network modelling were performed by Cainaniello et al (1967) who investigated collective modes of excitation in neural networks and the conditions leading to them. Wiener (1965) and also Cowan (1968) used the techniques of statistical mechanics and Ricciardi and Umezawa (1967) applied the formalism of the many-body problem to describe neural nets.

Smith and Davidson (1962) used a probablistic approach toward the analysis of networks of simulated neurons having properties similar to those of biological neurons. By using both digital computer simulations and theoretical analysis they showed that those networks were capable of supporting self-maintained activities.

Anninos et al (1970) studied the dynamic properties of probablistic nets using analytical methods and computer simulation for steady and slowly varying excitatory or inhibitory inputs and found out that there was a high probability for the existence of cyclic activity within the net. Harth et al (1970) used the mathematical method and the numeric results of Anninos et al (1970) to view the central nervous system (CNS) as a network made up of basic building blocks called the netlets out of which complex nets may be assembled. They presented a model based on this and on other anatomical and physiological data and applied the concepts of netlet interaction to information processing in the cortex.

Harth et al (1974) presented a dynamic model based on the neural net model of Anninos et al (1970) and Harth et al (1970) to explain the neuromuscular mechanisms involved in controlling the swimming escape sequence of the mollusk Tritonia and showed that the dynamic characteristics of the escape can be reproduced by a neural net model. They also showed that the functioning of the model depended on the macrostates of the neural population rather than on the detailed spatially and temporally defined microstates.

The effect of structure on the function of neural nets was investigated by Anninos and Elul (1974a) in which a theoretical analysis was made to study the effect of different connectivity laws (Poisson and Gaussian) on the dynamic behaviour of the nets. They found that nets with connections distributed according to the Poisson probability law exhibited sustained activity, whereas nets connected according to the normal probability law were not capable of sustained activity.

The use of network modelling to explain pathological and abnormal activities and to relate anatomical configurations to network activity has been investigated by very few workers.

Dichter and Spencer (1969) used a model consisting of an array of elements connected to one another through both positive and negative feedback of various strengths to explain the abnormal activity in the penicillin focus and demonstrated that a network of neurons could generate triggered self-limited responses showing many of the features of the experimental ictal discharges of the penicillin focus.

Rashevsky (1971a, 1971b, 1972) used models consisting of chains of neuron circuits acting as a centre for the production of spontaneous sustained non-periodic fluctuations of excitation and inhibition. From his studies of the role played by these circuits, he concluded that the mechanism of these circuits could serve as the origin of pathological fluctuations such as epilepsy and depression and of normal and creative thought.

Cyrulnik et al (1974) showed that a randomly interconnected neural net was capable of exhibiting the features of Parkinson's disease like cogwheel rigidity, resting tremor and dysdiadochokinesis.

The relationship between the anatomical structure and the normal and abnormal functioning of the mammalian nervous system is still largely unknown due to the complex structures of the mammalian brain and to the difficulty involved in such studies. One possible approach to this problem is through the use of nerve net models. In this thesis we are going to study the effect of the microscopic structure of a randomly interconnected artificial neural net on its function and examine whether or not differences in the microscopic structure and connectivities of these neural nets lead to differences in the operation and behaviour of these nets.

The general properties of the model we are going to use in this thesis are essentially the same as those used by Harth et al (1970). Whereas they were interested in the effect of varying the firing threshold of the neurons on the dynamic behaviour of neural nets we are concerned with the effect of the variations of microscopic structure on the dynamic behaviour of the nets. This is done by examining the effect of different connectivity patterns on network activity when the firing threshold of the neurons is kept fixed. Our main objectives are:

1. To compare the EEG histograms of the artificial neural nets with the normal distribution curve. The X^2 goodness-of-fit method will be employed to determine how well the EEG histogram of each net fits the normal distribution.
2. To classify the nets into normal and abnormal nets according to criterion 1. A net whose EEG histogram fits the normal distribution curve will be considered normal.
3. To identify the effect of evoked potential applied to the normal and abnormal nets on the cyclic activity and the number of synaptic delays before the start

of the cyclic activity. Our purpose is to observe if a difference in the reaction of the normal and abnormal nets exists and if this difference can be used as a criterion in classifying the nets.

4. To compare the microscopic connectivities of different randomly interconnected artificial neural nets and observe if differences in the connectivities of the normal and abnormal neural nets exist. The comparison will be made by constructing the frequency distribution curves of the number of the excitatory, the inhibitory, and the total number of incoming connections to the neurons in the nets employing the statistical method of moments to those curves.

A brief introduction to the fundamental neurophysiological properties of the neurons will be given in Chapter II. This chapter outlines the basic structural features and some of the fundamental concepts of the biological neurons that will be used in our model. Our assumptions regarding the neural net model along with the basic assumptions of Harth et al (1970) are discussed in Chapter III. This chapter also introduces a brief outline of the dynamics of single isolated neural nets and a description of the main features of the digital computer program used in this study for simulating neural networks. The results of the computer simulation are reported in Chapter IV along with the frequency distribution curves of the number of excitatory, inhibitory, and the total number of incoming connections to the neurons in the net. In Chapter V the criteria used to classify different neural nets are disccused. The criteria used in this study include the gaussian character of the EEG histograms and the presence of cyclic activity. A theoretical model of the gaussian character of the EEG is also presented in that chapter along with the discussion of the results reported in Chapter IV.

CHAPTER II

NEUROPHYSIOLOGY

The basic functional and morphologic unit of the nervous system is the neuron. It functions as an integrator, conductor, and transmitter of coded information, and through its processes forms an interconnected segment in the network of the nervous system. It consists of three regions on the basis of different electrical characteristics (Ruch and Patton, 1965). The first region comprises the cell body and the dendrites which conduct impulses from other nerve cells, and is characterized by low level potential changes. The second region consists of the axon and termination fibers, which conduct impulses with large amplitudes that are essentially identical, away from the cell body. The axon hillock, which consists of a conical elevation of the cell body from which the axon arises, comprises the third region. In this region the incoming impulses are integrated and the outgoing ones are initiated.

Although neurons differ widely in their size, shape, and the arrangements of their dendrites, they all have the fundamental functional properties of reacting to stimuli, transmitting excitations, and influencing other neurons or receptors (Noback, 1967).

A neuron can be activated by a stimulus coming from another neuron or a stimulus applied to a specialized receptor (House and Pansky, 1965). If a stimulus applied to the axon of a neuron lowers the membrane potential to a certain critical level called the threshold level value, a nerve impulse called the "action potential" will be produced (Katz, 1966). This nerve impulse propagates along the axon membrane without attenuation to other neurons.

The membrane potential, which is also called the resting potential, is maintained by the ionic concentration differences inside and outside the neuron. There is ten times greater concentration of (Na^+) ions outside the membrane than inside (Eccles, 1965), and on the inside the concentration of (K^+) ions is 20 - 50 times more than outside. There is also more concentration of (Cl^-) ions outside the membrane than inside. The distribution of (K^+) and (Cl^-) ions are roughly in equilibrium with the membrane potential which is about -70 to -90 millivolts (the inside negative); however, the (Na^+) ions are out of balance with the membrane potential because of the different concentrations of (Na^+) ions inside and outside the membrane. This creates a tremendous pressure for the flow of (Na^+) ions from the outside to the inside of the membrane. However, the resting membrane is impermeable to the (Na^+) ions in the absence of a stimulus and becomes highly permeable to the (Na^+) ions only when stimulated (Noback, 1967). With this increased permeability the (Na^+)

ions flow to the inside balancing and then reversing the polarity of the membrane potential. This depolarization spreads along the nerve fiber independently of the initial stimulus (Woodburne, 1967). Following this, a reduction in the entry of (Na^+) ions occurs followed by the flow of (K^+) ions to the outside. Then the (Na^+) ions are pumped out and the (K^+) ions are pumped into the neuron, and these differential concentrations are maintained again to produce the resting membrane potential.

The junction between the axon of one neuron and the dendrites or cell body of another neuron is called the synapse. It is characterized by a small gap or synaptic cleft between the presynaptic and postsynaptic membrane (Truex and Carpenter, 1969). In a sequence of neurons, the synapses act as one-way valves, allowing conduction of the impulses in only one direction (Noback, 1967). The synapses also serve as selective routing mechanisms and make possible selective excitation and inhibition (Woodburne, 1967).

The nerve impulse travelling along the axon membrane triggers the release of a chemical transmitter substance upon arrival at the presynaptic axonal termination (synaptic knob). This chemical transmitter substance is released from the surface of the presynaptic membrane. It diffuses across the synaptic cleft, and after a delay, called the synaptic delay τ, (Eccles, 1964) of 0.5 to 0.8 milliseconds, it causes a change in the potential of the postsynaptic membrane. This potential change is termed the postsynaptic potential (PsP). It is a smooth, low level, graded, local response that can be either excitatory or inhibitory. When the postsynaptic membrane responds to the transmitter substances by lowering its membrane potential (the potential is driven toward or beyond threshold) an excitatory postsynaptic potential (EPsP) results. In contrast, an inhibitory postsynaptic potential (IPsP) results when the postsynaptic membrane potential increases. (The postsynaptic membrane potential is driven toward a subthreshold level where no firing will occur (Katz, 1966)).

One of the important properties of synapses is summation which is an expression of the accumulative effects of a number of stimuli on a neuron (Noback, 1967). The summation of many stimuli received almost simultaneously at different locations on the postsynaptic membrane is called spatial summation. Another type of summation is called temporal summation which involves the addition of repetitive synaptic potentials generated by a single presynaptic neuron (Noback, 1967). The time that a given subthreshold PsP will persist is called the summation time (Purpura, 1965).

When the sum of the PsP's at the hillock exceeds a certain critical value "the threshold potential", a change in the permeability of the membrane occurs and a spike ensues and the neuron fires; otherwise, the neuron will not fire.

This is known as the "all-or-none" law, which means that a minimal strength of stimulus is required to evoke the propagation of the action potential, and when this threshold is reached, any additional increase in intensity of the stimulus has no effect on the amplitude or duration of the action potential (Rugh et al, 1965). Therefore, in receiving a stimulus a neuron either responds by firing an impulse, if the stimulus exceeds the threshold, or it does not send any impulse at all. (It should be noted that when the neuron receives a stimulus with a subthreshold value, it responds to it locally with a small potential that decays exponentially with distance from the point of simulation (Katz, 1966)). Following the initiation of the nerve impulse, the axon becomes refractory for about one millisecond, during which no stimulus, no matter how strong, can elicit an impulse. This refractory period is called the absolute refractory period, and is immediately followed by another refractory period called the relative refractory period. During this period an increased strength of stimulus would be required to fire a neuron. The refractory period lasts for a few milliseconds (Katz, 1961).

The neurons of the nervous system are organized in sequences of cells called neuron circuits. These circuits which are found in all levels of the central nervous system may be divided into five different categories (Noback, 1967):

1) The two-neuron chain.

2) The simple open circuit, which consists of a chain of neurons connected together in such a way that no neuron is connected (directly or indirectly) through an axon with a prior neuron in the chain.

3) The simple closed circuit which is formed when a neuron in the chain connects to a prior neuron. This circuit represents a simple feedback circuit in which an efferent neuron may influence itself.

4) The open multiple-chain circuit consisting of many neurons linked together and arranged in parallel.

5) The closed multiple-chain circuit which consists of many closed multisynaptic chains that form feedback circuits. These feedback loops could permit the reverberation of impulses which raises or lowers the excitability of various neurons in the chain.

The neural net model we are studying in this thesis is composed of a collection of closed and open multiple-chain neuron circuits and makes use of many of the properties of the biological neuron mentioned in this chapter.

CHAPTER III

THE NEURAL NET MODEL

A. The Netlet Model

The neural net model we are using in this research was originally developed by Harth and Edgar, 1967, and subsequently studied, used, and further developed by many workers (Harth et al, 1970; Anninos et al, 1970; Anninos and Elul, 1974). The basic assumptions of the netlet model are (Harth et al, 1970):

1) The structure of the nervous system could be approximated by sets of discrete populations of randomly interconnected neuron nets. These discrete nets are called netlets (Harth et al, 1970).

2) Connections between the neurons within the netlets and the distribution of efferent fibers to the netlet are chosen at random.

3) The level of activity within the netlet is considered to be the only significant dynamical variable in the netlet.

Many physiological and anatomical evidences seem to support the above mentioned assumptions. Mount Castle (1957) discovered that neurons that are located along the radial column in the somato-sensory cortex exhibit identical receptive fields. These were called "elementary units of organization". Asanuma and Sakata (1967) identified discrete colonies of neurons in the motor-sensory cortex of the cat and found that neurons belonging to a colony have projections that terminate in the same neuron pool. Many other studies (Penfield and Rasmussen, 1955; Sholl, 1956; Jasper, et al 1960; Morrell, 1961; Penfield and Perott, 1963: and Colonnier, 1967) have shown that simulation and recall of complex sensory events do not require a delicate special patterning of incident excitation on one particular neuron, but that reproducible responses could be elicited if the location of the stimulus was specified only to within a millimeter. These experiments have shown that the neuronal activity of the net can be set up simply by stimulating a given region. They also indicated that a specific response to a stimulus is not very dependent on the location of the neuron being simulated.

In addition to the above three basic assumptions of the netlet model, the following assumptions are included in our model:

4) The total input to any cell within the net is expressed as an algebraic sum of both excitatory and inhibitory inputs.

5) All the cells in the net have the same synaptic delay, τ.

The neurons fire only at integral multiples of τ.

6) Dendritic and axonal transmission delays over small distances are assumed to be negligible compared with synaptic delays.

7) Summation is essentially spatial, i.e. the effect of the excitatory postsynaptic potentials and the inhibitory postsynaptic potentials remain for a period less than the synaptic delay τ.

8) The absolute refractory period r, of any neuron in the net varies between τ and 2τ. This assumption is not unreasonable since the refractory periods take one or two milliseconds (Katz, 1961), and the synaptic delay τ, takes 0.5 to .8 milliseconds (Eccles, 1964). Relative refractory periods will not be considered in this study.

9) The refractoriness of a neuron at time τ is independent of the probability that the neuron receives threshold excitations at the same time. Although this is not true in general for biological neurons, this dependence is very small (Wilson and Cowan, 1972).

10) Each neuron in the net goes from the resting or inactive state to the firing or active state whenever the sum of all the excitatory and inhibitory postsynaptic potentials (PsP's) arriving at the neuron exceeds the threshold

value θ and providing that the neuron is not refractory.

11) The firing thresholds of the neurons are returned to normal 2τ after the neuron fires.

12) A neuron can be either excitatory or inhibitory. An excitatory neuron generates excitatory postsynaptic potentials (EPsP) only, and the axon branches of an inhibitory neuron generate inhibitory postsynaptic potentials (IPsP) only.

13) The axonal connections emanating from each neuron in the net are randomly distributed among all other neurons in the same net. The number of the axonal connections varies between two limits. The average number of axon branches is denoted by μ^+ for an excitatory neuron and μ^- for an inhibitory neuron.

14) The PsP's are produced by an active neuron after a time interval equal to the synaptic delay τ.

15) Inhibitory neurons within the net are chosen randomly. The fraction of the total number of the inhibitory neurons in the net is given by (h).

The connectivity of the net can be completely described by a connectivity matrix $\{K_{ij}\}$ made up of coupling coefficients k_{ij}'s (Harth and Edgar, 1967). The coupling coefficient represents the size of the postsynaptic potential in arbitrary units from the jth neuron to the ith neuron. The absence of a synaptic link between the ith and the jth neurons is characterized by $k_{ij} = 0$.

With the network assumption stated above, we can define the activity α_n as the fractional number of neurons that fire at $\tau = n\tau$. If the total number of neurons in the net is A, then the total number of active neurons at any time $\tau = n\tau$ is given by $A\alpha_n$. The activity α_{n+1} in the net at $\tau = (n+1)\tau$ is entirely determined by the activity α_n at $\tau = n\tau$. This is due to our assumptions regarding the refractoriness of the neuron and the summation time mentioned above. A neuron that fires at time $\tau = n\tau$ will not be sensitive at $\tau = (n+1)\tau$ to any stimulus, and the PsP's on any neuron at $\tau = (n+1)\tau$ depend only on the firing record of the net at $\tau = n\tau$.

B. Dynamics of Single Isolated Neural Nets

In this section we present some of the results described by Anninos et al (1970) regarding the dynamic properties of a randomly connected isolated neural net.

The structural parameters of the nets include the average number of neurons receiving postsynaptic potentials from an excitatory neuron (μ^+), the average number of neurons receiving postsynaptic potentials from an inhibitory neuron (μ^-), the average value of the PsP produced by an excitatory neuron (K^+), the value of the PsP produced by an inhibitory neuron (K^-), and the minimum numbers of EPsP's required to trigger a neuron in the absence of inhibitory inputs (η). This quantity is defined as

$$\eta = f\ (\ \theta/K^+\ )$$

where the function $f\ (\theta/K^+)$ is defined as the smallest integer which is equal to or greater than (θ/K^+).

Fig. 1 shows a series of curves of α_{n+1} versus α_n, for an isolated net with a Poisson connectivity law and for various values of η.

In this figure three different modes of behaviour can be distinguished. These are labelled class A, B, and C. The curve corresponding to a threshold $\eta = 1$ characterizes a class A net. It has the property that for low activity α_n, the subsequent activity α_{n+1} will always be larger than α_n,

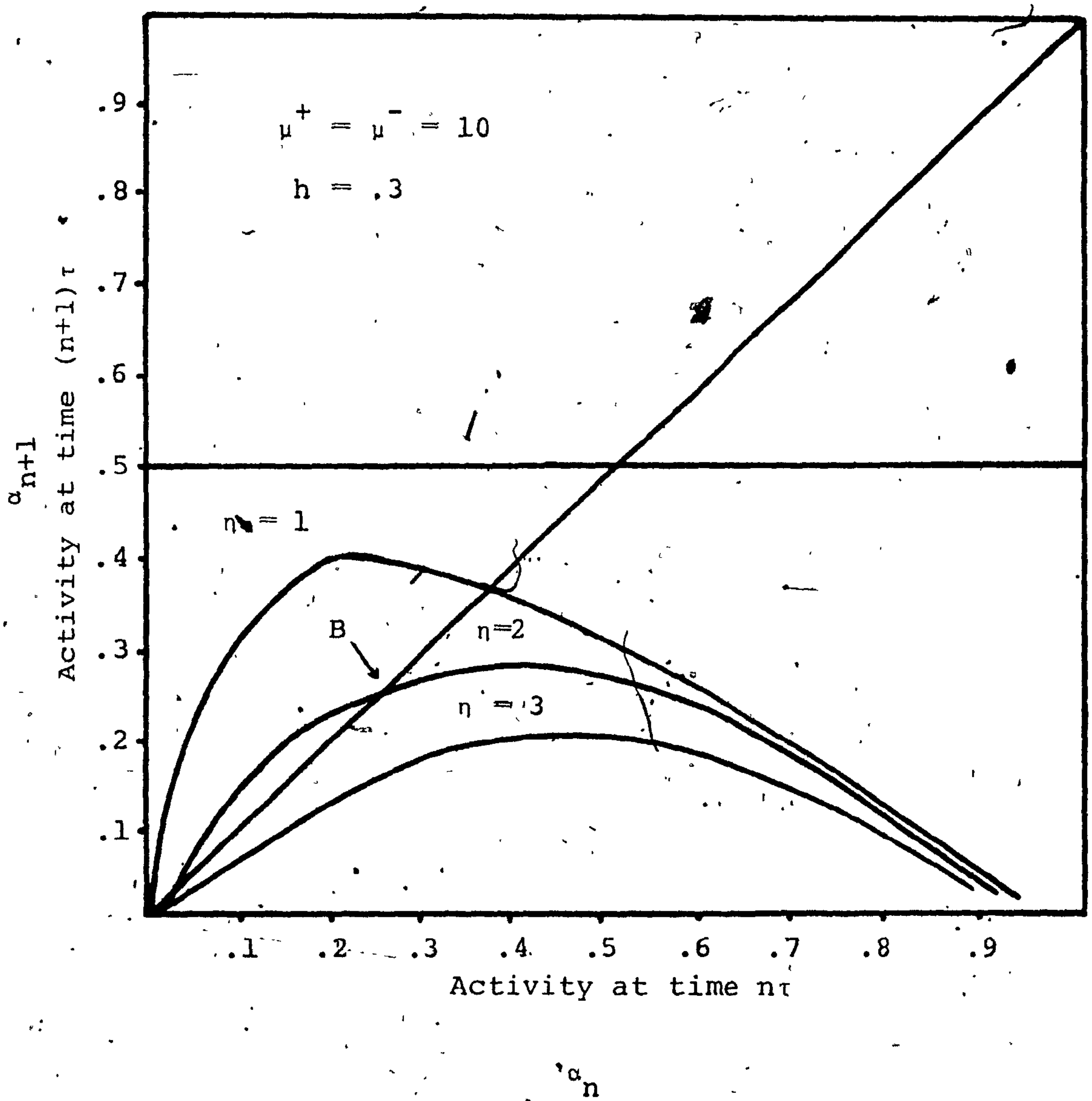
μ+ = μ- = 10
h = .3
η = 1
η=2
η = 3
B
.1
.2
.3
.4
.5
.6
.7
.8
.9
Activity at time nτ
αn
Activity at time (n+1)τ
αn+1

Fig. 1

up to the point where the curve crosses the 45° line (Harth et al, 1970). These nets can produce sustained activity because of their instability to small fluctuations for any value of the activity α_n in the region:

$$0 < \alpha_n < \alpha_B$$

A Class B net is defined as one in which there exists a threshold for being triggered into sustained activity. The curve $\eta = 2$ in Fig. 1 belongs in this category. It lies partly below the 45° line and partly above it. If the initial activity is below this threshold, then the subsequent activity will decay toward zero. However, when the initial activity is above the threshold level, the activity will be sustained.

The third curve in Fig. 1 is an example of Class C nets. These nets lie wholly below the 45° line, hence they are incapable of maintaining any form of sustained activity.

In this study we will consider Class A nets connected according to the Poisson distribution law.

C. Computer Simulation of Neural Nets

In this section we are going to point out the main features of the digital computer program used in this study for simulating neural networks.

The main program consists of two overlays. The first overlay is used to create the non-vanishing elements of the connectivity matrix $\{K_{ij}\}$. This is done by first specifying the total number of neurons in the net A, the fraction of inhibitory neurons in the net h, the minimum and maximum number of the outgoing connections for each excitatory neuron $\mu^{+}_{min.}$ and $\mu^{+}_{max.}$ respectively, and the minimum and maximum number of the outgoing connections for each inhibitory neuron $\mu^{-}_{min.}$ and $\mu^{-}_{max.}$ respectively. The specific neurons which will be inhibitory are determined by randomly selecting the appropriate number of neurons consistent with h. This information is stored and later, when the value of the coupling coefficients K is determined for a given connection, a negative sign is assigned to K whenever the connection originates from an inhibitory neuron.

The second step is to create the non-vanishing elements of the connectivity matrix, $\{K_{ij}\}$ which defines the connections between the individual elements. To do this, each non-zero element of the connectivity matrix is determined by three values: he number of the neuron from which the connection originates j, the number of the neuron on which the connection terminates i, and the value of the coupling coefficient K. This coupling coefficient K can be either positive or negative depending on whether the neuron is excitatory or inhibitory, respectively. The interconnection of the neurons in the net is accomplished by taking the neurons sequentially one by one (from j_1 to j_A) and randomly finding a terminal neuron (between i_1 and i_A) for each of its outgoing connections. When the process of interconnection is completed, the values of i, j, and K, and its sign are all stored in the computer as an array called KORK array.

In establishing the pattern of connection, the specific neurons that are inhibitory, and the number of excitatory and inhibitory connections originating from each neuron, extensive use is made of a random number generator "RANF" available in the CDC 6000 computer subroutine package at Sir George Williams University. The "RANF" subroutine generates a uniform distribution of random numbers between 0 and 1. A number selected by the user must be used as a first entry. This number determines the specific sequence of random numbers obtained from the subroutine. The number selected as a first entry together with the number of inhibitory neurons h, and the average number of outgoing excitatory and inhibitory connections μ^+ and μ^- determine the microscopic structure of the net. Changing any one of them will change the microscopic structure of a given net. Therefore, by using different initial numbers for the "RANF" subroutine, the detailed microscopic structure of the net can be modified without altering the statistical parameters μ^+, μ^-, h, K, and η. This feature of the program will be used to determine the effect of the specific microscopic net structure on the behaviour of the net.

The second overlay is used for simulating a neural network. The simulation starts by specifying the number of neurons which fire at $\tau = 0$. The neurons connected to these initially active neurons can be found from the KORK array formed by the first overlay. Those neurons which are connected to the initially active neurons receive excitatory or inhibitory inputs depending on their incoming connections. The algebraic sum of the coupling coefficients characterizing the active incoming connections represents the level of excitation in any one of these neurons. This information is stored by the program in an array called STATE. Whenever the sum exceeds the threshold, the neuron fires and after one synaptic delay it becomes the source of excitation or inhibition for all the neurons which are connected to it. If the sum of the coupling coefficients of the active connections is less than the threshold, then the neuron will not fire and all the active incoming connections return to their initial values one synaptic delay later. Therefore, excitation and inhibition last for one synaptic delay and disappear immediately thereafter.

The sum of the coupling coefficients of all the incoming connections to a neuron and its threshold θ, determines whether it will fire or not. The total value of the PsP arriving at a neuron can be represented by:

$$\sum_{j=1}^{n} k_j x_j$$

where k_j represents the value of the coupling coefficient between the jth axon branch and the neuron receiving the excitation. k_j can take any value between two limits $K^{+}_{min.}$ and $K^{+}_{max.}$ for excitatory neurons and $K^{-}_{min.}$ and $K^{-}_{max.}$ for inhibitory neurons, or can be fixed at one value. x_i is a binary number that can be either 1 or 0 depending on whether or not the jth axon is active.

If $$\sum_{j=1}^{n} k_j x_j - \theta \geq 0$$

then the neuron fires, i.e. its x will be equal to 1 and it will send out a signal which can be either excitatory or inhibitory, depending on its coupling coefficient k.

The McCullock-Pitts neuron model (1943) is similar to our model with the exception that the coupling coefficient in their model was always fixed at a constant level, whereas our model allows for the variation of the coupling coefficient between any two limits specified by the user.

The fact that a neuron fired in a particular firing cycle is recorded in an array called the FIRE array in which the elements could assume a value of 1 if the neuron fired or 0 if it did not fire. The number of the neurons that fired during each firing cycle is counted and stored in an array called RECORD. The successive elements of this array contain the neural activity in successive firing cycles. The instantaneous sum of the PsP's of all the cells in the net (the gross electroencephalogram of the net, EEG) is also stored after each firing cycle.

When a neuron fires, its threshold is raised to a maximum value (32000) and is then increased by 1 in each firing cycle until the number of the synaptic delays elapsed since the neuron fired is equal to the absolute refractory period specified by the user. During this period, the neuron can receive excitation but cannot fire due to its high threshold. Following this period, the neuron returns to its normal threshold and once more becomes capable of firing provided that the sum of the coupling coefficients is equal to or greater than its normal threshold.

The second overlay can also treat external inputs delivered at different time intervals for the study of the effect of evoked potentials on the behaviour of the net by specifying the number of active connections incident upon the net through a bundle of afferent fibers. Upon entering the net, each of these afferent fibers branches and makes excitatory synaptic connections with μ_o^+ different neurons if it comes from an external excitatory neuron or makes inhibitory connections with μ_o^- neurons when it comes from an external inhibitory neuron. The fraction of afferent fibers discharging at any time is denoted by σ^+ and σ^- for excitatory and inhibitory neurons respectively and the total number of active fibers is given by $\Lambda_o \sigma \mu_o$, where Λ_o is the total number of the incident fibers. The afferent fibers make random connections with the neurons in the net. The coupling coefficients of those neurons that receive outside connections increase by the coupling coefficients specified for the incident connections and if the sum is equal to and exceeds the threshold the neurons fire and then behave in exactly the same manner as if the excitation was coming from within the net.

The digital computer program used in this work is a modified version of a program written at Syracuse University by Mr. L.S. Edgar and members of the computing centre, and was used by Anninos et al (1970), and Harth et al (1970). The original program was written in Fortran with the transfer of the arrays from disk to core and vice versa accomplished with a set of subroutines written in assembler language. The program used in this thesis is entirely written in Fortran language and makes use of overlays to reduce storage requirements. Three new subroutines were added to the present program in order to compute and plot the EEG histograms and to fit them to the normal distribution curve. "RANDU" subroutine available from the IBM scientific subroutines package was used in the original program for the generation of random numbers. In our program the random numbers are generated with the aid of "RANF" subroutine available to the users of the CDC 6000 computer systems. A listing of the simulation program used in this thesis is given in the Appendix.

CHAPTER IV

COMPUTER SIMULATION RESULTS

In this chapter, we shall present the results obtained by simulating a neural network. Unless otherwise stated, the following parameters were used in all the simulation experiments.

$$\mu^{-}_{min} = \mu^{+/}_{min} = 1$$

$$\mu^{-}_{max} = \mu^{+}_{max} = 5$$

$$K_{ij} = 10$$

$$\theta = 10$$

$$\eta = 1$$

$$\alpha_o = 0.1$$

where $\alpha_o A$ = total number of neurons firing initially.

Table 1 shows the pattern of the cyclic activity

TABLE 1

CYCLIC ACTIVITY IN NEURAL NETS

Net	h	$\frac{\alpha_1}{\alpha_0}$	Period of Cycle	Total Number of Synaptic Delays Before Cycling	Initial Random Number
1	.35	1.7	2	17	27
2	.35	1.51	318	143	123456789
3	.4	1.57	16	36	27
4	.4	1.44	-	No cyclic activity	123456789
5	.4	1.17	38	29	194519463
6	.3	1.83	8	15	27
7	.3	1.68	2	31	999977775
8	.3	1.41	2	27	194519463
9	.3	1.59	2	38	187785495
10	.3	1.7	2	22	123456789
11	.35	1.39	8	40	187785495
12	.4	1.43	2	21	999977775
13	.4	1.17	38	29	194519463
14	.4	1.31	24	40	187785495

for a sample of fourteen nets out of a total of 30 different nets considered in this study along with the fraction of inhibitory neurons (h) in each, the ratio of the fraction of active neurons at $\tau = 1 (\alpha_1)$ to the fraction of active neurons at $\tau = 0 (\alpha_0)$, the number of states in each cycle, the period of the cyclic activity in multiples of τ, and the number of synaptic delays before the beginning of the cyclic activity. From this table it can be seen that nets with the same statistical parameters but with different initial pseudo-random number generators exhibit different dynamic properties. The dynamic properties can also be altered by simply changing the fraction of the inhibitory neurons (h) and keeping the same pseudo-random number generator.

