USE OF NEURAL NETWORKS IN THE BUSINESS PROCESS MODELING

Marian Sorin IONESCU [a*]

[a] *Politehnica University of Bucharest, Romania*

ABSTRACT

The emergence of Artificial Intelligence concepts and paradigms, of intelligent digital machines, able to elaborate the decision-making process without the interference of the human operator, modifies particularly fast, transparent and complex business models used by almost entirely of economic organizations.
Neural Artificial Networks are models for the implementation mathematical algorithms built mimetically according to the neural structure specific to the human brain.
The approach of business phenomenology as a systemic, integrated structure, based on operational and strategic entities that are based on the notion of knowledge are the fundamental priorities in the issue of elaborating the highest added value.
A neural network is not a mandatory inductive towards a classical cognitive process, using classical „if-then" computer structures, this should rather be perceived as an expert system.
Through a mimetic system, the aspects of information processing and physical structure of the human brain are simulated, the classification is made for a miniature, microscopic or even nano-scopic system for a „white-box" model or a macroscopic expert system in the „black-box" model.
The tendency to studying, understanding and shaping the economic activities with neural science provides effective solutions to economic actors, actors in a globalized business macro-environment with a high degree of complexity and extremely dynamic.
Numerous economic organizations have invested in the identification and development at the level of R & D (Reasearch and Development) departments of the business solving problems which they are confronted with, traditionally solved with the help of operational research, through algorithms and methods specific to neural networks and data exploration (data mining).
This paper presents, positions and offers solutions to the use of neural networks in the issue of modern business models.

KEYWORDS: *Artificial Intelligence, Black-Box, Business Phenomenology, Learning Processes, Neural Artificial Networks, White-Box.*

1. INTRODUCTION

The modern, highly globalized business environment, with its implicit characteristics, the concepts and the paradigms of the Artificial Intelligence, in whose field of scientific expertise we also find the neural networks, offers a broad and fertile operational field to this new type of approach.
We identify and nominate the following areas of business models, suitable to improve the performances generated by the implementation and the modeling with the help of neural networks:

- Sales force: Neural networks are recognized for flexibility and proactive in case of organizational punctual operations such as „sales force", sales prediction is fundamental to inventory decisions, number and professional training of the personnel, setting price policies (the price of the product), the consolidation of the existing client portfolio, its quantitative

* Corresponding author. E-mail address: marian.ionescu@man.ase.ro

and value extrapolation are also two trends in which the approach through neural concepts fully justifies its presence;

- Financial Management: electronic signature, bank identity, credit identification, exchange rate currency forecasts, risk rating, initial stock-market offers forecast, capital market, bankruptcy economic organization prognosis, scoring client financial-banking system, credit card limit, debt and credit, choice of investment capital market structure, diversity of financial instruments, financial economic forecasts at the level of economic organization, business macro- environment, bond ratings, credit policy, complex economic and financial forecasts, financial risk management;
- Logistics: risk management of distribution channels, risk management strategies for the expansion of transport capacity, storage of wares;
- Human resources: management of performance and human behavior strategies at the individual level, sizing of personnel structure allocation at economic organization level;
- I.T Department: managing existing hardware and software resources, prognoses (forecasting) correlation department, sustainable organizational economic growth;
- Accounting: identification of financial frauds, fiscal, conducting audit procedures, correct interpretation to the top organizational management of the elaborated results;
- Marketing: identifying end-user behavioral trends, analyzing policies and strategies for promoting innovative products, improving sales, segmentation and targeting the specialized market where the economic organization is operationally present and developing consolidation and development strategies.

A neural network is descriptive using a graph type structure in whose nodes we find neurons, the sides-to-edges making the connection between them, and within this type of network a defined set of processors that define a network of interconnections with a high degree density.

At the neural entity level, a weighted amount of information is perceived on their input positions, supply is made with the help of sides, edges.

The interest of the scientific study at this point is for the "feed-forward" networks, therefore the supply interconnections for the basic graph, do not contain cycles.

One of the innovative ideas brought about by the Artificial Intelligence paradigms for the business models used is the idea of "learning".

