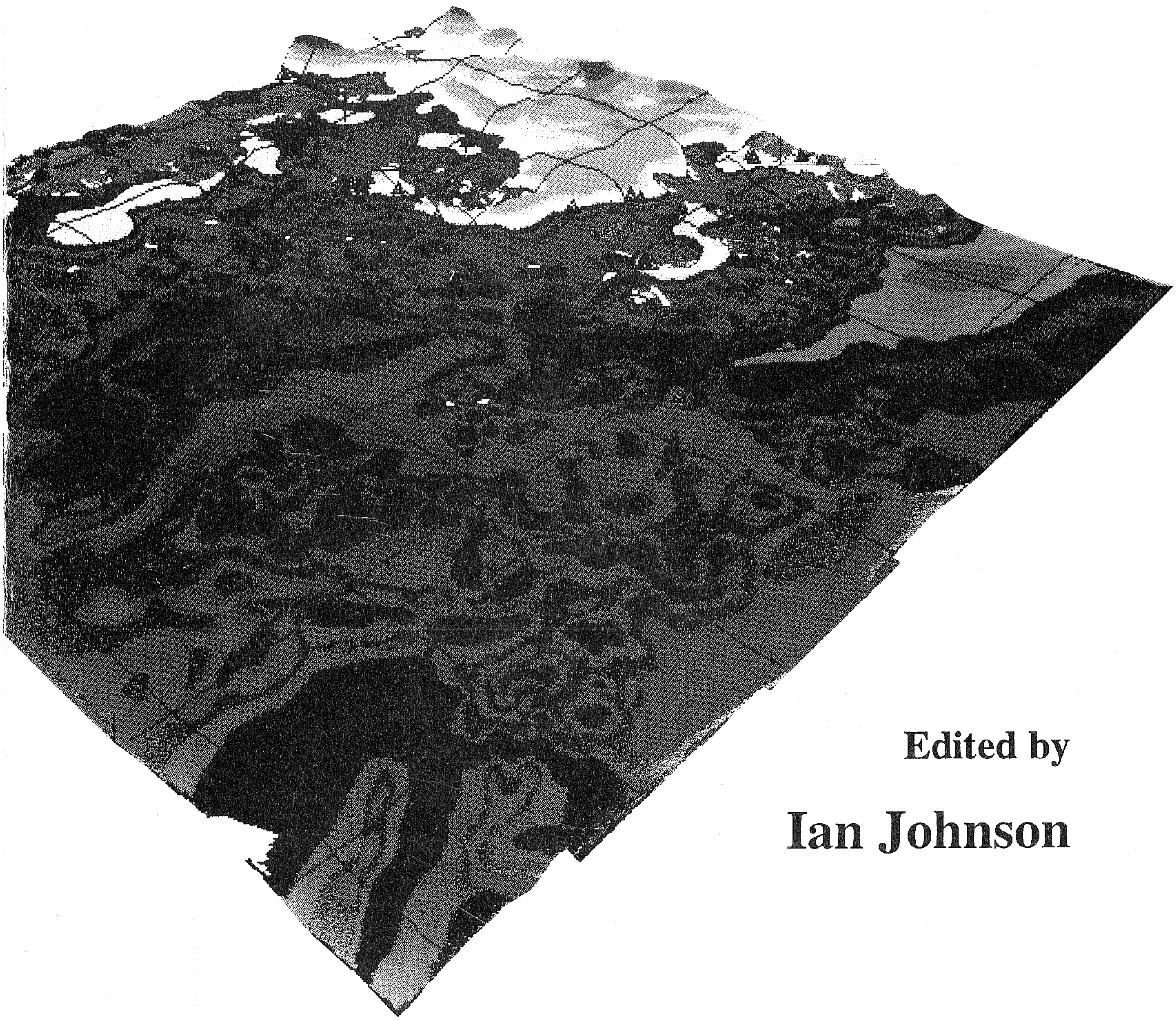
Methods in the Mountains

Proceedings of UISPP Commission IV Meeting
Mount Victoria, Australia. August 1993



Edited by

Ian Johnson

Sydney University Archaeological Methods Series # 2

1994

KIPA - A computer program to analyse the social position of women in hunter-gatherer societies

Juan A. Barcelo
Assumpcio Vila
Teresa Argeles

Archaeologists are currently using ethnological data to explain the archaeological record they found in excavations. However, there is no investigation of the nature of ethnological data and their relevance in the reconstruction of prehistoric societies.

