AFSCET

Res-Systemica

Revue Française de Systémique

Fondée par Evelyne Andreewsky

Volume 23, printemps 2022

Systémique quantique

Res-Systemica, volume 23, article 06

Faiblesse d'une intelligence artificielle vis à vis des bruits environnementaux en compatibilité électromagnétique

Olivier Maurice, Anne Louis, Alain Reineix

10 pages

contribution reçue le 09 avril 2023

AI weakness regarding one noisy environments and the EMC points of view - *faiblesse d'une IA vis à vis des bruits environnementaux en CEM*

Olivier MAURICE[1], Anne LOUIS[2], Alain REINEIX[3].
[1]AFSCET, Paris. olivier.maurice@e-nautia.com,
[2]CESI, Rouen. alouis@cesi.fr [3]Xlim, Limoges. alain.reineix@xlim.fr

April 9, 2023

Abstract

Artificial Intelligence includes both software and hardware sides. Clearly, the software part constitutes the AI fundamentals, but implicitly, it supposes that the hardware part is not disturbed. If the datas are facked due to electromagnetic interferences, the AI function may be drastically disturbed, making its analysis empty of meanings. The purpose of this article is to explore the impact of various data modifications during the AI working steps. The database itself can be changed, but also parameters values. We try looking to the impact of each kind of disturbance: on the datas or on the parameters, for submit some tracks of works more detailed on some practical cases. We give an example where the AI process is associated with a game involving persons.

L'intelligence artificielle comporte un aspect à la fois hardware et logiciel. Clairement, la partie logicielle porte les fondamentaux de l'IA mais implicitement, son bon fonctionnement suppose que la partie hardware n'est pas perturbée. Si les données sont falsifiées par une interférence électromagnétique, la fonction IA peut être drastiquement perturbée, rendant ses analyses hors propos. L'objet de cet article est d'explorer les impacts de diverses modifications de données dans le fonctionnement d'une IA. La base de données elle-même peut être changée, mais aussi divers paramètres de réglages. Nous essayons d'évaluer l'impact de chaque type de perturbation : sur les données, sur les paramètres, pour suggérer des pistes de travaux de façon à modéliser plus profondément ces effets. Nous donnons un exemple illustratif où le processus IA est associé à un jeu avec des personnes.

1 Characteristics and cascade classificator

We suppose that an object P can be present in an image O. For the image O made of a set of symbols s and for a set of canonical characteristics c, we look in a first step if a subset c' of c exists in s(O). If s doesn't include c', the research of P in O is stopped. Therefore, the AI lost of functionality depends on the difference between the subset $\tilde{c}'$ disturbed and the original

c' subset. The process robustness is linked to the distance between $\tilde{c}'$ and c'. The question is: how evaluating this distance for in final becoming unable to identify O?

It depends of both the typology of the characteristic and the mathematical operation used in the $\tilde{c}'$ into s inclusion determination.

If ψ is an application associating a vector of numbers to each object O, c, c', s, we can accept a distance D definition based on two vectors difference term by term structured by a metric g as:

$$D = \sqrt{\sum_{ij} g_{ij}(c'_i - \tilde{c}'_i)(c'_j - \tilde{c}'_j)} \tag{1}$$

If $dim(c') = n$ with $dim(s) = \alpha n \cdot dim(c')$, then we compute at step q:

$$R_q(m) = \{s\}[m,\, m + dim(\{s\})/n] \oplus {c'}_q \tag{2}$$

$R_q(m)$ being the identification function saying if the pattern c' belongs to the image s at step q. Now, considering the effect of disturbing the characteristic, the operation becomes:

$$R_q(m) = \{s\}[m,\, m + n] \oplus \tilde{c}'_q \tag{3}$$

with $D > \epsilon$. ϵ is the limit difference acceptable over which the characteristic is no more a pertinent signature for recognizing the image s.

By defining g and ϵ, this allows to compute the influence of a possible changing in c. It's clear that the corruption concerns firstly the characteristics that are fixed for a given AI, while the dataset varies. So disturbing the dataset can only provoke a limited loss of function in time, contrarily to the characteristics which involve a permanent failure.

2 Neighborhoods

Once an image $\{s\}[m,\, m + dim(\{s\})/n]$ is detected, a scale reduction f is often applied to confirm the detection:

$$f(\beta,\, \alpha) : \{s\}[m,\, m + dim(\{s\})/n] \rightarrow \{s\}[m + \beta,\, m + \beta + \frac{dim(\{s\})}{(\alpha n)}] \tag{4}$$

If the corrupted digits belongs to the set of symbols of the downscaled characteristic, the scale reduction will propagate the degraded identification process. This can affect the neighborhood detection performance, hense the detection one in final.

