



Neural Code in the Human Brain:
The Dynamical Coding Hypothesis

by

Bernard Beauzamy (Ph. D.),

and

Alain Bohec, Christian Colas (M.D.), Marc Mouly

Société de Calcul Mathématique, S.A.
111, Faubourg Saint Honoré, 75008 Paris, France

revised, November 1998

In application of Research Contract DGA/DSP/STTC 96 25 07 A 000,

Ministry of Defense, France.

I. What is a code ?

To code an information means to represent it in a symbolic manner, in a way that allows later access, transformations, manipulations, and further use when needed. For instance, we use Arabic numerals (0, . . . , 9), whereas the Romans used I, II, III, IV, V, . . . , IX. Computers use a binary code, a music note is represented on a staff : all these are examples of codes.

The main property of a code (at least apparently) is to be biunivocal : two distinct elements of information must be represented in distinct manners. But written English does not represent a correct coding of oral English, because there are cases (the homonyms) where two words with different meanings have the same spelling. The same is true for all human languages.

The way an information is used highly depends on the way it has been coded. One code can be quick, but will not be reliable ; another will be redundant, so will be reliable, but slow to operate. Another might require a lot of memory to keep the information. The example of a hard disk of a computer, and various algorithms for data compression, are illustrations of this remark.

The choice of the code corresponds to the orientation of the need. But there is an element which is much more important, and seldom understood. Depending on the code one uses, that is depending on the language that is used, some problems will make sense, some possibilities will appear, some aptitudes will be created.

Let's first take an example of mathematical nature : it is practically impossible to represent the multiplication using Roman numerals. Arabic numerals represent a code that allows to "invent" multiplication. Another example, slightly more sophisticated, is as follows : depending whether you denote a function by $f(t)$ or $\langle f|t \rangle$, you do not have the same point of view : in the first case, f is fixed and t is the parameter, in the second one, you are ready to consider both f and t as variables.

The same is probably true for all human languages (oral and written). Some languages probably favor the coding for some types of thought, and in this fact one might find the origin of the aptitudes of certain nations for particular domains : speculative thought, literature, poetry, music, and so on.

We do not know how the human brain codes the information : neither the immediate information (the one used, for instance, in a reflex), nor the long term information (the memory). The topic of the present work is to introduce an hypothesis on this question. Our starting point will be Von Neumann's observations [Von Neumann, 1957], which we recall in the next paragraph. The whole question is difficult, because all we can access, by experience, are "secondary languages" : speech, music, writing, mathematics. We have no access to the primary language used by the brain. To make this point very clear, let us take a comparison with the computer (as Von Neumann did) : you can look at programs written in "C", Fortran, and so on ; nothing will tell you the computer uses a binary code.

According to Von Neumann, our mathematics, which use a system of numeration that is 6,000 years old, have nothing to do with the code that is actually used by the human brain. His last words, written just before his death in 1957 [Von Neumann, 1957] are : “Whatever the code is, it cannot fail to differ considerably from what we explicitly know as mathematics”.

Von Neumann was a mathematician, his conclusions were personal and were not the result of a team work. They were not pursued after his death. Neurosciences explored other directions : the role of a cortical zone, transmission of spikes, where a given functionality can be found, and so on. This information is precious and quite hard to obtain : science, in this respect, has obtained considerable progresses, and we do have now precise data that were not available to Von Neumann (see in particular [Crow, 1988], [Dudai, 1989], [Squire, 1987], [Rieke and al., 1997]). Moreover, some mathematical modeling has already been realized, on different subjects (spike propagation, for instance).

One should realize, anyhow, that this is not the same question. To take again the comparison with the computer, to ask “where” the information is kept (the hard disk) is not the same as to ask “how” (magnetic data), and this is also different from the question “with what code” (binary). Neurosciences have attacked the first two questions; we are concerned here with the third.

II. Von Neumann’s observations

We present hereafter the reflections made by John Von Neumann. They are taken from his book “The Computer and the Brain” [Von Neumann, 1957]. Though they are more than 40 years old, these observations remain fully topical: the present development of high speed computing, the timid appearance of parallel programming, make even more pertinent than ever Von Neumann’s analyses. The differences he observed between the computer and the brain are now more important ; they did not reduce.

Von Neumann’s observations did not receive the attention they deserved among his spiritual descendants. Not only his questions were not understood, but they are even not tackled anymore.

In the sequel of this paragraph, we reproduce Von Neumann’s words (in italics). In rare cases, we have added a comment.

A. Background

Existing computing machines fall into two broad classes : “analog” and “digital”. In an analog machine, each number is represented by a suitable physical quantity, whose values, measured in some pre-assigned unit, is equal to the number in question. In a decimal digital machine, each number is represented in the same way as in conventional writing or printing, i.e. as a sequence of decimal digits. Each decimal digit, in turn, is represented by a system of “markers”.

The functioning of the nervous system is, at first look, of digital nature. The basic component of this system is the neuron, and the normal function of a neuron is to generate and to propagate a nerve impulse.

The nervous pulses can clearly be viewed as (two-valued) markers: the absence of a pulse represents 0, the presence represents 1. Upon receipt of pulses in certain combinations and synchronisms, a neuron will be stimulated to emit a pulse of its own, otherwise it will not.

The normal stimulation, when it comes from another pulse, emanates from a special ending of the axon, a synapse. The time of trans-synaptic stimulations is of the order of 10^{-4} s. Immediately after the stimulated pulse has become evident, the stimulated neuron has not yet reverted to its original condition: it is fatigued, for about 1.5×10^{-2} s. [In modern words, this is called “refractory period”]. So the reaction time of a neuron can be estimated to 10^{-2} s. Modern computers are much faster: they can execute several billions of operations per second, which gives 10^{-9} s per operation. The factor between both is of the order of 10^7 [10^4 at Von Neumann’s time].

The natural componentry favors automata with more, but slower, organs, while the artificial ones favors the reverse arrangement of fewer, but faster organs. Hence, large and efficient natural automata will tend to be parallel, while large and efficient artificial automata will tend to be serial. Hence, the logical approach and structure in natural automata may be expected to differ widely from those in artificial automata. Also, the memory requirements of the latter will be more severe than those of the former.

Basic logical operations. If a neuron is contacted, by way of their synapses, by the axons of two other neurons, and if its minimum stimulation requirement, in order to evoke a response pulse, is that of two simultaneous incoming pulses, then this neuron is an “and” organ. If the minimal requirement is the arrival (at least) of one pulse, the neuron is an “or” organ. The neurons appear, when thus viewed, as the basic logical organs, and hence also as the basic digital organs.

The more frequent situation is that the body of a neuron has synapses with axons of many other neurons. Things can be more complicated than just the “and”, “or” or threshold functions, and take into account spatial relations between neurons, or more complicated quantitative and geometrical relationships.

The memory. We do not know where in the physically viewed nervous system a memory resides. We are as ignorant of its nature and position as were the Greeks. The only thing we know is that it must be a rather large-capacity memory, and that it is hard to see how a complicated automaton like the human nervous system could do without one. An evaluation of the capacity of the memory in the brain is 2.8×10^{20} bits (number of digital impressions per second, times number of cells, times number of seconds in a lifetime).

There is every reason to believe that a very large-capacity memory is associated with the nervous system. We most emphatically do not know what type of physical entities are the basic components for the memory in question.

B. Notion of code

The nervous system has a nerve-pulse part, which is digital, and a part involving chemical changes, which is of analog type.