One of the most typical mistakes has been to consider that Paleolithic societies were egalitarian, *because comparable ethnological data suggests the absence of social differentiation in hunter-gatherer societies.* In this paper we try to demonstrate the opposite, that is to say that ethnoarchaeological research suggests that there are no egalitarian societies: even hunter-gatherer communities show a degree of social differentiation. Our hypothesis is that social differentiation is the consequence of a dialectical process affecting production and reproduction (both biological and sociological). The way societies use to control reproduction is the social division of work, and its related ideological justification. It adopts the appearance of sexual division of work, reducing the ideological value of work done by women, to increase social control. This socioeconomic phenomenon is only a way to discriminate women to gain control over reproduction. We have explained this theoretical framework elsewhere (Vila & Argelés 1993).

We intend to build a mathematical tool to measure, in the empirical record, the degree of discrimination experienced by women in a hunter-gatherer community. As a case study we have selected Yamana ethnography, as published by Gusinde (1937), because it is one of the best instances of a supposed *egalitarian* hunter-fisher-gatherer community. The methodological tool we use in this paper is Neural Network Technology. We think that it provides a better environment for

Methods in the Mountains: Proc. UISPP Comm. IV Mtg., Mt. Victoria, Australia, Aug. 1993
I.Johnson (ed) 1994 Sydney University Archaeological Methods Series # 2

social simulation, because it is not limited by the form of input vectors (Attribute-Value pairs), and because it is able to simulate dialectic processes and non-linear causal relationships.

In this paper we show only a preliminary test to study the potential of this technology.

What is a neural network?

A neural network is an information processing system that is non-algorithmic, non-digital, and intensely parallel. It consists of a number of very simple and highly interconnected processors called *units*, which are the analogues of the biological neural cells, or neurons, in the brain. These units are connected by a large number of weighted links, over which signals can pass. Each unit typically receives many signals over its incoming connections: some of these incoming signals may arise from other units, and others may come from the outside world, through an input device. The unit usually has many of these incoming signal connections; however, it never produces more than a single outgoing signal. That output signal transmits over the unit's outgoing connection (corresponding to the biological taxon of a neuron), which usually splits into a very large number of smaller connections, each of which terminates at a different destination. Each of these branches of the single outgoing connection transmits the same signal: the signal is not split or divided among them in any way. Most of these outgoing branches terminate at the incoming connection of some other unit in the network: Others may terminate outside the network and generate control or response patterns (Fig. 1)

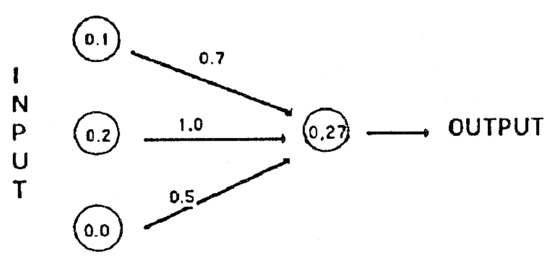


Fig. 1: The structure and functioning of a processing unit (computer neuron).

Each unit in a neural network is an extremely simple devicé. It receives input stimuli along its input connections and translates those stimuli into an output response, which is transmitted along the unit's output connection. The mathematical expression that describes the translation of input stimulus pattern to output response signal is called the *transfer function* of the unit, and it consists of a three-step process:

First step

The unit computes the net weighted input it is receiving along its input connections. A strong input signal arriving over a weakly weighted connection may have less effect than a weaker signal arriving over a strongly weighted connection. Even more, a weight may be negative instead of positive. In this case the connection is said to be *inhibitory*. It tends to reduce the overall stimulation of the receiving unit. Weights and inputs can be represented in several ways, but most commonly consists of computing the value I_i as shown here:

$$I_i = \Sigma w_{ij} x_j$$

In this expression, I_i is the net weighted input received by unit i from a total of n units in the network. The incoming signal from the j th unit is designated by x_j, and the weight on the connection directed from unit j to unit i is designated by w_{ij}.