How the digits of the characteristic can be changed through some electromagnetic interaction? We make the assumption that the characteristics are memorized in a fixed subpart of a memory, while the dataset belong to a dynamic memory. The question becomes: how this fixe memory can be modified by an electromagnetic interaction?

The memory can be changed by heavy ions or other ionizing radiations. But it's not our purpose to consider these effects. If we look only at radiofrequency interferences, this means that some interaction occurs when the characteristic is being stored. In practice this kind of event is very rare. One mechanism that can lead to this result goes through a non linear detection of very high frequency interaction.

This effect can be mathematically modeled by the function: $f(u) = \beta\,(\alpha s(u) + V_{DC})$. β is called the detection efficiency, α the enveloppe coefficient and V_{DC} which is a continuous signal having half $\alpha s(u)$ value.

When the disturbing signal comes in interaction with the target, the added level expressed by $f(u)$ can change a digit value, and by there change a memorized value like a characteristic. Nevertheless, the probability of occurrence between the electromagnetic disturbance and the memory flux is very low.

We may think that a similar probability anyway exists concerning datas. But if one data is modified in the whole collection of datas used by AI, we understand that the consequences in general won't be important for the AI performance.

3 Machine learning

When classification of dataset can be organized based on known datas: this is supervised classification in link with machine learning. We take a look to the classificator of the nearest neighboors. As classically we transform the matrix of values associated with the information into a column vector.

This kind of operation is not so easy to define. The components of the target vector are obtained using special connectivities. We define for 2x2 matrices:

$$\gamma_{1,2} = \begin{bmatrix} 1 \\ \\ 0 \end{bmatrix} \quad \gamma_{3,4} = \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix} \tag{5}$$

If V is the target vector and M the information matrix, we compute:

$$V_k, k \in \{i, j\}\ V_k = M\gamma_{i,j} \tag{6}$$

Once the target vector is constructed, once more the problem consists in a distance calculation between a reference data characteristic of one category and the studied data. Generalizing the problem, the question is to define the limit over which the modified vector cannot be more seen as a neighbor or a category component. Looking at our previous reasoning, we can try to find a processus giving tracks for explaining the case where the modified vector getout of its identification.

Taking the target vector as a reference, we can compute the distance of any solution, including neighbor ones. One distance d_1 can be defined using:

$$d_1(I_1,\, I_2) = \sum_p |I_1^p - I_2^p| \tag{7}$$

I_1 and I_2 beeing the target vector and the tested one. Another distance definition d_2 can be used with:

$$d_2(I_1,\, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2} \tag{8}$$

We can consider that the disturbance impacts one component only in the vector. Depending on the distance value, the modification can make a false candidate to become true, or that a true candidate becomes rejected.

If we take a look to the first assumption, it means that a vector that would have been rejected is then accepted, consecutively to its modification.

For transposing the problem in tne numerical space, we transform the information vector in a numerical one. In this format, the vector becomes a list of 0 and 1. The distance results from a summation of 0 or 1. This

time, under the kept assumption that the disturbance changes only one component, it means that one of the zeros or ones is changed into a one or zero respectively. For a rejected vector, more than one component is different from the target vector definition. But it can be sufficient that only one component is changed to reduce enough the distance and make the rejected vector acceptable.

Various cases can exist having this same configuration. If $\{V_1\}$ is the set of rejected vector due to strictly one component fault, the set $\{\tilde{V}_1\}$ groups accepted vector coming from the set $\{V_1\}$ where one incrrimined component is changed into a good value.

Finally, following this discussion, the set $\{V_1\}$ constitutes the set of possible weakness of the AI, it means that knowing $\{V_1\}$, we know how many errors can possibly occur at the maximum, for one component modification per vector.

3.1 Classification of the K nearest neighbors

One technique that appears more robust in front of the possible disturbance we have explore, is the classification of the K nearest neighbors. When comparing the data vector with the target, more than one candidate is retains. Then, by vote for identifying the dataset with one category, we classify the datas with this category. If the vote is wide enough, the vote process encloses the modified vector and may associate it with a bad category, but keeps the other datas in the good one. From the majority point of view, the treatment is safe. How can we write this processus mathematically?