Codes. A system of logical instructions that an automaton can carry out and which causes the automaton to perform some organized task is called a code. A code may be complete: one may have specified the sequence in which the impulses appear and the axons on which they appear. One may also speak of “short codes”.

A. M. Turing showed that it is possible to develop code instruction systems for a computing machine which cause it to behave as if it were another, specified, computing machine.

[At this level, the terminology of “short code” is improper: the list of instructions given to a machine A, in order to execute a given task, may be much longer if it is asked to imitate a machine B, than if it does it directly. Von Neumann refers to what we now call “macro-instructions”]

A short code must contain, in terms that the machine will understand, instructions that will cause it to examine every order it gets and determine whether this order has the structure appropriate to an order of the second machine. This way, the first machine can be caused to imitate the behavior of any other machine.

The order structure referred to may actually deal with orders of a much more complex character than those which are characteristic of the first machine. The reason for calling such a secondary code a short code is historical: these short codes were developed as an aid to coding, i.e. they resulted from the desire to be able to code more briefly for a machine than its own natural order system would allow. [Think of programming languages, vs machine languages]

Any artificial automaton that has been constructed for human use, and specifically for the control of complicated processes, normally possesses a purely logical part and an arithmetical part. This is due to the fact that, with our habits of thought, it is very difficult to express any truly complicated situation without having recourse to formulae and numbers. For instance, an automaton which is to control constancy of temperature will, if a human designer has to formulate its task, have that task defined in terms of numerical equalities or inequalities. On the other hand, there may be portions of the task which can be formulated in purely logical terms.

C. Importance of numerical procedures

The nervous system, viewed as an automaton, must have an arithmetical as well as a logical part. But what is the precision of the arithmetical part? This question is particularly crucial: all experience with computing machines shows that if a computing machine has to handle as complicated arithmetical tasks as the nervous system obviously must, facilities for high levels of precision must be provided. Calculations are likely to be long, and in the course of long calculations not only do errors add up, but also those committed early in the calculation are amplified by the latter parts of it. One would expect that the arithmetical part of the nervous system exists and, when viewed as a computing machine, must operate with considerable precision: ten or twelve decimal precision.

But in the nervous system, this is absolutely impossible. Intensities are translated into frequencies, which usually lie in the range 50 - 200 per second. Clearly, under these conditions, precisions of 10 to 12 decimals are out of question. Only precision levels to 2 or 3 decimals are possible. No known computing machine can operate reliably and significantly on such a low precision level.

The nervous system leads also to a rather high level of reliability. In a digital system of notations, if a single pulse is missing, nonsense may result. In a statistical system of notation, if a single pulse is lost, or even several pulses are lost, the relevant frequency, i. e. the meaning of the message, is only inessentially distorted.

D. Arithmetic and logical structure in the brain

In the course of a long calculation, the deterioration is due to the accumulation and amplification of errors. It depends on the arithmetical depth of the calculation, which, in turn, depends on the logical depth of the scheme.

The message system used in the nervous system is of an essentially statistical character. What matters are not the precise positions of definite markers, digits, but the statistical characteristics of their occurrence, i.e. frequencies of periodic or nearly periodic pulse-trains, etc.

Thus, the nervous system appears to be using a radically different system of notation from the ones we are familiar with in ordinary arithmetics and mathematics: the meaning is conveyed by statistical properties of the message. This leads to a lower level of arithmetical precision, but to a higher level of logical reliability.

E. Language

The nervous system is based on two types of communications: those which do not involve arithmetical formalisms, and those which do. The former may be described as language proper, the latter as mathematics.

It is only proper to realize that language is largely an historical accident. The basic human languages are traditionally transmitted to us in various forms, but their very multiplicity proves that there is nothing absolute and necessary about them. Just as languages like Greek or Sanskrit are historical facts and not absolute logical necessities, it is only reasonable to assume that logics and mathematics are similarly historical, accidental forms of expression. They may have essential variants, i.e. they may exist in other forms than the ones to which we are accustomed. Indeed, the nature of the central nervous system and of the message systems that it transmits indicate positively that this is so. We have now accumulated sufficient evidence to see that whatever language the central nervous system is using, it is characterized by less logical and arithmetical depth than what we are normally used to.

The following is an obvious example of this: the retina of the human eye performs a considerable reorganization of the visual image as perceived by the eye. This reorganization is effected on the retina, or, to be more precise, at the point of entry of the optic nerve, by means of three successive synapses only, i.e. in terms of three consecutive logical steps. The statistical character of the message system used in the arithmetics of the central nervous system and its low precision also indicate that the degeneration of precision cannot proceed very far in the message system involved. Consequently, there exist here different logical structures from the ones we are ordinarily used to in logics and mathematics [Von Neumann said earlier that the basic logical functions “and” and “or” were the same in the central nervous system and in our logics. He means here that these primitives are not arranged in the same manner, in more elaborate functions.]

Thus logics and mathematics, in the central nervous system, when viewed as languages, must structurally be essentially different from those languages to which our common experience refers.

The language here involved may well correspond to a short code in the sense described earlier, rather than to a complete code: when we talk mathematics, we may be discussing a secondary language, built on the primary language truly used by the central nervous

system. Thus the outward forms of our mathematics are not absolutely relevant from the point of view of evaluating what the mathematical or logical language truly used by the central nervous system is. However, the above remarks about reliability and logical and arithmetical depth prove that, whatever the system is, it cannot fail to differ considerably from what we consciously and explicitly consider as mathematics.

III. Propagation of action potentials

A. Signal propagation in the brain

According to the classical schemes, a neuronal “event” is characterized by the propagation of an action potential (a “spike”) along the axons. These action potentials (AP) all have roughly the same shape and the same size ; only the frequency differs from one event to the other. But a specific frequency will determine the path taken later by the signal, by allowing the apparition, on certain synapses, of certain neuro-transmitters, which themselves will permit the propagation of the AP to the next neuron. Other frequencies, on other synapses, will make inhibitors appear. So one sees that this code, built upon frequencies, realizes “switch” functions. Depending upon the frequency, the path of the AP will not be the same.

These facts are of experimental nature and cannot be disputed. However, standing alone, they do not provide a sufficient answer to the question : “what is the nature of the code ?”

To describe things in a rough manner, one has the impression of signals, propagating constantly, sometimes along some circuits, sometimes along other circuits. Let’s take an old analogy, from ancient electronic circuitry. One has the same impression when one looks at a vacuum tube, such as a triode : the grid potential, depending on how negative it is, blocks or does not block the current going from the cathode to the anode : this is also a switch, and not only a zero-one switch : it reacts in a quantitative manner. Much before vacuum tubes, man has built circuits containing switches, for instance relays. A switch is nothing but a component, no matter whether it is electronic or neuronal. One could easily realize circuits, made of billions of neurons, connected by switches, in which currents would propagate according to one scheme or another. But so what ?

Such circuits would not realize any of the functions of the brain. In other words, to put into evidence the functions of neuro-transmitters (the switches) is useful, but does not tackle the true problem. If we open a radio set and observe that there are vacuum tubes, we have acquired an information. If we study one of them, and put into light the influence of the grid upon the electron flow, as previously described, we have indeed understood something : it’s a non-trivial step. However, we are very far from having understood how the radio set works : a wave coming from the outside, its demodulation and amplification, and so on.