Second step

The translation operation represented by the unit's transfer function converts the net input to an activation level. This activation level is equivalent to the level of excitement of a biological neuron. It is also represented by a mathematical expression or *activation function*. In this case we have selected two different activation functions:

- a classic linear function. With this function, the activation level of a unit (its *potential*, or *confidence* we have in it) is set to the input, where the input is the weighted sum of all signals input to the unit. There is a linear, one-to-one correspondence between the unit's input and its output;

- a standard boolean threshold function. With this function a unit takes on only its maximum or minimum activation values. If the weighted sum of the incoming signals is greater than the unit's threshold, the unit's new activation will be its maximum value; otherwise, the new activation will be the unit's minimum value.

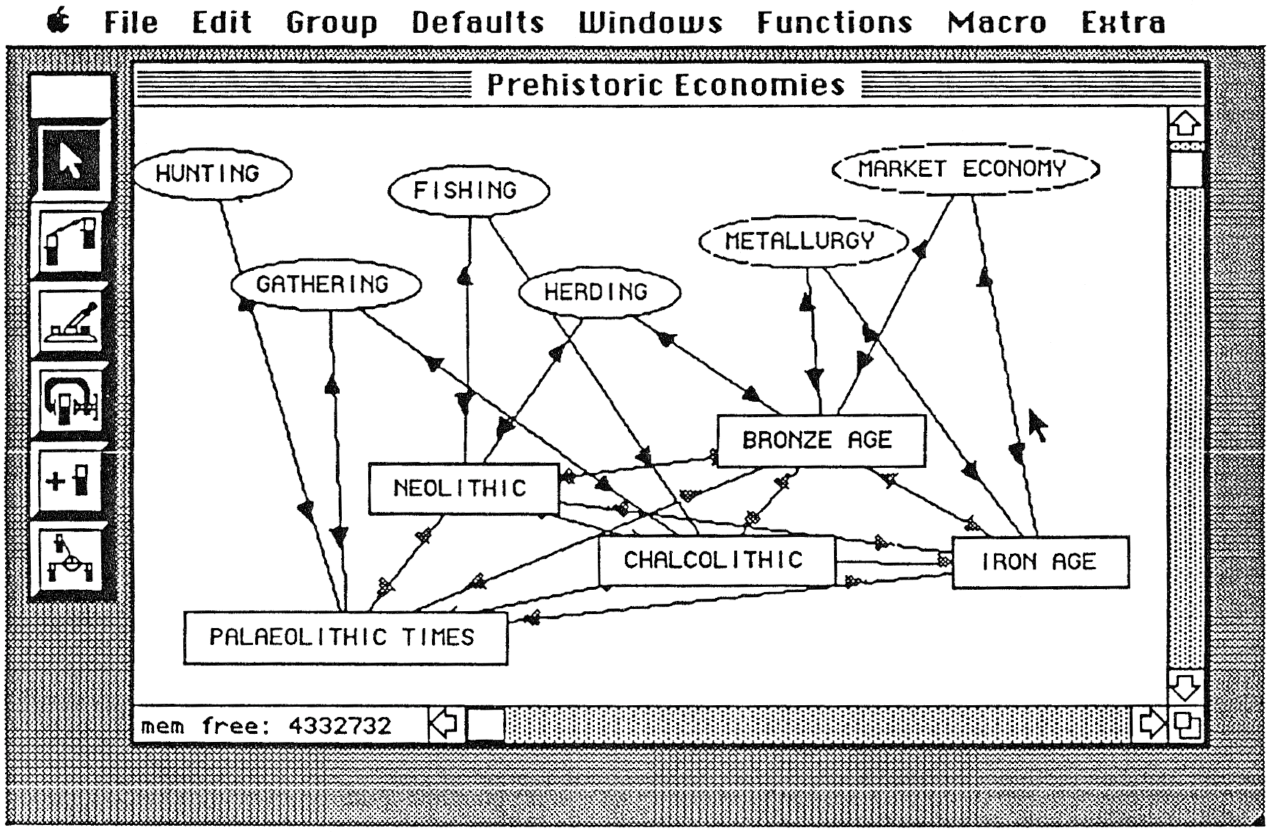


Fig. 2: Example of feedback in a localised neural network (hybrid semantic network).

Third step

The final step accomplished by the transfer function is to convert the unit's activation level to an output signal. Most commonly, this is done by setting the output signal to the following expression:

$$\text{If } f(l) > T: \quad y_i = f(l)$$

$$\text{otherwise:} \quad y_i = 0$$

where T is a threshold value. In other words, the unit's output is its activation level as long as that activation value exceeds a given threshold; otherwise the unit's output is nothing.

When a neural network is running, each unit receives some input from other units in the network. These signals are modified by their passage through the weighted connections leading to the middle layer. Since the weights on the connections are typically different, each of the units receives a different input from its neighbours. As a result, some combination of units will become active to varying degrees, depending on the weights on their input connections. In some cases, information (signals) flows in two directions: from the units and to the units. *Feedback* consists of sending each unit's output signal back to the input of every other unit for recycling (as well as out to the external world). The net output signal, which is fed back from any unit to any other, is determined by the weight of the link between them, just as any net input signal is determined by the weight on the incoming connection. It is important to realise that the weights of links in feedback networks do not need to be symmetrical.