We imagine a set of input vectors $\{V_k \dots \tilde{V}_m\}$ embedding one false vector. The target categories are C_n. From the set and the target we can compute the k distances d_2^{kn} between each vector of the set $\{V_k \dots \tilde{V}_m\}$ and a target category. Following the distances computation, we can establish a result vector made of a code number depending on the fact that a vector belongs or not to one target category. If a vector doesn't belong to any category, the code number is zero. As the decision is taken from a vote to the majority, two situations can exist:

- the fake vector can change the decision only if it increases the number of vector associated with one category ;
- whatever the fake vector exists or not, it cannot change the decision.

Let's study some situation to understand how it works. We can imagine first three objects: one belongs to the category 1, another to the category 2 and the third is the fake object. If the modified component makes the fake object as an element of the category 1 or 2, the majority vote can change of category. Now if we have four objects: one to category 2, two for the category 1 and one fake object. The transformation of the fake vector can either confirm the vote to choose category 1, either making the vote undetermined. Through these two simple examples, we understand that the fake object can change the decision simply by increasing the weight for one category.

If R is the result vector, made of components c_k which are the code numbers: as the vote chooses the maximum similar codes for identifying the category, the state of R under disturbances $\tilde{R}$ indicates what will be the consequence of this disturbance. The state vector $|c_1, c_2, c_1, \dots, c_3\rangle$ points out one category after having been changed by a disturbance. We can study the constitution of R, which gives all the possibilities of the

disturbance impacts. This avoid to detail the distances changes but we must remember that R components comes from modifications on datas that reduce or increase some distances.

4 On the conical projection

We can project the set of N code numbers as N graduations on a perimeter, the length of each code axis being proportional to the number of datas associated with each code by the AI. A vertical axis, perpendicular to the graduated disk plan, shows the evolution of the data projection on the disk during time. Time passing, the AI must select some solutions and reduces the possible axises as categories for the datas.

At the begining, the projections make a bent 2D curve. This kind of surface exists at each time step but the perimeter reduce with time if the AI converge to the solution. Finally the surface is reduced to one point, pointing out the reteined category. The whole form created by the marchine learning process can be assimilated with a manifold. Studying the manifold amounts to study the AI identification process. When the process converges, it takes the form of a leprechaun hat.

Looking to the manifold form gives immediately information on how the identification process evolves. The base pace is directly in relation with the number of categories. Its 2D curve shows the correspondance between the data collection and the set of possible categories. Using this property, we can list the various possibilities of disturbance consequancies.

One general rule can be written: usually, a functional AI creates a sort of leprechaun hat following the converging evolve of the AI into a single point. Significative disturbances will change the form presented without the disturbance. The fact that the disturbance is significative means that at least the final point change of location.

Typically, a falsified data changes at a time step, the category surface form. A change in the category codes affects the manifold at all times. These two major differences in the manifold aspect give a robust method to detect what kind of disturbance occurs. In this description we make the assumption that the training phase is finished. If the training phase is disturbed, it has a major consequence as the category base is modified. The next steps and manifold construction doesn't present abnormal functions, but the results will be meaningless.

5 Mathematical translation

We define a set of category codes $\{a, b, c, \ldots\}$. Each code is associated with one axis $\vec{c_i}$. For representing this space we distribute the axises projecting them as a circle graduation. It constitutes the manifold base reference. The base being numbered the components of one information vector say how many time each category is pointed out in a dataset. Various process can lead to such a projection depending on the dataset type. We can take as an example the dataset coming from an image.

For example we consider a lego play where the player (the artificial intelligence) must construct a coffee maker. The input vector containing all the lego pieces coded with a number. each component refers to the input natural space axis and is our input category. A neuronal network following a succession of images as instructions, selects one piece in the

collection in order to construct the coffee maker. The input vector is defined at time 0 by:

$$V_0 = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \end{bmatrix} \tag{9}$$

Each p_i value is associated with an angle in the circular projection of V_0 through the map $\phi : V_0 \to \theta_0$. The AI chosing one piece realize the operation $\gamma_i V_0$, γ_i being the chosing operator: a matrix made of only one component γ_{ii} equal to 1 in diagonal (all the other components being null) which selects the piece i in $V_0(i)$. By the fact, the new input vector $V_0(i+1)$ having one piece less is tranformed through $(1-\gamma_i)\, V_0(i)$. If the AI makes a bad choice, the operation is symbolized by $\tilde{\gamma}_i V_0(i)$. In that case, the AI can correct its choice after one shot making $V_0(i+1) = (1 + l_i C(i))\, V_0(i)$, $C(i)$ being the coffee maker. This correction transforms the solution (the coffee maker) becomes $C(i) = (1 - l_i)\, C(i-1)$.