In this sense, a set of strategic operational hypotheses is elaborated, predictions for the neural network, their totality shares the same graphical network structure, the difference is made by the load (weight) on the edges.

For a set of n variables we introduce a "predictor" implementable in T(n), similar to a "neural predictor" (for a neural network of defined length), with the length of O(T(n)^2), the cardinal of the nodes set represents the size of the network.

In the case of neural networks of polynomial dimensionalities, the actual processes and learning tasks are dependent on the use of certain types of selected predictors, subsequent operationally implemented, the degree of complexity of the learning sample, under accepted assumptions is bordered on the dimensionality of the network.

By addressing strategically developed business models and subsequently operationally transposed by economic organizations, we assert that identifying the sets of assumptions under which we develop the mathematical reasoning presented as procedures and algorithms by the predictors of neural networks are very difficult to quantify.

We introduce a particularly useful mathematical concept, the "Gradient Stochastic Descendent", defined as a descending incremental gradient process, an iterative method for optimizing a differential objective function, used in stochastic approximation methods.

G.S.D. has a wide range of uses in the processes of neural network development with operability in modern economic issues.

Neural networks are subjected to some learning processes adapted to the intermediate and final objectives pursued by the studied economic organization.

2. LEARNING PROCESSES FOR NEURAL NETWORKS

We consider a neural network as (V, E, σ, w) subsequently results the function $h_{V,E,\sigma,w}: R^{|V_0|} \rightarrow R^{|V_T|}$, any set of these types of functions are used in the learning processes, we introduce a set of assumptions for the predictor network, fixing a graph (V, E) used to activate the function σ, making the assumptions for the sets as having the totality of the mathematical form $h_{V,E,\sigma,w}$ in specialized scientific literature is called the network architecture, the set (class) of hypotheses is

$$H_{V,E,\sigma} = \{h_{V,E,\sigma,w}: w \text{ is cartography} - \text{gradient from } E \text{ to } R\} \quad (1)$$

An aspect to which special attention must be paid is the process of approximating the induced error, its estimation, its possible optimization, under the set of selected hypotheses.
The complexity of the development of neural networks offers alternatives that many neurons are connected by communication of information links, the formal classical presentation is of a visual-graphic entity, in the nodes of the network neurons are located, the edges (margins) are from the exit of a neuron, to the entry into another neuron, the present study addresses the graphic representation which does not contain cycles.

2.1 Neural network feed-forward in multiple layers

A neural network of the feed-forward type admits the mathematical representation of an acyclic graph (without cycles), $G = (V, E)$ the weight function positioned over the margins, $w: E \rightarrow R$, in the nodes of the graph are located neurons, each neuron is represented as a scalar function $\sigma: R \rightarrow R$, at the level of applications for economic models, we identify three possibilities for the functions σ: sign function, $\sigma(a) = sign(a)$, threshold function, $\sigma(a) = 1_{[a>0]}$, the sigmoid function, a smooth approximation of the threshold function, $\sigma(a) = \frac{1}{1+\exp(-a)}$, this function σ is called "neural activation function", we notice that the totality of neurons within a network are connected, in the system "output" – "input", with the help of the margins (sides) of the graph, the input to a neuron is obtained as the sum of the output weights of the total neurons connected to it, the weights are determined in accordance with w.
The organization of the neural network is structured in "layers", the set of nodes is decomposable in a union of disjoint subsets, $\cup_{t=0}^{T} V_t$, so that any edge E connects the node V_{t-1} the node V_t for $t \in [T]$, V_0 is called the input layer, we identify $(n+1)$ neurons, where n is the dimensionality of the inputs space for which the scientific study is performed, $\forall i \in [n]$, the neural exit i in V_0 is x_i, the last neuron as a position in V_0 is a neuron with the constant output value, equal to 1, noting the $v_{t,i}$ neuron in the t layer, let $o_{t,i}(x)$ the output of $v_{t,i}$ with the network that includes the entry in vector x..
We choose $a_{t+1,j}(x)$ the output of $v_{t+1,j}$ when in the network is received the input into the x vector, results:

$$a_{t+1,j}(x) = \sum_{r:(v_{t,r},v_{t+1,j})\in E} w\left((v_{t,r}, v_{t+1,j})\right) o_{t,r}(x)$$
$$o_{t+1,j}(x) = \sigma(a_{t+1,j}(x)) \quad (2)$$