Fig. 2 shows an ideal example of a feedback network. At the top of the screen there are six feature units and at the bottom five chronological ages. The eleven units are computationally the same. The only difference between the two groups is that the chronological units are connected to each other with negative weights (so they are mutually inhibitive), while the production units are not connected to each other at all. There are positive weights between some production (or descriptive) units and some chronological units. These weights represent the corresponding knowledge we have about each age. For instance, a positive weight from 'Gathering' to 'Paleolithic' means that 'Gathering' is a feature associated with 'Paleolithic'. Likewise, a positive weight from

'Paleolithic' to 'Gathering' allows the network to reason in the opposite way: if 'Paleolithic' is active, then 'Gathering' should also be active.

The units in the network compete for the right to generate an output signal through the feedback couplings. The activation level of each unit changes as the signals iterate through the network and reflects the current state of the network during that iteration. After some iterations, the activation of units stabilises at some specific level.

To sum up, a neural network is a computational model that is a directed graph composed of nodes (the *units*) and connections between the nodes. With each unit or node is associated a value, referred to as the node's activation. Similarly, a value is associated with each connection in the network, called its weight. Each node's activation is based on the activation of the nodes that have connections directed at it, and the weights on those connections. The goal of the method is to update the activations *simultaneously* (or in parallel).

Why is neural or connectionist inference 'different'? We are trying to find the best explanation for our data and this 'best' explanation may be substituted by the more penetrating notion of 'activation of the most appropriate activation vector'. Activating the most appropriate available activation vector is what our network does as a matter of course, and it does it directly, in response to the input, without canvassing a single alternative vector.

For any set of observations there is a literal infinity of possible hypotheses that might be posed in explanation. How can we possibly search a space of infinite size? We can assume, as an analogy, that a scientific theory - the search space - is like the human brain: a multilayered network of interconnected units. We can uniquely specify its current position in conceptual space by specifying the individual strengths or weights of its myriad synaptic connections. That configuration of weights can be directly represented by a specific point in a multidimensional space. This space has a distinct axis for each link among units (a unit in the network is a 'concept' or theoretical entity). There is a second space to consider here: the space of possible *activation patterns* across the concepts in the theory. This 'activation-vector' space has a distinct axis for each unit or concept, an axis that measures the level of that concept's activity. A specific configuration of synaptic weights will partition the activation space of a given neuronal layer into a taxonomy of distinct *explanations*.