All this can be synthesized:

1. the coffee maker construction is $C(i) = \gamma_i V_0(i)$ and the collection becomes $V_0(i+1) = (1-\gamma_i)\, V_0(i)$;
2. a bad choice is noticed $\tilde{\gamma}_i V_0(i)$;
3. a correction means to retire a piece from the coffee maker: $C(i+1) = (1 - l_i)\, C(i)$ and replace it in the collection: $V_0(i+1) = V_0(i)\, (1 + l_i)$;
4. a lengthening of the time to play for one choice is $\tau \to \tilde{\tau}$.

All player actions are symbolized by a couple $(\gamma_i,\, \tau)$. From these relations we can model the play using a payoff matrix where appear the nominal or exceptional noise and the focused or disturbed player. This payoff matrix is defined for each play time. We retrieve in it the four possible couples of actions and times:

nominal noise	amplified noise	time i
$(\gamma_i,\, \tau_i)$	$(\gamma_i,\, \tilde{\tau}_i)$	focused player
$(\tilde{\gamma}_i,\, \tau_i)$	$(\tilde{\gamma}_i,\, \tilde{\tau}_i)$	disturbed player

The player is the AI and we can now describe how it choose the pieces. The lego image are in black and white. In a first step we develop the succession of pixels in a single image vector d^k. If the white pixels are coded with a 0 and the black ones with a 1, and if each piece has a unique black pixel number, the operation $d^k d_k$ gives this number. The AI knows the sequence of construction of the coffee maker. So at each time step it will explore all the pieces and stop once it detects the good black pixel number on one sample. The AI has learn in a previous learning step what is the good sequence of number to construct the coffee maker. Afterwhat it just have to retrieve this sequence in the pieces selection. This simple AI is perfect for our illustration. Let's write all this mathematically.

The image is a matrix $I_{aa} \in \{0,\, 1\}_{n \times n}$. From this image we make the image vector $J \in \{0,\, 1\}^N$, $N = n^2$ by:

$$\begin{cases} (i,k) \in [1,\ldots,n] \times [3],\, q \in [1,\ldots,n \times n]\,,\; I_{i3} \to J_{q=i*k} \\ \\ \Rightarrow I_{i2} \to J_{q=i*k-1}, \quad \Rightarrow I_{i1} \to J_{q=i*k-2} \end{cases} \tag{10}$$

Starting from the vector J we then apply an operator g having in each line the signature of each piece. This is a matrix made of N columns and M lines if the coffee maker groups M lego. Each line of g is a neuron.

Making the product $g_u^k J_k$ we obtain a vector where each line is the correlation of a piece signature g^k with the unknown input vector J. The output S of this first stage neural network gives for each neuron a binary number equal to the correlation product value. If $\phi_{10,2}$ is the map giving the binary correspondence of any decimal value: $\phi_{10,2} : g_q^k J_k \rightarrow S_q$. A second neural network stage makes the identification of all the components of S_q with the piece numbering, each neuron taking in charge one piece number. If N_q is the number attached with each neuron, their outputs R_q are obtained by making $R_q = \overline{S_q \oplus N_q}$. Only one neuron will have its output equal to 1 after multiplying all the components of an R_q. This identity points out the number of the chosen lego. Figure 1 shows the neuronal network structure.

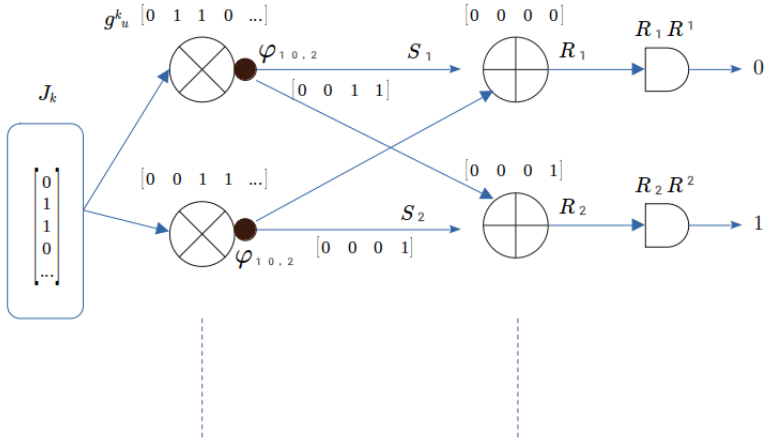

Figure 1: Neuronal network structure

6 Modeling the described AI disturbance

Having described completely a simple AI we can now explore our previous listed disturbance mechanisms by applying them to it. After what we can explore the modelization of real intelligence. AI can be used as a simple way for modeling the basic mechanism of any living been. But another step beyond needs to be implemented for taking into account the minding that leads to a choice, far from the simple neuronal network used for identifying the piece. That's the object of the game theory layer added to the AI action.