Let us already point out one thing which might be seen as a problem, when one looks at this vast set of neurons between which currents are flowing : these currents are constantly changing. Nothing seems to be fixed, even for the shortest amount of time. This does not seem compatible with a code. A code implies an idea of stability, even for a very short time. An addition, in a computer, is made by combining the data carried by one memory register with other data, carried by another register. At a given time, these data are fixed : this bit is at 0, this one is at 1. If we realize (as we can do in present circuitry)

several billions of operations per second, each operation acts indeed upon something which is stable and fixed: the state of the memory just before the operation. Of course, the operation itself modifies the contents of the memory, but, in order to be performed, it requires fixed data to work on them. We cannot conceive that the contents of the memory change constantly before the operation : upon which value would the operation be performed ?

This absence of fixity at certain instants, no matter how brief they might be, causes a problem in the purely propagative scheme we just described. One does not see at first glance anything fixed, and a code “needs” something fixed between two operations (the word “needs” is ambiguous. Let’s say that no code realized by a man can proceed otherwise. To assert that this is impossible would be presumptuous. Anyway, our aim is not to explore all possible coding schemes, but to try to get an idea of the procedure nature used in the brain).

This observation, as pointed out by Dr. Vibert (Hôpital Pitié-Salpêtrière, Paris), is not completely correct : it refers only to digital codes, or numerical codes (each quantity is represented by a number). If one uses an analogue code (for instance quantities being represented by a flow (of water, for instance), one can easily conceive operations which would be continuously performed.

An element of code, clearly brought into light by Von Neumann, is the frequency of the signal. This element (which is of statistical nature) has the fixity properties we were looking for : the signal constantly propagates, but its frequency is fixed during a certain amount of time. Moreover, well-studied neuronal events (synaptic transmissions, threshold effects, and so on) imply a modification of the frequency of the signal. Therefore, it seems legitimate, following Von Neumann, to consider that frequency is an element of the code.

So we have a very complex set of neurons, and we know (at least partially) that on a certain net, a certain branch will carry a frequency of 100 Hz, and if some event occurs (let’s say for instance : the eye sees a red disk), during a certain amount of time, on certain branch of the circuit, the frequency will fall to 90 Hz during a certain time, and on some other branch it will raise to 120 Hz during another time. Of course, this is a simplification : there are billions of circuits, and the variations may be extremely complex on each of them.

Some experimental facts have been observed, concerning frequency variations. In the case of the cat, some groups of neurons develop signals at 14 Hz when the animal is just waiting ; this frequency rises to 30 Hz if the animal is attentive. This frequency, common to a whole group of neurons, represents a simultaneous behavior of a whole net, something that could be called “resonance vibration” [work done by Dr. Rougeul-Büser. Communicated by Dr. Véron].

Let us now imagine that, for some animal with few neurons, we have observed all circuits and performed all measures : we know the variation of all frequencies, on all branches of all circuits, when some event occurs (for instance : the animal sees a light). Have we understood the brain code of the animal ? Not at all. Let’s imagine we open a computer and measure all intensities on all cables, on all processors, when the computer performs elementary actions (an addition, for instance). This is a kind of cartography ; it is useful but does not tell us about the kind of code the computer uses.

B. Coding and cartography : neuronal zones

Today, the fact that, in the brain, specific functions are associated with (one or several) specific zones is commonly accepted. Rather than to speak about a “zone”, in the strict sense of the word, one should speak of a “succession of zones”, because signals usually go through several consecutive centers, where they are transformed. For instance, in the case of vision, which is probably the most studied and the best known among neuronal processes, one knows rather well the nature of cortical zones which are crossed by the signal ; one knows how to locate each zone with precision, and one knows rather well what is the “use” of each zone, in the sense that one may observe, and sometime measure, the malfunctions of the signal which are induced by a deficiency of the corresponding area. A lesion of a certain zone, for instance, will affect the vision of colors ; another area is mostly concerned with movements, and so on.

In the case of vision, recent work [Fuyita and al., 1992] seems to indicate an interesting concept : that of “retinotopic” coding. The set of neurons that codes a given shape (for instance a stick) has, in the brain, a similar shape. The cortical area has, geographically speaking, a resemblance with the form it has to code : the brain contains a map topologically equivalent to what is projected upon the retina, in the sense that two neighboring elements in the retina are represented as two neighboring neurons in the primary visual area.

Of course, scale corrections are necessary and boundary effects have to be taken into account. The same is true, apparently, for the ear (tonotopic coding).

These observations or assumptions are interesting. Do they inform us about the code used by the brain ? Not really. This is another type of cartography. Let’s imagine we are looking at the hard disk of our computer (its memory). By performing certain experiments (for instance neutralizing some sectors of the hard disk), we will discover that each sector is associated to some function : if this sector is damaged, the word processor does not work anymore ; if the maths coprocessor is damaged, floating points operations will not be as fast, and so on. All this information, of cartographic type, is interesting, but it does not inform us about the type of code used by the computer.

One should strongly insist upon the difference between cartography and code. These are distinct concepts.

Of course, a given type of code will have to be implemented in a specific manner, which in turn will impose a given cartography. There are links between the two, and the knowledge of the cartography may inform us about the type of code, mostly by allowing us to eliminate some types of codes which would look incompatible with the observed cartography. But, altogether, the cartographic information is not sufficient to understand the code.

C. The notion of “Logical depth” of an operation

This concept was introduced by Von Neumann. It represents the number of transformations the signal has to go through, between the input and the output. When these transformations are performed by “elementary transformers” (here, the neurons), of which the individual action is well-known, the concept is directly linked to the “complexity” of the operation. The more numerous are the neurons the signal passes, the more complex the operation is. To come back to our electronic analogy, this would be the same for a signal going through transistors. To take an example, a reflex normally uses few neurons ; it is

a kind of automatic reaction, which can be inhibited by a feed-back loop situated “above” (that is, controlling it). One sees that evolution has added circuits of command, superposed one upon the others, thus increasing the possible logical depth for each operation.

The interest of this notion is that it is directly accessible to experiments, at least in simple cases. As we said earlier, one can follow the signal sent by the optical receivers, as it proceeds to the cortex, and one can count the number of transformations it undergoes (though these transformations are not understood).

The possible decrease in precision during the operations, the system’s ability to provide correct answers even if data are incomplete (for instance identification of someone from an incomplete photography), all this, as Von Neumann said, helps us understand the nature of the code.

It seems experimentally accepted that, in the case of vision, the number of synapses crossed by the signal is rather small : 5 to 10. This means that the signal undergoes few operations : the logical depth of the transformation is small. Let us observe, by comparison, that 10 to 20 elementary transistors are necessary in order to perform the addition of two numbers written in base 2 (the precise number of transistors depends on the way the operation is performed).

This observation was used in two ways, quite contradictory :

- Von Neumann : “The code used in the central nervous system uses a logical and arithmetical depth smaller than that of our mathematics. For instance, the retina realizes a considerable reorganization of the image seen by the eye, and this reorganization is made at the point of entry of the optical nerve by 3 synapses only, that is in three logical steps only. The drop in precision is not important, and so the logical structures must be different”.
- Conversely, Simon Thorpe [Thorpe, 1990] (see also Geoffrois, Edeline, Vibert [Geoffrois and al., 1994]) considers that the coding, at least in some cases, cannot be realized using frequency only. In the case of vision, some reactions require a time of 100 to 140 ms, but the signal has crossed 10 synapses at total. Each synapse is thus crossed in 10 to 14 ms on average. But since frequency is of the order of 100 Hz, this means that only one or two spikes are emitted in 10 ms, and frequency cannot enter. This author proposes as element of code the arrival times of the spikes.