Figs. 2 and 3 show the dependence of the activity α_1 on α_0 for two of the nets considered in this study. The number of neurons firing initially was changed every run and the resulting activity, one synaptic delay after the initial activity, was observed and plotted against the initial activity. The curves show that these nets belong to Class A nets. All the nets considered in this study showed similar variations with initial acitivity, although the point where the curves cross the 45° line depended on the connectivity and the statistical parameters of each net.

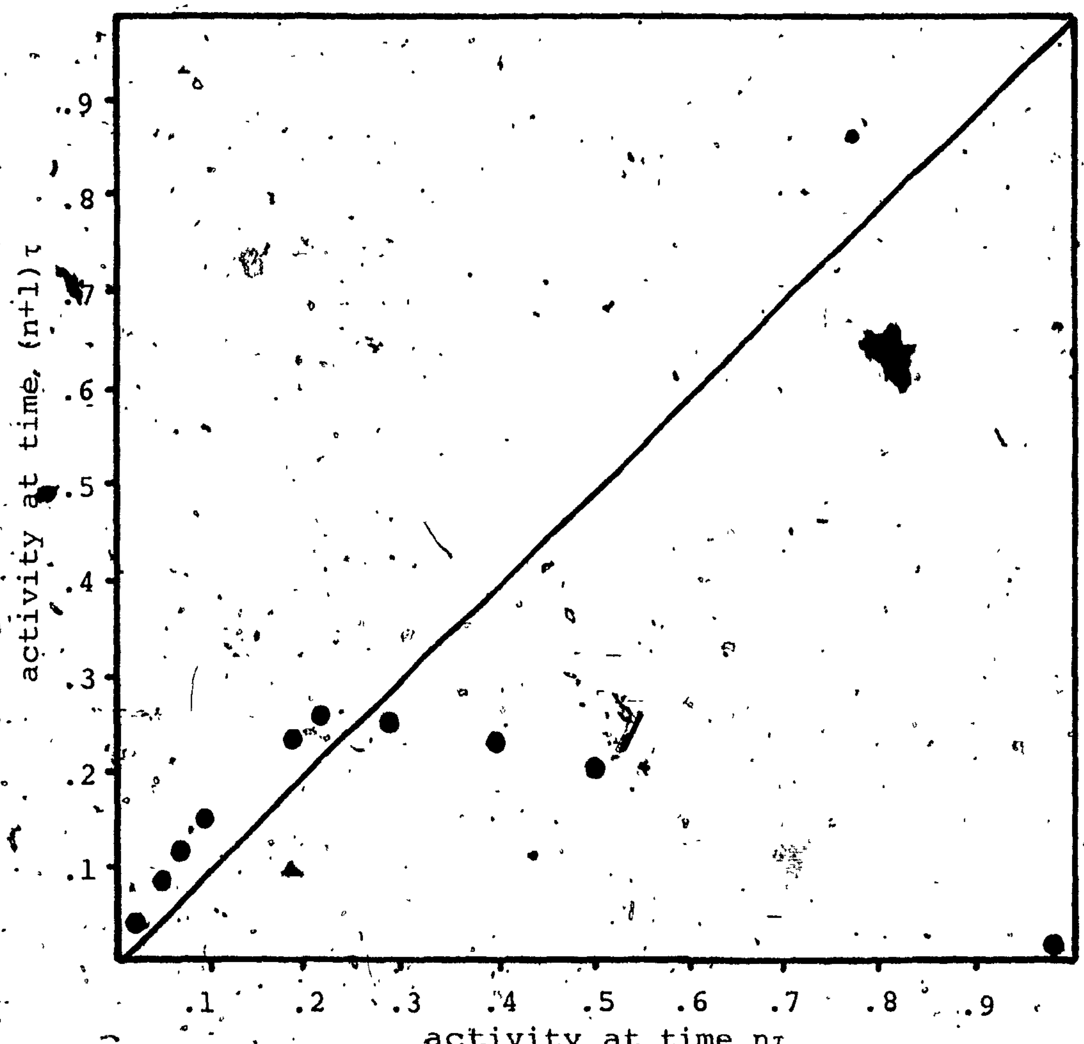

Fig. 2. α_1 vs. α_0 for Net 2

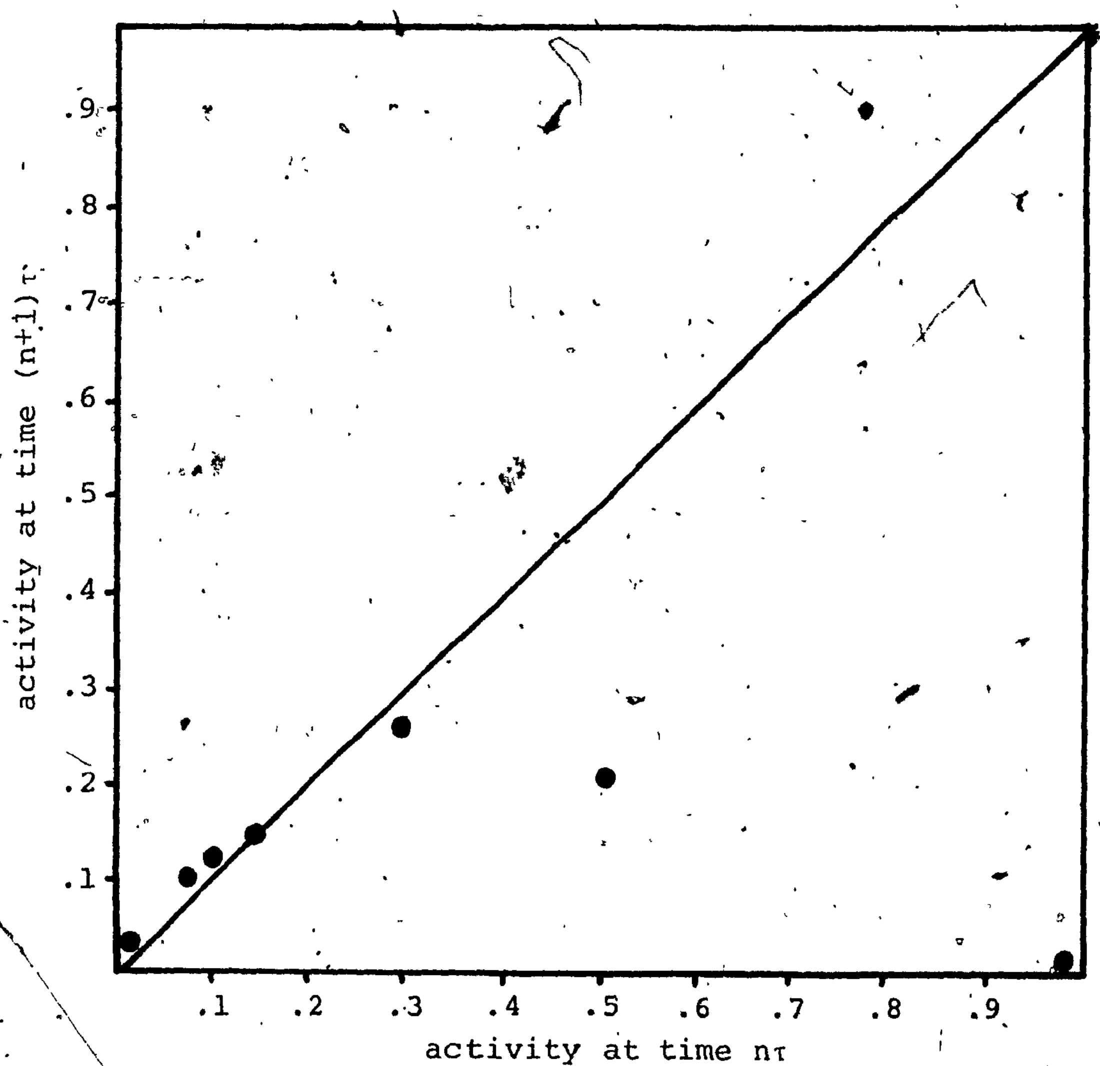

Fig. 3. α_1 vs. α_o for Net 5

Figs. 4 through 8 show the computer printout of the inhibitory arrays for each of the first five nets of Table 1. In these figures, 0 represents an excitatory neuron, while ● represents an inhibitory neuron. The neurons are arranged sequentially from left to right in 20 rows with 50 neurons in each row. The inhibitory array for each net was formed by randomly selecting a specific number of neurons consistent with the value of the fraction of inhibitory neurons (h) given at the start of the simulation run. This was then stored in the inhibitory array. The number of connections originating from each neuron in the net was then computed and the coupling coefficient assigned to them with a negative value given to all connections originating from the inhibitory neurons. The location and sequence of the inhibitory neuron was frozen for the entire run. The sequence and location of the inhibitory neurons within the net can only be changed by changing the number supplied by the operator of the simulation program as a first entry to the random number generator.

00●000●●00	0●00●00●0●	00●000000●	00●00●0●0●	00●0●00●00
00●0●0●00●	●00●0●0●00	000000000●	00000●●000	0●00●●0000
0●●00●0000	000●0●00●0	0000●00●●0	00●0000●00	0000000●0●
00●●0●00●●	0000●●000●	0000●●0●●0	000●00●0●0	000●●0●●●0
00●000●00●	000000●00●	0●000●●●0●	00●●00●000	●●●00000●0
●●●00●●000	●0●●●●0●00	0●000000●●	000●00●0●0	00●0●0000●
●000●00●00	000●0●●000	0000●●●0●●	0000000●00	0●0●00●0●0
●●0000000●	0000●●0●0●	00●00●000●	●●000●●000	0●●●0●0●0●
000●●●00●●	●●000●000●	000●●00●●0	00●000●●00	●00000000●
0●00●●0●0●	●000●00000	0●00●●●0●0	●00000●●●●	●0●●0000●0
●●0●●00●00	00●0●00●00	0●0●●●0●0●	0●00●●00●●	0●00●000●0
0●●00●●●00	00●0●000●●	000●●●0000	000000●0●●	0●0●0000●0
0●000000●0	0●0●00●●●0	00●000●000	●0●00●●000	000●0●●00●
00000●0000	0●●000●●00	000●0●00●0	0●00000●0●	0●0●●000●0
0●●0●00000	00●●0000●0	00●00●0●00	0●00●0●●●0	●000●●0●00
0●000●0●●0	0●●●●000●●	00●●●●●00●	000000●●●0	●●●0●00000
●●0000●000	0●●0●00000	00●0000000	0●00●●0●●●	00000●00●0
000●00●000	0●0●0●0●00	●0000●0●●0	00●000●●00	000●0000●0
0000●●●00●	0●00000000	●000●●0●00	0●●●00●●00	000000●00●
000●●00●00	0●●00●0000	●●0●00●0●0	0●000●0●00	000●●0000●

Fig. 4. Inhibitory Array for Net 1

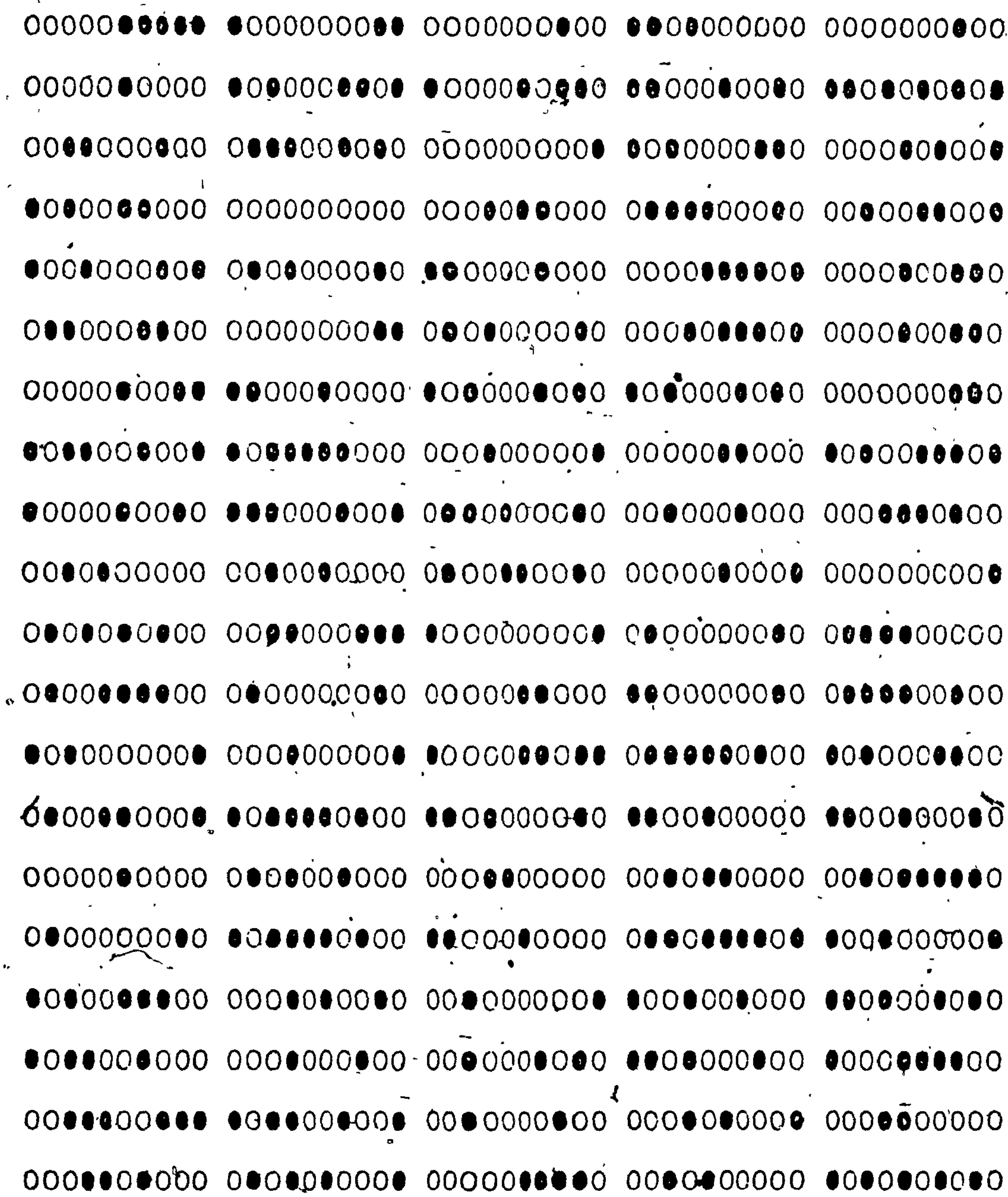

Fig. 5. Inhibitory Array for Net 2

```
00●0000●00 0●000●0●0● 00●000000● 00●00●0●0● 00●0●00●00
00●0●0●00● ●00●0●0●00 0000●0000● 00000●●000 0●0●●●0000
0●●00●0000 000●0●0●●0 000●●●0●●0 00●0000●00 0000000●0●
00●●0●00●● 000●●●000● ●●0●●●0●●0 000●00●0●0 0●0●●0●●●0
00●000●00● 000●00●00● 0●000●●●0● 0●●●00●000 ●●●●0000●0
●●●00●●000 ●0●●●●0●00 0●000000●● 000●●0●0●0 00●0●0000●
●000●00●00 ●00●0●●0●0 0000●●●0●● 0000000●00 ●●0●00●0●0
●●00●0000● 0000●●0●0● 00●00●00●● ●●00●●●000 0●●●0●0●0●
000●●●●0●● ●●000●000● 000●000●●0 00●000●●00 ●●0000000●
0●000●●0●● ●00●000000 0●00●●●●●0 ●000●0●●●● ●0●●0000●0
●●0●●00●00 0●●0●00●00 0●0●●●0●0● 0●000●00●● 0●00●000●0
0●●0●●●●00 ●0●0●000●0 000●●●0000 000000●0●● 000●[illegible]
0●000000●0 0●0●●0●●●0 ●0●00●●000 ●0●00●●0●0 000●[illegible]
00000●●000 0●●000●●00 000●0●00●0 ●●●0000●0● 000●[illegible]
0●●0●00●●0 00●00000●● 00●00●0●0● 0●00●0●●●0 ●000●●0●00
0●000●0●●0 0●●●●000●● 00●●●●●●0● 00000●●●●0 ●●●0●0000●
●●000●0000 0●●0●00000 ●0●0000000 0●●0●●0●●● ●●000●00●0
000●00●00● 0●0●0●0●00 ●0000●0●●0 00●0000●00 000●0●0●●0
0000●●●00● 0●00000000 ●000●●0●00 0●●●00●●00 000000●00●
000●●00●00 0●●00●0000 ●●0●00●0●0 0●000●0●00 000●00000●
```

Fig. 6. Inhibitory Array for Net 3

00000●●●●●	●0000000●●	0000000●00	●●0●●00000	0000●00●00
00000●0000	●0●000●●0●	●0000●0●●0	●●000●00●●	●●●●0●●●0●
00●0000●00	0●●●00●0●0	000000000●	●0●0000●●0	0000●0●00●
●0●00●●000	0000000000	●00●0●●000	0●●●●000●0	00●●0●●●0●
●●0●000●0●	0●0●0000●0	●●0000●0●0	0000●●●●0●	0000●00●●0
0●●0000●●0	0000●000●●	●●0●0000●0	0●●0●●●●0●	0000●00●●●
0000●●00●●	●●0000●000	●0●00●●0●0	●0●000●0●0	0000000●●0
●0●●●0●00●	●0●●●●●000	000●00000●	●0000●●000	●0●0●●●●0●
●0000●00●0	●●●000●00●	0●●0●000●0	00●000●000	000●●●0●00
00●0●00000	00●00●0000	0●00●●00●0	0●000●000●	000●00000●
0●0●0●●●0●	●0●●00●●●●	●0000●000●	0●●00000●0	0●●●●00000
0●●0●●●●00	0●000000●0	00000●●●0●	●●000●0●●0	0●●●●00●00
●0●00000●●	000●00000●	●0000●●0●●	0●●●●●0●00	●0●000●●00
0●00●●000●	●0●●●●0●00	●●●●0000●0	●●0●●0000●	●●00●●00●0
00●00●0000	0●0●00●000	000●●00000	00●0●●0000	●0●0●●●●●0
0●0●0000●0	●0●●●●0●00	●●000●0000	0●●0●●●●0●	●●0●00000●
●0●00●●●00	000●●●00●0	00●●00000●	●000●0●000	●●0●000●●0
●0●●00●0●0	000●000●00	00●000●0●0	●●0●000●00	●000●●●●00
00●●●00●●●	●0●●000●●●	00●0000●●0	000●0●●00●	000●●0●0●0
000●●0●0●0	0●0●0●000●	0●●0●●●●●0	00●0●00000	●0●0●●0●●0

Fig. 7. Inhibitory Array for Net 4

```
0●0000●●●0 0●●●000●●0 00●0000000 000●●0●00● ●●0●0●●●●0
●●●00000●● 0●000●●●00 0●0●0●●00● ●00●000●●0 00●0●00●●●
●0●●0●0000 ●0●●●●●●00 ●0●●000●0● 0●●0●0000● 00000●0000
●0●0●0●●00 0●0●0●●0●0 0●●0●000●0 0●00●●●0●● 0●000●●000
00000●0●●0 00●0●0000● ●0000●●●0● 0●00●●●000 ●0000●00●0
0000000●00 0●00000000 0●0000●0●0 0000000000 00000●00●0
00●0●00●●0 ●●00●000●● ●●●●000●0● 0●000●●000 000●●0●000
0●000●0000 000●0●0●●● 000000●●0● 000●0●0●●● 000●●●0●00
00●●●000●0 0●●●●0●●●0 ●0●000●00● ●00●00●00● ●00000●●00
0●00●00●●● 0●000●00●● 00●000●●0● 0●00000●00 0●●●00●●00
●●●0●00●●● 00000●●0●0 000●●0●●0● 00000●00●0 ●0●●00000●
●●●0●000●0 000●●0●●●0 000●000000 0●00000●●0 ●●●00●0●0●
0●000000●0 00●0000●0● 000●●0000● 0●00●00●●0 0●00●●0●●●
000●●0●●0● ●●●00●000● ●●0●0●0●00 0●●●00●00● 0000●000●0
0000000●0● 0●0000000● 0●0●0●●0●● ●00●●00000 0●●00●●●●●
0●0●●0●0●● 0●●000●●00 ●000000000 ●00000●●0● ●●00●0●●●●
●00000●000 ●●●0●0●000 ●0●●0●●000 0●●●●00●00 0●0●00000●
000●000●●0 ●00●●0000● 0●●00●●●0● 000●●●00●0 0●000●000●
0000●00●0● ●●●000●0●● 0●0●000●●0 ●000●0●●●0 0●●●00●0●0
00●●●●●0●● 0●00●●0000 0●●00●●000 0●00000●00 ●000●00●●0
```

Fig. 8. Inhibitory Array for Net 5

The total number of the excitatory and inhibitory incoming connections and the total number of the inhibitory incoming connections incident on each neuron were printed by the computer from the KORK array with the aid of one of the Fortran subroutines AMINO available for the users of the computer at this University. From this information, the neurons with a specific number of incoming connections were obtained and their frequency distribution calculated. Similarly, the frequency distribution of the total number of inhibitory incoming connections was calculated. A third group of frequency distribution data can be obtained by subtracting the number of the inhibitory incoming connections from the total number of incoming connections for each net. The difference between the two numbers for each neuron represented the total number of excitatory connections incident on each neuron.

The three groups of frequency distribution data are shown graphically in Figs. 9 to 23, inclusive, for each of the five nets listed in table 1. In these figures, the ordinate represents the total number of neurons that have a particular connection. Therefore, the height of the line x=1 represents the number of neurons receiving one connection only. Similarly, the height of the lines above x=2, x=3, etc., represent the total number of neurons receiving two, three, or more connections respectively.