The input towards $v_{t+1,j}$ represents the sum of the outputs weights of the neurons from V_t connected to $v_{t+1,j}$, the $V_1, \dots, V_{T-1}$ layers within the neural network adapted for an economic organization are sometimes also known as "hidden layers", V_T located in the top of the layers is also called the exit layer, in the predictive economic strategy problems, the output layer contains a single neuron whose output coincides with the exit from the neural network.

We designate T as the number of layers in the network, except for V_0, or the "depth" of the network, the size of the network is expressed with the help of the value $|V|$, "width", the scale of the neural network is defined by $max_t|V_t|$

2.2 Neural network, Hopfield model

The approach of business organizational models with the help of Hopfield's neural network models, implies the lack of a stratified architecture, the weights of specific neural entities remaining unchanged, the Hopfield networks are totally interconnected for those n existing neurons, the weight of the W_{ij} network is fixed and symmetrical, $w_{ij} = w_{ji}$, the storage of information at the memory level provides stability to this type of neural network, each neuron has a x_i state, value margin of $0,1$, the neurons are completely in agreement with a differential equation, the energy function over time is minimized.

Hopfield networks are particularly important for modern business models because, these, looking for the performance of economy, of the generation of added value in significant quantities, difficult to imitate by competition, need conceptually close methods and paradigm of classical operational research for optimizing system processes.

Hopfield & Tank have demonstrated that weights in a Hopfield neural network are selected such that updating neural processes minimizes Hopfield energy function and optimizes the identified problem, each neuron i is updated according to the differential equation:

$$dnet_i/dt = -\frac{net_i}{\tau} + \sum_{j=1}^{n} W_{ij}x_j + I_i, x_i = f(net_i) \quad (3)$$

where $f(.)$ is a sigmoidal output function margined between $0,1$ and τ is constant.

The previous equation is equivalent to calculating the output of a neuron in the exception of "Multiple Feedforward Neural Networks" ("M.F.N.N.s"), where a constant term I_i is added to the network inputs for each neuron, the dynamic times are now continuous, the systemic process is assimilable to a Euler-discrete type approximation, each time a neuron is updated in this manner, the energy function is:

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}x_ix_j - \sum_{i=1}^{N} I_ix_i \quad (4)$$

We define the energy function as a Liapunov function for the system, there is a guarantee that it does not increase, the reasoning developed based on the fact that the updating process of the neurons diminishes the value of the energy function, analog process to the operationality of the rules of updating the neural weightings, within the M.F.N.N. approach with retroactive propagation ("back propagation"), induce a significant, most abrupt decrease in the error function.

The methodology of optimization problem solving, using Hopfield neural network models requires the selection of the weights W_{ij} and the constant I_i terms in accordance with the target that the force of the energy function and the optimization of the objective function are equivalent.

The optimization problem is expressed as a single function which must be minimized, incorporating the totality of the costs and constraints to which it is subjected.

Using modeling with the help of a penalty function, weights W_{ij} are simple coefficients of the quadrantic terms x_ix_j in the expression of the energy function, while the constant terms I_i are the coefficients of the linear terms x_i.

With the selection of neural network weights and constants, the state of a neuron x_i is randomly initialized and the totality of neurons are updated in accordance with equation (3), with time passing, energy function is minimized down to stabilization of the neurons network, the state the

last neuron reaches, the final one, corresponds to a minimal local solution belonging to the optimization problem.
Hopfield Neural Network approach is a descending technique for resolving the optimization problem using modeling through a penalizing function, improving the economic performances of business models which are fundamented on this type of approach, it is done by incorporating "hill-climbing" strategies in neuronal updating equations (3) which leads to systemic processes simulated by "annealing".
For a more complete characterization of the economic processes approach using Hopfield's neural networks, we recall that Boltzman machines and information technology are used of "mean-field annealing" type.