In this description, each spike has an individual importance : the fact that one spike arrives before another may for instance block a given neuron. So, there is no more statistical property connected with frequency, and the questions asked by Von Neumann come back : if there are 10 consecutive synapses, and if the code is the instant of arrival of a given spike (or the fact that a spike arrives before another), the precision at the end of the 10 operations should be extremely weak : everything depends in a crucial way on the arrival time of the first spike on the first neuron. However, this decrease in precision does not exist.

These authors do not come to the question of propagation of numerical errors, though such questions are essential when one deals with coding.

Let us note, however, that the code using arrival times may be converted into a code using frequencies later in the sequel of the treatment (this changes nothing to our remark concerning the fall of precision, though).

The owl is able to detect differences of localization for its preys ; these differences are obtained by comparisons of arrival times for the spikes, which can differ by a micro-second (comparisons of phases, and not of frequencies). It allows a perception by the animal of angular differences of 2 or 3 degrees [Dr. Vibert and Collaborators].

IV. Dynamical Coding

A. Overview

The experimental elements mentioned by Simon Thorpe (if they are correct) show that coding built upon frequencies cannot be sufficient to describe all situations. There are cases when “one does not have time enough” to code using frequencies. Frequency is essentially a statistical data ; it makes sense only if a dozen of spikes are emitted during the considered interval of time.

The notion of “zone cartography”, which we discussed earlier, shows that all zones are not equivalent : certain zones are associated to certain functions : this is now a well-established fact.

The Dynamical Coding Hypothesis that we describe here is that the path followed by the nervous influx is an essential part of the code.

Here, the word “path” is to be taken in the sense of “itinerary”. Let us explain first what we mean, from a childish example.

There are many ways of going from Paris (France) to St Petersburg (Russia). You can fly, or you can go through Frankfurt (Germany) and then Warsaw (Poland), or you can go through Stockholm (Sweden).

All these itineraries have the same endpoints (Paris and St Petersburg), but certainly they do not carry the same information. For instance, if you passed through Stockholm, you have seen that city, and you may have sent a postcard to your friends. We should also take into account the time you spent in each place, and the speed with which you traveled in between. All this carries information.

Also, you should not think of just one person: this is misleading. You should think of a whole group, all going from Paris from St Petersburg. This group may split at some places, gather at some others, so one has a lot of different possible situations. Each of them carries specific information.

In the case of the nervous influx, it seems that the speed of propagation is rather constant, so we are interested only in the itinerary it takes; that’s what we call the path.

The Dynamical Coding Hypothesis does not reduce to the fact that the signal may go through certain zones, having particular functions. In fact, it means that information is represented (coded) by the type of path taken by the signal : this implies that, for some functions, some zones will be active and some other zones will be inactive, but our concept does not reduce to that. Activity or inactivity for some zones are cartographical concepts, that is they are of static nature. Our hypothesis is of dynamical nature : it also takes into account the order in which the zones are touched, and, more generally, of the ordering of the paths taken by the influx. For instance, let’s look at the picture below :

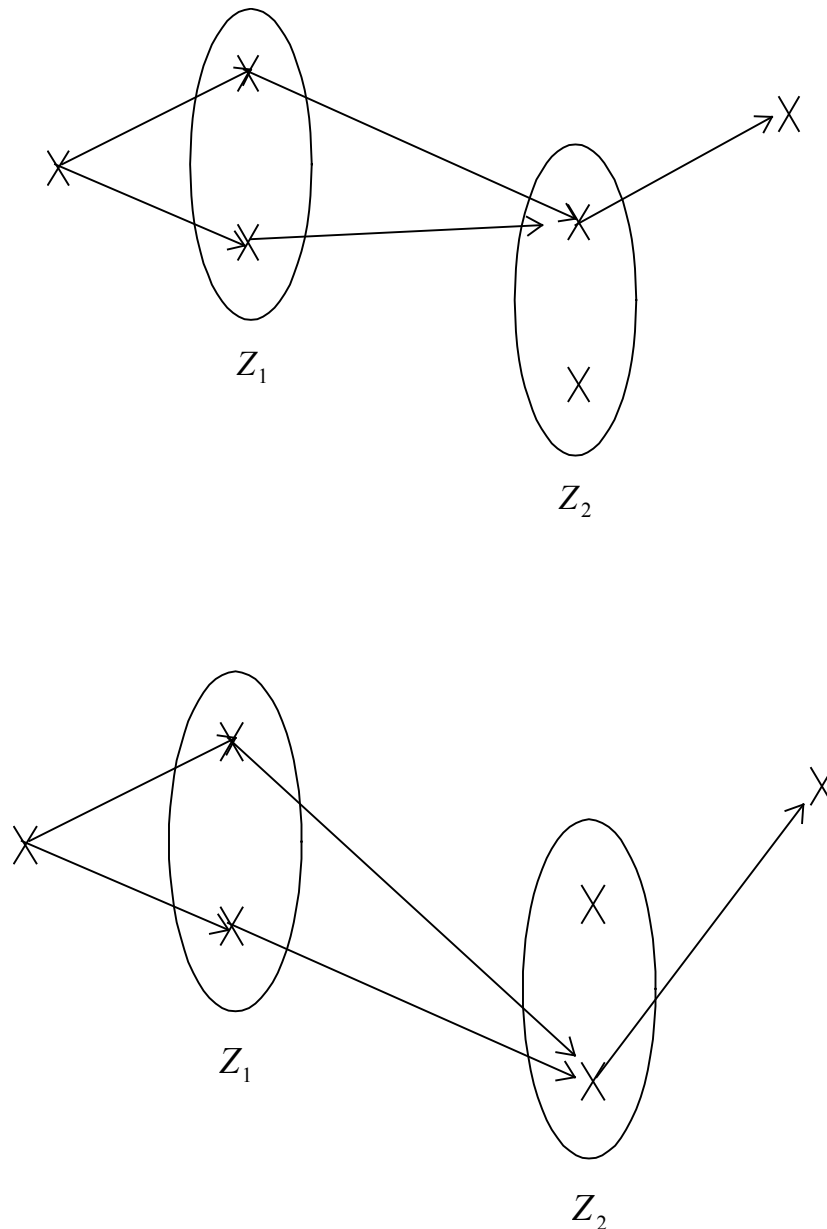


figure 1: Two different paths

The two diagrams above correspond to different paths (in the mathematical sense of the word). However, in both cases, the two zones Z₁ and Z₂ are active and are reached in the same order. To come back on our childish example, zone Z₁ might be Germany, zone Z₂ might be Poland. In zone Z₁, you have two cities, Frankfurt and Bonn, and in Z₂ two cities, Warsaw and Gdansk. A group of people is going from Paris to St Petersburg; half of them go through Frankfurt and half to Bonn, but in one case all go through Warsaw and in the other all go through Gdansk: This is not the same.

The spike frequency might be an element of the code, but certainly is not the primary one. If one permutes the two neurons in zone Z₂ above, this leads to different paths (different trajectories), so it leads to different encodings. Even with the same spike frequency on both, the two diagrams of figure 1 say represent different events.