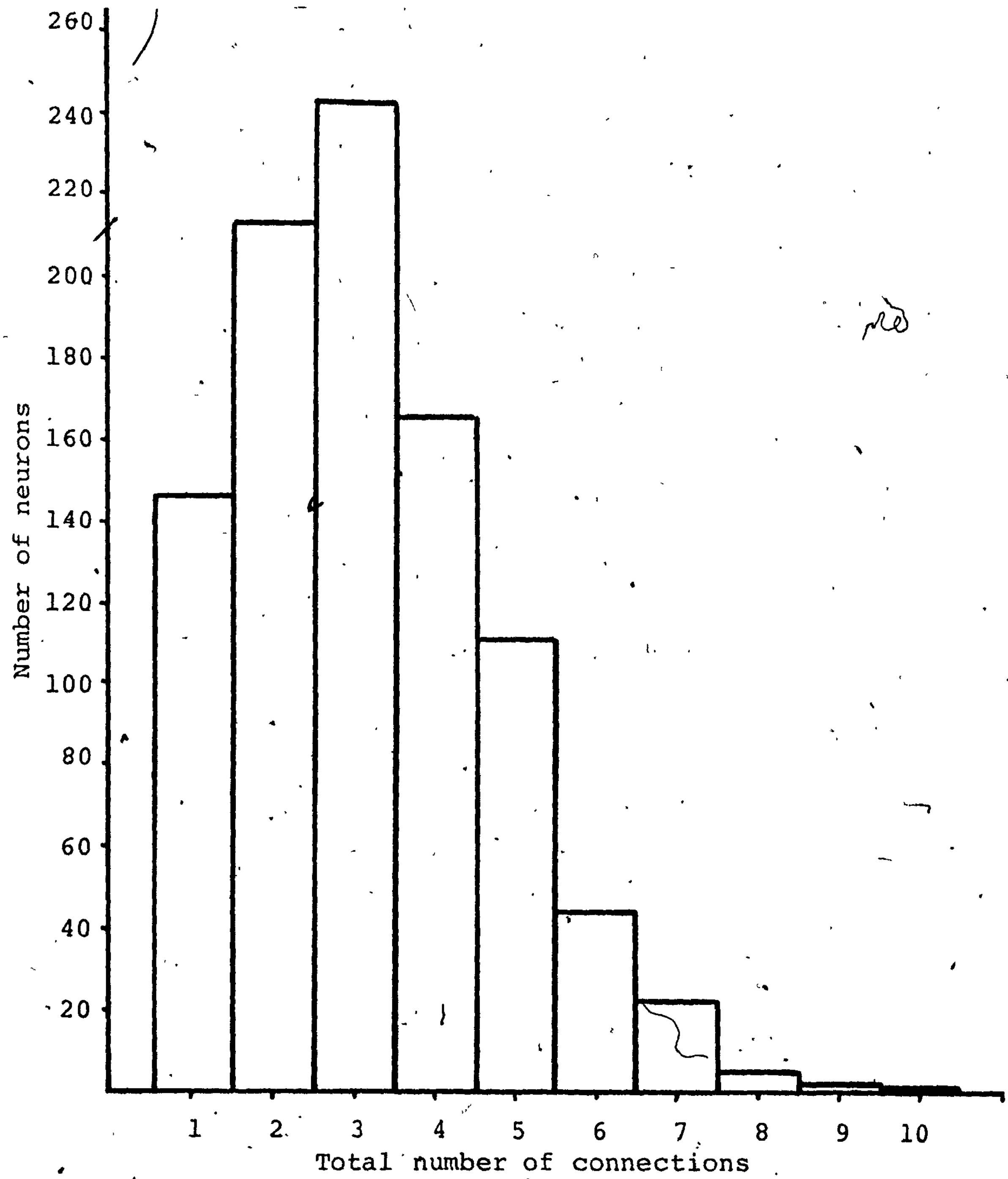

Fig. 9: Total Number of Incoming Connections for Net 1

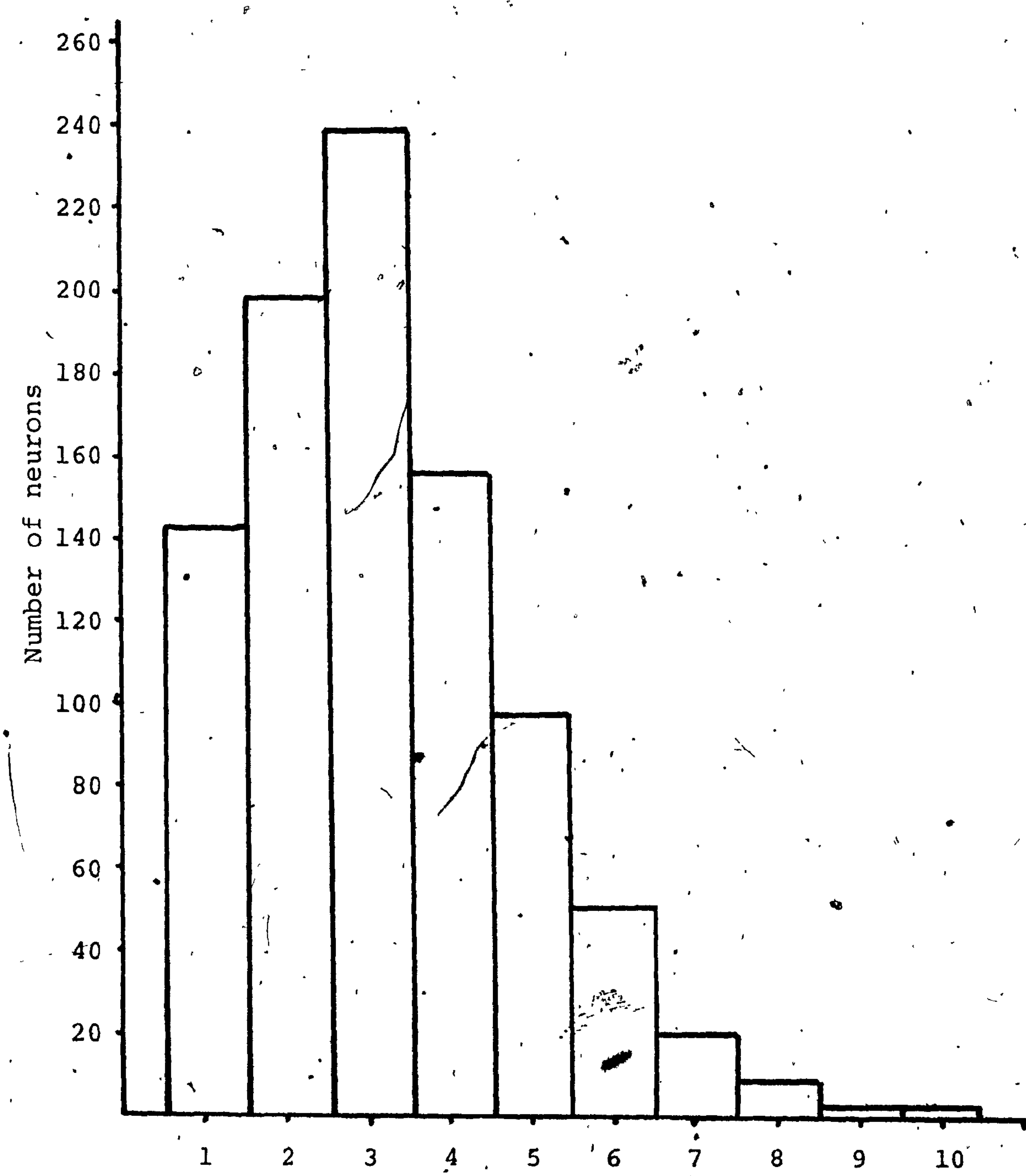

Fig. 10: Total Number of Incoming Connections for Net 2

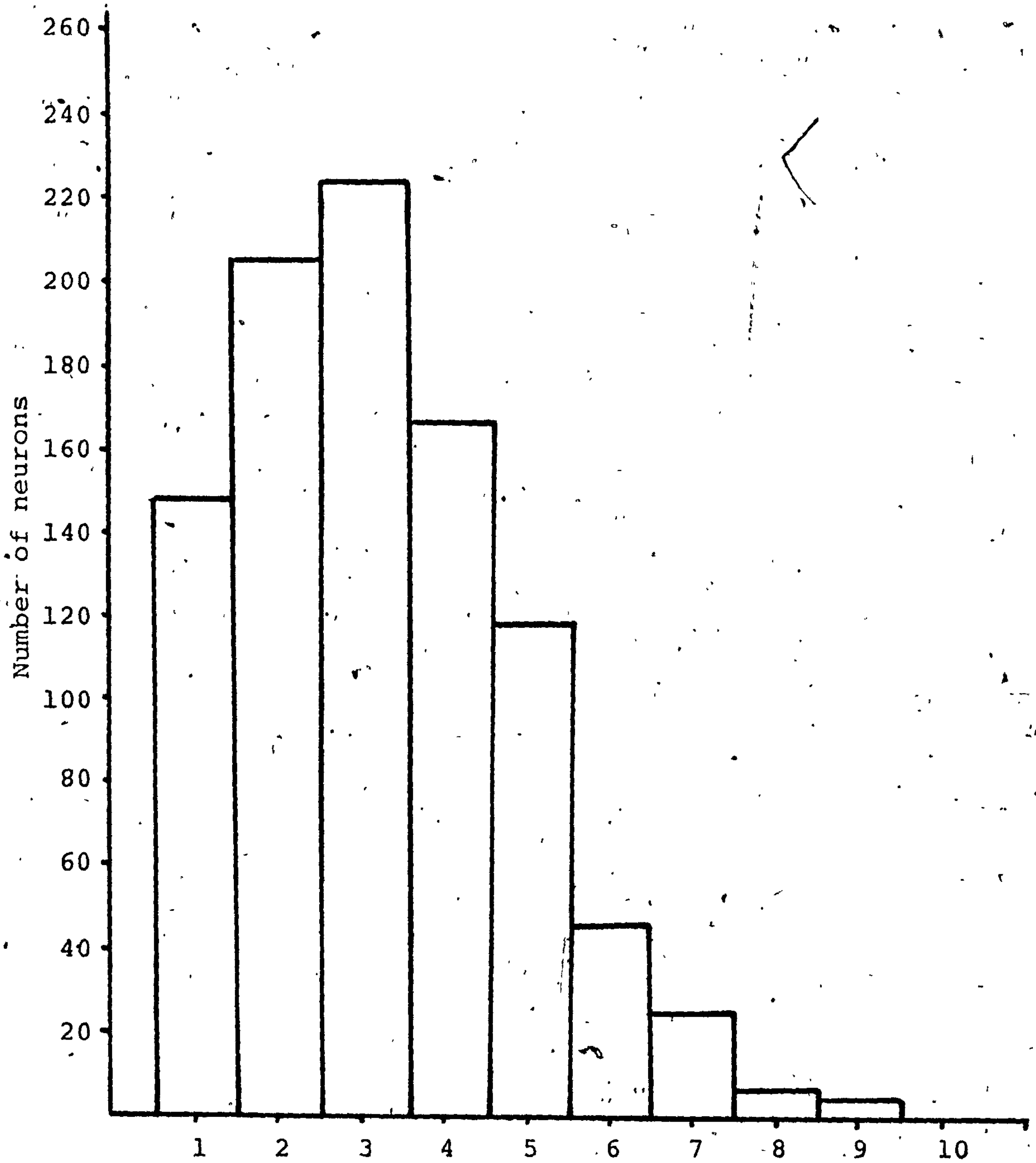

Fig. 11: Total Number of Incoming Connections for Net 3

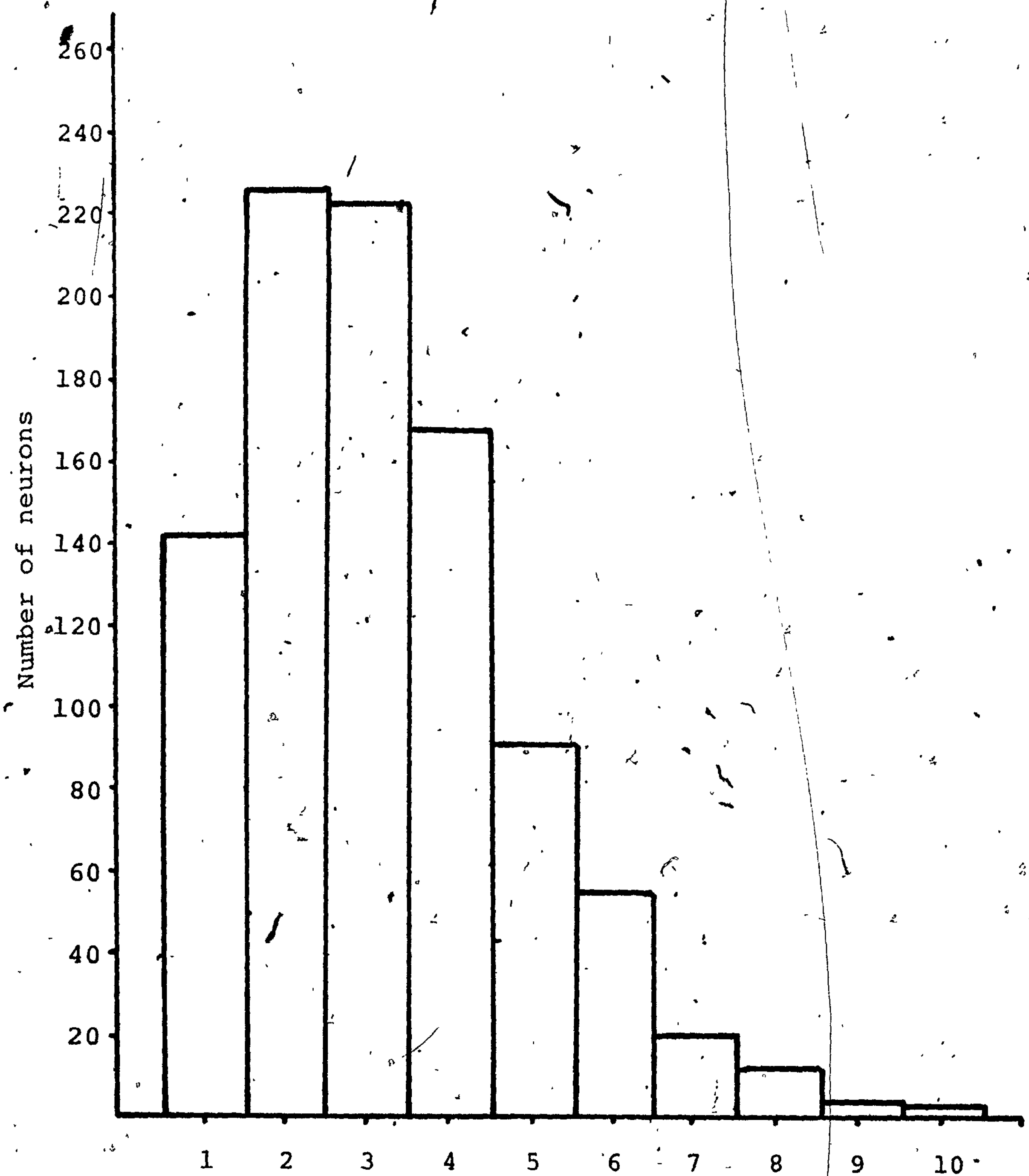

Fig. 12: Total Number of Incoming Connections for Net 4

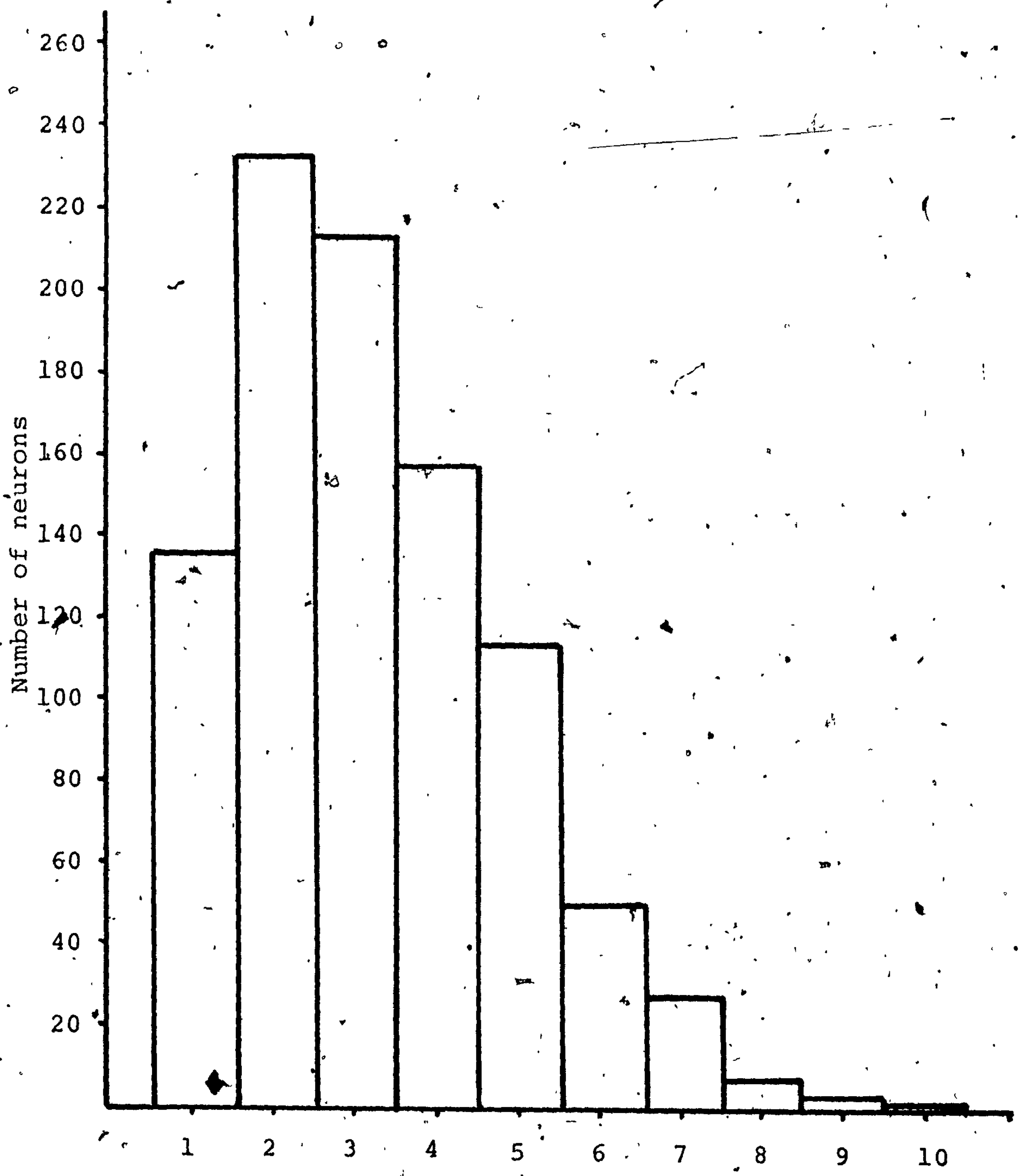

Fig. 13: Total Number of Incoming Connections for Net 5

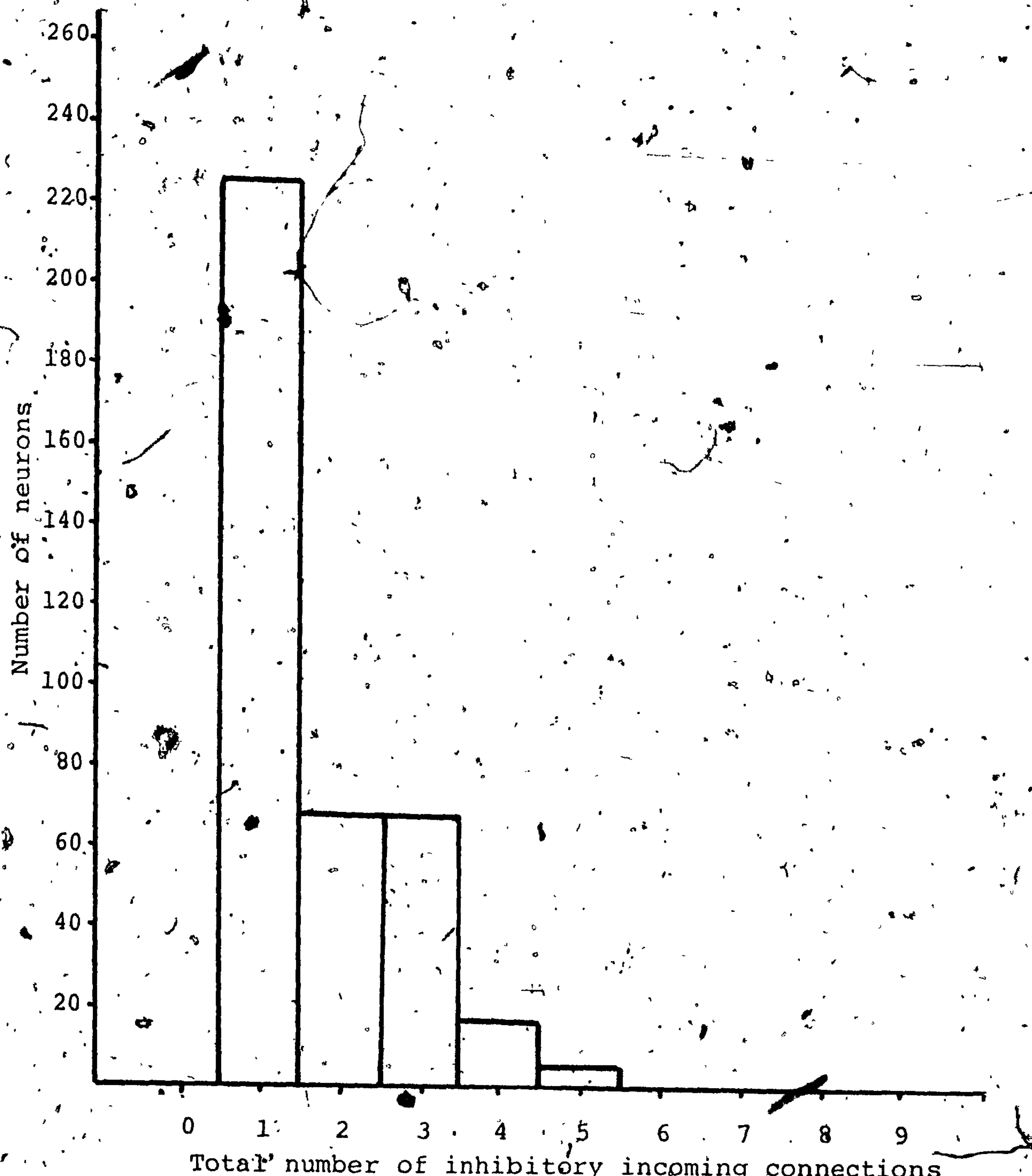

Fig. 14: Total Number of Inhibitory Incoming Connections for Net 1

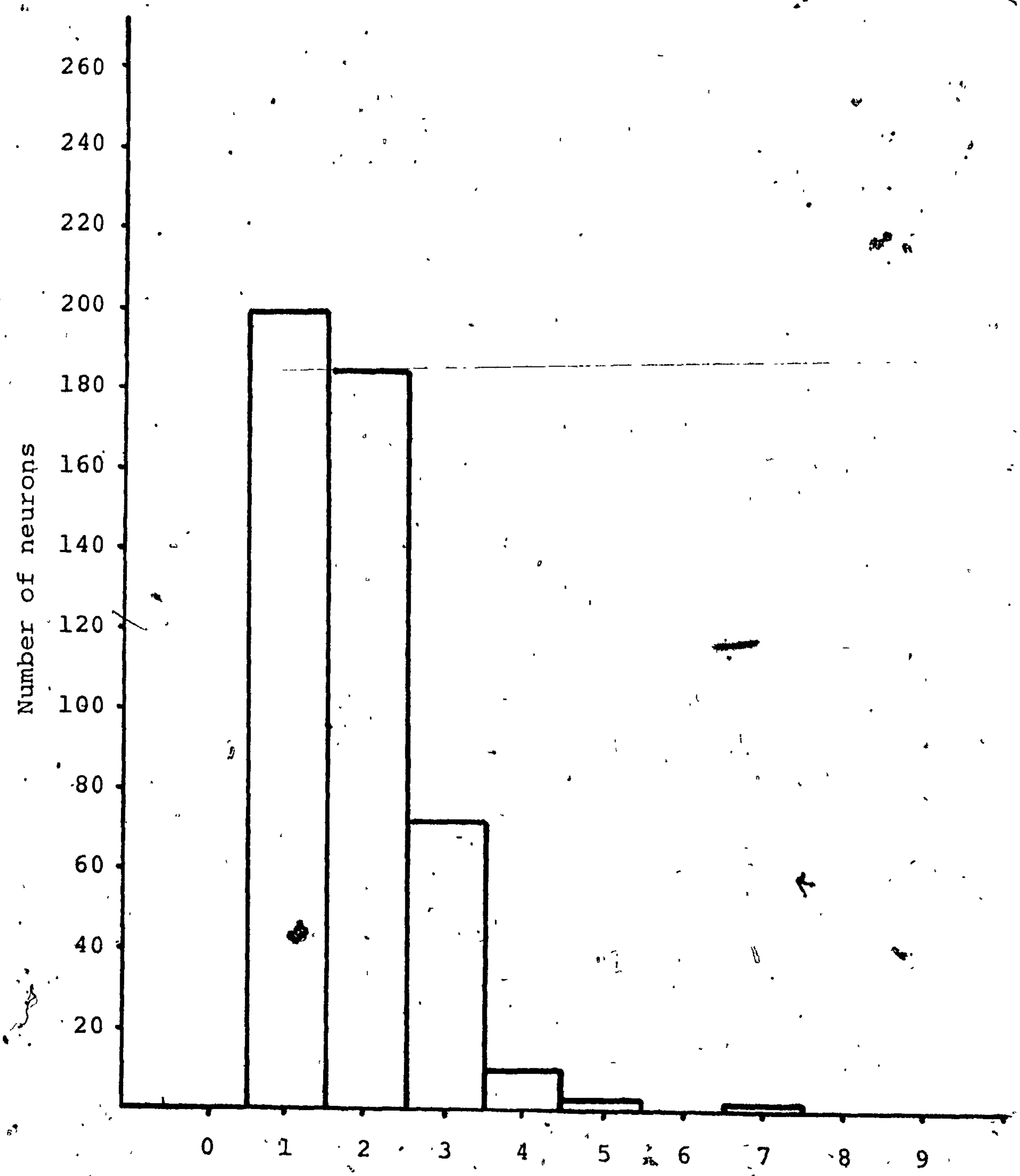

Fig. 15. Total Number of Inhibitory Incoming Connections for Net 2

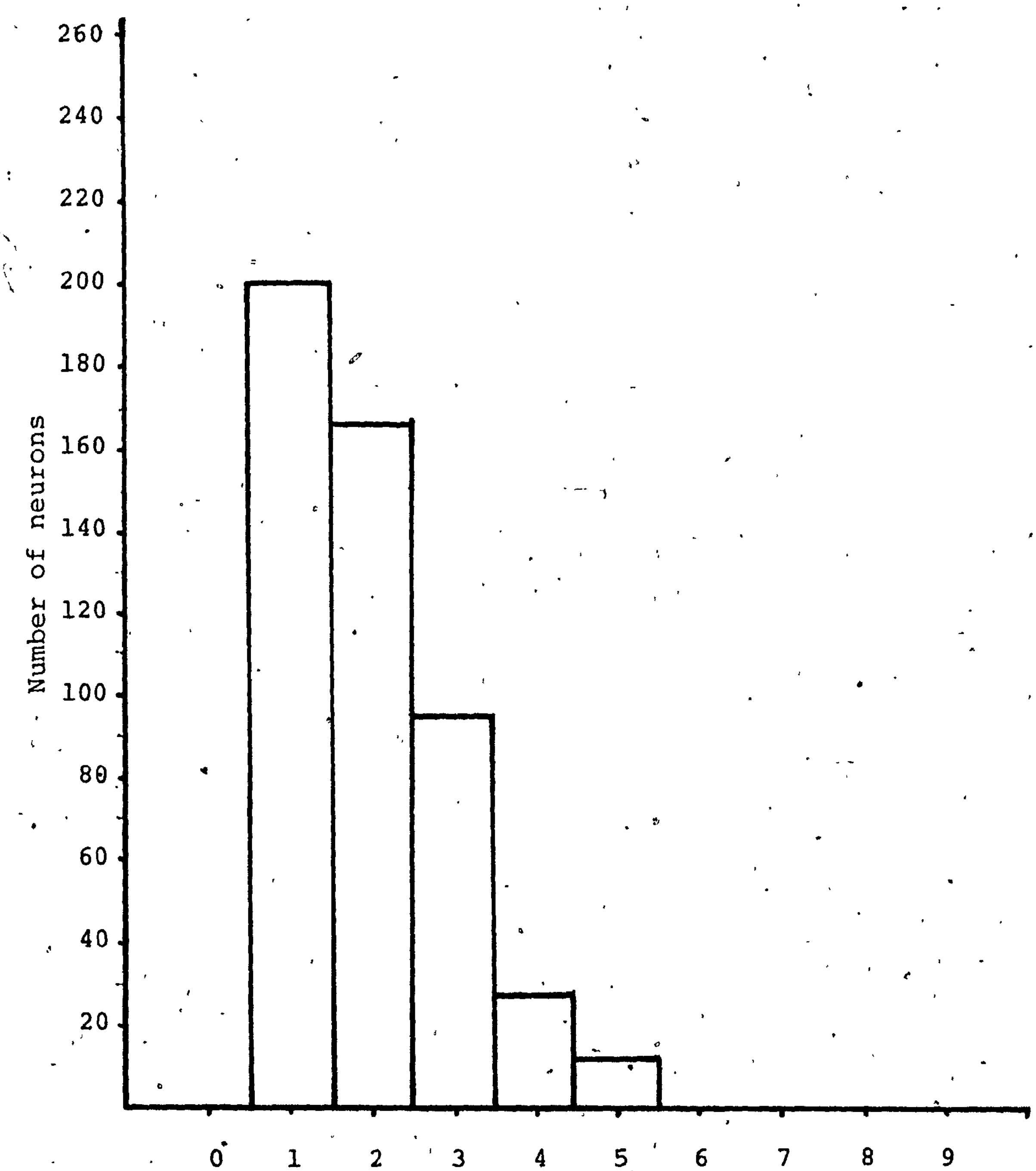

Fig. 16. Total Number of Inhibitory Incoming Connections for Net 3

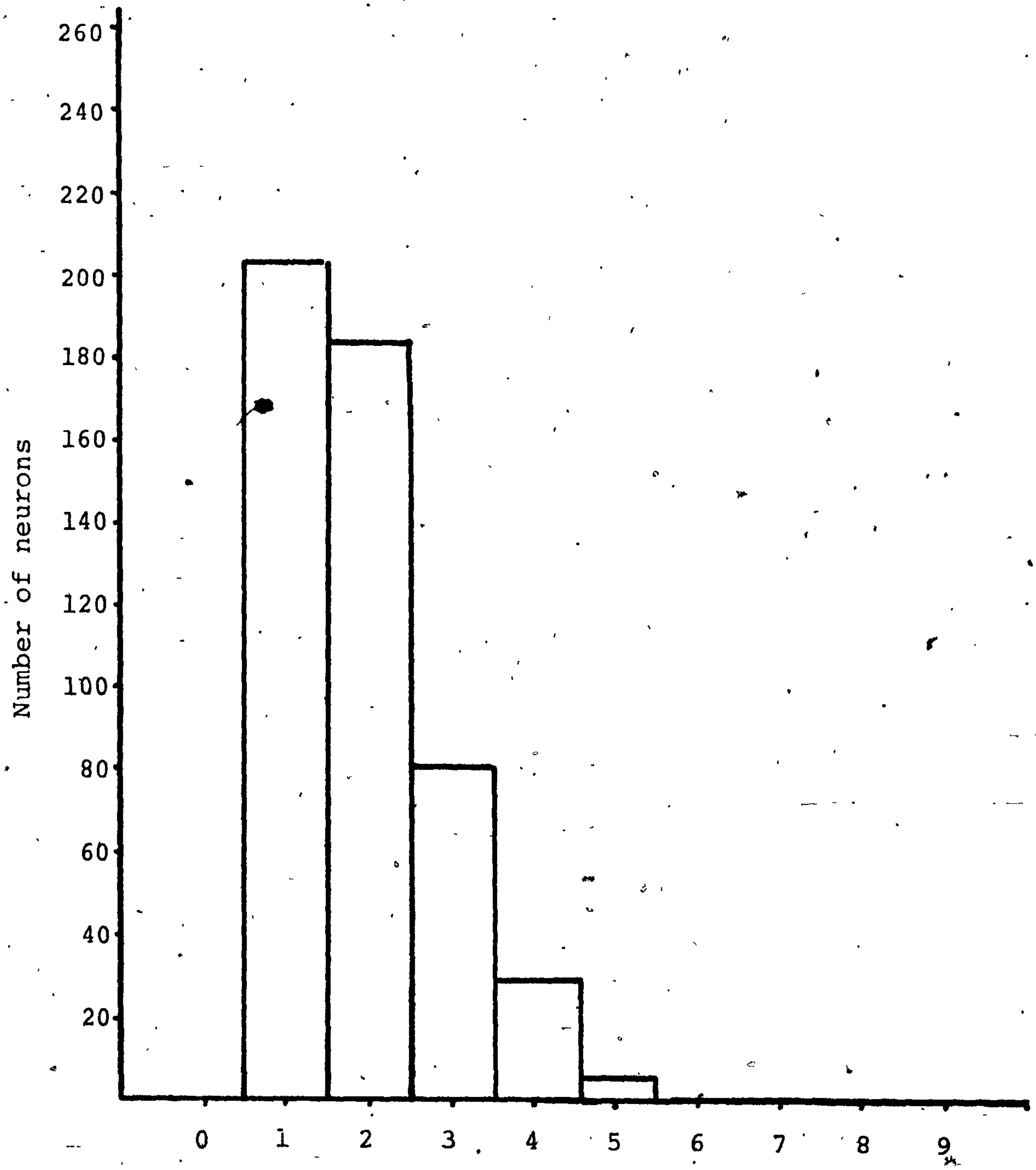

Fig. 17. Total Number of Inhibitory Incoming Connections for Net 4

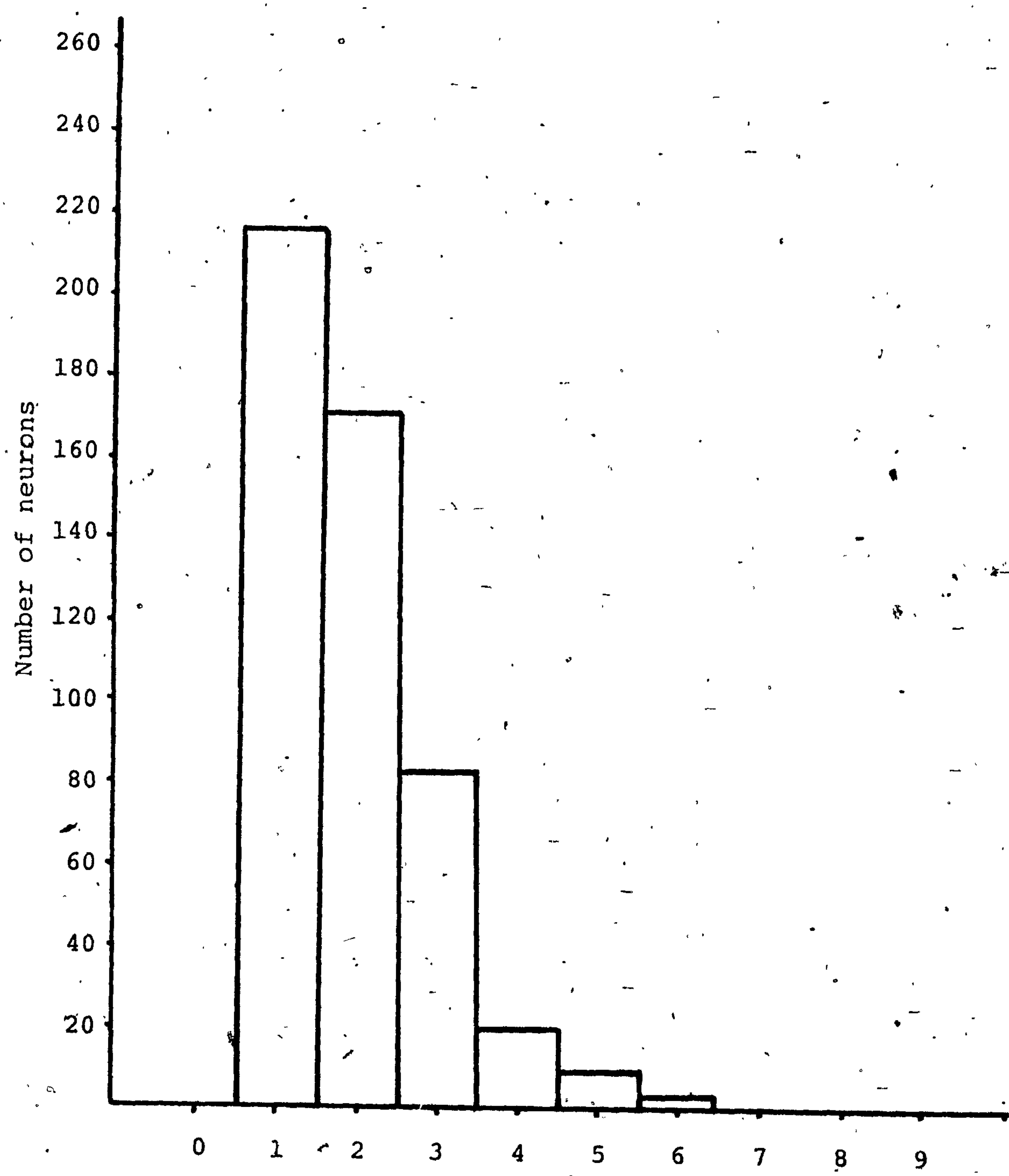

Fig. 18. Total Number of Inhibitory Incoming Connections for Net 5

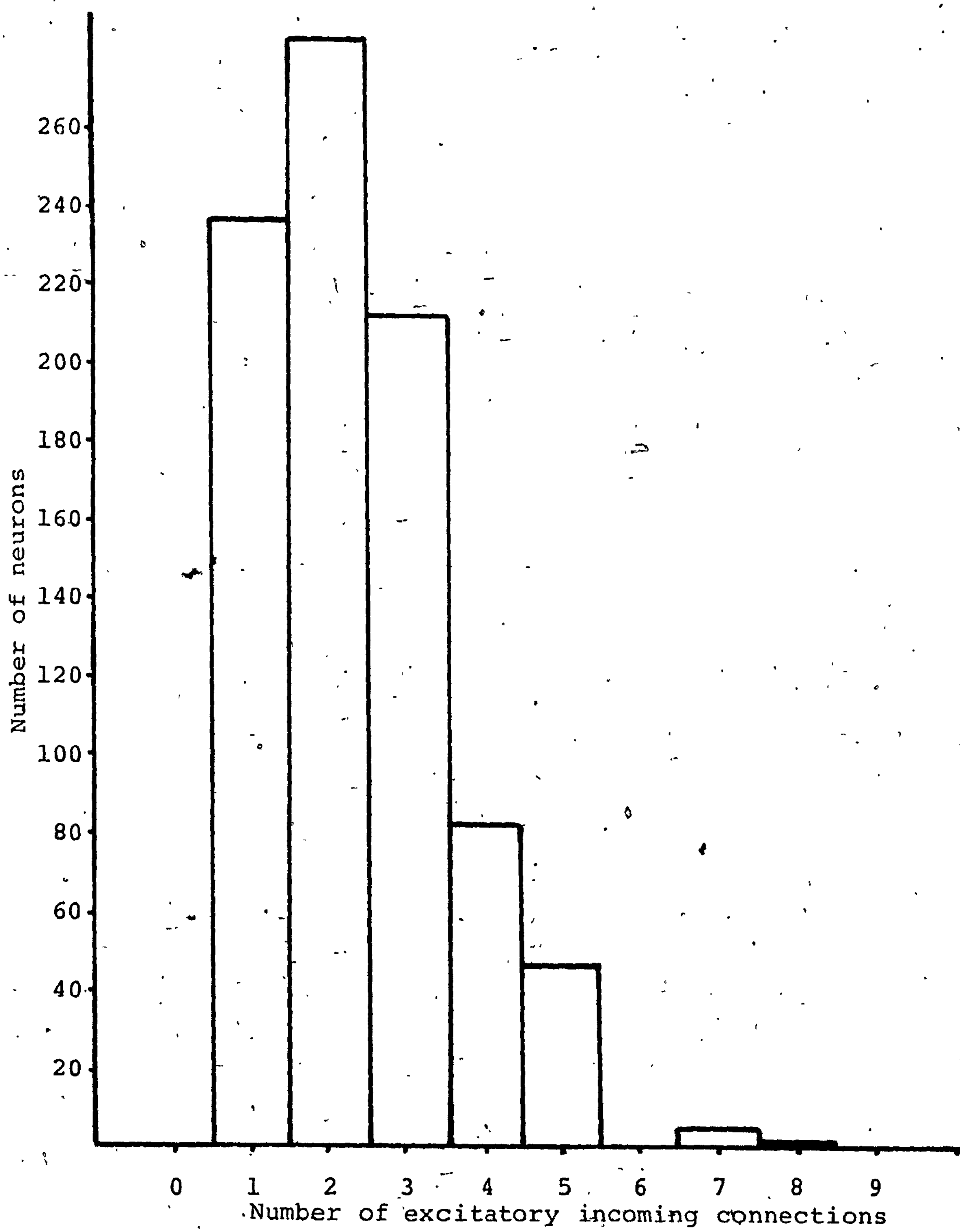

Fig. 19. Total Number of Excitatory Incoming Connections for Net 1

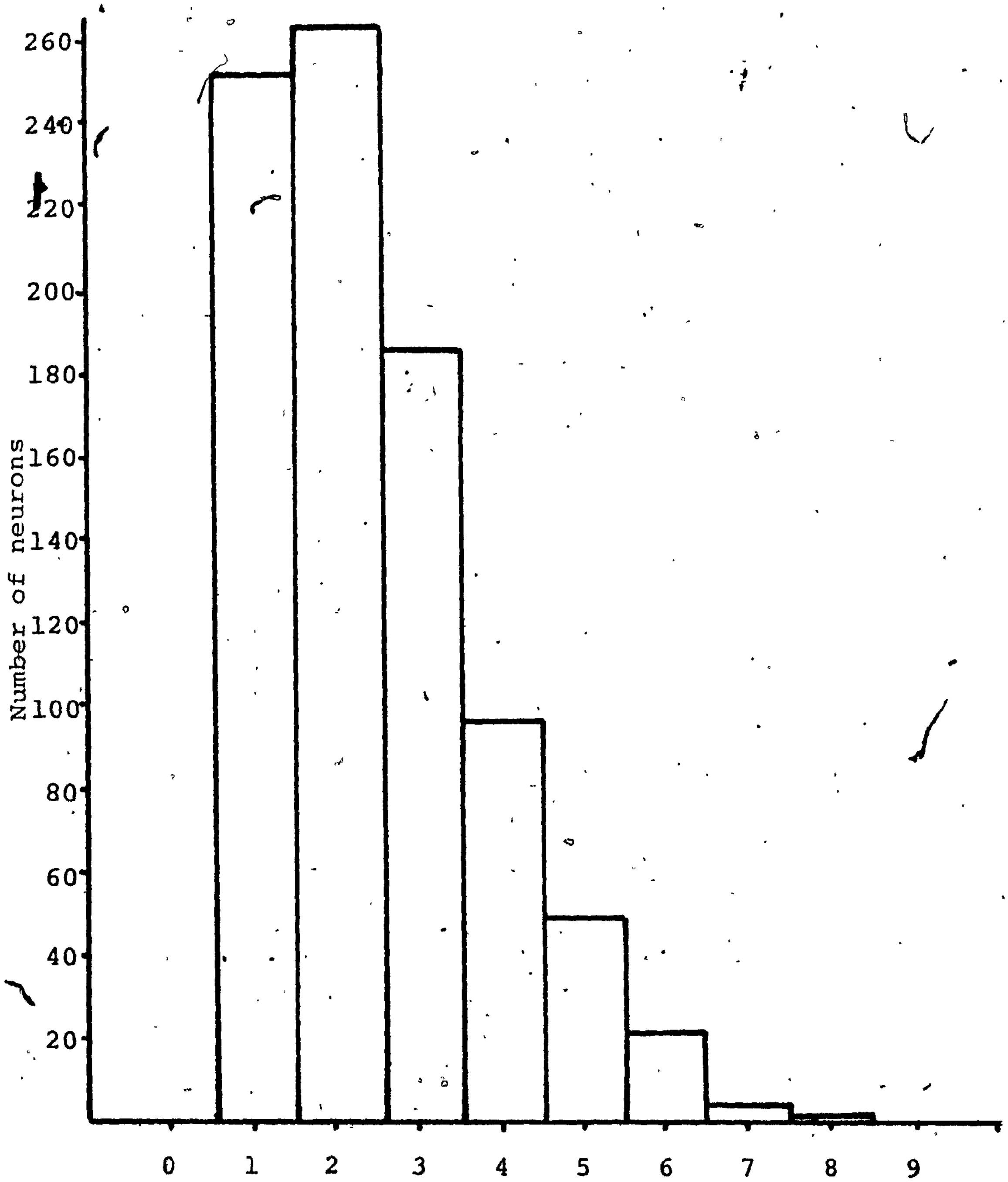

Fig. 20. Total Number of Excitatory Incoming Connections for Net 2

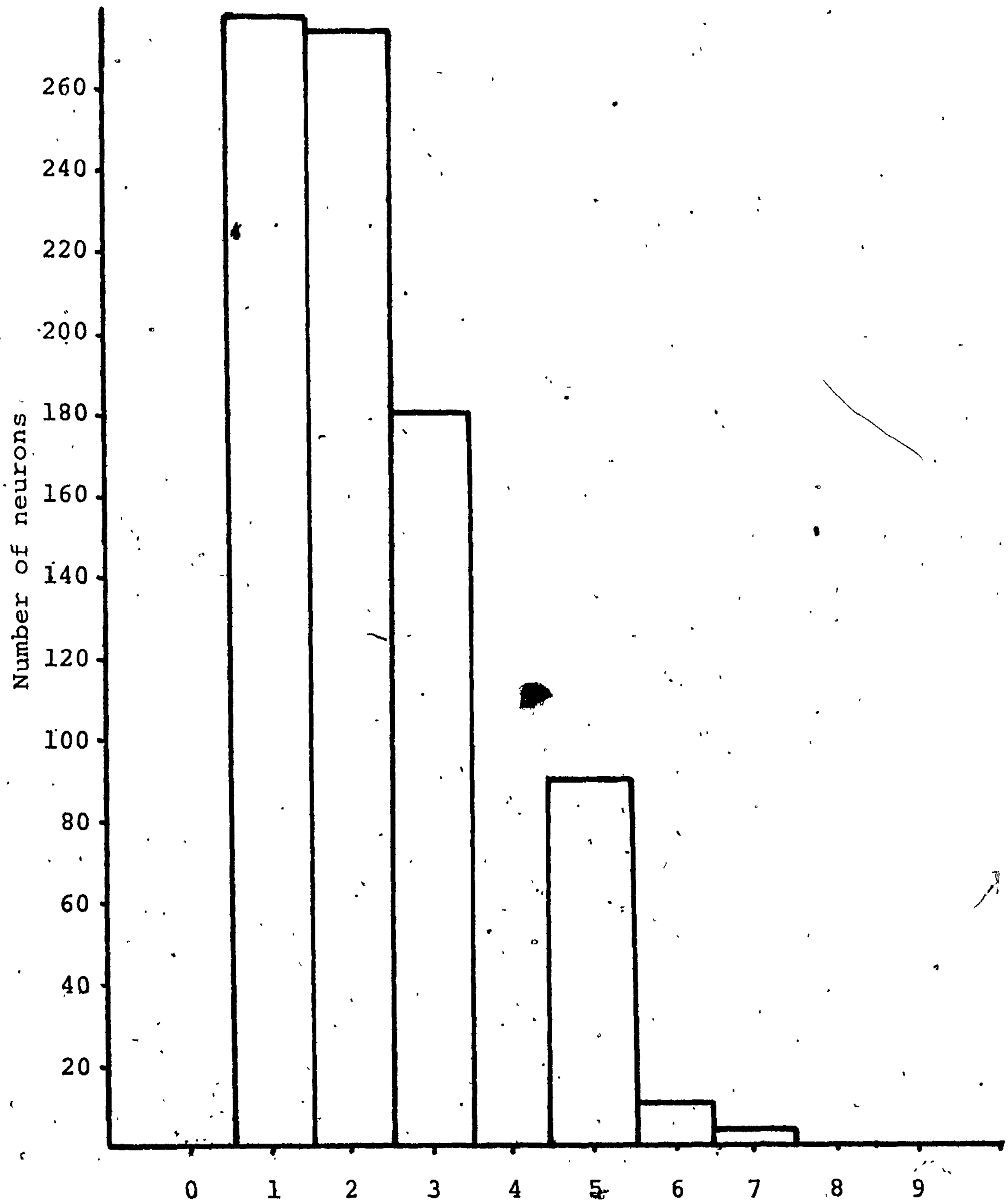

Fig. 21. Total Number of Excitatory Incoming Connections for Net 3

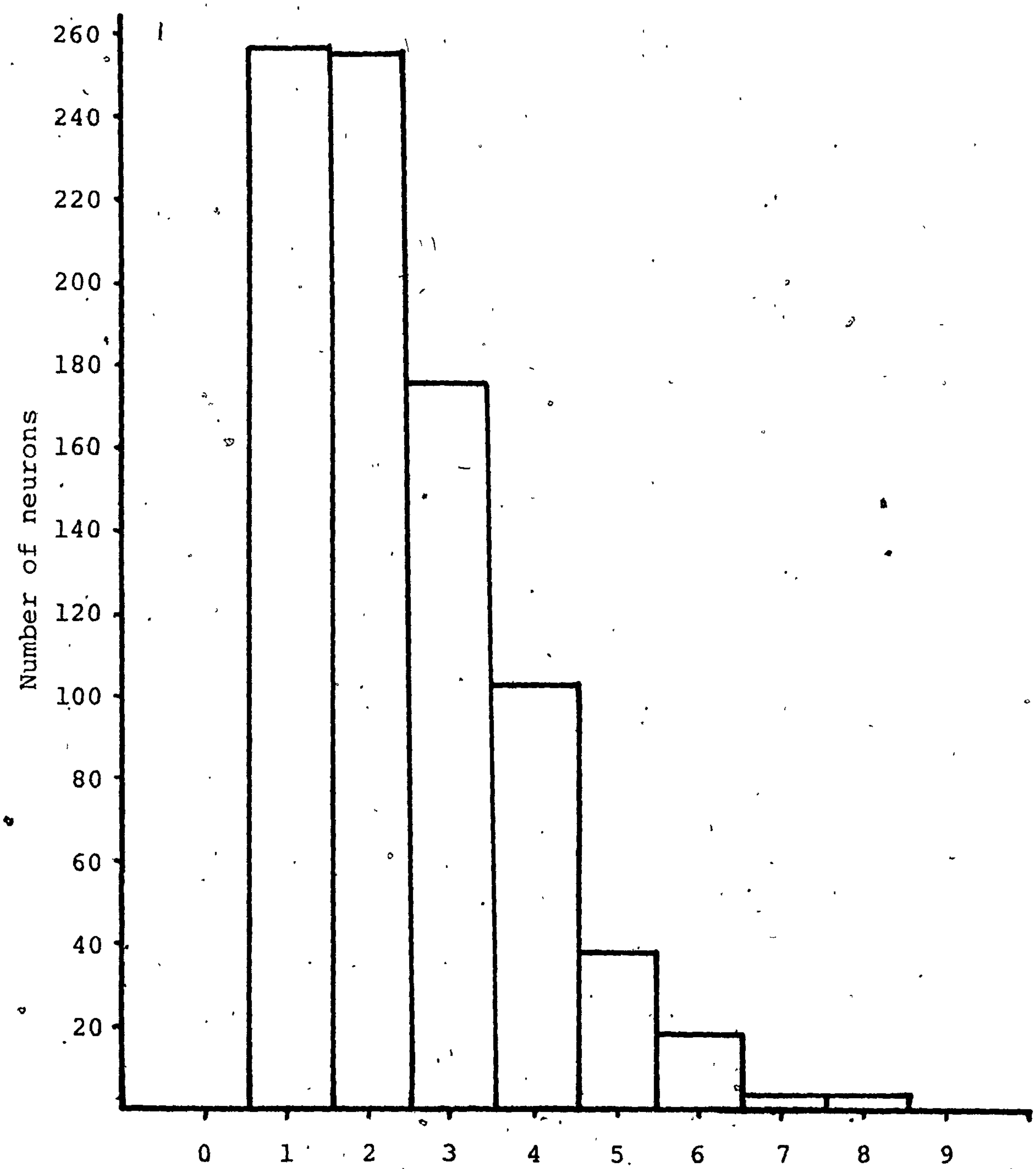

Fig. 22. Total Number of Excitatory Incoming Connections for Net 4

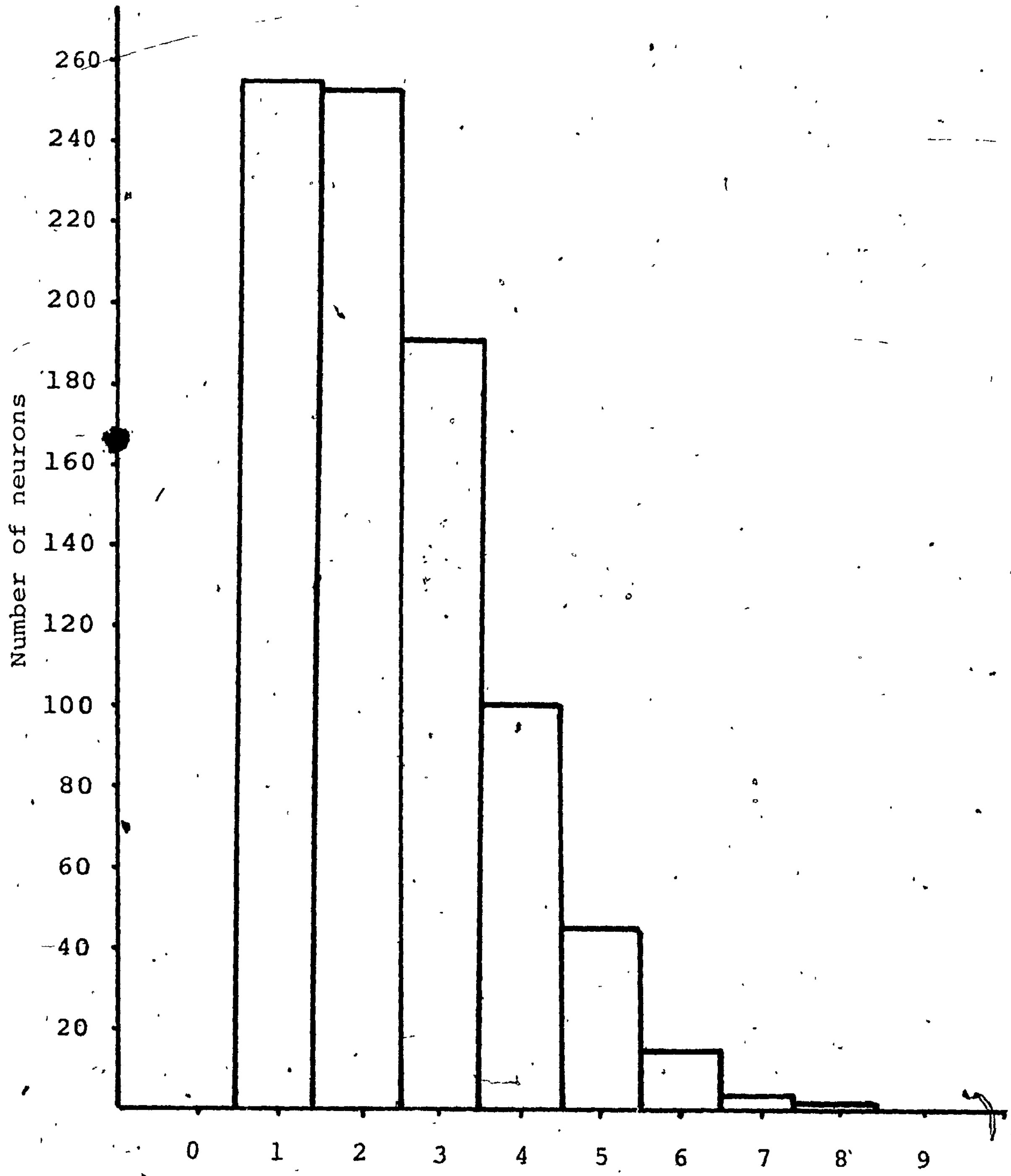

Fig. 23. Total Number of Excitatory Incoming Connections for Net 5

Figs. 24 to 27 show a computer printout of the amplitude histogram of the sum of the PsP's (the gross EEG of the net) for a period of 200 τ for four of the nets considered in table 1 and with 100 neurons firing initially. This is done with the aid of one of the subroutines belonging to the second overlay of the computer program used in this work. The range of the amplitudes of the PsP is divided into ten