A neural network to investigate social processes

A neural network is only a representative device, not very different from classical statistical models, because units are like statistical variables, and activation levels may be seen as values. Instead of representing causal relationships though linear equations, by using a neural network we can represent non-linear and very complex interrelationships among the units.

There are two kinds of networks according to the representational nature of units:

- **Distributed Networks** store information - the represented fact - in a *distributed way*, that is, across many units and connections. Therefore, single units do not represent anything. Information is represented redundantly, because a single unit may participate in the representation of several pieces of information (Rumelhart *et al.* 1985, Caudill & Butler 1992)

- **Localised Networks** store information in single units. That is, each unit means something. In this case, units are often referred to as *microfeatures* to emphasise the fact that individual concepts can be decomposed into them (Churchland 1989; Thagard 1988, 1989). Microfeatures are often chosen on the basis of a researcher's feeling that the chosen concepts are somehow basic to a complete semantics of the concepts being represented (Shastri 1988; Zeidenberg 1990).

We have selected a *Localised Network*, because we are interested in *decomposing* a social concept (social differentiation) by detecting its formation (or causal) processes: the social differentiation of work, reproductive control, and the ideological justification for power. The units in our mathematical model represent agents, actions and beneficiaries. Connections represent the specific *association* between variables or components in the social model.

The activation level of any unit represents the *potential* of some variables to become a component of a causal or formation process. In other words, we are coding mathematically the variable, and the *confidence* we have in the relationship between that variable and the others in the model.

We intend to simulate how the social preeminence of men and women depends on the number of productive and reproductive actions they do, and the number of benefits they receive from the actions done by others. We have built a localised neural network where activation spreads from the People units to the Actions units, and back to the People units (feedback). The first spreading activation simulates the number of actions done by each biological category. The second one represents, and describes, the *social* (and not biological) process that conducts people from biological diversity to social diversity.

The final state of the system can be described by saying that one unit (Old Male, for instance) is the winner in the competition among all other units. There are negative weights among all the Social Position units: through a mathematical process called *mutual inhibition*, the unit with the strongest activation *after* each unit has sent to others a negative signal suppresses the activation of the rest of units.

KIPA

Kipa is the Yamana word for woman. We are experimenting our ideas about social differentiation among age and gender categories using a computer program called KIPA. This program reproduces the dialectic process of social diversity formation in Yamana society. The units represent Yamana social categories (as described by the ethnologist, and not as seen by the Yamana) and Yamana working activities.

We are working simultaneously with two versions of KIPA, although they are not yet finished programs. In this section we will describe the architecture of both systems. We will begin by describing the model's architecture, that is to say the type of units, the network's *topology* (connectivity structure), *learning rules*, and *activation functions*. KIPA is implemented in a Neural Network shell called MacBrain (Vsn. 3.0) by Neurix Inc.

Units

Both versions of KIPA contain *p*-type units (Feldman & Ballard 1982). Any unit in KIPA is a computational entity comprising

- ***p*** a continuous value in (0..Max. input), called *potential*
- ***v*** an output value that is proportional with the unit potential
- ***i*** a vector of inputs

We know that:

$$p = f(i) \quad \text{(activation function)}$$

and that:

$$v = f(p) \quad \text{(transfer function)}$$

In KIPA we have considered the activation value (or *potential*) of a unit and the output of a unit to be the same thing. Thus, the activation function is also the transfer function.

Topology

We are doing some experiments with two different network models. The first one (KIPA 1.0, very similar to our current test version) is a *feed-forward layered network* in which no feed-back occurs between any set of units, and where units are only connected to units in the next layer. KIPA 2.0 is an *irregularly connected network*, where feed-back occurs, and where all units in the first layer are connected to all units in the second layer, where all units in the second layer are completely connected, but there is no connectivity among units in the first layer.