Another example :

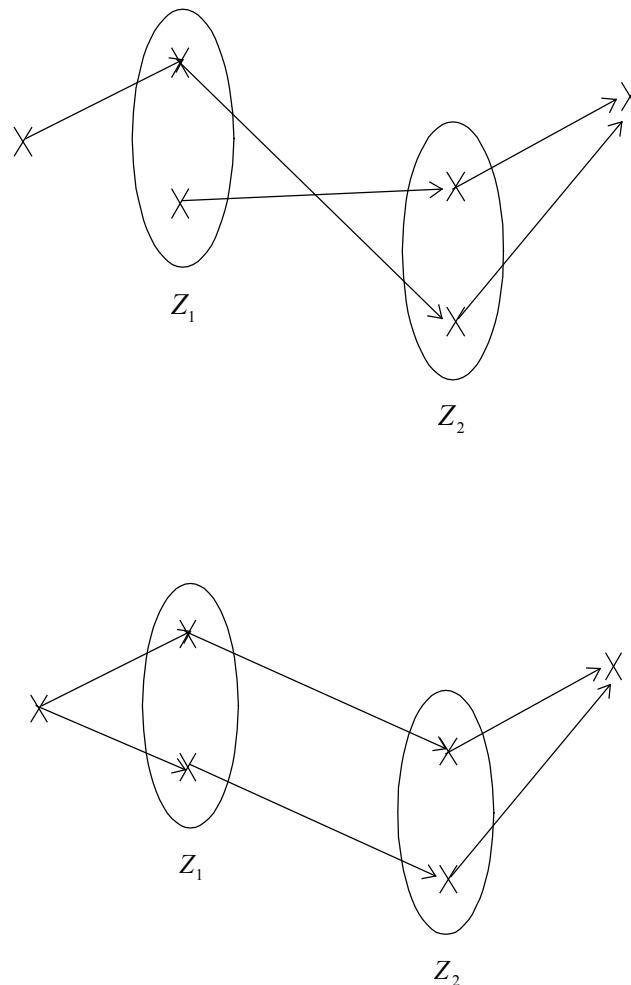


figure 2: Different paths through the same zones

Here again, the above diagrams correspond to different paths, though the same zones are active and in the same order. Even if the signals carry the same frequency, both diagrams of the above figure do not represent the same event.

The Dynamical Coding Hypothesis takes into account the concept of “switch”, which we met earlier.

The neuro-transmitters, activated by a given signal, activate or inhibit a given circuit. The fact that a circuit is, at a given time, activated or inhibited represents (in our hypothesis, but let us use the present and not the conditional, to simplify) the fixed element (that is : constant over a brief interval of time) we need for a code.

Of course, frequency coding also plays a role : along a given path, the signal may have one frequency or another. But this is a secondary code, inside the primary code which is the choice of the path. Secondary does not mean of lesser importance, because we know that the frequency of the signal will orient the next switch, by freeing or inhibiting neuro-transmitters, and will therefore determine the sequel of the path followed by the

influx. The choice of the path is of qualitative nature, the choice of frequency is of quantitative nature.

Dynamical coding allows us to remove the difficulty mentioned by Thorpe : the transit time is too short to allow frequency coding. In this precise case, the only coding is a choice of paths : a switch, which can be done very quickly and with no loss of information (it is binary).

Let us note an experimental observation, which gave birth to an hypothesis anticipating our hypothesis of Dynamical Coding. In the case of a complex coding, for instance a round shape associated to a color, one noted a phase coupling (something one might call a “resonance”) between the neurons coding the shape and those coding the color [Dr. Vibert and collaborators]. The words “phase coupling” are probably too restrictive, except in simple cases : shapes and colors may be very complex, and a simple correlation of phase cannot suffice to describe them in full generality. However, this fits very well with the Dynamical Coding Hypothesis : the path taken by the influx goes through the zone “shape” and through the zone “color”, and the various possible paths (associated to various possible frequencies) may very well suffice to represent all possible diversities of shapes and colors.

Assume the information “red” was input at a certain time : then the signal took a certain path. If this information is changed to green the path will be modified (at least in part).

B. Degree of complexity for Dynamical Coding

This type of coding (independently of frequency) allows to code an information much larger than just choices of zones would permit.

Indeed, let us take n neurons. There are 2^n possible subsets, counting the whole set and the empty subset. Let’s take a simple example :

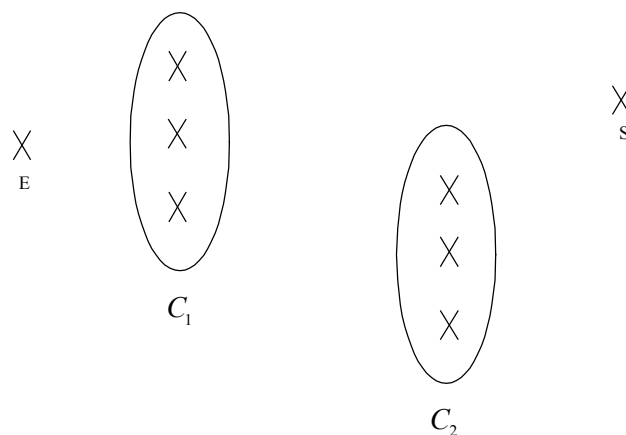


figure 3: Description of the circuit

This circuit is made of an entry neuron E, an output neuron S, and two layers C_1 , C_2 , each made of 3 neurons.

The signal comes from E, arrives at S, and must pass through C_1 and C_2 .

1) If the coding is made “by zones”, one has to choose a subset of C_1 and a subset of C_2 . There are $2^3 - 1 = 7$ subsets of C_1 (the empty subset is excluded), 7 for C_2 , and so this gives a total of $7 \times 7 = 49$ possible codings.

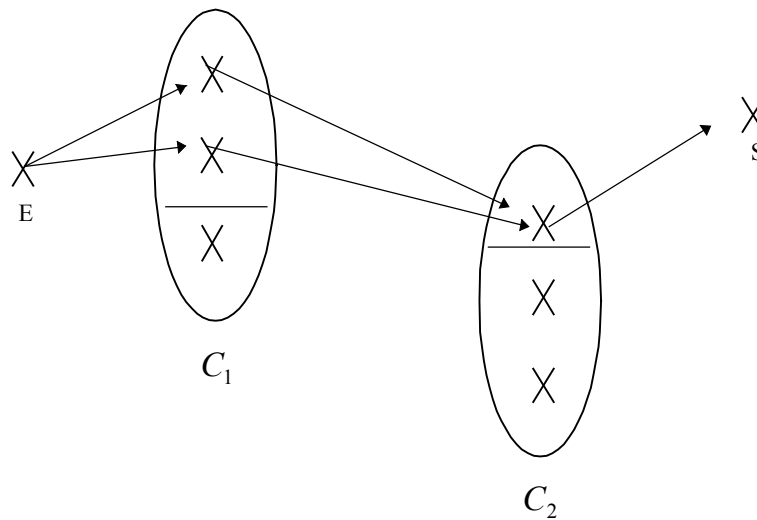
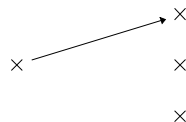


figure 4: An example of a code

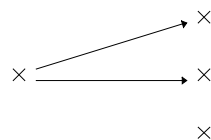
(Here is an example of one of the 49 possible codes. Here we chose the top two neurons of C_1 and the top neuron of C_2).

2) If the coding is dynamical, the number of paths is much higher, and the counting is more delicate. We now give it.