class intervals and a count is made of the number of PsP's falling within each amplitude class interval. These frequency distributions are normalized by dividing the frequency of each class by the total frequency of all classes (total number of readings) so that the total area limited by the frequency curve is equal to 1. The arithmetic mean ($\bar{X}$) and the standard deviation (S) are then calculated for the normalized histograms. A fit between these normalized histograms and a theoretical normal distribution with the same arithmetic mean and standard deviation is examined by the use of the χ^2 goodness-of-fit method which is given by (Croxton, 1959):

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - T_i)^2}{T_i}$$

where O_i is the number of counts (observed frequencies) in the ith bin of the normalized EEG histogram and the T_i is the number of counts in the ith bin of a

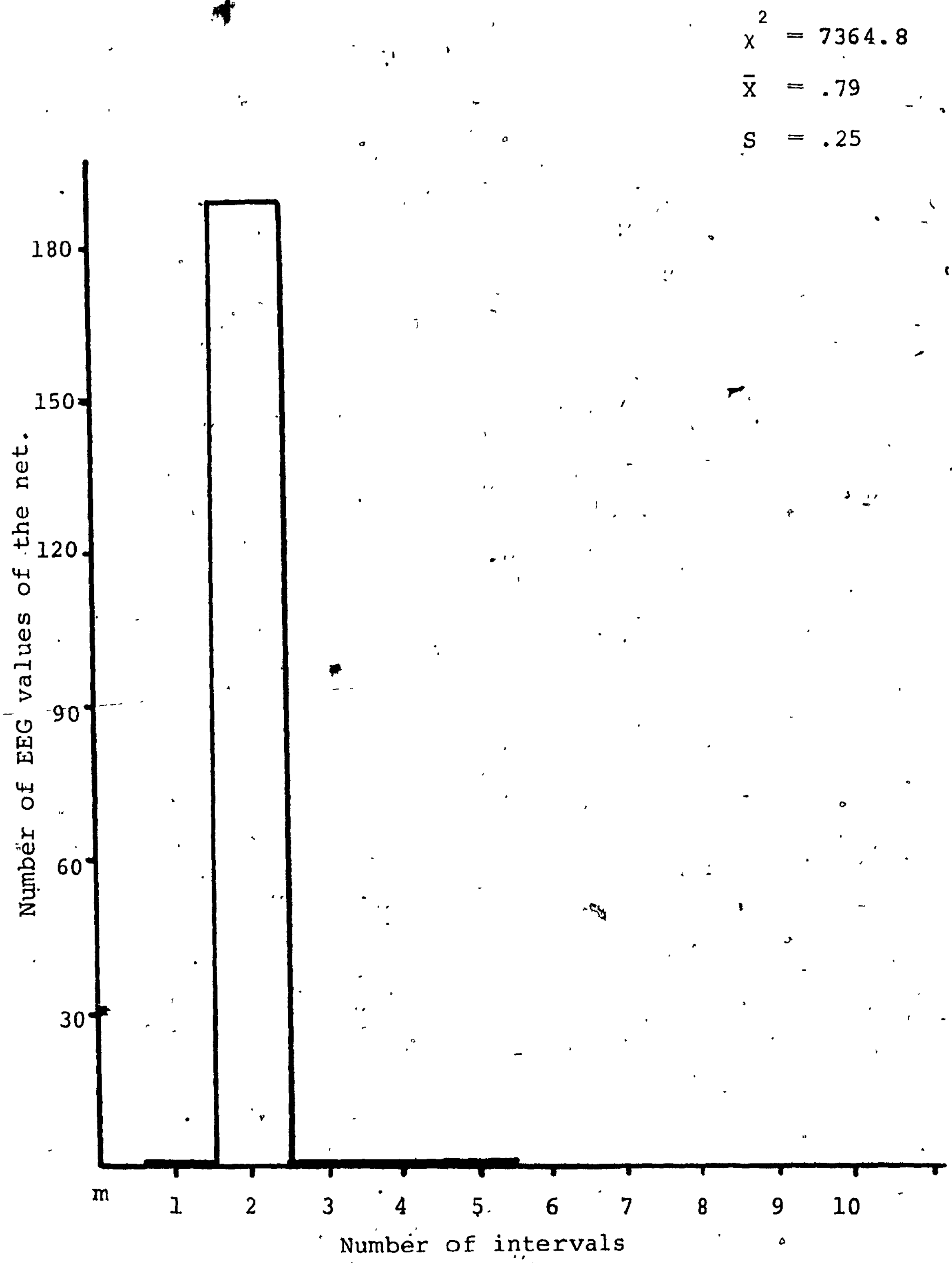

Fig. 24. EEG Histogram for Net 1

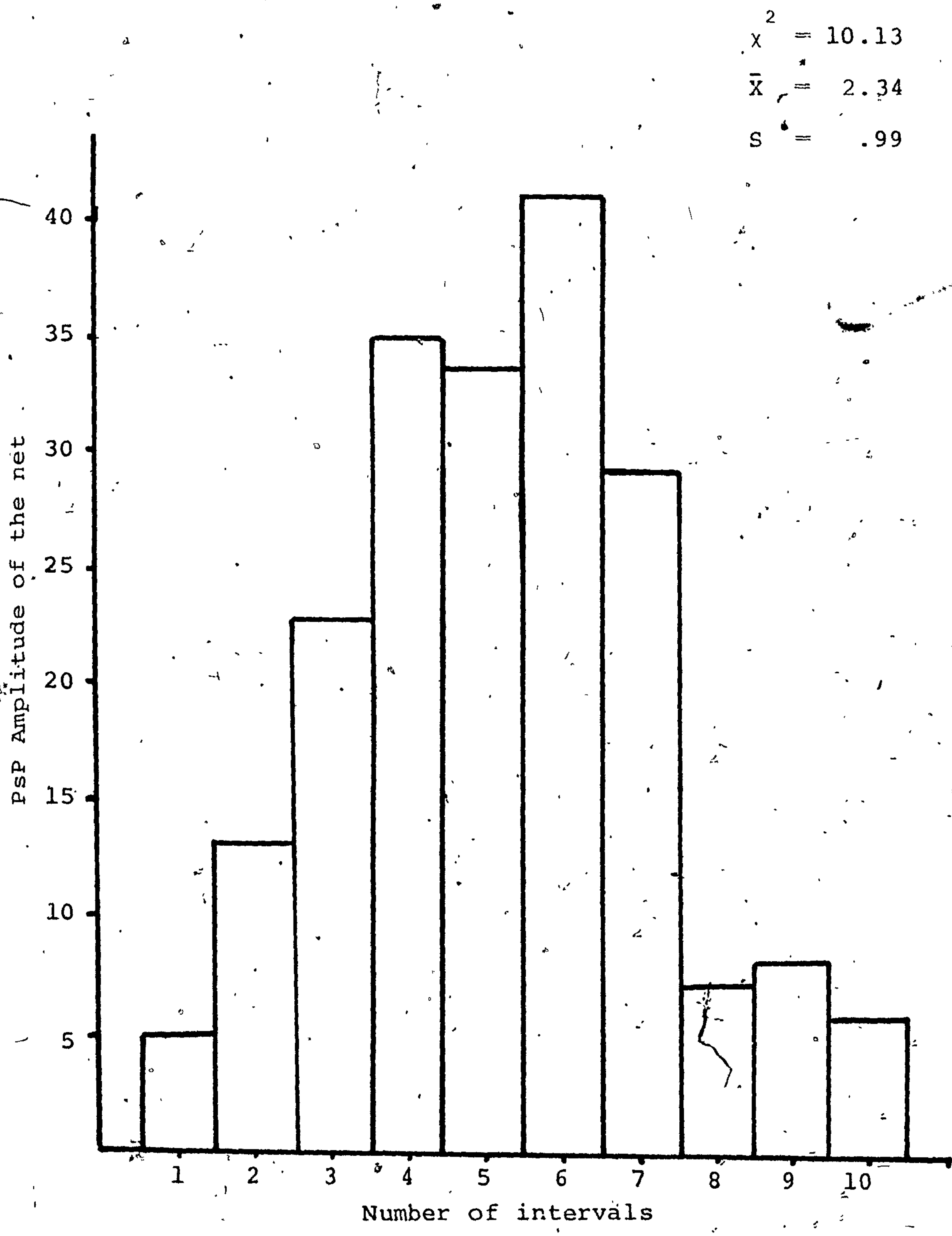

Fig. 25. EEG Histogram for Net 2

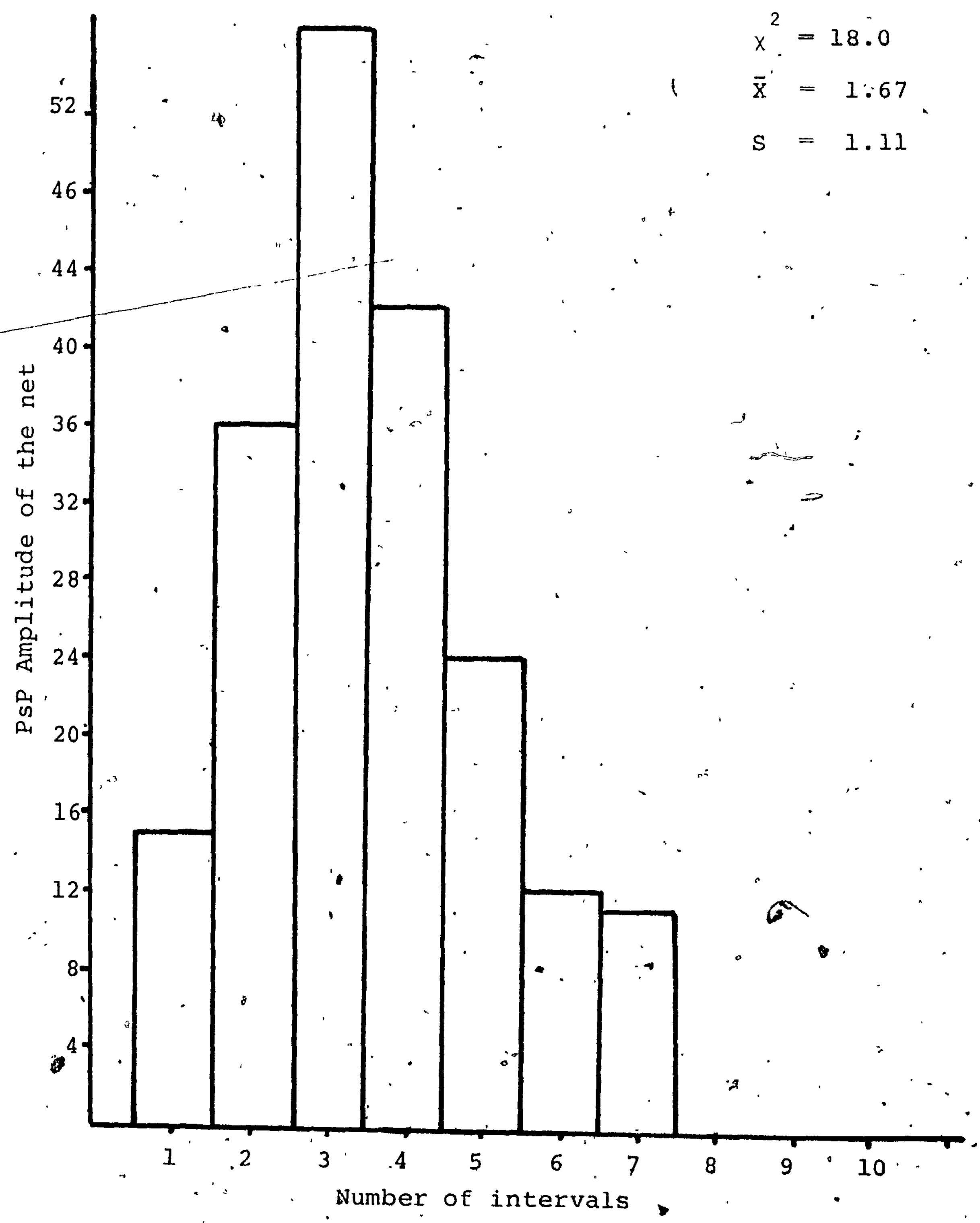

Fig. 26. EEG Histogram for Net 3

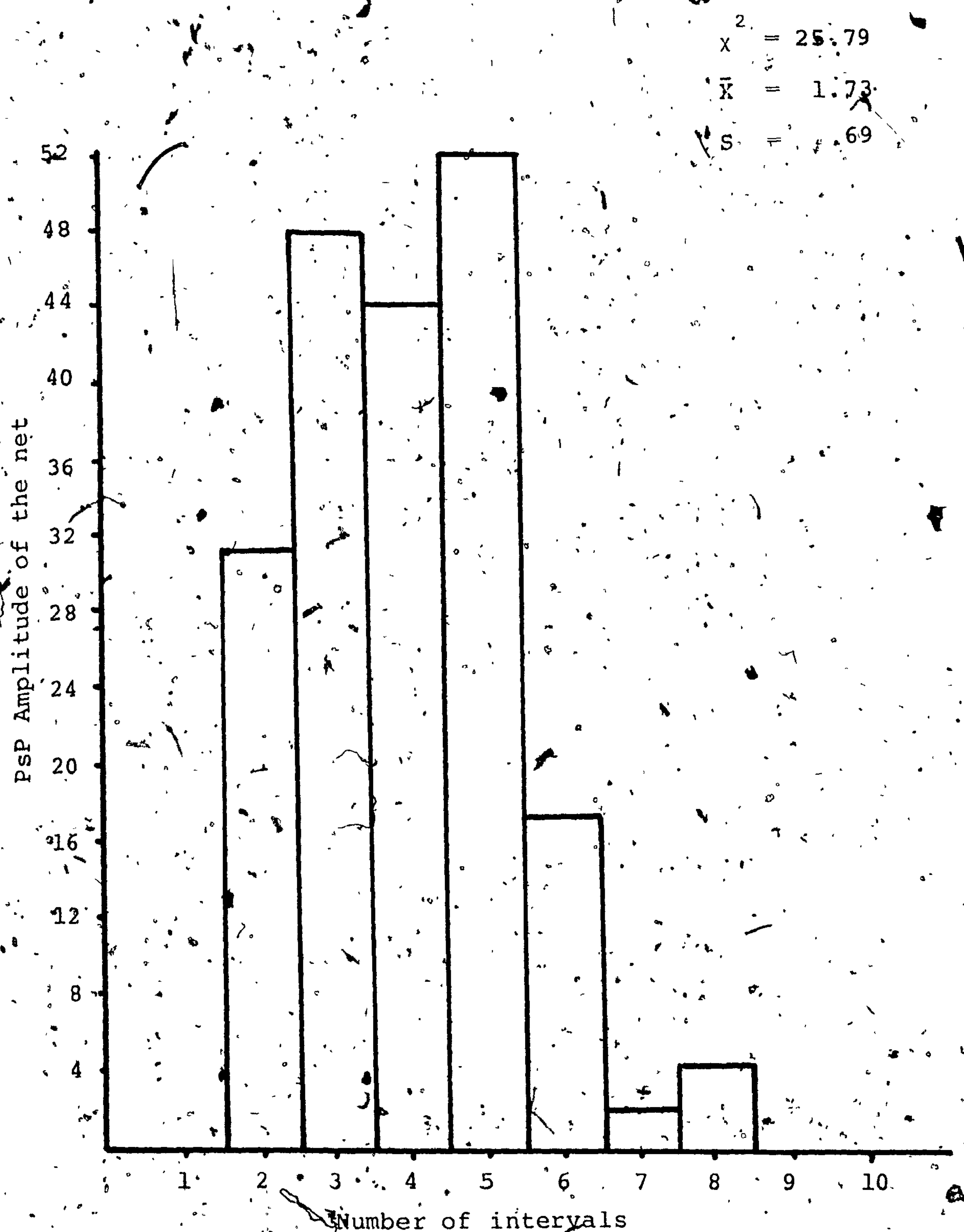

Fig. 27. EEG Histogram for Net 5

normal distribution (theoretical or expected frequencies) having the same mean and standard deviation as the normalized EEG histograms and k is the number of bins. The number of degrees of freedom n is given by $n = k - 1$, if no population parameters have to be estimated from sample statistics and by $n = k - 1 - m$ when the expected frequencies can be calculated only by estimating m population parameters from sample statistics. In our case, $m = 2$, since both the arithmetic mean and the standard deviation are computed from the samples. When $k = 10$, then $n = 7$ for the EEG histograms. The value of n, together with the value of χ^2, determines the probability that the distribution tested is normal. χ^2 values and the standard deviation (S) of each net are also shown in those figures.