6.1 Fake images in the source database

Having fake images means that the input vector is disturbed. The disturbance can lead to the result that none of the correlation $S_u = g_u^k \tilde{J}_k$ reaches a maximum. It means that the decisions R_ν is undecided and no clear identification get out the NN process.

In the lego exercise, this was simulated by the presence of a lego piece that had a small difference with the figure in the mounting instructions. The impact was a longer time delay to identify the concerned piece. Or it

may have been a longer delay to identify the fact that the searched piece was not present. It appears like a noise and the corresponding operator couple is (γ, $\tilde{\tau}$) for a focused player. In case of a disturbed player, the time is extended and in more, he chooses a bad piece in place of the good one. Operators become ($\tilde{\gamma}$, $\tilde{\tau}$).

The time extension cannot be understood through the undecided result from J_k to $R_u R^u$. The NN takes the same time for processing whatever the result. It comes from the added layer of interpreting the NN result by the decision organ. For a focused player, to choice one piece between others having the same NNoutput score calls for other criteria (remaining pieces, geometry similarity, ...) which take more time than if the NN output has no ambiguity.

What we call a "disturbed player" can be also a player who tries some combination, so makes a process $\tilde{\gamma} \to l \to \gamma$ before to success. This show that the idea saying that a trial error process can be better than a deeper leading to a direct success has no meaning. The trial error approach reflects ignorance on the theory which allows to predict the error.

6.2 Errors in the machine learning process

Having errors in the machine learning process means that the operator g_u^k is falsified. In our lego exercise, this means that the instructions are not corrects somewhere. A first possibility leads to the same result as previously, i.e. a time increase for reaching one decision $\tau \to \tilde{\tau}$. Another possibility is that g_u^k provokes a bad identification on the output $R_u R^u$, and in that case, the player choice will be always false, even in a nominal time ($\tilde{\gamma}$, τ).

This can be observed also for a focused player and in this case, it becomes a disturbed player.

To simplify this simulation, we add a piece in the instructions that doesn't exists in the collection. For this configuration, the sequence $\tilde{\gamma} \to l \to \gamma$ is impossible for the focused player, but at least it provokes a time extension to understand that the required piece is not needed.

6.3 Manifold representation: from the NN to (γ, τ)

The AI outputs can be studied finally through the use of a graph where we symbolize the operations done and associated with a system of equations previously involving γ, etc. The couple of the graph representation and its equations constitute a topology written $T(G, M)$. G is a geometry here projected on a graph and M a manifold made of the system of equations. In place of a graph we can also use any representation showing the observables on a drawing. Figures 2 and 3 give an example applied on the coffee maker construction process in both case without and with an error. The advantage of such a representation is to synthesize very efficiently the game trajectory. Rather than studying the game through the outputs values, it is possible to study it looking to its drawing.

An AI can very fastly make an analysis of this trajectory under this form and in some application, this can give access for fast decision in real time. In this spirit we may imagine a screen showing the pieces used for the coffeee maker. Immediately you may react if an error occurs while showing the sequence of piece numbering would ask you more effort for the same reaction. The representation chosen figure 2 gives immediately informations on γ and τ. For studying big data grouping thousands of

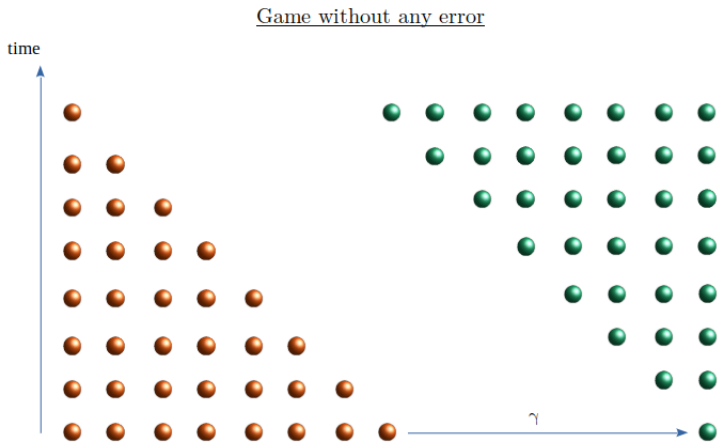

Figure 2: Game manifold representation

similar games, the image synthesis is probably one of the more efficient way. But the best technique for justifying the method of image construction goes through a topological approach identifying the geometry and the manifold involved.