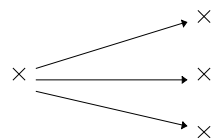
a) We have three simple ways of entering C_1 (path reduced to one arrow)



3 double ways (path consisting in two arrows)



1 triple way (path consisting in three arrows)



So at total 7 ways.

b) Each neuron in C_1 may be connected to 0, 1, . . . , 5 neurons (the two others in C_1 and the three in C_2). That makes $2^5 = 32$ possible connections. Since there are 3 neurons in C_1 , that makes 32^3 possible connections. But not all of them are acceptable : the path must reach C_2 , and should not remain in C_1 .

The number of paths that “loop” in C_1 is 3^3 (each neuron may be connected to any of the two others, or to both, which makes 3 choices per neuron). So the total of admissible paths, starting at C_1 and arriving at C_2 is $32^3 - 3^3 = 32741$.

c) Now we have to connect C_2 to S. Each neuron can be connected to 0, 1, 2, 3 neurons (the two others in C_2 and S), so there are 2^3 possible choices. Since there are 3 neurons in C_2 , the number of paths is $(2^3)^3 = 512$. One should eliminate the paths that do not leave C_2 : their number is 3^3 as before. So we are left with $512 - 27 = 485$ admissible paths, connecting C_2 to S.

Finally, the number of paths from E to S, passing through C_1 and then through C_2 is :

$$7 \times 32741 \times 485 = 111\,155\,695.$$

Example :

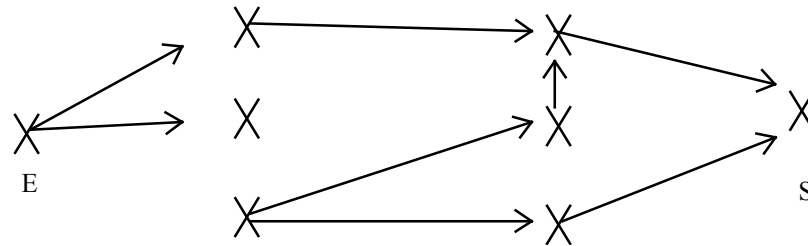


figure 5: Example of a path

We can easily extend this reasoning, and this extension will give an idea of the order of magnitude for the number of paths.

Consider layers C_0, C_1, \dots, C_N , having respectively k_0, k_1, \dots, k_N neurons. A neuron in C_j may be connected to 0, . . . , $k_j + k_{j+1} - 1$ neurons (the $k_j - 1$ others in C_j and the k_{j+1} in C_{j+1}). So there are $2^{k_j + k_{j+1} - 1}$ possible connections.

For the k_j neurons in C_j , there are $(2^{k_j + k_{j+1} - 1})^{k_j} = 2^{k_j^2 + k_j \cdot k_{j+1} - k_j}$ possible connections.

The number of paths which are not acceptable (because they stay in C_j) is $(2^{k_j - 1})^{k_j}$.

So the number of admissible transitions from C_j to C_{j+1} is

$$2^{k_j^2 + k_j \cdot k_{j+1} - k_j} - 2^{k_j^2 - k_j} \tag{1}$$

and the total number of paths is :

$$A = \prod_{j=0}^{N-1} \left(2^{k_j^2 + k_j \cdot k_{j+1} - k_j} - 2^{k_j^2 - k_j} \right). \quad (2)$$

If all layers have the same number of neurons, say k , we find :

$$A \approx \left(4^{k^2} - 2^{k^2 - k} \right)^n \approx 4^{nk^2}. \quad (3)$$

So, as a conclusion, we see that the number of paths, obtained by dynamical coding, is considerably higher than the cartographic coding, which would give here $(2^k)^n = 2^{nk}$.

Let us observe that, though A is already extremely large, we have only taken into account the paths which go from C_0 to C_1 , then from C_1 to C_2 , and so on in this order : there is no feed-back between zones (there may be a feed-back inside a zone). However, one knows (and we saw it earlier) that nature has implemented feed-back circuits (which, for instance, would come back from C_3 to C_1). In these conditions, the number of paths will be still higher. Recall also that there is a secondary coding, using frequencies, atop the primary coding ! As we already said, frequency coding will pilot the sequel of the path, by orienting switches in a given direction.

V. Discussion

A. Ties to existing data and neuroscience literature

The dominating idea, in neuroscience literature, is that spike frequency is the main element of code. A detailed study can be found in the recent book [Rieke and al., 1997], where statistical data from the spike train are studied, using concepts from probability theory, such as Bayes formula. It also investigates information theory, entropy, and so on, with the underlying assumption that the nervous signal must be more or less built on the same pattern as man-made signals (radio waves, for example).

On one hand, this approach does not prove to be satisfactory when there are only few spikes (as we already said), so that statistical data are not numerous enough. On the other hand, to believe that Bayes formula may be of some help to understand the coding in the brain looks very naive, to say the least. It certainly contradicts Von Neumann's words, quoted above : "whatever the system is, it cannot fail to differ considerably from what we consider as mathematics". As we already said, and as this example shows clearly, Von Neumann's ideas were lost.

In the book we just quoted, the idea that the itinerary of the signal might be an element of the code appears nowhere. To the best of our belief, we present it here for the first time. On the other hand, the notion of cortical zone appears frequently in the literature, though only in a static manner (we explained the difference above).

We are now going to develop several items which support our hypothesis.

B. *The memory*

If we admit, as proposes the Dynamical Coding Hypothesis, that coding is done by the designation (activity or inhibition) of some paths, the mechanism of memory fits very well in this scheme.

Let us send the reader to the book by Georges Chapouthier [Chapouthier, 1994] for a brief summary of known facts. One does not know how memory operates (chemically, electric potentials ?), nor where it is localized (and even not if it is).

Dynamical Coding we just described possesses latency properties which make it apt to realize some operations. “A certain path is active” is the basic element of the code. In order that it can serve as a memory, a path should have to be designated as active on a permanent basis, and some other might be inactivated on a permanent basis. But it is quite possible, taking into account the very large number of neurons, synapses, to qualify some paths in this manner. Inside a group of neurons, the fact that certain synapses connect with certain neurons would represent the basic “element of memory”.

Von Neumann gives an estimate of 2×2^{20} bits for the capacity of the memory, in order to keep all information recorded during a lifetime. Dynamical Coding ensures well beyond this capacity : with just one billion neurons, organized into 10 layers, without taking into account feed-backs and frequency coding, one gets $4^{10^{16} \cdot 10} = 4^{(10^{17})}$ bits of information.

The fact that memory is linked to the activation or inhibition of some circuits is, up to a certain extend, an experimental fact. For instance, when a child learns to speak, one sees experimentally, by RMN imaging, that some cortical zones become active. The influence exerted by the, the environment, upon synaptic evolution, is known as epigenesis [J.-P. Changeux, 1983]. Hebb’s law describes the aptitude of two neurons, having worked together, to continue this collaboration. These facts go in the same sense as our hypothesis, which, let us repeat, goes much beyond : for us, the activation or deactivation of the circuits is the element of memory.

Of course, in order to keep a circuit active in the long term, one needs a secondary circuit, the duty of which is to “refresh” the first one from time to time and keep it alive. But nature has provided several secondary circuits (feeding, waste elimination), so to suppose the existence of that one is not absurd.