Activation function

We have implemented different activation functions for the two main classes of units.

People units are built following a standard *Linear threshold function*, selected because of practical advantages (it is easier to compute than the continuous sigmoidal), and it seems to be more adjusted to the model.

Given that the Input is:

$$I_i = \Sigma\, w_{ij}\, x_j$$

We will compute the Activation level (A) of a unit i:

$$A = k\, I_i$$

where k is a constant.

Action units use a *boolean threshold* activation function. With this function, a unit takes on only its maximum or minimum activation values. If the weighted sum of the incoming signals is greater than the unit's threshold (T = 0), the unit's activation will be its maximum value (max. act. = 1); otherwise, the new activation will be its minimum activation value (min. act. = 0). We have selected this activation function, because we wanted to give equal relevance to all the actions we have extracted from ethnographical records. Consequently, all their outputs must be 1 or 0.

Learning and generalisation

Most neural networks are not programmed to solve problems. They have to learn to do so.

Learning is achieved not by modifying the units in the network but by modifying the weights on the interconnections which allow the system to update activation levels. In other words, learning consists of making systematic changes to these weights in order to improve the network's response performance to acceptable levels.

Training is the procedure by which the network learns; learning is the result of that procedure. Training is done by example, and it can take place in two ways: *supervised*, in which the network is provided with an input stimulus pattern along with the corresponding desired output pattern, and *unsupervised* or 'self-organisation', in which the network is presented only with a series of input patterns and is given no information at all about its performance levels.

The current release of KIPA is an untrained network. The objective is only to experiment some properties of the model, and to evaluate the representation format. We do not intend to create a universal model of social differentiation, nor to calculate the parameters (weights) of a single case model. We already know all these parameters, because Yamana society has been described by ethnologists (Gusinde 1937). We have used this information to link units in our model. For instance, if an agent (a young male) makes a specific action (fishing), then the weight of the link between these two units is 1. If an agent (an adult male) does not make a specific action (cooking), then the weight of the link between these two units is 0. To simplify the model's topology we have preferred to delete all 0 weights.

When the program starts (Fig. 3a) people units are already connected to action units, because we know from the ethnological record which actions were made by which actors. The same is true for

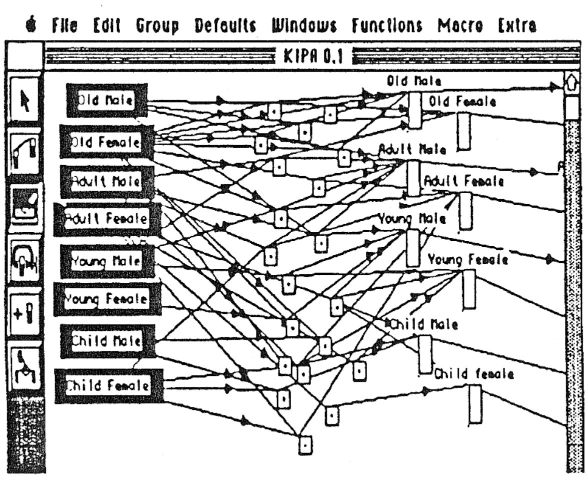


Fig. 3a: Initial activation state.