CHAPTER V

DISCUSSION

A. Criteria

The results of the computer simulation reported in Chapter IV suggest that nets with different microscopic structures behave in different ways. In this section, we are going to discuss the criteria that we will be using to determine whether different microscopic structures lead to different behaviour patterns in artificial neural nets. The following factors will be considered in determining whether the behaviour of a particular net is normal or abnormal.

1. The Gross Encephalogram of the Nets EEG

The EEG of the net is obtained by summing the membrane potentials of all the neurons in the net, where the membrane potential of a neuron is equal to the instantaneous sum of all its synaptic inputs. Differences in the characteristics of the EEG activity with different nets having the same statistical parameters can be due to differences in the microscopic structures of these nets.

2. The Presence of Cyclic Activity

It has been found (Anninos et al, 1970) that Class A nets are capable of sustaining cyclic activity. Therefore, the presence or absence of the activity could serve as an indicator of different nets' behaviour. If such a difference can be detected in two Class A nets having the same overall statistical parameters (h, μ^-, μ^+, and θ), then it might be due to differences in the microscopic connectivities of these nets.

3. Delay Prior to the Entry into the Cyclic Mode

The number of the synaptic delays before the ensuing of the cyclic activity depends on the characteristics of the net and could be used as an indicator of variations in these characteristics.

B. Gaussian Character of the EEG

It has been shown (Elul, 1969) that the EEG of the mammalian brain can be viewed as the random sum of non-linearly related and almost independent generators. It was also shown (Elul, 1972) that the EEG exhibits the statistical properties of a normal random process when viewed over a relatively long period of time. The probability distribution of the amplitude of scalp EEG has been investigated by Elul (1969) who made an amplitude analysis of the EEG of an adult subject in the idle state and during the performance of mental tasks. By using the χ^2 goodness-of-fit test, he concluded that the EEG of the subject in the idle state follows a Gaussian distribution. He also found that the number of failures in the χ^2 test during the performance of the mental task was more than twice that in the idle state. He concluded that amplitude analysis of the EEG can provide significant information on mental functions.

Goldstein et al (1963) and Goldstein et al (1965) used a "Drohocki" integrator for the study of the amplitude analysis of the EEG of male chronic schizophrenics and a normal group. The area under the EEG curves of the subjects was automatically measured and the standard deviation was obtained. Comparison of the analyses of monopolar EEG's from the left occipital area in the two groups showed that the coefficient of variation ($V = S/\bar{X}$) for the patient group was one-half of that for the normal group, but no significant differences were detected with respect to the overall amplitude.

The Gaussian dependence of EEG activity on interneural coupling has been investigated by Anninos and Elul (1974b) who found that increasing the level of connectivity ($\bar{\mu}^{+}$) in a neural net resulted in the shift of the EEG activity from the Gaussian toward the non-Gaussian range.

We think that the EEG pattern and its amplitude histogram of a neural net can be used to classify the neural nets according to their behavioural characteristics into normal and abnormal nets. Although it is clear that such a classification into normal and abnormal nets is completely relative, it is an important one if these nets are to represent elementary units of the nervous system. This distinction will be used to characterize the different behaviour patterns of the nets and their dependence on interneural connectivities.

Referring to Figs. 24 to 27 of Chapter IV, the χ^2 goodness-of-fit will be used to determine how well the EEG histogram of each of the artificial nets considered here fits the Gaussian distribution.

We are going to test the artificial nets' histograms at the 0.05 significance level. This means that if the computed value of χ^2 is greater than the value $\chi^2_{.95}$ the observed frequencies differ significantly from the theoretical frequencies and we will reject the hypothesis that the histogram being tested is normal. Otherwise, we accept the hypothesis.

Since the $\chi^2_{.95}$ value for $n = 7$ is equal to 14.1 (Lindgren and McElrath, 1967), we can see from Figs. 24 to 27 that the only net with a value of $\chi^2 < 14.1$ is net 2.

Therefore, we conclude, according to our hypothesis, that this net is Gaussian since its EEG corresponds to a normal process. The χ^2 of net 1, which has identical statistical parameters to net 2, is clearly non-Gaussian.

If we accept the assumption that a Gaussian characteristic of the amplitude distribution of the EEG activity is associated with normal operations, then we can consider net 2 to be a normal net while net 1, for example, can be considered to be abnormal. We are going to see if this distinction can be supported by other criteria, like the presence of cyclic activity and its period. Finally, the microscopic structure and the neural connectivities will be examined for possible differences between the two groups (normal and abnormal).

C. Theoretical Model Of The EEG Activity

In this section we are going to present a mathematical model of the EEG activity of a neural network based on the neu ral netlet assumptions of Chapter III. According to those assumptions a given cell in a net with A neurons will be active and will fire at time $(n+1)\tau$ if the following two conditions are satisfied simultaneously:

1) The cell did not fire at time $n\tau$, i.e. it is non-refractory.
2) The sum of the incoming PsP's is equal to or greater than the threshold.

The above two conditions indicate that the number of active neurons at time $(n+1)\tau$ depends on the number of active neurons at time $n\tau$. In order to find the theoretical value of EEG activity of the net it is necessary to know the relationship between the activity of the net at time $n\tau$ and its activity at time $(n+1)\tau$. If the fraction of active neurons in the net at time $n\tau$ is equal to α_n then the total number of active neurons is $A\alpha_n$ and there will be $A(1-\alpha_n)$ neurons that will not be refractory at time $(n+1)\tau$. The probability that a given neuron fires at time $(n+1)\tau$ is equal to the product of the probabilities that it is non-refractory and that the sum of the incoming

PsP's is equal to or greater than the threshold. Therefore the probability that a given neuron in the net will fire at time $(n+1)\tau$ is given by

$$P(X_{n+1} = 1 \mid \alpha_n) = (1-\alpha_n)\ P(W_{n+1} \geq \theta \mid \alpha_n)$$

where the variable X_{n+1} represents the activity of the neuron at time $(n+1)\tau$ and can have the following values.

$X_{n+1} = 1$ if the neuron is firing at $(n + 1)\tau$

$X_{n+1} = 0$ if the neuron is resting.

W_{n+1} is the overall input to the cell at time $(n+1)\tau$, and θ is the firing threshold.

The expected value of the activity at $t = (n+1)\tau$, given the activity at $t = n\tau$, is

$$\langle \alpha_{n+1} \mid \alpha \rangle = \langle \frac{1}{A} \sum^{A} X_{n+1} \mid \alpha_n \rangle$$

$$= P(X_{n+1} = 1 \mid \alpha_n)$$

$$= (1 - \alpha_n)\ P(W_{n+1} \geq \theta \mid \alpha_n) \quad . \; . \; . \quad (1)$$

If we let e_{n+1} denote the PsP generated by a given cell at time $(n+1)\tau$, where $e_{n+1} = 0$ if the cell fires at time $n\tau$ or if the sum of the incoming PsP's is less than the threshold θ, and $e_{n+1} = K$ if the sum of the PsP's is equal to or greater than the threshold and the cell is not refractory. K can be either positive or negative depending on whether the neuron is excitatory or inhibitory respectively. The probability that the PsP of the cell at time $(n+1)\tau$ is equal to zero is given by

$$P(e_{n+1} = 0) = P(e_{n+1} = 0 \mid X_n = 1) \, . \, P(X_n = 1)$$

$$+ \; P(e_{n+1} = 0 \mid W_n \leq \theta) \, . \, P(W_n < \theta \text{ and } X_n = 0)$$

Since according to our netlet assumption

$$P(e_{n+1} = 0 \mid W_n < \theta) = 1$$

and

$$P(e_{n+1} = 0 \mid X_n = 1) = 1$$

The above equation reduces to

$$P(e_{n+1} = 0) = P(X_n = 1) + P(W_n < \theta \text{ and } X_n = 0)$$

$$= P(\alpha_n) + P_n \tag{2}$$

The first term in this equation is equal to the probability that the neuron is active and is referred to as $P(\alpha_n)$. The second term, which is denoted by P_n, can be written as:

$$P_n = P(X_n = 0 \mid W_n < \theta) . P(W_n < \theta)$$

$$= [P(X_{n-1} = 1) + P(W_{n-1} < \theta \text{ and } X_{n-1} = 0)] .$$

$$P(W_n < \theta)$$

$$P_n = (P(\alpha_{n-1}) + P_{n-1}) P(W_n < \theta) \tag{3}$$

and the probability that the PsP of the cell is equal to K is given by

$$v_{n+1} = 1 - P(e_{n+1} = 0)$$

Since the average number of the outgoing connections from an excitatory cell is μ^+ and that from an inhibitory cell is μ^-, then the total number of excitatory and inhibitory axon collaterals at time $(n+1)\tau$ is $A\alpha_n(1-h)\mu^+$ and $A\alpha_n h\mu^-$ respectively and the amplitude of excitatory and inhibitory PsP's impinging on any given neuron at time $(n+1)\tau$ is given by $\alpha_n(1-h)\mu^+k^+$ and $\alpha_n h\mu^-k^-$. The amplitude of the input to any cell in the net at time $(n+1)\tau$ given the activity at time $n\tau$ is

$$\alpha_n(\mu^+(1-h)k^+ - \mu^- hk^-)$$

Therefore the gross EEG generated by the net at time $(n+1)\tau$ is given by

$$V_{n+1} = A\alpha_n[\mu^+(1-h)k^+ - \mu^- hk^-]\, v_{n+1} \qquad (4)$$

It can be seen from this equation that the EEG amplitude depends on the activity at time $n\tau$ and also on the probability that the sum of the PsP is equal to or greater than the threshold. The probability that the sum of the incoming PsP's is equal to or exceeds the threshold depends on the type of connections that exists between the neurons in the net. For a net connected according to the poisson probability distribution the mean value of the amplitude of the excitatory and inhibitory PsP's arriving at any given cell at time $(n+1)\tau$ is given by

$$\delta_1 = \alpha_n(1-h)\mu^+k^+ \text{ and } \delta_2 = \alpha_n h\mu^-k^-$$

respectively. Therefore the frequency functions for a net whose excitatory and inhibitory PsP's are distributed according to the poisson probability distribution law are given by

$$f(\ell) = \frac{\delta_1^{\ell}\, \bar{e}^{\delta_1}}{\ell!} \qquad \ell = 0,1,2, \ldots$$

and

$$f(m) = \frac{\delta_2^{m}\, \bar{e}^{\delta_2}}{m!} \qquad m = 0,1,2, \ldots$$

where ℓ and m represent the PsP amplitudes from excitatory and inhibitory cells respectively. The probability that the sum of the PsP amplitudes is equal to or exceeds the threshold, given the activity at time $n\tau$, and in the presence of inhibitory inputs, is given by

$$P(W \geq \theta | \alpha_n) = P(\alpha_n) = \sum_{\ell=\theta+m}^{A\delta_1} \sum_{m=0}^{A\delta_2} f(\ell) \; f(m) \qquad (5)$$

$$= \sum_{\ell=\theta+m}^{A\delta_1} \sum_{m=0}^{A\delta_2} \frac{\bar{e}^{(\delta_1+\delta_2)} \delta_1^{\ell-m} (\delta_1\delta_2)^m}{\ell! \, m!}$$

and the activity at time $(n+1)\tau$ for a net connected according to the poisson probability distribution, given the activity at time $n\tau$, is equal to (equation 1)

$$< \alpha_{n+1} | \alpha_n > = (1-\alpha_n) \sum_{\ell=\theta+m}^{A\delta_1} \sum_{m=0}^{A\delta_2} \frac{\bar{e}^{(\delta_1+\delta_2)} \delta_1^{\ell-m} (\delta_1\delta_2)^m}{\ell! \, m!} \qquad (6)$$

A computer program was specifically written by us to calculate the time course of the EEG activity of a net connected according to the poisson probability distribution law. This was done by calculating the probability of triggering a neuron given that $A\alpha_n$ neurons are initially active from equation 5, and then computing the value of the new level of activity α_{n+1} at time $(n+1)\tau$ from equation 6. This value is then used to calculate a new value for the probability from equation 5 which is then used to get another value for the activity from eq. 6. The process is repeated for 500 iterations. In computing the value of the probability from equation 5 the upper limit in the inner sum was set at m=50 since it was found that the additional terms contributed very little to the sum.

After calculating the value of P_n and keeping only the first four terms in eq. 3 since it was found that the series converges rapidly. The probability that the PsP of the cell is equal to K and then the gross EEG generated by the net at time $(n+1)\tau$ can be calculated from equation 4. The value of V_{n+1} was computed for 200 iterations for a net with

$$\mu^+ = 10$$
$$\alpha_o = 0.1$$
$$h = 0$$
$$k^+ = 10$$
$$A = 1000$$

The EEG amplitude histogram for this net is shown in Fig. 28. The frequency distribution was obtained by dividing the range of V_{n+1} values into ten equal class intervals and then by determining the number that falls into each class interval.

The theoretical EEG histogram of Fig. 28 shows similar characteristics to the EEG histogram obtained through computer simulations as is evidenced by comparing the shape of the theoretical curve with the EEG histogram of net 3 shown in Fig. 2[illegible]. It was not possible to get a direct comparison between the theoretical values of the EEG activity obtained from equation 4 and the computer simulation results since it was found that the variation of α_n with time showed highly damped oscillation and reached a steady value after only few iterations for nets with low connectivities and also in the presence of inhibitory connections.

Fig. 29 shows the theoretical relationship between the fraction of active neurons at time $(n+1)\tau$, which was computed from equation 6, and the fraction of active neurons at time $n\tau$ for the net mentioned above. The theoretical curve of Fig. 29 shows good agreement with with shape of the curve obtained from computer simulations shown in Fig.'s 2 and 3.

The broken line curve which is superimposed on the theoretical EEG histogram of Fig. 28 shows the expected

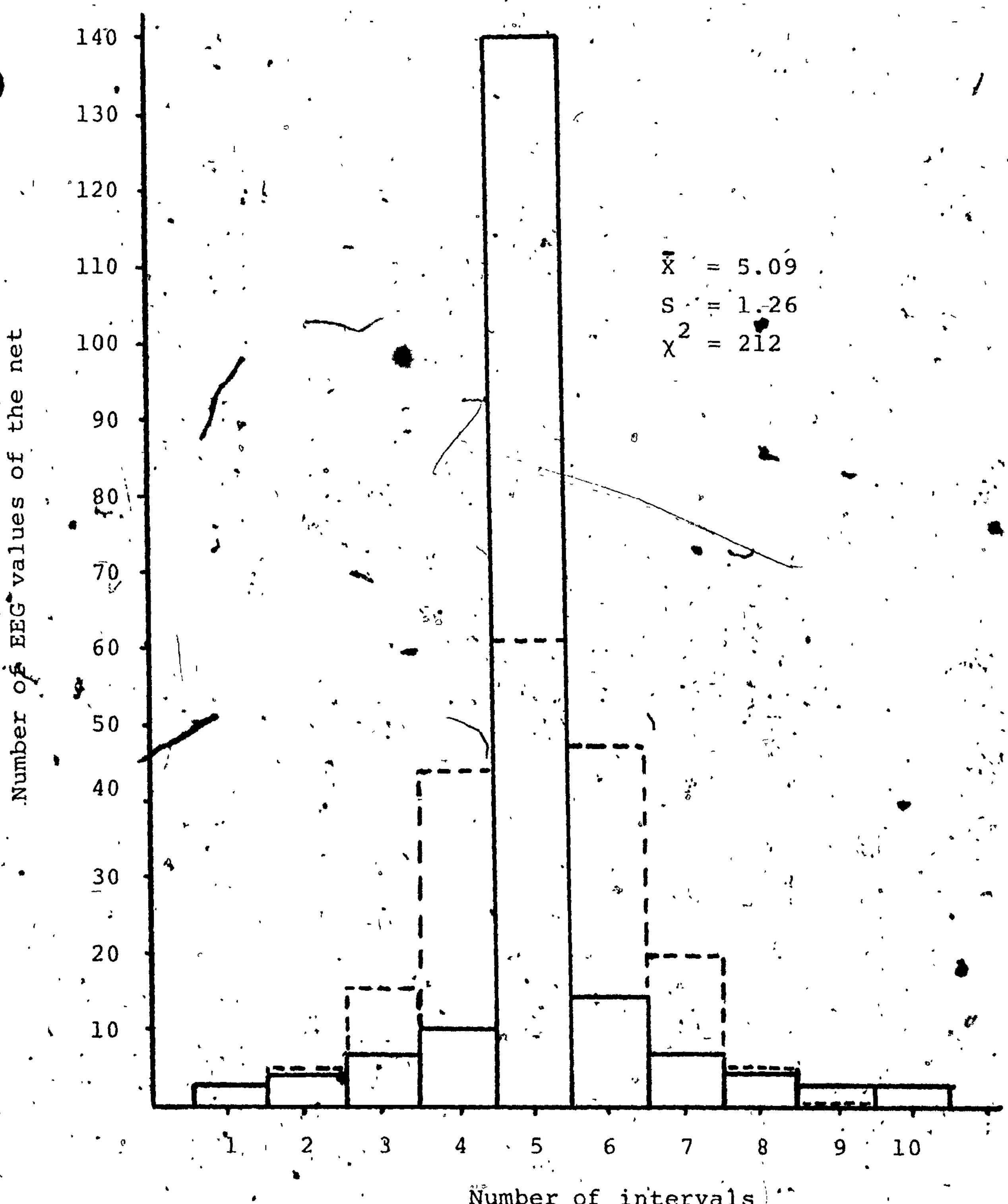

Fig. 28: Theoretical EEG histogram of a net connected according to the poisson distribution law. The broken line represents the expected EEG frequency if they are normally distributed.

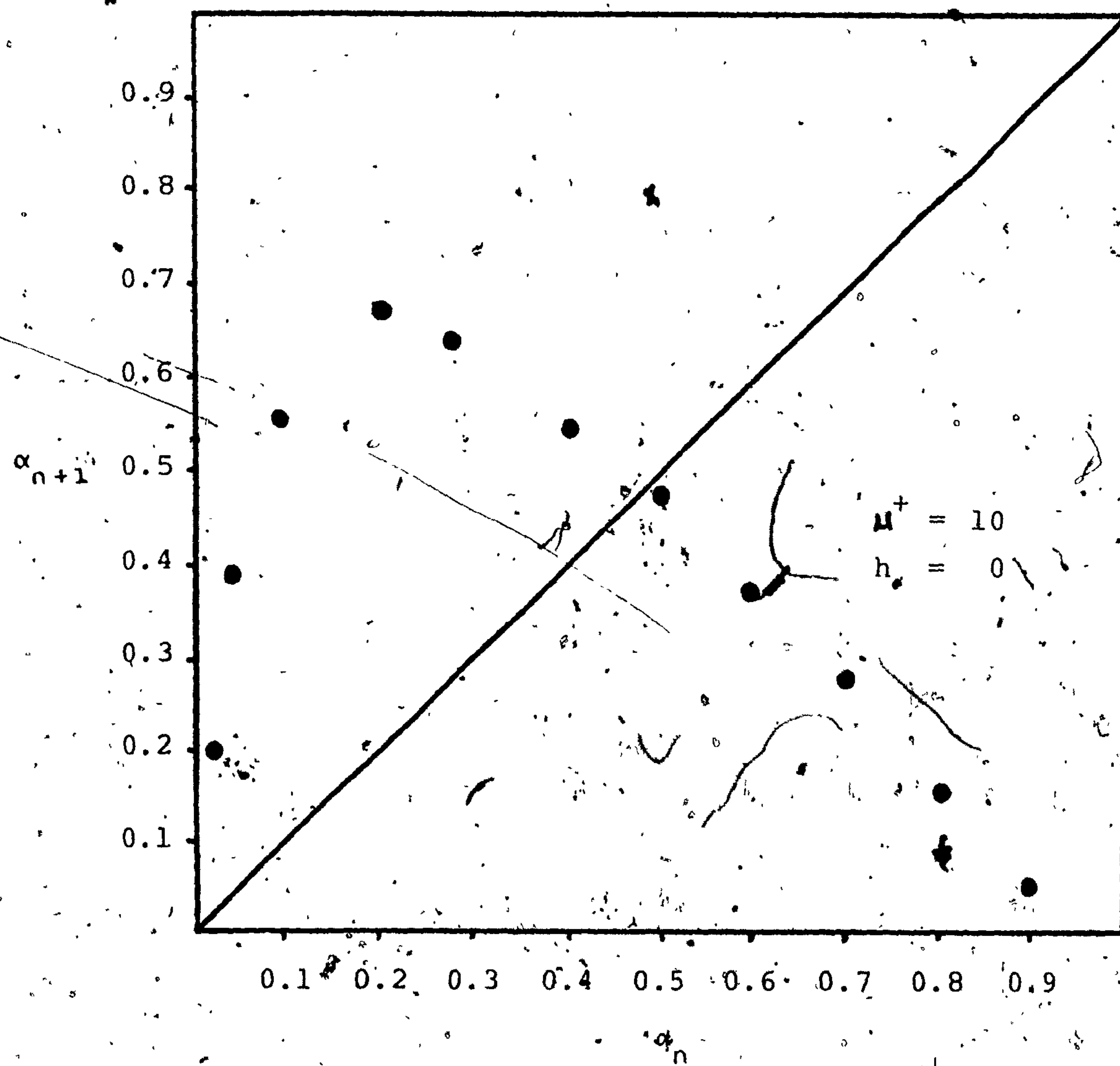

Fig. 29: Theoretical values of α_{n+1} vs. α_n for a net connected according to the poisson distribution law.

frequency of occurrence of EEG values obtained from the normal distribution curve that has the same arithmetic mean $\overline{X}$, and standard deviation S as the theoretical histogram. The expected frequencies were found by calculating the arithmetic mean for the theoretical EEG histogram and the standard deviation which was found by using the equation

$$S = \sqrt{\frac{\sum_{i=1} f_i \, (X_i - \overline{X})^2}{N}}$$

where f_i is the frequency of occurrence of certain EEG values X_i, and N is the total number of X_i. The deviation of each interval from the mean $\overline{X}$ was then written in units of the standard deviation by using

$$Z = \frac{X - \overline{X}}{S}$$

The area bounded by the normal curve

$$Y = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} Z^2}$$

and any two values of Z determines the proportion of the expected values of the EEG activity. When this is multiplied by the total number (N=200), the expected number of EEG values that fall between the two values of Z is obtained if the EEG activity is normally distributed. The area under the normal curve and the expected number of EEG values was obtained for the ten intervals of Fig. 28 and was then plotted. The χ^2 value along with the values of the arithmetic mean ($\overline{X}$) and the standard deviation (S) are also shown in this Figure. The high χ^2 value could be due to the fact that even with this net the time course of the activity α_n showed highly damped oscillation.

D. Presence of Cyclic Activity

It has been found by Anninos (1972) that sustained cyclic activity in a net seems to depend only on the statistical parameters of the net (such as the average number of outgoing connections, the percentage of inhibitory neurons, and the threshold), the nets considered in this study have similar statistical parameters to those considered by Anninos. The fact that one of the nets considered in this study (Net 4) not only showed the absence of the cyclic activity but also the lack of any sustained activity indicates that this net does not behave according to the findings of Anninos (1972). It was also found that increasing or decreasing the number of neurons firing initially did not affect the response of the net except for the fact that the delay until the activity died completely was dependent on the initial number of neurons firing. For example, with $\alpha_0 = .2$, the number of interactions (in synaptic delays) before the activity died completely was equal to 23, while with $\alpha_0 = .05$, it was equal to 20. The behaviour of the net was the same in the presence of evoked potentials delivered by any number of active neurons (αA_0) between 100 to 500 neurons. The fact that this net has similar statistical parameters to all the other nets considered clearly shows that the abnormal behaviour of this net is mainly due to the pattern of connectivities of the neurons in it. All other nets considered in this study showed the presence of sustained cyclic activity in the absence of external stimuli (Table 1) and also in the presence of external excitatory or inhibitory inputs applied to each net 100 synaptic delays after the initialization process.

Table 2 shows the effect of evoked potentials applied through a cable of active afferent fibers (A) connected randomly at the net. The average number of synaptic connections that each fiber makes with the neurons in the net is considered to be equal to

$$\mu_0^+ = \mu_0^- = 3$$

depending on whether it comes from an excitatory or inhibitory neuron respectively. In this table, σ^+ and σ^- represent the fraction of active excitatory or inhibitory fibers respectively. It can be seen from the table that an evoked response is elicited in each case. The stimulus was applied 100 synaptic delays after the initialization process and after the onset of the cyclic activity in each case. After each administration a deflection was

TABLE 2

EFFECT OF EVOKED POTENTIAL ON CYCLIC ACTIVITY

Net	σ^+	σ^-	Period of Cycle	Number of Synaptic Delays Before Cycling
1	.2		2	17
	.3		2	13
	.5		2	13
		.2	2	5
		.3	2	5
		.5	2	5
2	.1		318	284
	.2		318	143
		.1	318	243
		.2	318	263
3	.1		16	130
	.2		16	69
	.3		16	69
	.5		16	69
		.1	16	20
		.2	16	89
		.3	6	8
		.5	3	2
5	.1		38	36
	.2		38	17
	.3		38	47
		.1	38	20
		.2	38	24
		.3	38	20

observed one synaptic delay after. This was followed by a latency period before the reappearance of the cyclic activity. It was generally found that increasing the strength of the excitatory stimulus resulted in the decrease of the latency period as measured in term of the number of synaptic delays, while increasing the strength of the inhibitory stimulus resulted in an increase in the latency period. Although all the nets considered showed such a general behaviour, it was found that nets whose EEG activity did not show gaussian characteristics deviated from such a general behaviour more than the nets whose EEG activity was gaussian. Although such a criteria can be used to distinguish between the normal and abnormal nets we feel that more nets should be tested and more data collected to find out if this criteria is general.

E. Effect of Microscopic Structure on Net Behaviour

In the preceeding sections we discussed the similarities and differences that exist between the various nets considered in this study. In this section, we are going to see whether differences in nets behaviour are mainly due to the different connectivity patterns of their incoming and outgoing connections and whether some general properties can be found that characterize a particular behaviour. To do this, use will be made of frequency distribution of the total number of incoming connections (Figs. 9 to 13), the histogram of the total number of the incoming excitatory connections (Figs. 14 to 18), and the histograms of the total number of the incoming inhibitory connections to each neuron in the net (Figs. 19 to 23).

A statistical analysis will be made of these histograms to estimate their central tendency, their variance and skewness using the method of "moments". In physics, the position of the centre of mass for a discrete distribution of mass $m(x_j)$ at different values of x_j is given by:

$$x_{c.m} = \frac{\Sigma x_j m(x_j)}{\Sigma m(x_j)}$$

where x_j (j=1, 2 ...) is the distance of the mass $m(x_j)$ from the origin from which the centre of mass is measured.

In the terminology of mathematical statistics, this quantity is called the first moment and is given by (Pall, 1971):

$$m = \frac{\Sigma X_j}{N}$$

where X_j (j=1, 2 ...N) is a random variable with N values. It is also called the arithmetic mean ($\bar{X}$).

In general, the ith moment about the mean ($\bar{X}$) is defined as:

$$m_i = \frac{\Sigma_j (X_j - \bar{X})^i}{N}$$

m_2 (i=2) is called the variance (Speigel, 1961). The ith moment about any origin A is given by

$$m'_i = \frac{\Sigma_j (X_j - A)^i}{N}$$

The distribution properties of random variables can be measured by using a dimensionless ratio of the moments about the mean given by

$$a_i = \frac{m_i}{\sqrt{m_2^i}}$$

The skewness which is defined as the degree of asymmetry of a distribution is given by Speigel, 1961)

$$a_3 = \frac{m_3}{\sqrt{m_2^3}}$$

a_3 can be compared with the value of a_3 of the normal distribution which is equal to 0. A positive value of a_3 indicates that the frequency curve of a distribution has a larger tail to the right (Pall, 1971) and is called positively skewed. A negative value of a_3 indicates that it is negatively skewed.