So we get a partial answer to a question which at first glance looks strange : how is it that, up to now, we know nothing about the way memory works ? Indeed, if the brain analogue of a hard disk existed, that is a specialized place where data are stored in good order, with corresponding input-output devices, we would have found it long ago. But if memory designation is obtained by activating/deactivating certain circuits, that’s much harder to observe. Apparently, all circuits look identical, and the piece of information given by the fact that one circuit is active, rather than some other, can be seen only if one looks into that direction.

C. *Comparison with a known natural code*

In fact, only one code is known, at least partly : that’s DNA, which deals with the transmission of genetic patrimony. DNA’s code is simply made with four basic letters, and the code is made from the words one can build using these letters. Of course, one should not extrapolate from just one example, but let’s do it and argue this way : nature codes from

very simple basic elements, and the code is made from the complexity of all possible repetitions of these basic elements (conversely, to measure a signal with 10 decimals is a complex code, with no repetition : this is not the method used by nature). Dynamical Code, described earlier, fits very well in this scheme. Indeed, one has a very simple basic element (a path), and the code is obtained by juxtaposing the basic elements. Of course, here, the choice is not just between four possibilities : there are many more. But at each end point, there are only few possible choices, and there resides the analogy with DNA. Let's see in this analogy a little extra argument in favor of our hypothesis : it is of the same type, conceptually speaking, as what nature does elsewhere.

D. Operations on the code

If the basic coding element is the choice of a path, what are the operations that can be performed on such elements ? On DNA, the operation is to concatenate : put one element at the end of the other.

The same makes sense for paths : the “sum” is the path obtained by taking the first path, then the second.

There are obvious compatibility conditions : the second path should start where the first one finishes (that's more restrictive than for DNA, where any two words can be concatenated).

We also mention the possibility of parallelism, already seen by Von Neumann : the above evaluation of $4^{(10^{17})}$ leaves obviously considerable room for redundancy : one operation can be performed several times simultaneously, so as to reduce the errors (numerical errors, due to propagation of rounding errors, and measurement errors).

Finally, there is an obvious possibility for conditional switches (if something, do A, else do B). This is realized by means of a switch in one position, the influx is oriented towards one path, in the other, towards another path.

To summarize, we see two types of operations :

- Concatenate two paths : first pass through some zone, and then through some other zone. This corresponds to the information “and”. First to go through the zone corresponding with “circle” and then through the zone corresponding with “red” would mean “red circle”. Of course, the two features “red” and “circle” are not perceived at different times : our perception is global.

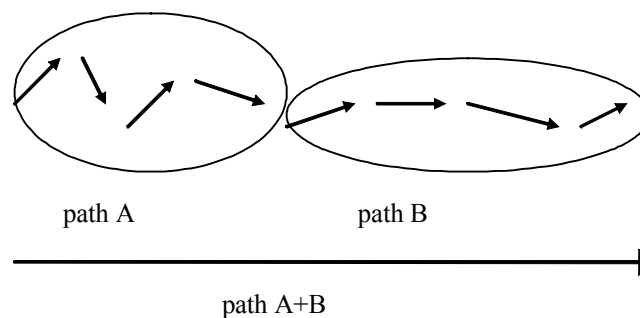


figure 6: Concatenation

- Conditional branching : orientation into one direction or another. This corresponds to an “if”.

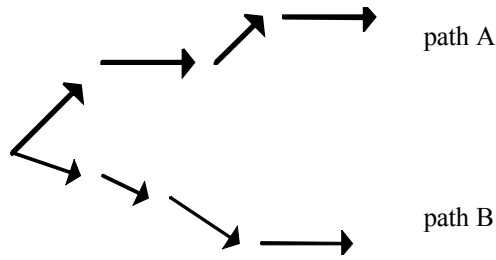


figure 7: Conditional Branching

Conversely, a logical procedure which requires to come back on acquired data (what we would call a recursive procedure) is not easily realized in such a scheme : one has to modify the whole information and code it again differently.

In the next section, we will see that the fact that some operations are easy and natural in this scheme, whereas some others are harder, has very important consequences in practice.

E. Possible experiments in order to validate the hypothesis

The Dynamical Coding Hypothesis may be submitted to experimental validation, the following way.

Let's take an animal having few neurons, and let's try to measure the activity on all connections between all neurons. Let's assume that the animal learns something, for instance the presence of an obstacle. Some circuits should be activated in a permanent manner (or perhaps deactivated, we do not know which one is the coding element : mathematically speaking, that's equivalent). If, later, the animal forgets about the obstacle, either because it disappeared or was moved, the initial circuitry should be modified, perhaps returning to its original condition. This is probably hard to observe experimentally, because one should observe all the potentials on all the connections, but it is theoretically feasible.

Some experiments realized on crayfish seem to indicate that modifications do occur in the paths followed by the influx, when the animal learns [M. Renault].

F. Practical consequences of the Dynamical Coding Hypothesis

1. The format of the input data

Let's assume here, without taking more precautions, that our hypothesis is correct : the coding element is the choice of a path, with the operations described above : concatenation (to put two paths one at the end of the other), parallelization, and orientation (switching, choice of a path among two or more). What might be the practical consequences ?

Let's also admit the following idea, at least temporarily : the closer data are (in terms of format) from what is used in the brain, the more easily they get assimilated and memorized.

For computers, this idea is wrong : if you give to a "C compiler" a sequence of instructions directly written in the machine language, the C compiler understands nothing. A "C compiler" wants to read instructions written in C, and it converts them into machine-understandable instructions.

But for the human brain, this idea is not absurd. Some elements make us think that there is a direct connection between what one sees and the way it is stocked in the brain (retinotopic coding, in particular).

Quite obviously some facts, some information is easier to remember : a face is easier to remember than a phone number. A phone number is easier to remember if it is written as a word than as a sequence of figures (in some countries, phone companies know this, in other they don't). So there are coding laws, and laws for understanding and memorizing. In our brain, our "C compiler" does not request a "C" language ; it can work more or less with many things, however it works better in some cases than in some others.

2. Levels of abstraction and of understanding

One often believes that what is best understood, best felt, best memorized is what is given to us by our five senses : blue sky, noise of cars, and so on. However, this is wrong. The human brain is naturally equipped with circuits allowing it a high degree of abstraction. Of course, education develops these circuits, but obviously nature has provided their existence. The sentence "this toy belongs to me" is meaningful for a child, as well as its adult equivalent "I own my flat" [B. Gaveau]. These sentences, however, represent a high degree of abstraction ; they do not refer to anything that can be handled by our five senses (it would be interesting as B. Gaveau points out, to know how such a notion is coded in the brain). Of course, they have some cultural content, but, for anyone who has the required cultural data, they offer no difficulty in understanding, in manipulating, nor in memorizing.

A recent book by S. Dehaene [Dehaene 97] asks several questions about understanding and memorizing arithmetical concepts. Very few people, indeed, can perform mentally a multiplication of the type 73×9 , though it is not "very" abstract : nine shepherds having each 73 sheep, how many sheep altogether ? This question looks more related to our five senses than that of ownership of a toy or of a flat.

So, the present authors claim that, contrarily to what people generally think, it is not the degree of abstraction that counts : the human being is perfectly equipped (after proper training) to manipulate highly sophisticated concepts. What counts is the coding of these concepts.

If the Dynamical Coding Hypothesis is correct, and if it is true that our understanding capabilities reflect the coding, what we like most is what makes the coding easiest : pieces of information put at the end one of the other, sequentially, with no backwards return.

Let us present this hypothesis under shortened form :

What the human being assimilates best is linear and non recursive logics : each information follows another, completing it, with no need to come back on a previous piece of information. There may be conditional branching : if something, do A, else do B.