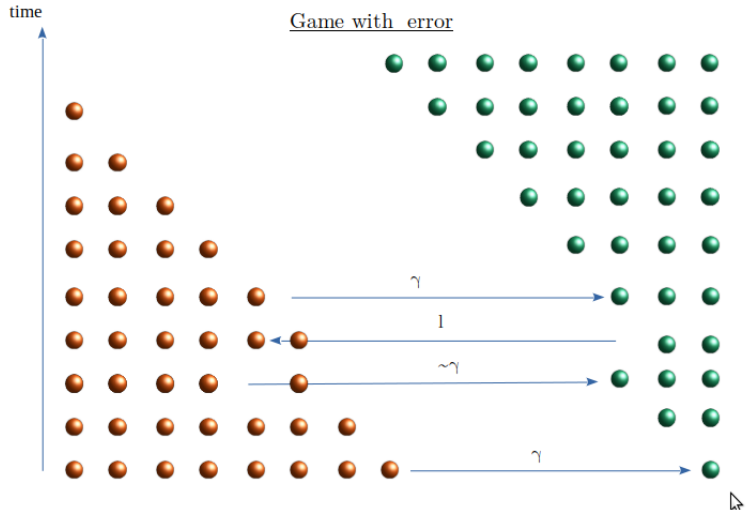

Figure 3: Uncorrect game manifold representation

7 Game theory, AI and EMC

If we want to generalize the AI modeling, we can consider a manifold of equation $e_\nu = \zeta_{\nu\mu} k^\mu$ where e_ν is the generalized source covector; k^μ the natural space vector link with the system dynamics and $\zeta_{\nu\mu}$ a metric

wearing the correspondencies between the system dynamics and its scalar measurements e_ν. This whole description can be realized using Laplace's formalism. It is framed by a temporal sequence of events changing the source: $e_\nu(t+dt) = \gamma^\nu_\nu e_\nu$. And in this sequence, once the dynamic is solved (i.e. the speed k^μ) a game theory allows to model the step between some outputs AI observables and the decision. This added layer of decision can also be present at the end of the temporal loop.

In this global schematic, the EMC disturbance, or in fact any noisy disturbance, can impact e, k, γ or the game theory (GT) layer. The whole processus can be represented by a set including a topology $\mathcal{T}$ and a game theory GT: $\{\mathcal{T},\, GT\} \rightarrow \{(G,\, M)\,,\, GT\}$.

The game theory is principally modeled by the payoff matrix and the associated analysis. The payoff matrix has for components in each players (of any kind, including nature, contexts, etc.) combination a couple $(\gamma\, \tau)$.

8 Experiment

We realize an experiment to give some material to our arguments. The play is similar to the one exposed previously. The gamer must realize a little coffee machine in lego. It makes it in two periods: in a first period the gamer learns how to make the reduced model reading instructions step by step. In a second period, he makes again the same process of mounting, but this time various sounds are played while the gamer reads the instructions. The sounds is the noise that can disturb the gamer. Knowing that the gamer has already mounted the coffee machine one time before in a smarter environment, he should go faster during the second period.

Seven gamers make the test. Some of them were lego users, others have never play with lego. For adding some difficulty, there was an error in the instructions, speaking of a piece that doesn't exist.

Globally the gamers, whatever their competencies, have lost between 70 and 300 seconds due to the sounds influence. This simple experiment already shows that the attention can be disturbed by sounds, even for rigorous persons. Probably our far conditions for survive got use to us for listening any strange noise appearing in any situations. Our attention being partly disturbed by this inconscious surveyor, we loose time for making the first activity faster. We don't have the opportunity to extedn the game on many players. applying the sutdy on seven persons would not be serious if we may want to compare the results with some modelisation. For this reason and as we won't have the facilities for going further in this experiment we stop here the comparison. Anyway, this small experiment has given the opportunity to develop the corresponding model and to make the correspondance between IA process and brain simple reasonings.