The moment coefficient of Kurtosis is given by (Speigel, 1961)

$$a_4 = \frac{m_4}{m_2^2}$$

and is a measure of the degree of peakedness of a distribution. With respect to the normal distribution which has a value of $a_4=3$, a distribution with a value of $a_4>3$ is called leptokurtic distribution and is characterized by a relatively high peak. In contrast, if $a_4<3$ for a given frequency distribution, then it is called platykurtic and has a flatter top than the normal distribution.

Table 3 lists the values of the second third and fourth moments m_2, m_3, and m_4 about the mean $\bar{X}$ for the histogram of Figs. 9 to 23. Also shown are the values of a_3, a_4, and S.

It can be seen from this table that differences do exist in the skewness and Kurtosis of these frequency distributions. Examination of the values of a_3 and a_4 for the total number of excitatory incoming connections reveals that net 2 has a higher value of a_3 and a lower value of a_4 than net 1. This means that net 2 has a greater positive skewness than net 1 and it is platykurtic while net 1 is leptokurtic. Similarly, net 3 is skewed more to the right than net 4 and is less leptokurtic than net 4 relative to the normal distribution.

a_3 and a_4 values for total inhibitory incoming connections for these nets also show some variations. For example, a_4 for net 2 is greater than that for net 1 while a_3 for net 1 is greater than that for net 2. This means that the frequency distribution of the inhibitory incoming connections for net 2 is less skewed and more leptokurtic than the frequency distribution of the inhibitory incoming connections of net 1. Net 3 is also more leptokurtic than net 4. However, this net is more skewed than net 4.

TABLE 3

TOTAL EXCITATORY CONNECTIONS

Net	S	m_2	m_3	m_4	a_3	a_4
1	1.18	1.40	1.43	7.04	.86	3.59
2	1.34	1.8	2.22	6.88	0.91	2.12
3	1.31	1.74	2.39	11.04	1.04	3.65
4	1.31	1.74	2.20	11.28	0.95	3.72

TOTAL INHIBITORY CONNECTIONS

Net	S	m_2	m_3	m_4	a_3	a_4
1	1.01	1.03	1.24	3.69	1.19	3.48
2	.9	0.81	.8	4.55	1.09	6.93
3	1.01	1.02	0.95	3.51	0.92	3.37
4	0.88	0.79	0.47	1.96	0.67	3.14

TOTAL INPUT

1	1.59	2.53	2.78	21.78	0.69	3.4
2	1.63	2.67	3.60	26.07	1.64	3.66
3	1.58	2.51	2.38	19.52	0.59	3.09
4	1.66	2.75	4.05	26.00	0.89	3.44

Variations in the values of a_3 and a_4 is also apparent in the analysis of the histograms of total numbers of incoming connections of these nets.

Values a_3 and a_4 for the histograms of the excitatory incoming connections and also for the inhibitory incoming connections for net 2 indicate that the frequency distributions of net 2 show a definite pattern which differs from the pattern associated with net 1 which has identical statistical parameters to that of net 2.

The analysis of the histograms of the excitatory incoming connections of net 3 is similar to that of net 2, but the histogram of its inhibitory incoming connections differs from that of net 2 in being more skewed (relative to net 4). We may therefore conclude that net 3 has some normal tendencies. Examination of the value of χ^2 of net 3 shows that it is equal to 18.0. Although this is greater than the critical value of $\chi^2_{.95}$. The curve shows a greater normal tendency than other nets.

The behaviour of net 3 is also similar to that of net 2 in the presence of external stimuli (Table 2) increasing the fraction of the external active incoming connections for excitatory neurons (σ^+) had the effect of decreasing the number of synaptic delays before the start of the cyclic activity, while increasing (σ^-) had the opposite effect. Net 3 showed a similar behaviour to that of net 2 for a fixed period of the cyclic activity. The fact that net 4 was not able to sustain any activity is due to the particular pattern of the incoming connection of its cells. Fig. 12 shows that more neurons received a total number of two incoming connections than any other number of connections while net 3 (Fig. 11) shows that a greater number of neurons received a total of three connections than any other value. This fact might indicate that this net is not capable of sustained activity because of the presence of a large number of networks exerting negative feedback control on the cells of this net.

The mathematical analysis of the frequency distribution of the excitatory and inhibitory incoming connections indicate that differences do exist in the connectivity pattern of these nets and that these different connectivity patterns could be responsible for the different behavioural patterns of these nets.

F. The existence of neural multivibrators

One of the interesting phenomena observed in the dynamic behaviour of some of the simulated neural nets is the presence of two state periodic activity as can be seen from Table 1. An investigation of the operation of these two state cycles showed their similarity to the operation of the free running or the astable electronic multivibrator. The astable multivibrator circuit is a resistance-capacitance coupled two-stage amplifier with positive feedback. It has two quasi-stable states and the circuit alternates between these two states without requiring an external input trigger.

The basic circuit is shown in Fig. 30. Note that regeneration and therefore oscillation is provided by the closed loop consisting of the two transistors Q_1 and Q_2. The output of the first transistor is fed to the input of the second transistor while the output of Q_2 is fed to the input of Q_1.

When d-c. power is applied to the circuit the transistors will go into one of two possible unstable states due to random variations or to an unbalance in the components with one transistor going ON into conduction, and the other transistor going OFF. Let us assume that

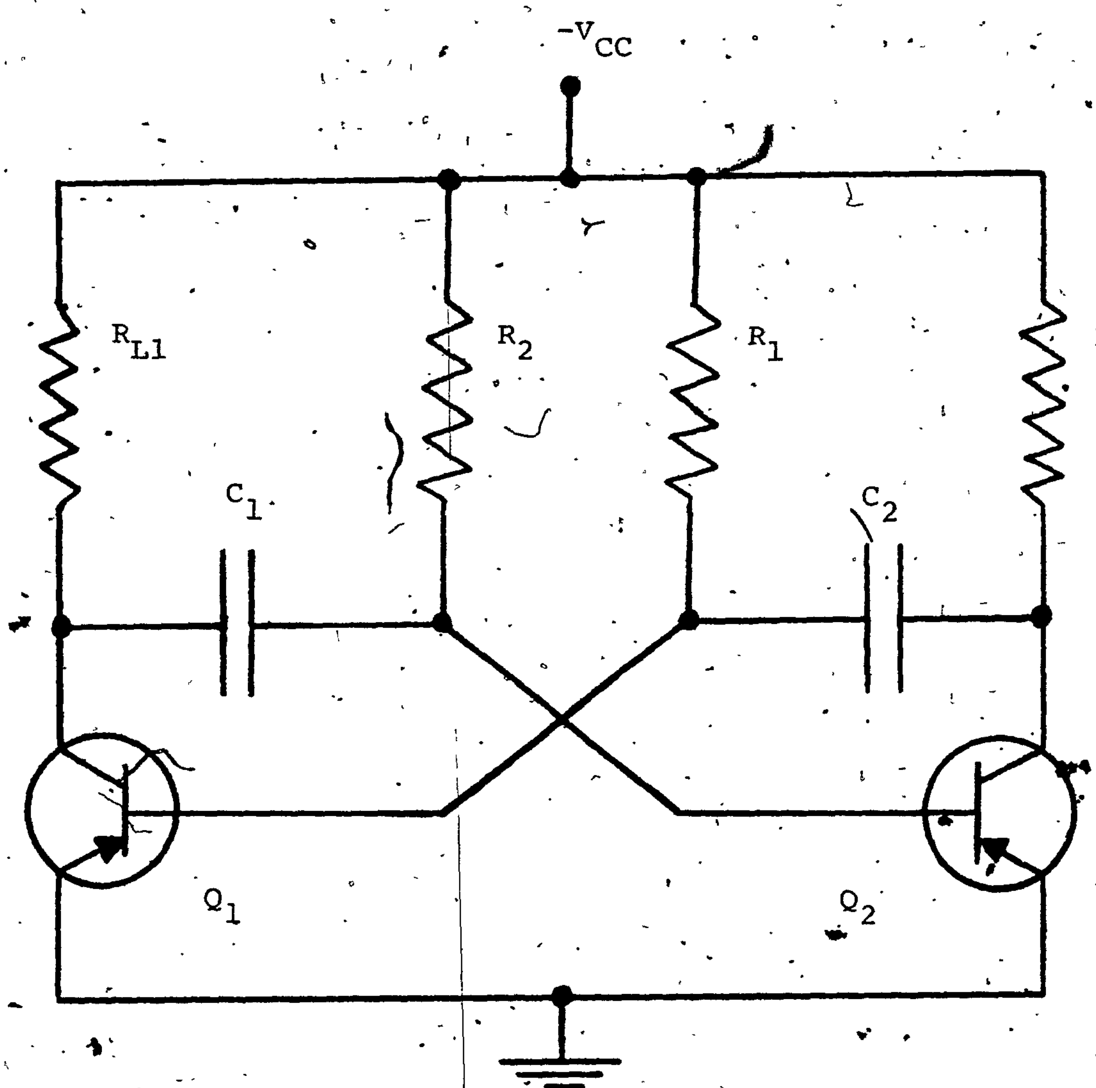

Fig. 30. An astable (free running) multivibrator circuit diagram.

transistor Q_1 starts conducting more heavily than transistor Q_2. This causes the collector of Q_1 to become more positive (close to ground potential). This positive voltage at the collector of Q_1 causes the base of Q_2 to become positive. The positive base voltage of Q_2 causes a decrease in the forward bias and a decrease in the collector current of Q_2 making the voltage at the collector to become more negative. This is fed back through capacitor C_2' to the base of Q_1 increasing its forward bias. The process continues until the base of Q_2 becomes so positive with respect to the emittor that Q_2 is cut off and Q_1 is saturated. This means that the collector of Q_1 is essentially at ground potential and the collector of Q_2 is essentially at $-V_{cc}$ since it is OFF and the voltage across C_2 is essentially equal to V_{cc} and will be fully charged as long as Q_2 is OFF. The process happens so quickly that C_1 does not have enough time to charge. After the saturation of Q_1, capacitor C_1 starts to charge through R_2 so that the base voltage of Q_2 goes negative to the point where forward bias is reestablished across Q_2 bringing it into conduction and causing the collector voltage to quickly become positive. This gives rise to a positive pulse to the base of Q_1 thus cutting off Q_1. With Q_1 OFF capacitor C_2 charges until the base of Q_1 is sufficiently negative with respect to the emitter for Q_1 to come ON again, and for the cycle to repeat. The periodic time for a complete cycle of the astable multivibrator operation is equal to the sum of the times that Q_1 is OFF and Q_2 is OFF.

The operation of two states cyclic activity in a randomly connected neural net is similar to the operation of the astable multivibrator. The period for a complete cycle is equal to the sum of the times that the first state and the second state are OFF. The first state triggers the second state into activity while it becomes inactive, and the second state after one synaptic delay (equivalent to the time that one transistor remains ON) becomes inactive while the first state because of feedback from the second state becomes active. The neural oscillator just like the electronic astable multivibrator continues to alternate between these two states without requiring an external input trigger.

Table 4 shows the connectivity matrix, which was constructed by using computer simulation, for 10 neurons of a class A net that exhibits the two state cycle. In this table each row represents the outgoing connections and each column the incoming connections for each neuron. Therefore the values of the first row represent the number of outgoing connections from neuron 1 to all the neurons in the net while the values in the first column give the number of the incoming connections to neuron 1. In this table a connection originating from an inhibitory neuron is indicated by a negative value. The circuit can be brought into the oscillating mode by triggering some of its neurons into action. If, for example, neurons 6 and 10 are triggered first, then the activity within the net will progress and will enter into its periodic oscillation three synaptic delays later. The cycle involves neuron 5 in one state and neuron 1, 4, and 7 in the other. These two states continue to alternate because the output of neuron 5 is connected to the input of neuron 1 while the output of neuron 1 is connected to the input of neuron 5 as can be seen from table 4. The connections between neuron 1 and 5 are excitatory which is analogous to the positive feedback that exists between the two transistors of the astable multivibrator. Another similarity that exists between the neural and the electronic multivibrators is the fact that the neural oscillator just like the electronic one will fail to operate if the two neurons are made active at the same time.

TABLE 4

The connectivity matrix

Neurons with incoming connections

Neurons with outgoing connections	1	2	3	4	5	6	7	8	9	10
1			1	1	1			1		
2			1	1	1					
3		-1		-1	-1				-1	
4				1		1	2	1		
5	1			1			1			
6			1		1					
7			-1			-1		-2	-1	
8		-2					-2			-1
9			-1			-1				
10			1	1						

Other pathways can also lead to the same two state cyclic activity, for example, triggering neurons 6 and 8 lead to the same cycle after two synaptic delays. Neurons 3 and 4 give the same effect after three synaptic delays.

Once the cyclic activity starts it will continue indefinitely until it is modified, modulated or destroyed by an external input delivered to the net. For example, if after the onset of the cyclic activity, with neurons 5 firing at one time and neurons 1, 4, and 7 firing in the next time, neuron 10 is activated by an external excitatory input so that it fires when neurons 1, 4, and 7 are firing then the periodic two state oscillations will cease but the net will continue to be active with different neurons firing every time for a total time of 8τ after which the same two states periodic activity comes back. Such a modulating effect can only come from outside the net since all the internal incoming connections of neuron 10 are inhibitory which means that this particular neuron can not be activated by any stimulus coming from within the net. If, on the other hand, neuron 10 is made to fire when neuron 5 fires then the periodic activity will be completely destroyed and will not come back unless another external input is applied to the right neurons to activate the neural oscillator. It can be seen from the above discussion that neuron 10 can act as a control unit for modulating or destroying the cyclic activity within the net and that such function can only be triggered from the outside.

Although the presence of oscillating circuits within the nervous system has been demonstrated in lower animals to control rhythmic functions (Wooldridge, 1963) the existence of complex networks which may behave like the electronic multivibrator circuit within the mamallian central nervous system and their biological significance should be investigated.

CHAPTER VI

CONCLUSION

It has been shown that nets having the same overall statistical parameters and connected according to the Poisson distribution law show completely different behavioural patterns. An investigation of the connecitivity patterns of these nets showed some differences in their connectivities. Although all these nets belong to Class A which should exhibit sustained activity and cyclic activity, one of these nets showed neither of these two phenomena. This behaviour can be due to the connectivity pattern and the inhibitory feedback networks acting within the net.

The results also showed that the frequency distribution curves of the excitatory and inhibitory incoming connections can be used to distinguish between normal and abnormal nets. It was found that nets whose frequency distribution curves of the excitatory incoming connections are platykurtic (relatively flat) and whose frequency distribution of the inhibitory incoming connections were leptokurtic (highly peaked) relative to the normal distribution curve showed gaussian EEG activity, while nets with other frequency distribution curve patterns showed an EEG activity that was not gaussian in character. Such a distinction between normal and abnormal nets based on the frequency distribution curves should be further investigated with nets connected according to other distribution laws to determine the limits and the conditions under which a net will have a gaussian EEG activity.

The theoretical results of the EEG activity of a net whose excitatory and inhibitory connections are distributed according to the poisson distribution law showed that theoretical EEG histograms of such nets were not distributed according to the gaussian probability distribution law.

It was also found that in general the application of an inhibitory stimulus to a normal net exhibiting cyclic acitivity resulted in an increase in the number of synaptic delays before the reappearance of the cycle. Since similar results but with more deviations were obtained for abnormal nets, we feel that such a criteria can not be generalized without testing more nets to determine whether the variations in the number of synaptic delays before the reappearance of the cyclic activity is due to the differences in connectivities of the nets or some other variables.

It was also shown that the criteria listed in Section A of Chapter V can be used to classify the nets according to their behavioural chracteristics and from the operational point of view into normal and abnormal nets depending on their capability to exhibit sustained activity and on how well the amplitude distribution of the EEG of the nets fit the Gaussian distribution. From the structural point of view, this classification was also supported by the variations of the frequency distribution of their microscopic connectivities.

Some of the nets considered in this study showed the existence of two states periodic activity that operated in a similar manner to the operation of the electronic astable multivibrator. The existence of such oscillating and logic circuits in the central nervous system and their biological significance should be investigated.

Our results suggest that the microscopic structure plays an important role in determining the behaviour of artificial nets. They also show that some variations in the connectivities are admissible without appreciably modifying the behaviour of the nets, (e.g. nets 2 and 3). The results reported here may provide some indications of the constraints which are imposed by specific anatomical

REFERENCES

Anninos, P.A., 1972, Kybernetik 11:5.

Anninos, P.A., B. Beak, T.J. Csermely, E.M. Harth, and G. Pertile, 1970, J. Theor. Biol. 26:121.

Anninos, P.A., and R. Elul, 1974a, Biophys. J. 14:8.

Anninos, P.A., and R. Elul, 1974b, In press.

Asanuma, H.H., and Sakata, 1967, J. Neurophysiol. 30:35.

Burele, R.L., 1956, Phil. Trans. Roy. Soc., London, Ser. B. 240:55.

Cainaniello, E.R., A. DeLuca, and L.M. Ricciardi, 1967, Kybernetik 4:10.

Colonnier, M.L., 1967, Arch. Neurol. 16:651.

Cowan, J.D., 1968, In Neural Networks, E.R. Cainaniello (ed.) Springer-Verlag, New York, pp. 181-188.

Croxton, F., 1959, Elementary Statistics, Dover, New York.

Cyrulnik, R.A., P. Anninos, and P. Marsh, 1974, Can. J. Neurol. Sci. 1:17.

Dichter, M., and W. Spencer, 1969, J. Neurophysiol. 32:663.

Eccles, J.C., 1964, The Physiology of Synapses, Academic Press, New York.

Eccles, J.C., 1965, Scientific American, January, pp. 166-176.

Elul, R., 1969, Science 164:328.

Elul, R., 1972, Synchronization of EEG Activity in Epilepsies, H. Pettsche and M. Brazier (eds), Springer-Verlag, New York.

Farley, B.G., and W.A. Clark, 1961, Activity in Networks of Neuron-like Elements, In Information Theory (Fourth London Symposium), C. Cherry (ed.), Butterworth and Co., Ltd., London.

Goldstein, L., H. Murphree, A. Sugerman, C. Pfeiffer, and E. Jenney, 1963, Clin, Pharmacol. Therap. 4:10.

Goldstein, L., A. Sugerman, W. Stolberg, H. Murphree, and C. Pfeiffer, 1965, Electroenceph. Clin. Neurophysiol. 19:350.

Harth, E.M., and S.L. Edgar, 1967, Biophys. J. 7:689.

Harth, E.M., J.J. Csermely, B. Beak, and R.D. Lindsay, 1970, J. Theor. Biol. 26:93.

Harth, E.M., N.S. Lewis, and T.J. Csermely, 1974, In press.

Hodgkin, A.L., and A.F. Huxley, 1952, J. Physiol. (London) 117:500.

House, E.L., and B. Pansky, 1967, A Functional Approach to Neuroanatomy, McGraw-Hill, New York.

Jasper, M.M., G. Ricci, and B. Doane, 1960, Microelectrode analysis in cortical cell discharges during avoidance conditioning in monkey. In the Moscow Colloquium on electroencephalography of higher nervous activity. M.M. Jasper and G.D. Smirnov, (eds.) Encephalog. and Clin. Neurophysiol. Suppl. 13.

Katz, B., 1961, Scientific American Sept., pp. 245-254.

Katz, B., 1966, Nerve, Muscle and Synapse, McGraw-Hill, New York.

Leake, B.D., and P. Anninos, 1974, In press.

Lewis, E.R., 1964, An electronic model of the neuron based on the dynamics of potassium and sodium ion fluxes. In Neural Theory and Modelling, R.F. Reiss (ed.), Stanford: Stanford Univ. Press.

Lindgren, B., and G. McElrath, 1967, Introduction to Probability and Statistics, The MacMillan Co., New York.

McCullock, W.S., and W. Pitts, 1943, Bull. Math. Biophys. 5:115.

Minsky, M.L., 1956, Some universal elements for finite automata, In Automata Studies, C.E. Shannon and J. McCarthy, (eds.), Princeton: Princeton Univ. Press.

Morrell, F., 1961, Physiol. Rev. 41:443.

Mount Castle, V.B., 1957, J. Neurophysiol. 20:408.

Noback, C.R., 1967, The Human Nervous System, McGraw-Hill, New York.

Pall, G., 1971, Introduction to Scientific Computing, Meredith, New York.

Penfield, W., and P.H. Perott, 1963, Brain 86:595.

Penfield, W., and T. Rasmussen, 1955, The Cerebral Cortex of Man, The MacMillan Co., New York.

Purpura, D.D., 1965, J. Neurophysiol. 28:925.

Rashevsky, N., 1971a, Bull. Math. Biophys. 31:539.

Rashevsky, N., 1971b, Bull. Math. Biophys. 33:561.

Rashevsky, N., 1972, Bull. Math. Biophys. 34:213.

Ricciardi, L.M., and H. Umezawa, 1967, Kybernetik 4:44.

Rochester, N., J.H. Holland, L.H. Haibt, and W.L. Duda, 1956, IRE Trans. Inform. Theory. 2:80.

Ruch, T.C., and H.D. Patton, 1965, Physiology and Biophysics, W.B. Saunders, Philadelphia.

Rugh, T.C., H.D. Patton, J.W. Woodbury, and A.L. Towe, 1965, Neurophysiology, W.B. Saunders, Philadelphia.

Sholl, D.A., 1956, The Organization of the Cerebral Cortex, Methuen, London.

Smith, D.R., and C.H. Davidson, 1962, J. Assoc. Comput. Math. 9:268.

Speigel, M.R., 1961, Statistics, McGraw-Hill, New York.

Truex, R.C., and M.B. Carpenter, 1969, Human Neuroanatomy, Williams and Wilkins, Baltimore.

Von Neumann, J., 1958, The Computer and the Brain, Yale University Press, New Haven, Conn.

Wiener, N., 1965, Prog. in Brain Res. 17:398.

Wilson, H.R., and J.D. Cowan, 1972, Biophys. J. 12:1.

Woodburne, L.S., 1967, The Neural Basis of Behavior, Merrill, Columbus.

Wooldridge, D.E., 1963, The Machinery of The Brain, McGraw Hill, New York.