A sentence of the type “I am the owner of my flat” obeys this rule : there is a subject, a verb, a complement and a complement of the complement.

Conversely, we do not like recursive formulas, which oblige us to come back on some previous data, to reevaluate the situation.

This explains quite well the difficulty, mentioned by S. Dehaene, that most people meet in arithmetical operations : the result is not obtained at once, one has to come back in order to compute some digits. It is the carry over which embarrass the children.

If, for example, we want to compute 73×9 , we perform $3 \times 9 = 27$ (which everyone does mentally), $7 \times 9 = 63$, and then you have to shift this and add the previous “2”, from which we get the result 657. This does not come from a linear logic, because (in our example), the box for tenths has been occupied twice : once by the 2 during the first elementary multiplication, and then by the 3 of the second multiplication, after shifting left. A non recursive procedure would allow to fill each box once and for all, using the previous ones.

This observation is substantiated by the choice of the base : to perform an addition in base 2 is perhaps ideal for a computer, but not for a human being, because there are many carry over. Conversely, the higher the base is, the less frequent the carry over will be.

The addition becomes “natural” if we present it the following way : the first quantity is represented by a rope of length A, the second by a rope of length B, put at the end of the first (as we concatenate paths), so the sum is the rope of length $A + B$.

So we see here –this meets the opinion given by Von Neumann and answers S. Dehaene’s question– that the difficulty does not come from the degree of abstraction of our mathematical concepts, but from the fact that the coding they use strongly differs from the one used by the brain.

The present study does not intend to look at the problem of presentation of mathematics: it plays only here an illustrative role, which is rather significant, anyhow, of one considers that mankind has been performing arithmetic operations for thousands of years.

The hypothesis we present here applies to any kind of concept. Let us repeat it in other terms : a concept will be better assimilated if it is presented in a linear manner (each “brick” having a link with the previous one and with the following one), may contain conditional statements, but will avoid coming back on previous bricks.

This can be applied to a situation where people have to obey orders. A single order “if this light gets on, you push this button” proceeds from linear logic, and this is well-understood.

However, if the situation becomes complex, with various lights, possibly going on at the same time, long sequences of orders linked to every possibility, one gets easily lost : this is because one has to re-evaluate the situation : what I did when this light was on does not apply when another light goes on. So, in such a case, it is better to explain the orders in more abstract terms (which, as we saw, will be perfectly understood), rather than try to list all possible occurrences and, in front of them, the corresponding behavior that has to be taken.

Let's take a simple example. In a nuclear plant, an order such that "if a fire occurs, call the firemen" will be perfectly understood and executed. However, a complete list of tasks to perform is the fire lights switch on is impossible to realize (a partial list exists, of course, and training is performed) because the fire may be connected with another event (earthquake, fall of a plane), the nearest firemen brigade may be busy somewhere else, and so on.

VI. Conclusion

We have tried, in the above pages, to present and substantiate an hypothesis dealing with the nature of the code used by the human brain. Our presentation is obviously extremely simplistic : between a principle and its application, the way it is put in practice, there is a large gap, to which we understand absolutely nothing. The principle itself might be absolutely wrong, in the sense that what nature does is not at all what we presented here. It can be partly wrong : Dynamical Coding really exists, but there are other phenomena, more or less important, perhaps more important than this coding itself.

The words "true" and "wrong" do not have much sense in this context. The hypotheses made by the scientists are only attempts of an explanation, with our words and within our culture, of natural phenomena which have no reason to reduce themselves to this explanation. It is the old "cavern myth" of Plato which gives the most appropriate view : we only describe the shades we see on the cavern's wall ; these shades are pale reflections of the outer world.

The hypothesis presented here seems to fit with experimental facts (at least those we know !), seems to continue in a satisfactory manner Von Neumann's observations, and provides a frame for reflexion (certainly not for conclusions !), about questions related to acquisition of knowledge, depending on the presentation and on the degree of abstraction.

References

Abeles, M. - de Ribaupierre, F., de Ribaupierre, Y., 1983 : Detection of single unit responses which are loosely time-locked to a stimulus. *IEEE Transactions on systems, man and cybernetics*, vol. SMC 13, no 5, pp. 683-691, sept-oct. 1983.

Abeles, M. - Gerstein, G., 1988 : Detecting spatiotemporal firing patterns among simultaneously recorded single neurons. *Journal of Neurophysiology*, vol. 60, no 3, sept. 1988, pp. 909-924.

Bialek, William - Rieke, Fred - de Ruyter, Rob - Warland, David, 1991 : Reading a Neural Code. *Science*, vol. 252, 28 June 1991, pp. 1854-1857.

Changeux, J. P., 1983 : *L'homme neuronal*. Paris, Ed. Fayard.

Changeux, J. P. - Connes, A. : *Matière à pensée*. Editions Odile Jacob.

Chapoutier, Georges, 1992 : *La biologie de la mémoire*. Que sais-je ? P.U.F.

Cherruault, Y : *Modélisation mathématique de la physiologie*. Ed. Médimat.

- Crow, T, 1988 : Cellular and Molecular analysis of associative learning and memory in *Hermisenda*. *Trends Neurosci.*, 11, 1988, pp. 136-142.
- Dehaene, S., 1997 : *La bosse des maths*. Editions Odile Jacob, Paris.
- Dudai, Y, 1989 : *The neurobiology of memory : Concepts, findings, trends*. Oxford University Press.
- Dupuy, Jean-Pierre, 1994 : *Aux origines des sciences cognitives*. Ed. de la Découverte, 1994.
- Fujita, Ichiro - Tanaka, Keiji - Ito, Minami - Cheng, Kang, 1992 : Columns for visual features of objects in monkey inferotemporal cortex. *Nature*, vol. 360, 26 nov. 1992, pp. 343-346.
- Geoffrois, Edouard - Edeline, Jean-Marc - Vibert, Jean-François, 1994 : Learning by delay modifications. *Computation in Neurons and Neural Systems*, Franck H. Eeckman editor, Kluwer Academic Press, pp. 133-138.
- Hopfield, J. J., 1995 : Pattern recognition computation using action potential timing for stimulus representation. *Nature*, vol 376, 6 July 1995, pp. 33-36.
- Konishi, Masakazu, 1991 : Deciphering the Brain's Codes. *Neural Computation*, 3, 1991, pp. 1-18.
- Rieke, Fred -Warland, David - de Ruyter van Steveninck, Rob - Bialek, William, 1997 : *Spikes, exploring the neural code*. MIT Press.
- Rind, F. Claire, 1997 : *Neural Circuits for Collision Avoidance*. Rapport INRIA, pp. 22-31.
- Sejnowski, Terrence, 1995 : Time for a new neural code ? *Nature*, vol 376, 6 July 1995, pp. 21-22.
- Squire, L. R., 1987 : *Memory and brain*. Oxford University Press.
- Stryker, Michael, 1992 : Elements of visual perception. *Nature*, vol. 360, 26 Nov. 1992, pp. 301-302.
- Thorpe, Simon, 1990 : Spike arrival times : a highly efficient coding scheme for neural networks. *Parallel Processing in Neural Systems and Computers*. Elsevier, 1990, p. 91-94.
- Véron, Guy : *Psychologie de l'action*. Ed. Vigot.
- Von Neumann, John, 1957 : *The Computer and the Brain*. Yale University Press.