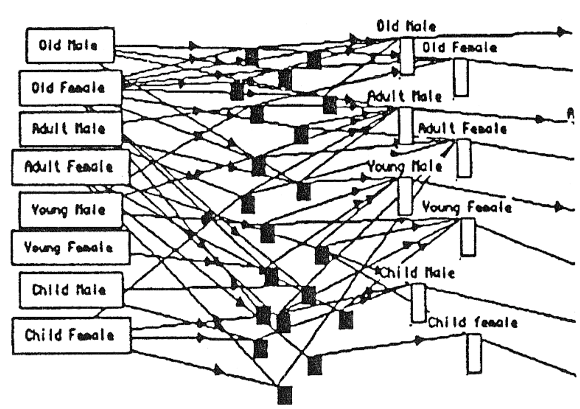
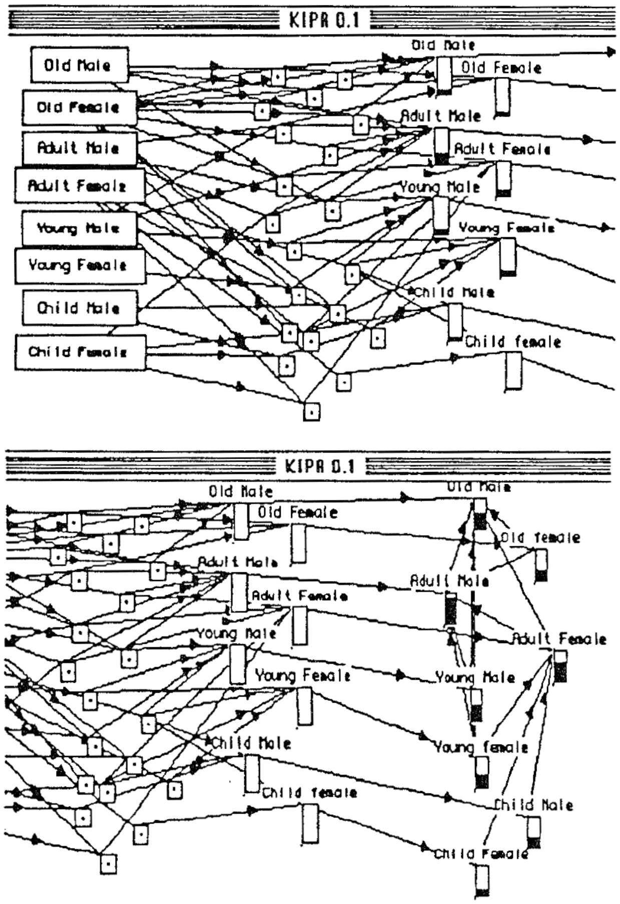


Fig. 3b: Second activation state.

Fig. 3: Running KIPA. Symbols represent agents, tasks and people receiving the profit of tasks. Framed rectangles represent the initial state of the system (equal activation levels for all agents). White symbols represent units before activation. Solid dots represent the activation level of task activities. Vertical rectangles are used as 'thermometers': black space indicates activation level. The higher this level, the higher the domination of one agent over the others.



Figs. 3c & 3d: Third & fourth activation states.

the beneficiaries. Activation spreads from the agent units (first layer) to the action units (second layer) through certain directed links. Given that action units use a boolean threshold function, their activation is always 1 (the maximum level), irrespective of the number of agents involved in that action. Actors (or beneficiaries) units use a *linear threshold function.* Therefore their maximum activation level is equal to the number of actions from which an actor benefits. In the third layer, social differentiation emerges: each social category shows a different activation level. The more the activation, the more preeminence this category has in social life.

However, social differentiation is not just the product of the ratio:

$$\frac{\text{number of benefits}}{\text{number of working actions}}$$

The fourth layer in the model represents the way power relationships affect social preeminence. If the ethnological record suggests the existence of power differences between two categories (for instance, between husband and wife, between mother and children, etc.), then it implies an inhibitory link· (a negative weight), which suppresses the activation of the dominated category and increases the activation level of the dominant one.

Conclusions

We cannot use the word conclusions to finish a paper about such work in progress. The current version (KIPA 0.1 has been replaced by KIPA 0.2 since the conference, in October 1993) has introduced some minor improvements in the architecture of the fourth layer, but it is already an ideal network. Experiments with these two programs have shown to us the advantages of this simulation technology. We can now build a real network, using ethnological information. The number of people units (agents, actors, social categories, that is to say, the first, third and fourth layers) will be the same as in the prototype KIPA 0.1 (due to limitations in the ethnographic description by Gusinde), but the number of actions has to be increased (300 different working actions). The network's topology will then be very complex (more than 1,000 links and connections between layers). Although the size of the model changes to accommodate real data, the architecture of the simulation system remains like the prototype shown in this paper.