APPENDIX

```
      OVERLAY(BRAIN,0,0)
      PROGRAM MAIN(INPUT,OUTPUT,TAPE1,TAPE2,TAPE5,
     1TAPE6=OUTPUT)
00003 REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
00003 REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AN,V,NOISE(100)
00003 REAL NSTD(10),TRESH3(1000),H
00003 INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
00003 INTEGER VNET(1000),FIRST,ITEM(20)
00003 INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
00003 INTEGER RECORD(1,1000)
00003 INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
00003 INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
00003 INTEGER PEINF(100),INC(100)
00003 INTEGER PRNT(10)
00003 DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
00003 DIMENSION LSTORN(5,1000),LSTORC(5,1000)
00003 LOGICAL LEARN,NULL,IFLAG,TMOD(10)
00003 LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
00003 COMMON/PARAM/ IOWORK,TITLE,INFO1,INFO2
00003 COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
00003 COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
00003 COMMON/PARAM/ NKXSUM,LCCSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
00003 COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
00003 COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
00003 COMMON/PARAM/LSTORN,LSTORC
00003 COMMON/PARAM/NUETS,NBLOCK,NTOTAL,NUMBER
00003
00003 COMMON/PARAM/IDENT,NSTIM,INFO,IOITER,NITER
00003 COMMON/PARAM/TAVG,TDCAY1,SDCAY,INDEX
00003 COMMON/PARAM/ FIRST,LAST,NFIRE
00003 COMMON/PARAM/IMUP,INUM,H,ICELL,VNET
00003 COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
00003 COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
00003 COMMON/PARAM/ PEINF,INC
00003 COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
00003 COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
00003 COMMON/PARAM/ FIRE,JOIN
00003 CALL OVERLAY(6HRICONS,1,0,0)
00006 CALL OVERLAY(6HNETSIM,2,0,0)
00011 STOP $ END
```

```
      OVERLAY(RICONG,1,0)
      PROGRAM RICONG
C*===================================================================*C
C* RICONGN2 - RANDOMLY INTERCONNECTED COMPOUND NET GENERATOR    NEW *
C*===================================================================*C
C
****DECLARATIONS
C
      REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
      REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
      REAL NSTD(10),TRESH3(1000),H
      INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
      INTEGER VNET(1000),FIRST,ITEM(20)
      INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
      INTEGER RECORD(1,1000)
      INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
      INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
      INTEGER REINF(100),INC(100)
      INTEGER PRNT(10)
      DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
      DIMENSION LSTORN(5,1000),LSTORC(5,1000)
      LOGICAL LEARN,NULL,IFLAG,TMOD(10)
      LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
      COMMON/PARAM/ IDWORK,TITLE,INFO1,INFO2
      COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
      COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
      COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
      COMMON/PARAM/JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
      COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
      COMMON/PARAM/LSTORN,LSTORC
      COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER

      COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
      COMMON/PARAM/TAVG,TDCAY1,SDCAY,INDEX
      COMMON/PARAM/ FIRST,LAST,NFIRE
      COMMON/PARAM/IMUP,IDUM,H,ICELL,VNET
      COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
      COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
      COMMON/PARAM/ REINF,INC
      COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
      COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
      COMMON/PARAM/ FIRE,JOIN
      DATA ISTAR/4H****/
C
C*****FORMAT STATEMENTS
C
    5 FORMAT(*1START OF RUN*)
  105 FORMAT(I8,15A4/I2,I5,F12.2)
  115 FORMAT(*1NET NO.*,I8,15A4///5X,* TOTAL NUMBER OF SUBNETS*,6X,*
     1NETS=*,I9/6X,* TOTAL NUMBER OF NEURONS*,6X,* NTOTAL=*,I9/6X,*
     2RAND. NO. GENERATOR. ,X*,5X,*=*,F14.2///)
  705 FORMAT(15A4/2I5,F5.0)
  755 FORMAT(6X,* SUBNET NO.*,I2,15A4//31X,* JFROM=*,I5/32X,
     1*JJFROM=*,I5/31X,* INH    =*,F9.3/31X,* INN    =*,I5/31X,* IFRCTN=
     2*,F9.3//)
  805 FORMAT(15A4/2I5,8I3)
  815 FORMAT(6X,* BLOCK NO.*,I2,15A4//31X,* ITO    =*,I5/31X,* IITO   =
```

```
              1*,I5/31X,* NKXMAX=*,I5/31X,* NKXMIN=*,I5/31X,* INKMAX=*,I5/31X,*
              2 INKMIN=*,I5/31X,* LCOMAX=*,I5/31X,* LCOMIN=*,I5/31X,* ICOMAX=*,
              3I5/31X,* ICOMIN=*,I5//)
000003     2405 FORMAT(*OSBRTN CONECT FAILED --- BRAIN SUPRESSED*)
000003     3005 FORMAT(* BRAIN ALREADY EXISTS --- NOT ALTERED, IDKORK=*,I8)
000003     9005 FORMAT(*-END OF RUN*)
000003     1200 FORMAT(6(5X,3I5))
           C
           C
000003          IRUN=0
000004          WRITE(6,5)
           C
           C*****READ IN STARTING PARAMETERS
           C
000010      100 IRUN=IRUN+1
000012          IF(IRUN.NE.1) GO TO 9000
000013          READ(2,105) IDKORK,TITLE,NNETS,NTOTAL,X
000030     9001 WRITE(6,115) IDKORK,TITLE,NNETS,NTOTAL,X
           C
           C*****INITIALIZE
           C
000046          YFL=RANF(X)
000051          X=0
000052          DO 200 K=1,4096
000053          INHIB(K)=.FALSE.
000054      200 CONTINUE
000056          DO 300 K=1,10
000057          NNEUR1(K)=0
000060          NNEUR2(K)=0
000061      300 CONTINUE
000063          NBLOCK=NNETS**2
000065          DO 400 K=1,100
000066          NKFRST(K)=0
000067          NKLAST(K)=0
000070      400 CONTINUE
           C
           C*****DEFINE SUBNETS IN COMPOUND NET
           C
000072          INN=0
000073          NKFRST(1)=1
000074          DO 900 K=1,NNETS
000075          READ(2,705) INFO1,JFROM,JJFROM,INH
000110          NNEUR1(K)=JFROM
000112          NNEUR2(K)=JJFROM
000114          IF(INH.EQ.0) GO TO 750
           C
           C*****DECIDE WHICH NEURONS ARE INHIBITORY IN K-TH SUBNET
           C
000115      500 INN=IFIX(FLOAT(JJFROM-JFROM+1)*INH+0.5)
000123          DO 700 L=1,INN
000124      600 YFL=RANF(X)
000127          J=JFROM+IFIX(FLOAT(JJFROM-JFROM)*YFL+0.5)
000135          IF(INHIB(J)) GO TO 600
000137          INHIB(J)=.TRUE.
000140      700 CONTINUE
000142      750 IFRCTN=FLOAT(INN)/FLOAT(JJFROM-JFROM+1)
000147          WRITE(6,755)K,INFO1,JFROM,JJFROM,INH,INN,IFRCTN
           C
```

```
              1*,I5/31X,* NKXMAX=*,I5/31X,* NKXMIN=*,I5/31X,* INKMAX=*,I5/31X,*
              2 INKMIN=*,I5/31X,* LCOMAX=*,I5/31X,* LCOMIN=*,I5/31X,* ICOMAX=*,
              3I5/31X,* ICOMIN=*,I5//)
000003    2405 FORMAT(*OSBRTN CONECT FAILED --- BRAIN SUPRESSED*)
000003    3005 FORMAT(* BRAIN ALREADY EXISTS --- NOT ALTERED, IOKORK=*,I8)
000003    9005 FORMAT(*-END OF RUN*)
000003    1200  FORMAT(6(5X,3I5))
          C
          C
000003          IRUN=0
000004          WRITE(6,5)
          C
          C*****READ IN STARTING PARAMETERS
          C
000010     100  IRUN=IRUN+1
000012          IF(IRUN.NE.1) GO TO 9000
000013          READ(2,105)IOKORK,TITLE,NNETS,NTOTAL,X
000030    9001  WRITE(6,115)IOKORK,TITLE,NNETS,NTOTAL,X
          C
          C*****INITIALIZE
          C
000046          YFL=RANF(X)
000051          X=0
000052          DO 200 K=1,4096
000053          INHIB(K)=.FALSE.
000054     200  CONTINUE
000056          DO 300 K=1,10
000057          NNEUR1(K)=0
000060          NNEUR2(K)=0
000061     300  CONTINUE
000063          NBLOCK=NNETS**2
000065          DO 400 K=1,100
000066          NKFRST(K)=0
000067          NKLAST(K)=0
000070     400  CONTINUE
          C
          C*****DEFINE SUBNETS IN COMPOUND NET
          C
000072          INN=0
000073          NKFRST(1)=1
000074          DO 900 K=1,NNETS
000075          READ(2,705)INFO1,JFROM,JJFROM,INH
000110          NNEUR1(K)=JFROM
000112          NNEUR2(K)=JJFROM
000114          IF(INH.EQ.0) GO TO 750
          C
          C*****DECIDE WHICH NEURONS ARE INHIBITORY IN K-TH SUBNET
          C
000115     500  INN=IFIX(FLOAT(JJFROM-JFROM+1)*INH+0.5)
000123          DO 700 L=1,INN
000124     600  YFL=RANF(X)
000127          J=JFROM+IFIX(FLOAT(JJFROM-JFROM)*YFL+0.5)
000135          IF(INHIB(J)) GO TO 600
000137          INHIB(J)=.TRUE.
000140     700  CONTINUE
000142     750  IFRCTN=FLOAT(INN)/FLOAT(JJFROM-JFROM+1)
000147          WRITE(6,755)K,INFO1,JFROM,JJFROM,INH,INN,IFRCTN
          C
```

```
      C
C*****INTERCONNECT NEURONS OF K-TH SUBNET WITH REST IN COMPOUND NET
C
           L1=1+(K-1)*NNETS
           L2=K*NNETS
           DO 800 L=L1,L2
           READ(2,805)INFO2,ITO,IITO,NKXMAX,NKXMIN,INKMAX,INKMIN,LCOMAX,LCOMI
          1N,ICOMAX,ICOMIN
           IF(INFO2(1).EQ.ISTAR) GO TO 800
           WRITE(6,815)L,INFO2,ITO,IITO,NKXMAX,NKXMIN,INKMAX,INKMIN
          1,LCOMAX,LCOMIN,ICOMAX,ICOMIN
           CALL CONECT(I1)
           IF(I1.EQ.1) GO TO 2400
           NUMBER=NKXSUM+INKSUM
           NKLAST(L)=NUMBER
           IF(L.LT.NBLOCK) NKFRST(L+1)=NKLAST(L)+1
       800 CONTINUE
       900 CONTINUE
C
C*****PRINT KORK ARRAY
C
           DO 950 I=1,NBLOCK
           IF((NKLAST(I)-NKFRST(I)).GT.0) GO TO 950
           NKFRST(I)=0
           NKLAST(I)=0
       950 CONTINUE
           LX=NKXMAX
      1000 CONTINUE
           WRITE(6,1200)(J,(LSTORN(L,J),LSTORC(L,J),L=1,LX),J=1,NTOTAL)
           CALL LIST
           GO TO 100
      2400 WRITE(6,2405)
           GO TO 9000
      3000 PRINT 3005,IDKORK
           GO TO 100
      9000 WRITE(6,9005)
           WRITE(6,9010) NTOTAL
      9010 FORMAT(5X,* NTOTAL =*,I10)
           RETURN
           END
C
```

```
      SUBROUTINE CONECT(I1)
C
C*****DECLARATIONS
C
      REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
      REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
      REAL NSTD(10),TRESH3(1000),H
      INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
      INTEGER VNET(1000),FIRST,ITEM(20)
      INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
      INTEGER RECORD(1,1000)
      INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
      INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
          INTEGER REINF(100),INC(100)
      INTEGER PRNT(10)
      DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
      DIMENSION LSTORN(5,1000),LSTORC(5,1000)
      LOGICAL LEARN,NULL,IFLAG,TMOD(10)
      LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
      COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
      COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
      COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
      COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
      COMMON/PARAM/JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
      COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
      COMMON/PARAM/LSTORN,LSTORC
      COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER

      COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
      COMMON/PARAM/TAVG,TDCAY1,SDCAY,INDEX
       COMMON/PARAM/ FIRST,LAST,NFIRE
      COMMON/PARAM/IMUP,IMUM,H,ICELL,VNET
      COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
      COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
      COMMON/PARAM/ REINF,INC
      COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
      COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
      COMMON/PARAM/ FIRE,JOIN
C
C*****FORMAT STATEMENTS
C
  555 FORMAT(31X,* NXNEUR=*,I6/31X,* NKXSUM=*,I6/31X,* XNKAVG=*,F9.2/3
     11X,* XCOAVG=*,F9.2//31X,* INNEUR=*,I6/31X,* INKSUM=*,I6/31X,* IN
     2KAVG=*,F9.2/31X,* ICOAVG=*,F9.2/31X,* IFRCTN=*,F10.3///)
  605 FORMAT(* KORK ARRAY OVERFLOWED*)
C
C*****INITIALIZE
      I1=0
      NKXSUM=0
      INKSUM=0
      NXNEUR=0
      INNEUR=0
      LCOSUM=0
      ICOSUM=0
      XNKAVG=0
      INKAVG=0
      XCOAVG=-0.0
```

```
000014        ICOAVG=-0.0
      C
      C*****TAKE NEURONS ONE-BY-ONE WITHIN LIMITS
      C
000016     60 DO 500 J=JFROM,JJFROM
000020        IF(INHIB(J)) GO TO 100
      C
      C*****COMPUTE NO. OF EFFERENT SYNAPSES FOR EXCITATORY NEURONS
      C
000022        YFL=RANF(X)
000024        NKX=NKXMIN+IFIX(FLOAT(NKXMAX-NKXMIN)*YFL+0.5)
000032        NKXSUM=NKXSUM+NKX
000034        NXNEUR=NXNEUR+1
000035        XNKAVG=FLOAT(NKXSUM)/FLOAT(NXNEUR)
000040        NK=NKX
000041        MIN=LCOMIN
000043        MAX=LCOMAX
000044        GO TO 200
      C
      C*****COMPUTE NO. OF EFFERENT SYNAPSES FOR INHIBITORY NEURONS
      C
000046    100 YFL=RANF(X)
000051        INK=INKMIN+IFIX(FLOAT(INKMAX-INKMIN)*YFL+0.5)
000057        INKSUM=INKSUM+INK
000061        INNEUR=INNEUR+1
000062        INKAVG=FLOAT(INKSUM)/FLOAT(INNEUR)
000065        NK=INK
000066        MIN=ICOMIN
000070        MAX=ICOMAX
      C*****ASSIGN COUPLING COEFFICIENT
      C
000071    200 DO 400 L=1,NK
000074        YFL=RANF(X)
000076        LSTORC(L,J)=MIN+IFIX(FLOAT(MAX-MIN)*YFL+0.5)
000106        IF(LSTORC(L,J).LT.0)GO TO 250
000111        LCOSUM=LCOSUM+LSTORC(L,J)
000114        GO TO 300
000114    250 ICOSUM=ICOSUM+LSTORC(L,J)
000120    300 YFL=RANF(X)
      C
      C*****FIND POSTSYNAPTIC NEURON AND STORE I,J,COE
      C
000123        LSTORN(L,J)=ITO+IFIX(FLOAT(IITO-ITO)*YFL+0.5)
000133    400 CONTINUE
000136    500 CONTINUE
      C
      C*****PRINT AND RETURN
      C
000140        IF(NKXSUM.NE.0.0) XCOAVG=FLOAT(LCOSUM)/FLOAT(NKXSUM)
000144        IF(INKSUM.NE.0.0) ICOAVG=FLOAT(ICOSUM)/FLOAT(INKSUM)
000150        IFRCTN=FLOAT(INNEUR)/FLOAT(JJFROM-JFROM+1)
000156        WRITE(6,555)NXNEUR,NKXSUM,XNKAVG,XCOAVG,INNEUR,INKSUM,INKAVG,
             1ICOAVG,IFRCTN
000204        RETURN
000205    600 WRITE(6,605)
000211        I1=1
000213        RETURN
000214        END
```

```
             SUBROUTINE LIST
          C
          C*****DECLARATIONS
          C
 00002          REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
 00002          REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
 00002          REAL NSTD(10),TRESH3(1000),H
 00002          INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
 00002          INTEGER VNET(1000),FIRST,ITEM(20)
 00002          INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
 00002          INTEGER RECORD(1,1000)
 00002          INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
 00002          INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
 00002              INTEGER REINF(100),INC(100)
 00002          INTEGER PRNT(10)
 00002          DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
 00002          DIMENSION LSTORN(5,1000),LSTORC(5,1000)
 00002          LOGICAL LEARN,NULL,IFLAG,TMOD(10)
 00002          LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
 00002          COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
 00002          COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
 00002          COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
 00002          COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
 00002          COMMON/PARAM/JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
 00002          COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
 00002          COMMON/PARAM/LSTORN,LSTORC
 00002          COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER
 00002
 00002          COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
 00002          COMMON/PARAM/TAVG,TDCAY1,SDCAY,INDEX
 00002           COMMON/PARAM/ FIRST,LAST,NFIRE
 00002          COMMON/PARAM/IMUP,IMUM,H,ICELL,VNET
 00002          COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
 00002          COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
 00002          COMMON/PARAM/ REINF,INC
 00002          COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
 00002          COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
 00002          COMMON/PARAM/ FIRE,JOIN
          C
          C*****FORMAT STATEMENTS
          C
 00002        5 FORMAT(*1DISTRIBUTION OF CONNECTIONS IN COMPOUND NET*///)
 00002       21 FORMAT(10X,I10/10X,I10/)
 00002       22 FORMAT(20X,I10/20X,I10/)
 00002       23 FORMAT(30X,I10/30X,I10/)
 00002       24 FORMAT(40X,I10/40X,I10/)
 00002       25 FORMAT(50X,I10/50X,I10/)
 00002       26 FORMAT(60X,I10/60X,I10/)
 00002       27 FORMAT(70X,I10/70X,I10/)
 00002       28 FORMAT(80X,I10/80X,I10/)
 00002       29 FORMAT(90X,I10/90X,I10/)
 00002       30 FORMAT(100X,I10/100X,I10/)
 00002       41 FORMAT(*1DISTRIBUTION OF NEURONS IN SUBNETS*///)
 00002       45 FORMAT(10X,10I10)
 00002       46 FORMAT(10X,10I10/)
 00002       55 FORMAT(*1KORK ARRAY: IDKORK=*,I10,*(PRT ORDER:K,I,J,COE)*//)
 00002      555 FORMAT(5(5X,4I5))
```

```
00002      655 FORMAT(' END OF ARRAY')
00002      615 FORMAT(*1INHIB ARRAY.  IDKOPK=*,I1(//(10(1X,10I1)))
          C
          C*****PRINT NNEUR1 AND NNEUR2 ARRAYS
          C
00002          PRINT 5
00006          INDEX=0
00007       10 INDEX=INDEX+1
00011          IF(INDEX.GT.NNETS) GO TO 40
00014          GO TO (11,12,13,14,15,16,17,18,19,20),INDEX
00031       11 PRINT 21,NNEUR1(1),NNEUR2(1)
00041          GO TO 10
00042       12 PRINT 22,NNEUR1(2),NNEUR2(2)
00052          GO TO 10
00053       13 PRINT 23,NNEUR1(3),NNEUR2(3)
00063          GO TO 10
00064       14 PRINT 24,NNEUR1(4),NNEUR2(4)
00074          GO TO 10
00075       15 PRINT 25,NNEUR1(5),NNEUR2(5)
00105          GO TO 10
00106       16 PRINT 26,NNEUR1(6),NNEUR2(6)
00116          GO TO 10
00117       17 PRINT 27,NNEUR1(7),NNEUR2(7)
00127          GO TO 10
00130       18 PRINT 28,NNEUR1(8),NNEUR2(8)
00140          GO TO 10
00141       19 PRINT 29,NNEUR1(9),NNEUR2(9)
00151          GO TO 10
00152       20 PRINT 30,NNEUR1(10),NNEUR2(10)
00162          GO TO 10
          C
          C*****PRINT NKFRST AND NKLAST ARRAYS
          C
00163       40 PRINT 41
00167          DO 48 I=1,NNETS
00171          DO 44 J=1,NNETS
00172          ITEM(J)=NKFRST(I+(J-1)*NNETS)
00177          K=10+J
00201          ITEM(K)=NKLAST(I+(J-1)*NNETS)
00207       44 CONTINUE
00210          PRINT 45,(ITEM(L),L=1,NNETS)
00223          PRINT 46,(ITEM(L),L=11,NNETS)
00236       48 CONTINUE
          C
          C*****PRINT KCPK ARRAY
          C
          C
          C*****ROUTINE
          C
00241          K=0
00242          LL=0
00243          PRINT 55,IDKORK
00250      100 DO 200 I=1,NKYMAX
00252          ITEM(I)=0
00253      200 CONTINUE
00255          K=K+1
00257          IF(K.GT.NUMBER) GO TO 500
00262          DO 400 J=JFROM,JJFROM
```

```
      DO 350 L=1,NKXMAX
      ITEM(L)=K
      ITEM(L+1)=LSTCRN(L,J)
      ITEM(L+2)=J
      ITEM(L+3)=LSTORC(L,J)
      IP=L+3
C     WRITE(6,505)(ITEM(I),I=L,IP)
      K=K+1
      IF(K.GT.NUMBER)GO TO 500
  350 CONTINUE
      DO 550 I=1,NKXMAX
      ITEM(I)=0
  550 CONTINUE
  400 CONTINUE
  500 WRITE(6,605)
      WRITE(6,615)IDKORK,(INPIB(I),I=1,NTOTAL)
      RETURN
      END
```

```
              OVERLAY(NETSIM,2,0)
              PROGRAM NETSIM
000003        REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
000003        REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
000003        REAL NSTD(10),TRESH3(1000),H
000003        INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
000003        INTEGER VNET(1000),FIRST,ITEM(20)
000003        INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
000003        INTEGER RECORD(1,1000)
000003        INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
000003        INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
000003        INTEGER REINF(100),INC(100)
000003        INTEGER PRNT(10)
000003        DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
000003        DIMENSION LSTORN(5,1000),LSTORC(5,1000)
000003        LOGICAL LEARN,NULL,IFLAG,TMOD(10)
000003        LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
000003        COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
000003        COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
000003        COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
000003        COMMON/PARAM/ NKXSUM,LCCSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
000003        COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
000003        COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
000003        COMMON/PARAM/LSTORN,LSTORC
000003        COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER
000003
000003        COMMON/PARAM/IDENT,NSTIM,INFO,IOITER,NITER
000003        COMMON/PARAM/ TAVG,TDCAY1,SDCAY,INDEX
000003        COMMON/PARAM/FIRST,LAST,NFIRE
000003        COMMON/PARAM/INUP,INUM,H,ICELL,VNET
000003        COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
000003        COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
000003        COMMON/PARAM/ REINF,INC
000003        COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
000003        COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
000003        COMMON/PARAM/ FIRE,JOIN
        C*****FORMAT STATEMENTS
        C
000003
000003  50005 FORMAT(*1START OF RUN*)
000003  50016 FORMAT(5I4)
000003  50025 FORMAT(I8,F12.2)
000003  50026 FORMAT(2I3)F5.3,I4)
000003  50055 FORMAT(I4,15A4)
000003  50056 FORMAT(5I5,2F8.6)
000003  50145 FORMAT(I2,15A4/I4,30Z0)
000003  50150 FORMAT(2I2)
000003  50175 FORMAT(2I4)
000003  50905 FORMAT(*-LINK TO KORK FAILED --- RUN TERMINATED*)
000003  60000 FORMAT(*-END OF RUN*)
        C
        C*****IDENTIFICATION,INITIALIZATION AND PARAMETER ENTRY
        C
000003        WRITE (6,50005)
000007    100  READ(5,50025)IDKORK,X
000017        IF(IDKORK.EQ.999) GO TO 1000
000021    110 FORMAT(5X,I8,F14.2)
```

```
00021           LX=NKXMAX
00023           WRITE(6,110) IDKORK,X
00032      1001 INDEX=0
00033           CALL ZERO
00034           YFL=RANF(X)
00037           X=0
00040           READ(5,50026)IMUP,IMUM,H,ICELL
      C
      C*****IDENTIFY RUN
      C
00053       150 READ(5,50055) IDENT,TITLE
00063           INDEX=INDEX+1
00065           IF(IDENT.EQ.-1) GO TO 100
      C
      C*****SET UP THRESHOLDS
      C
00067           DO 250 I=1,NNETS
00070       200 READ(5,50056)TMOD(I),TMIN(I),TMAX(I),THR1(I),ARP(I),TDCAY1(I),
               1SDCAY(I)
00112           IF (TMOD(I)) CALL THRSET(I)
00116       250 CONTINUE
      C
      C*****PRINT PARAMETERS AND THRESHOLD ARRAY
      C
00121           CALL LIST1
      C
      C*****READ IN STIMULUS AND COMPUTE ALFA-N STATES
      C
00122       300 READ(5,50145)NSTIM,INFC,NITER,LEARN,NULL,IFLAG
00142           IF(NSTIM.EQ.-1) GO TO 150
00144
00146
00147
00157
00162
00164           CALL ZERO1
00165           IF(.NOT.IFLAG) GO TO 620
00167           DO 600 I=1,NNETS
00170       550 READ(5,50016) NSPONT(I),NEXT(I),INHIB(I),EXTCOE(I),IEXT(I)
00206       600 CONTINUE
00211
00212
00221
00224       620 CALL LIST2
00225       650 READ(5,50175)FIRST,LAST
00235           IF(FIRST.EQ.-1) GO TO 700
00237           CALL SET1(I1)
00241           IF(I1.EQ.1) GO TO 650
00243       700 CALL CYCLES
00244           CALL LIST3(I2)
00246           IF(I2.EQ.1) GO TO 300
      C
      C*****EXIT
      C
00250       900 WRITE(6,50905)
00254      1000 WRITE (6,60000)
00260           RETURN
00262           END
```

```
              SUBROUTINE ZERO
        C
        C*****DECLARATIONS
        C
00002         REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
00002         REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AN,V,NOISE(100)
00002         REAL NSTD(10),TRESH3(1000),H
00002         INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
00002         INTEGER VNET(1000),FIRST,ITEM(20)
00002         INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
00002         INTEGER RECORD(1,1000)
00002         INTEGER TMIN(10),TMAX(10),TDROP(10),APP(10),THR1(10),VT(1000)
00002         INTEGER NSPONT(10),NEXT(10),EXTCOE(10),PCRD,IEXT(10)
00002         INTEGER REINF(100),INC(100)
00002         INTEGER PRNT(10)
00002         DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
00002         DIMENSION LSTORN(5,1000),LSTORC(5,1000)
00002         LOGICAL LEARN,NULL,IFLAG,TMOD(10)
00002         LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
00002         COMMON/PARAM/ IDWORK,TITLE,INFO1,INFO2
00002         COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
00002         COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
00002         COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
00002         COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
00002         COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
00002         COMMON/PARAM/LSTORN,LSTORC
00002         COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER
00002
00002         COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
00002         COMMON/PARAM/ TAVG,TDCAY1,SDCAY,INDEX
00002         COMMON/PARAM/FIRST,LAST,NFIRE
00002         COMMON/PARAM/IMUP,IMUM,H,ICELL,VNET
00002         COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
00002         COMMON/PARAM/ TMIN,TMAX,TDROP,APP,THR1,PCRD
00002         COMMON/PARAM/ REINF,INC
00002         COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
00002         COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
00002         COMMON/PARAM/ FIRE,JOIN
        C
00002         DO 100 I=1,NTOTAL
00004         IF(INDEX.EQ.0) TRESH1(I)=0
00006         STATE(I)=0
00010         TRESH2(I)=TRESH1(I)
00011         FIRE(I)=.FALSE.
00012         JOIN(I)=.FALSE.
00013     100 CONTINUE
00016         RETURN
00016         END
```

```
      SUBROUTINE ZERO1
C
C*****DECLARATIONS
C
      REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
      REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
      REAL NSTD(10),TRESH3(1000),H
      INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
      INTEGER VNET(1000),FIRST,ITEM(20)
      INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
      INTEGER RECORD(1,1000)
      INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
      INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
      INTEGER REINF(100),INC(100)
      INTEGER PRNT(10)
      DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
      DIMENSION LSTORN(5,1000),LSTORC(5,1000)
      LOGICAL LEARN,NULL,IFLAG,TMOD(10)
      LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
      COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
      COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
      COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
      COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
      COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
      COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
      COMMON/PARAM/LSTORN,LSTORC
      COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER

      COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
      COMMON/PARAM/ TAVG,TDCAY1,SDCAY,INDEX
      COMMON/PARAM/FIRST,LAST,NFIRE
      COMMON/PARAM/IMUP,IMUN,H,ICELL,VNET
      COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
      COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
      COMMON/PARAM/ REINF,INC
      COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
      COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
      COMMON/PARAM/ FIRE,JOIN
      DO 5 I=1,NITER
      VNET(I)=0
    5 VT(I)=0
      DO 200 K=1,NITER
      DO 100 J=1,NNETS
      RECORD(J,K)=0
  100 CONTINUE
  200 CONTINUE
      RETURN
      END
```

```
         SUBROUTINE CYCLES
      C
      C*****DECLARATIONS
      C
00002    REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
00002    REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AH,V,NOISE(100)
00002    REAL NSTD(10),TRESH3(1000),H
00002    INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
00002    INTEGER VNET(1000),FIRST,ITEM(20)
00002    INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
00002    INTEGER RECORD(1,1000)
00002    INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
00002    INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
00002    INTEGER REINF(100),INC(100)
00002    INTEGER PRNT(10)
00002    DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
00002    DIMENSION LSTORN(5,1000),LSTORC(5,1000)
00002    LOGICAL LEARN,NULL,IFLAG,TMOD(10)
00002    LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
00002    COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
00002    COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
00002    COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
00002    COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
00002    COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
00002    COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
00002    COMMON/PARAM/LSTORN,LSTORC
00002    COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER
00002
00002    COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
00002    COMMON/PARAM/ TAVG,TDCAY1,SDCAY,INDEX
00002    COMMON/PARAM/FIRST,LAST,NFIRE
00002    COMMON/PARAM/IMUP,IMUN,H,ICELL,VNET
00002    COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
00002    COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
00002    COMMON/PARAM/ REINF,INC
00002    COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
00002    COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
00002    COMMON/PARAM/ FIRE,JOIN
      C
00002    NN=NITER+1
00004    DO 700 K=1,NN
00006    IDITER=100000*IDENT+1000*NSTIM+(K-1)
00014    IF(K.EQ.1) GO TO 100
00016    CALL SET
00017    CALL THINK(K)
00021 100 IF(.NOT.IFLAG) GO TO 400
00023    DO 200 L=1,NNETS
00024
00034    IF(NEXT(L).NE.0.AND.K.NE.1) CALL SET3(L,K)
00046    IF(IEXT(L).NE.0.AND.K.NE.1) CALL SET3(L,K)
00060 200 CONTINUE
00063
00064
00075
00100 400 CALL COUNT(K)
00102
00104
```

```
000106
000107
000113
000116            IF(RCRD.EQ.0.AND..NOT.IFLAG) GO TO 800
000122        700 CONTINUE
000125
000127
000131
000135
000140        800 RETURN
000140            END
```

```
              SUBROUTINE THRSET(I)
        C
        C*****DECLARATIONS
        C
000003        REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
000003        REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
000003        REAL NSTD(10),TRESH3(1000),H
000003        INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
000003        INTEGER VNET(1000),FIRST,ITEM(20)
000003        INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
000003        INTEGER RECORD(1,1000)
000003        INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
000003        INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
000003        INTEGER REINF(100),INC(100)
000003        INTEGER PRNT(10)
000003        DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
000003        DIMENSION LSTORN(5,1000),LSTORC(5,1000)
000003        LOGICAL LEARN,NULL,IFLAG,TMOD(10)
000003        LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
000003        COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
000003        COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
000003        COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
000003        COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
000003        COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
000003        COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
000003        COMMON/PARAM/LSTORN,LSTORC
000003        COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER
000003
000003        COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
000003        COMMON/PARAM/ TAVG,TDCAY1,SDCAY,INDEX
000003        COMMON/PARAM/FIRST,LAST,NFIRE
000003        COMMON/PARAM/INUP,INUM,H,ICELL,VNET
000003        COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
000003        COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
000003        COMMON/PARAM/ REINF,INC
000003        COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
000003        COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
000003        COMMON/PARAM/ FIRE,JOIN
        C
000003        SUM=0
000004        LL1=NNEUR1(I)
000006        LL2=NNEUR2(I)
000007        NNEUR=NNEUR2(I)-NNEUR1(I)+1
000012        DIFF=TMAX(I)-TMIN(I)
000014        DO 100 J=LL1,LL2
000016        YFL=RANF(X)
000020        TRESH1(J)=TMIN(I)+IFIX(DIFF*YFL+0.5)
000026        IF(INDEX.EQ.1) TRESH2(J)=TRESH1(J)
000032        SUM=SUM+TRESH1(J)
000035    100 CONTINUE
000037        TAVG(I)=(SUM)/FLOAT(NNEUR)
000042        RETURN
000043        END
```

```
             SUBROUTINE SET
       C
       C*****DECLARATIONS
       C
000002       REAL INF,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
000002       REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
000002       REAL NSTD(10),TRESH3(1000),H
000002       INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
000002       INTEGER VNET(1000),FIRST,ITEM(20)
000002       INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
000002       INTEGER RECORD(1,1000)
000002       INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
000002       INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
000002       INTEGER REINF(100),INC(100)
000002       INTEGER PRNT(10)
000002       DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
000002       DIMENSION LSTORN(5,1000),LSTORC(5,1000)
000002       LOGICAL LEARN,NULL,IFLAG,TMOD(10)
000002       LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
000002       COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
000002       COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
000002       COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
000002       COMMON/PARAM/ NKXSUM,LCCSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
000002       COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
000002       COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
000002       COMMON/PARAM/LSTORN,LSTORC
000002       COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER
000002
000002       COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
000002       COMMON/PARAM/ TAVG,TDCAY1,SDCAY,INDEX
000002       COMMON/PARAM/FIRST,LAST,NFIRE
000002       COMMON/PARAM/IMUP,IMUM,H,ICELL,VNET
000002       COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
000002       COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
000002       COMMON/PARAM/ REINF,INC
000002       COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
000002       COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
000002       COMMON/PARAM/ FIRE,JOIN
       C
000002       ER=0.0001
000004       DO 500 J=1,NNETS
000005       LL1=NNEUR1(J)
000007       LL2=NNEUR2(J)
000010       DO 400 I=LL1,LL2
000012       DST=TRESH2(I)-32000
000015       IF(DST.LT.ARP(J)) GO TO 100
000020       TDROP(J)=TRESH1(I)+THR1(J)
000023       TRESH2(I)=TDROP(J)
000025       STATE(I)=0
000026       GO TO 400
000026   100 IF(DST.LT.0) GO TO 200
000030       TRESH2(I)=TRESH2(I)+1
000032       GO TO 400
000033   200 STATE(I)=STATE(I)*SDCAY(J)
000040       DTHR=TRESH2(I)-TRESH1(I)
000042       IF(DTHR.LE.1) GO TO 300
000045       TRESH3(I)=TDROP(J)-TRESH1(I)
```

```
000050          IF(TRESH3(I).LT.ER) GO TO 250
000052          TRESH3(I)=TRESH3(I)*TDCAY1(J)
000055      250 TRESH2(I)=TRESH1(I)+IFIX(TRESH3(I)+0.5)
000061          GO TO 400
000062      300 TRESH2(I)=TRESH1(I)
000064      400 CONTINUE
000067      500 CONTINUE
000071          RETURN
000072          END
```

```
                SUBROUTINE THINK(K)
        C
        C*****DECLARATIONS
        C
000003          REAL  INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
000003          REAL  TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
000003          REAL NSTD(10),TRESH3(1000),H
000003          INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
000003          INTEGER VNET(1000),FIRST,ITEM(20)
000003          INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
000003          INTEGER RECORD(1,1000)
000003          INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
000003          INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
000003          INTEGER REINF(100),INC(100)
000003          INTEGER PRNT(10)
000003          DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
000003          DIMENSION LSTORN(5,1000),LSTORC(5,1000)
000003          LOGICAL LEARN,NULL,IFLAG,TMOD(10)
000003          LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
000003          COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
000003          COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
000003          COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
000003          COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
000003          COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
000003          COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
000003          COMMON/PARAM/LSTOR I,LSTORC
000003          COMMON/PARAM/NNCTS,NBLOCK,NTOTAL,NUMBER
000003
000003          COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
000003          COMMON/PARAM/ TAVG,TDCAY1,SDCAY,INDEX
000003          COMMON/PARAM/FIRST,LAST,NFIRE
000003          COMMON/PARAM/IMUP,IMUM,H,ICELL,VNET
000003          COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
000003          COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
000003          COMMON/PARAM/ REINF,INC
000003          COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
000003          COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
000003          COMMON/PARAM/ FIRE,JOIN
        C
000003          LL=0
000004          LL=LL+1
000006          DO 400 J=JFROM,JJFROM
000010          DO 350 L=1,NKXMAX
000011          NITO=LSTORN(L,J)
000014          NJFROM=J
000015          NCOE=LSTORC(L,J)
000017          IF(TRESH2(NITO).GE.32000) GO TO 140
000022          IF(FIRE(NJFROM)) GO TO 15
000024          GO TO 150
000025      140 IF(TRESH2(NITO).EQ.32000) GO TO 150
000030          IF(.NOT.FIRE(NJFROM)) GO TO 150
000032       15 STATE(NITO)=STATE(NITO)+NCOE
000035          VNET(K)=VNET(K)+STATE(NITO)
000037          IF(NITO.EQ.ICELL) GO TO 200
000040          GO TO 150
000041      200 VT(K)=STATE(NITO)
000044      150 LL=LL+1
```

```
      IF(LL.GT.NUMBER) GO TO 500
  350 CONTINUE
  400 CONTINUE
  500 RETURN
      END
```

```
              SUBROUTINE COUNT(K)
        C
        C*****DECLARATIONS
        C
000003        REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
000003        REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
000003        REAL NSTD(10),TRESH3(1000),H
000003        INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
000003        INTEGER VNET(1000),FIRST,ITEM(20)
000003        INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
000003        INTEGER RECORD(1,1000)
000003        INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
000003        INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
000003        INTEGER REINF(100),INC(100)
000003        INTEGER PRNT(10)
000003        DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
000003        DIMENSION LSTORN(5,1000),LSTORC(5,1000)
000003        LOGICAL LEARN,NULL,IFLAG,TMOD(10)
000003        LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
000003        COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
000003        COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
000003        COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
000003        COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
000003        COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
000003        COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
000003        COMMON/PARAM/LSTORN,LSTORC
000003        COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER
000003
000003        COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
000003        COMMON/PARAM/ TAVG,TDCAY1,SDCAY,INDEX
000003        COMMON/PARAM/FIRST,LAST,NFIRE
000003        COMMON/PARAM/IMUP,IMUM,H,ICELL,VNET
000003        COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
000003        COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
000003        COMMON/PARAM/ REINF,INC
000003        COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
000003        COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
000003        COMMON/PARAM/ FIRE,JOIN
        C
000003        RCRD=0
000004        DO 300 J=1,NNETS
000005        NFIRE=0
000006        LL1=NNEUR1(J)
000010        LL2=NNEUR2(J)
000011        DO 200 I=LL1,LL2
000013        FIRE(I)=.FALSE.
000014        IF(TRESH2(I).GE.32001) GO TO 200
000017        IF(STATE(I).GE.TRESH2(I)) GO TO 100
000021        GO TO 200
000022    100 NFIRE=NFIRE+1
000024        STATE(I)=0
000025        TRESH2(I)=32000
000027        FIRE(I)=.TRUE.
000030        IF(JOIN(I)) GO TO 200
000031        JOIN(I)=.TRUE.
000032    200 CONTINUE
000035        RECORD(J,K)=NFIRE
```

```
000037          RCRD=RCRD+RECORD(J,K)          130
000041    300 CONTINUE
000044          RETURN
000044          END
```

```
      SUBROUTINE SET1(I1)
C
C*****DECLARATIONS
C
      REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
      REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),SCAM,V,NOISE(100)
      REAL NSTD(10),TRESH3(1000),H
      INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
      INTEGER VNET(1000),FIRST,ITEM(20)
      INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
      INTEGER RECORD(1,1000)
      INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
      INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
      INTEGER REINF(100),INC(100)
      INTEGER PRNT(10)
      DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
      DIMENSION LSTCRN(5,1000),LSTORC(5,1000)
      LOGICAL LEARN,NULL,IFLAG,TMOD(10)
      LOGICAL FIRE(1000),JOIN(1000),INHIP(4096)
      COMMON/PARAM/ IWORK,TITLE,INFO1,INFO2
      COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
      COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
      COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
      COMMON/PARAM/ JFROM,JJFFOM,ITO,IITO,COE,INH,IFRCTN,ITEM
      COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
      COMMON/PARAM/LSTORN,LSTORC
      COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER

      COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
      COMMON/PARAM/ TAVG,TDCAY1,SDCAY,INDEX
      COMMON/PARAM/FIRST,LAST,NFIRE
      COMMON/PARAM/IMUP,IMUM,H,ICELL,VNET
      COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
      COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
      COMMON/PARAM/ REINF,INC
      COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
      COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
      COMMON/PARAM/ FIRE,JOIN
C
      I1=0
      WRITE(6,50)
   50 FORMAT(5X,46HALFA-0 STATE: THE FOLLOWING NEURONS ARE FIRING//)
      DO 100 I=FIRST,LAST
      STATE(I)=TRESH2(I)
  100 CONTINUE
      WRITE(6,200) FIRST,LAST
  200 FORMAT(2I4)
      I1=1
      RETURN
      END
```

```
      SUBROUTINE SET3(L,K)
C
C*****DECLARATIONS
C
000005      REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
000005      REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
000005      REAL NSTD(10),TRESH3(1000),H
000005      INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
000005      INTEGER VNET(1000),FIRST,ITEM(20)
000005      INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
000005      INTEGER RECORD(1,1000)
000005      INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
000005      INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
000005      INTEGER PEINF(100),INC(100)
000005      INTEGER PRNT(10)
000005      DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
000005      DIMENSION LSTORN(5,1000),LSTORC(5,1000)
000005      LOGICAL LEARN,NULL,IFLAG,TMOD(10)
000005      LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
000005      COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
000005      COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
000005      COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
000005      COMMON/PARAM/ NKXSUM,LCCSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
000005      COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
000005      COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
000005      COMMON/PARAM/LSTORN,LSTORC
000005      COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER
000005
000005      COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
000005      COMMON/PARAM/ TAVG,TDCAY1,SDCAY,INDEX
000005      COMMON/PARAM/FIRST,LAST,NFIRE
000005      COMMON/PARAM/IMUP,IMUM,H,ICELL,VNET
000005      COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
000005      COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
000005      COMMON/PARAM/ PEINF,INC
000005      COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
000005      COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
000005      COMMON/PARAM/ FIRE,JOIN
000005      IF(K.LT.100) RETURN
000010      IF(K.EQ.101) GO TO 10
000012      GO TO 300
000012   10 YFL=RANF(X)
000015      X=0
000016      LL3=IEXT(L)
000021      HFRCN=INHIB(L)/100
000024      NN=IFIX(NEXT(L)*HFRCN+0.5)
000030      DIFF=NNEUR2(L)-NNEUR1(L)
000032      LL1=NN
000034      LL2=NEXT(L)-NN
000035      IF(NN.EQ.0) GO TO 150
000036      DO 100 I=1,LL1
000037      YFL=RANF(X)
000041      J=NNEUR1(L)+IFIX(DIFF*YFL+0.5)
000047      IF(STATE(J).GE.32000) GO TO 100
000052      STATE(J)=STATE(J)-EXTCOE(L)
000054  100 CONTINUE
000057         IF(LL2.EQ.0) GO TO 200
```

```
000060     150 DO 200 I=1,LL2
000062         YFL=RANF(X)
000064         J=NNEUR1(L)+IFIX(DIFF*YFL+0.5)
000072         IF(STATE(J).GE.32000) GO TO 200
000075         STATE(J)=STATE(J)+EXTCOE(L)
000077     200 CONTINUE
000102         IF(LL3.EQ.0) GO TO 300
000103         DO 250 I=1,LL3
000104         YFL=RANF(X)
000106         J=NNEUR1(L)+IFIX(DIFF*YFL+0.5)
000114         IF(STATE(J).GE.32000) GO TO 250
000117         STATE(J)=STATE(J)-EXTCOE(L)
000121     250 CONTINUE
000124     300 RETURN
000125         END
```

```
      SUBROUTINE LIST1
C
C*****DECLARATIONS
C
000002      REAL TNF,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
000002      REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
000002      REAL NSTD(10),TRESH3(1000),H
000002      INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
000002      INTEGER VNET(1000),FIRST,ITEM(20)
000002      INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
000002      INTEGER RECORD(1,1000)
000002      INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
000002      INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
000002      INTEGER REINF(100),INC(100)
000002      INTEGER PRNT(10)
000002      DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
000002      DIMENSION LSTORN(5,1000),LSTORC(5,1000)
000002      LOGICAL LEARN,NULL,IFLAG,TMOD(10)
000002      LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
000002      COMMON/PARAM/ IOWORK,TITLE,INFO1,INFO2
000002      COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
000002      COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
000002      COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
000002      COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
000002      COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
000002      COMMON/PARAM/LSTORN,LSTORC
000002      COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER
000002
000002      COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
000002      COMMON/PARAM/ TAVG,TDCAY1,SDCAY,INDEX
000002      COMMON/PARAM/FIRST,LAST,NFIRE
000002      COMMON/PARAM/IMUP,IMUM,H,ICELL,VNET
000002      COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
000002      COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD,
000002      COMMON/PARAM/ REINF,INC
000002      COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
000002      COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
000002      COMMON/PARAM/ FIRE,JOIN
C
000002   15 FORMAT(3X,8H RUN NO.I4,15A4///3X,*NETWORK PARAMETERS:*///12X,*SUB
           1NET 1  SUBNET 2  SUBNET 3  SUBNET 4  SUBNET 5  SUBNET 6  SUBNET 7
           2 SUBNET 8  SUBNET 9  SUBNET 10*/)
000002   35 FORMAT(3X,*TMOD  =*,10I10)
000002   45 FORMAT(3X,*TMIN  =*,10I10)
000002   55 FORMAT(3X,*TMAX  =*,10I10)
000002   65 FORMAT(3X,*THR1  =*,10I10)
000002   75 FORMAT(3X,*ARP   =*,10I10)
000002   85 FORMAT(3X,*TDCAY1=*,10F10.4)
000002   90 FORMAT(3X,*SDCAY=*,10F10.4)
000002   95 FORMAT(3X,*TAVG  =*,10F10.4)
000002  105 FORMAT(3X,*TRESH1 ARRAY*//5(5X,5I3))
C
000002      WRITE(6,15) IDENT,TITLE
000012      WRITE(6,35) (TMOD(I),I=1,NNETS)
000025      WRITE(6,45) (TMIN(I),I=1,NNETS)
000040      WRITE(6,55) (TMAX(I),I=1,NNETS)
000053      WRITE(6,65) (THR1(I),I=1,NNETS)
```

```
000065         WRITE(5,75)  (ARP(I),I=1,NNETS)
000100         WRITE(6,85)  (TDCAY1(I),I=1,NNETS)
000112         WRITE(6,90)  (SDCAY(I),I=1,NNETS)
000125         WRITE(6,95)  (TAVG(I),I=1,NNETS)
000140         WRITE(6,105) (TRESH1(I),I=1,NTOTAL)
000153         RETURN
000153         END
```

```
              SUBROUTINE LIST2
        C
        C*****DECLARATIONS
        C
000002        REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
000002        REAL TAVG(10),TDCAY1(10),SDCAY(10),STD(100),S,AM,V,NOISE(100)
000002        REAL NSTD(10),TRESH3(1000),H
000002        INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
000002        INTEGER VNET(1000),FIRST,ITEM(20)
000002        INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
000002        INTEGER RECORD(1,1000)
000002        INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
000002        INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
000002        INTEGER REINF(100),INC(100)
000002        INTEGER PRNT(10)
000002        DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
000002        DIMENSION LSTORN(5,1000),LSTORC(5,1000)
000002        LOGICAL LEARN,NULL,IFLAG,TMOD(10)
000002        LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
000002        COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
000002        COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
000002        COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
000002        COMMON/PARAM/ NKXSUM,LCOSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
000002        COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
000002        COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
000002        COMMON/PARAM/LSTORN,LSTORC
000002        COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER
000002
000002        COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
000002        COMMON/PARAM/ TAVG,TDCAY1,SDCAY,TNDEX
000002        COMMON/PARAM/FIRST,LAST,NFIRE
000002        COMMON/PARAM/IMUP,INUM,H,ICELL,VNET
000002        COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
000002        COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
000002        COMMON/PARAM/ REINF,INC
000002        COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
000002        COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
000002        COMMON/PARAM/ FIRE,JOIN
        C
000002     15  FORMAT(3X,13H STIMULUS NO.I2,15A4//5X,12HPARAMETERS: 5HLEARN3X,O2
              10/17X,4HNULL4X,O20/17X,5HIFLAG3X,O20/17X,5HNITER3X,I4///)
000002     25 FORMAT(5X,34HREINFORCE IN THE FOLLOWING BLOCKS://13X,*SUBNET 1  SU
              1BNET 2   SUBNET 3   SUBNET 4   SUBNET 5   SUBNET 6   SUBNET 7   SUBNET 8
              2  SUBNET 9 SUBNET 10*//)
000002    110 FORMAT(1X,*SUBNET*,I3,5X,F9.7)
000002    115 FORMAT(1X,*SUBNET*,I3,5X,I5)
000002    118 FORMAT(1X,*SUBNET*,I3,5X,F6.4)
000002    225 FORMAT(5X,40HREINFORCEMENT PARAMETERS ARE AS FOLLOWS://13X,*SUBNET
              11  SUBNET 2   SUBNET 3   SUBNET 4   SUBNET 5   SUBNET 6   SUBNET 7   SUB
              2NET 8   SUBNET 9 SUBNET 10*//)
000002    505 FORMAT(5X,64HPARAMETERS OF SPONTANEOUS AND EXTERNAL ACTIVITY INCID
              1ENT ON NET://13X,*SUBNET 1   SUBNET 2   SUBNET 3   SUBNET 4   SUBNET 5
              2   SUBNET 6   SUBNET 7   SUBNET 8   SUBNET 9 SUBNET 10*//)
000002    521 FORMAT(3X,*NSPONT  *,10I10)
000002    522 FORMAT(3X,*NEXT    *,10I10)
000002    523 FORMAT(3X,*INHIB   *,10I10)
000002    524 FORMAT(3X,*EXTCOE  *,10I10)
```

```
000002     525 FORMAT(3X,*IEXT    *,10I10)
000002     605 FORMAT(5X,44HMEAN AND STANDARD DEVIATION OF NOISE IN NET*//13X,*SU
              1BNET 1   SUBNET 2   SUBNET 3   SUBNET 4   SUBNET 5   SUBNET 6   SUBNET 7
              2  SUBNET 8   SUBNET 9 SUBNET 10*//)
      C
000002         DO 10 I=1,10
000004         NSTD(I)=0.0
000005         PRNT(I)=0
000006      10 CONTINUE
000010         WRITE(6,15) NSTIM,INFO,LEARN,NULL,IFLAG,NITER
000027         IF(.NOT.LEARN) GO TO 500
000031         WRITE(6,25)
000034         DO 200 I=1,NNETS
000036         DO 100 J=1,NNETS
000037         PRNT(J)=REINF(I+NNETS*(J-1))
000044     100 CONTINUE
000046         WRITE(6,115) I,(PRNT(K),K=1,NNETS)
000062     200 CONTINUE
000065         WRITE(6,225)
000070         DO 400 I=1,NNETS
000072         DO 300 J=1,NNETS
000073         PRNT(J)=INC(I+NNETS*(J-1))
000100     300 CONTINUE
000102         WRITE(6,115) I,(PRNT(K),K=1,NNETS)
000116     400 CONTINUE
000121     500 IF(.NOT.IFLAG) GO TO 2000
000123         WRITE(6,505)
000126         INDEX=0
000127     510 INDEX=INDEX+1
000131         IF(INDEX.GT.5) GO TO 600
000134         GO TO (511,512,513,514,515),INDEX
000144     511 WRITE(6,521) (NSPONT(I),I=1,NNETS)
000157         GO TO 550
000160     512 WRITE(6,522) (NEXT(I),I=1,NNETS)
000173         GO TO 550
000174     513 WRITE(6,523) (INHIB(I),I=1,NNETS)
000207         GO TO 550
000210     514 WRITE(6,524) (EXTCOE(I),I=1,NNETS)
000223         GO TO 550
000224     515 WRITE(6,525) (IEXT(I),I=1,NNETS)
000237     550 GO TO 510
000240     600 WRITE(6,605)
000244         DO 800 I=1,NNETS
000246         DO 700 J=1,NNETS
000247         NSTD(J)=NOISE(I+NNETS*(J-1))
000254     700 CONTINUE
000256         WRITE(6,110) I,(NSTD(K),K=1,NNETS)
000272     800 CONTINUE
000275         DO 1000 I=1,NNETS
000276         DO 900 J=1,NNETS
000277         NSTD(J)=STD(I+NNETS*(J-1))
000304     900 CONTINUE
000306         WRITE(6,118) I,(NSTD(K),K=1,NNETS)
000322    1000 CONTINUE
000325    2000 RETURN
000326         END
```

```
                SUBROUTINE FIT(D,EAREA,CHISQR)
000006          DIMENSION XMARK(100),DX(100),XMSQ(100),DXSQ(100),
               1Z(101),AREA(101)
000006          DIMENSION BCOUNT(100)
000006          DIMENSION D(20)
000006          REAL EAREA(101)
000006          DO 5 I=1,100
000007          XMARK(I)=0.0
000010          DX(I)=0.0
000011          XMSQ(I)=0.0
000012          DXSQ(I)=0.0
000013     5    CONTINUE
000015          IZERO=0
000015          SUMDX=0.0
000016          SUMD=0.0
000017          SUMDXS=0.0
000021          DO 10 I=1,10
000022          IF(D(I).EQ.0.0) GO TO 10
000024          XMARK(I)=(I/2.)-0.25
000027          DX(I)=D(I)*XMARK(I)
000032          XMSQ(I)=XMARK(I)**2
000034          DXSQ(I)=XMSQ(I)*D(I)
000036          SUMD=SUMD+D(I)
000040          SUMDX=SUMDX+DX(I)
000042          SUMDXS=SUMDXS+DXSQ(I)
000045    10    CONTINUE
000047          XAVE=SUMDX/SUMD
000051          STD=SQRT((SUMDXS/SUMD)-XAVE**2)
000056          IF(STD.EQ.0.0) RETURN
000062          WRITE(6,20) XAVE,STD
000072    20    FORMAT(* *,* XAVE=*,F8.5,* STD=*,F8.5)
000072          SUM2=0.0
000073          SUM3=0.0
000073          SUM4=0.0
000075          DO 200 J=1,10
000100          TM2=((XMARK(J)-XAVE)**2)*D(J)
000103          SUM2=SUM2+TM2
000105          TTM3=(XMARK(J)-XAVE)**2
000110          TM3=D(J)*TTM3*(XMARK(J)-XAVE)
000113          SUM3=SUM3+TM3
000115          TTM4=(XMARK(J)-XAVE)**2
000120          TM4=((XMARK(J)-XAVE)**2)*D(J)*TTM4
000124          SUM4=SUM4+TM4
000126   200    CONTINUE
000130          GM2=SUM2/SUMD
000132          GM3=SUM3/SUMD
000133          GM4=SUM4/SUMD
000135          A3=GM3/SQRT((GM2**2)*GM2)
000141          A4=GM4/GM2**2
000144          WRITE(6,300) A3,A4
000154   300    FORMAT(* *,* A3=*,F7.3,* A4=*,F7.3)
000154          WRITE(6,401) GM2,GM3,GM4
000166   401    FORMAT(* *,* GM2=*,F9.3,* GM3=*,F9.3,* GM4=*,F9.3)
000166          IF(STD.EQ.0.0) RETURN
000172          DO 30 I=1,10
000174          IF(D(I).EQ.0.0) GO TO 35
000176          GO TO 40
```

```
[illegible]  30   CONTINUE
000202    40   BOUNT(1)=IZERO/2.0
000205         DO 45 J=1,11
000206         BOUNT(J)=(IZERO/2.0)+(J/2.0)-0.5
000214    45   CONTINUE
000216         DO 55 J=1,101
000220         AREA(J)=0.0
000221         EAREA(J)=0.0
000222         Z(J)=0.0
000223    55   CONTINUE
000225         DO 50 J=1,11
000227         Z(J)=(BOUNT(J)-YAVE)/STD
000232         TA=ABS(Z(J))
000234         AREA(J)=ERF(TA)
000240    50   CONTINUE
000244         DO 70 JJ=1,11
000245         IF(Z(JJ).GE.0.0) GO TO 80
000247         K=JJ+1
000250         IF(Z(K).GE.0.0) GO TO 85
000253    90   L=JJ+1
000255         EAREA(JJ)=ABS(AREA(JJ)-AREA(L))
000262         GO TO 70
000262    85   LL=JJ+1
000264         EAREA(JJ)=AREA(JJ)+AREA(LL)
000270         GO TO 70
000270    80   KK=JJ+1
000272         IF(Z(KK).GE.0.0) GO TO 90
000274         GO TO 85
000275    70   CONTINUE
000277         CHISQR=0.0
000300         DO 400 I=1,10
000301         IF(EAREA(I).EQ.0.0) GO TO 400
000303         SUM=(O(I)-EAREA(I)*SUMO)**2/(EAREA(I)*SUMO)
000311         CHISQR=CHISQR+SUM
000313   400   CONTINUE
000315         RETURN
000316         END
```

```
              SUBROUTINE HISTG(VOUT,NU,D,ICUM)
000007        REAL D(20)
000007        INTEGER VOUT(201)
000007        INTEGER VMAX,VMIN
000007        DIMENSION FREQ(20)
000007        VMIN=VOUT(1)
000007        VMAX=VOUT(1)
000011        DO 5 I=2,200
000012        VMAX=AMAX0(VMAX,VOUT(I))
000017     5  CONTINUE
000021        DO 6 I=2,200
000022        VMIN=AMIN0(VMIN,VOUT(I))
000027     6  CONTINUE
000031        DO 15 J=1,20
000032        D(J)=0.0
000034        FREQ(J)=0.0
000034    15  CONTINUE
000036        DO 10 I=1,200
000040        D1=10*(VOUT(I)-VMIN)
000044        IN=1*(D1/(VMAX-VMIN))+1
000053        IF(IN.GT.10) GO TO 10
000057        D(IN)=D(IN)+1.
000061        FREQ(IN)=D(IN)
000063    10  CONTINUE
000065        CALL HIST(NU,FREQ,10,ICUM)
000070        RETURN
000071        END
```

```
000007         DIMENSION JOUT(20),FREQ(20)
000007    1    FORMAT(6H EACH ,A1,8H EQUALS ,I2,7H POINTS,/)
000007    2    FORMAT(I6,4X,20(2X,A1))
000007    3    FORMAT(9HOINTERVAL,2X,19(I2,1X),I2)
000007    4    FORMAT(1H1,47X,11H HISTOGRAM ,I3)
000007    5    FORMAT(10HOFREQUENCY,20I5)
000007    6    FORMAT(6H CLASS)
000007    7    FORMAT(113(*-*))
000007    8    FORMAT(1H )
000007    9    FORMAT(A1)
000007   10    FORMAT(1H*)
000007      11 FORMAT(* *,47X,11H HISTOGRAM ,I3)
000007         REWIND 1
000011         WRITE(1,10)
000015         REWIND 1
000017         READ(1,9) K
000025         REWIND 1
000027         WRITE(1,8)
000033         REWIND 1
000035         READ(1,9) NOTH
000043         REWIND 1
000045         IF(ICUM.EQ.1) GO TO 13
000052         WRITE(6,4) NU
000057         GO TO 14
000062      13 WRITE(6,11) NU
000070      14 DO 12 I=1,IN
000074   12    JOUT(I)=FREQ(I)
000101         WRITE(6,5)(JOUT(I),I=1,IN)
000117         WRITE(6,7)
000123         FMAX=0.0
000124         DO 20 I=1,IN
000130         IF(FREQ(I)-FMAX) 20,20,15
000133   15    FMAX=FREQ(I)
000135   20    CONTINUE
000140         JSCAL=1
000140         IF(FMAX-50.0) 40,40,30
000143   30    JSCAL=(FMAX+49.0)/50.0
000147         WRITE(6,1)K,JSCAL
000157   40    DO 50 I=1,IN
000163   50    JOUT(I)=NOTH
000167         MAX=FMAX/FLOAT(JSCAL)
000172         DO 80 I=1,MAX
000174         X=MAX-(I-1)
000177         DO 70 J=1,IN
000201         IF(FREQ(J)/FLOAT(JSCAL)-X) 70,60,60
000206   60    JOUT(J)=K
000210   70    CONTINUE
000213         IX=X*FLOAT(JSCAL)
000216   80    WRITE(6,2)IX,(JOUT(J),J=1,IN)
000244         DO 90 I=1,IN
000246   90    JOUT(I)=I
000251         WRITE(6,7)
000254         WRITE(6,3)(JOUT(J),J=1,IN)
000275         WRITE(6,6)
000301         RETURN
000302         END
```

```
      SUBROUTINE LIST3(I2)
C
C*****DECLARATIONS
C
      REAL INH,XNKAVG,INKAVG,XCOAVG,ICOAVG,YFL,IFRCTN
      REAL TAVG(10),TODAY1(10),SDDAY(10),STD(100),S,AM,V,NOISE(100)
      REAL NSTD(10),TRESH3(1000),H
      REAL D(20),AMNFIR,TERM,SUM,SDNF
      REAL CAREA(101)
      INTEGER VNOT(201)
      INTEGER VOUT(201)
      INTEGER TITLE(15),INFO(15),INFO1(15),INFO2(15)
      INTEGER VNET(1000),FIRST,ITEM(20)
      INTEGER STATE(1000),TRESH1(1000),TRESH2(1000)
      INTEGER RECORD(1,1000),NNFIR(201)
      INTEGER TMIN(10),TMAX(10),TDROP(10),ARP(10),THR1(10),VT(1000)
      INTEGER NSPONT(10),NEXT(10),EXTCOE(10),RCRD,IEXT(10)
      INTEGER REINF(100),INC(100)
      INTEGER PRNT(10)
      DIMENSION NKFRST(100),NKLAST(100),NNEUR1(10),NNEUR2(10)
      DIMENSION LSTORM(5,1000),LSTORC(5,1000)
      LOGICAL LEARN,NULL,IFLAG,TMOD(10)
      LOGICAL FIRE(1000),JOIN(1000),INHIB(4096)
      COMMON/PARAM/ IDKORK,TITLE,INFO1,INFO2
      COMMON/PARAM/ NKX,NKXMIN,NKXMAX,XNKAVG,LCOMIN,LCOMAX,XCOAVG
      COMMON/PARAM/ INK,INKMIN,INKMAX,INKAVG,ICOMIN,ICOMAX,ICOAVG
      COMMON/PARAM/ NKXSUM,LCCSUM,NXNEUR,INKSUM,ICOSUM,INNEUR
      COMMON/PARAM/ JFROM,JJFROM,ITO,IITO,COE,INH,IFRCTN,ITEM
      COMMON/PARAM/NKFRST,NKLAST,NNEUR2,NNEUR1
      COMMON/PARAM/LSTORM,LSTORC
      COMMON/PARAM/NNETS,NBLOCK,NTOTAL,NUMBER

      COMMON/PARAM/IDENT,NSTIM,INFO,IDITER,NITER
      COMMON/PARAM/ TAVG,TODAY1,SDDAY,INDEX
      COMMON/PARAM/FIRST,LAST,NFIRE
      COMMON/PARAM/IMUP,IMUM,H,ICELL,VNET
      COMMON/PARAM/ STATE,TRESH1,TRESH2,TRESH3,VT
      COMMON/PARAM/ TMIN,TMAX,TDROP,ARP,THR1,RCRD
      COMMON/PARAM/ REINF,INC
      COMMON/PARAM/ NSPONT,NEXT,INHIB,EXTCOE,IEXT,RECORD
      COMMON/PARAM/ LEARN,NULL,IFLAG,TMOD
      COMMON/PARAM/ FIRE,JOIN
C
C
      I2=0
   15 FORMAT(*1JOIN ARRAY.IDJOIN=*,I9//(10(1X,10I1)))
   25 FORMAT(3X,25HRECORD ARRAYS FOR SUBNETS/)
  115 FORMAT(*1 RECORD ARRAY.IDREC=*,I9//(20(1X,I5)))
      ICUM=0
      DO 5 II=1,200
      VOUT(II)=0
      VNOT(II)=0
      NNFIR(II)=0
    5 CONTINUE
      NU=0
      KC=1
      KM=1
```

```
000024        WRITE(6,15) IDJOIN,(JOHN(I),I=1,NTOTAL)
000040        WRITE(6,25)
000044        J=0
000045     30 J=J+1
000047        IF(J.GT.2) GO TO 204
000053        WRITE(6,400)(VT(M),VNET(M),M=1,NITER)
000067    400 FORMAT(* *,10I6,4X,10I6)
000067        IDREC=IDJOIN+J
000071        GO TO (101,102),J
000077    101 WRITE(6,115) IDREC,(RECORD(1,I),I=1,NITER)
000114    102 DO 50 M=1,NITER
000117        VOUT(KC)=VT(M)
000121        KC=KC+1
000123        IF(KC.NE.201) GO TO 50
000125        WRITE(6,200)(VOUT(K),K=1,200)
000136    200 FORMAT(* *,10I6)
000136        NU=NU+1
000140        CALL HISTG(VOUT,NU,D,ICUM)
000143        CALL FIT(D,EAREA,CHISQR)
000146        WRITE(6,147)(EAREA(JJ),JJ=1,10)
000160    147 FORMAT(* *,* EAREA=* 10F8.5)
000160        WRITE(6,157) CHISQR
000166    157 FORMAT(* *,* CHISQR=*,F10.4)
000166        KC=1
000167        DO 1 IJ=1,201
000172        VOUT(IJ)=0
000173      1 CONTINUE
000175     50 CONTINUE
000200        NU=0
000201        ICUM=1
000202        DO 40 MM=1,NITER
000203        VNOT(KN)=VNET(MM)
000205        KN=KN+1
000207        IF(KN.NE.201) GO TO 37
000211        NU=NU+1
000212        WRITE(6,60)
000215     60 FORMAT(1H1,47X,*CUMULATIVE HISTOGRAM*)
000215        CALL HISTG(VNOT,NU,D,ICUM)
000220        CALL FIT(D,EAREA,CHISQR)
000223        WRITE(6,117)(EAREA(JJ),JJ=1,10)
000235    117 FORMAT(* *,* EAREA=*,10F8.5)
000235        WRITE(6,116) CHISQR
000243    116 FORMAT(* *,* CHISQR=*,F20.4)
000243        DO 2 IK=1,201
000246        VNOT(IK)=0
000247      2 CONTINUE
000251        KN=1
000252        GO TO 38
000252     37 MN=KN-1
000254        NNFIR(MN)=RECORD(1,MM)
000257        GO TO 40
000257     38 ISUM=0
000260        SUM=0.0
000261        DO 39 L=1,200
000262        ISUM=ISUM+NNFIR(L)
000264     39 CONTINUE
000265        ANNFIR=ISUM/200.0
```

```
      DO 41 LL=1,200
      TERM=(NNFIR(LL)-ANNFIR)**2
      SUM=SUM+TERM
   41 CONTINUE
      SDNF=SQRT(SUM/200.0)
      WRITE(6,4)ANNFIR,SDNF
    4 FORMAT(* *,* ANNFIR=*,F10.5,* SDNF=*,F10.5)
   40 CONTINUE
      GO TO 30
  204 I2=1
      RETURN
      END
```

```
                 FUNCTION ERF(X)
000003           DIMENSION PHI(100),CAPPHI(100),XI(100)
000003           N=10
000004           DXI=X/N
000006           NP1=N+1
000010           DO 1 JP1=1,NP1
000012           FJ =JP1-1
000014           XI(JP1)=FJ*DXI
000016           R=1.0/SQRT(2.0*3.14159)
000021         1 PHI(JP1)=R*EXP(-XI(JP1)*XI(JP1)/2.0)
000035           CAPPHI(1)=0
000036           DO 2 JP1=3,NP1,2
000037     2     CAPPHI(JP1)=CAPPHI(JP1-2)+(PHI(JP1-2)+4.0*PHI(JP1-1)+
                1PHI(JP1))*DXI/3.0
000051           ERF=CAPPHI(NP1)
000052           WRITE(6,20)ERF
000060    20     FORMAT(F10.4)
000060           RETURN
000062           END
```