2.3 Self-organized neural networks

Self-organized neural networks find a wide range of application within operational business models, these identify analogous systemic processes where "Multilayered Feedforward Neural Networks" – M.F.N.N.s are used in the case of availability of learning-training data with the "backpropagation" property.
The cluster is always used to generate a data grouping phenomenon based on the natural data structure, the pursued objective is that of developing an operationally implementable algorithm for grouping the data used by the business model, the degree of similarity of the possible operational models from the content of the cluster is maximized, the similarity of the fixed forms used for these structures, coming from different clusters is minimized.
In the practice of operational implementation of business models by the help of neuronal networks of self-organized type, we identify a synergistic structure with a high degree of complexity for positioning them in a high-dimensional entry space.
For the implementation in a characteristically space of one to three dimensions, the operational structure grows in transparent, in specialized scientific literature, Kohonen's Simple organizational features-characteristic maps (S.O.F.M.'s) are particularly useful "tools" in the complex process of decision elaboration, by detecting through automated system processes fundamental characteristics structured in large dimensionalities data sets, mapping through S.O.F.M. procedures transforms the overwhelming dimensional space input into a much smaller one, the cluster groups are visible in the case of diminishing the dimensionalities, easily accessible and redeemable for the operational implemented structures.
Approaching organizational business models with S.O.F.M. methodology involves adapting weightings to enhance learning processes that are further developed, learning is carried out unattended, neural network searches are unknown.
S.O.F.M. compared to previous neural network models suggest an innovative architectural structure.
The role of neuron locations in the process the learning process is decisive in terms of its quality, the input vector in the identified network is connected to a series of neurons, a dimension –"line" or two dimensions – "lattice", within the operational systemic processes, the neuron ordering plays a very important role, the references are made to regions of "neurons firing" - a neural firing process.
Fundamental to this approach is the notion of ordering and arranging neurons, operating the S.O.F.M.'s reference is made to so-called neurons firing regions, the process of burning a neuron is very likely to be interconnected with similar processes, "Firing Neuron" located in neighboring neural entities.
A systemic process of physical localization of neurons is developed, the idea is based on biological considerations, mimetism after the human brain induces large neural regions that function in a synergistic and centralized manner, thus the central task imposed on the human brain becomes the main objective.

Human brain analogy, S.O.F.M. implemented to developing, operationally implementing and improving the economic performance of business models, identifies a winning neuron, the "winning neuron", which is responsible for triggering the highest-intensity input signal, neurons located in his proximal vicinity are directly influenced, the whole region of the neural network becoming proactive.

We introduce the fundamental concept of "neighborhood" in interdependence with neuronal architecture developed by S.O.F.M., for a linear set of neurons, neighbors and interconnected.

The neurons positioned within the network to the left, and the right, the "winning neuron" represents its neighborhood, in specialized scientific literature, this definition is called "neighborhood size of one".

For the analysis and understanding of the effect induced by a large region of neurons, we consider a neighborhood of the neural network, significantly dimensional, operationalized for a rectangular vector, at the level of a geometrically describable neighborhood as a hexagonal structure.

We consider the neighborhood dimension around the "winning neuron" m at time "t" as being $Nm(t)$, the amount of learning data that each neuron " i " within the m neighborhood receives, is mathematically expressed with the relation:

$$c = \alpha(t)\exp\left(-\frac{||r_i - r_m||}{\sigma^2(t)}\right) \quad (5)$$

where $(r_i - r_m)$ is the physical distance between the i neuron and the "winning neuron", m, the two functions $\alpha(t)$ and $\sigma^2(t)$ are used in controlling the amount of learning information that each neuron receives in the relationship with the "winning neuron".

The learning algorithm developed for neural network approach using S.O.F.M. is structured on basic steps starting from inputs, calculates the outputs within the neural network and specifically updated weights, this has the following architecture:

Stage 1→ Initialization: weights with randomly low values, adjacent to the size of $N_m(0)$ which has the possibility to be increased, but smaller than the number of neurons expressing the matrix size, the parameter functions, $\alpha(t), \sigma^2(t)$are between $0, 1.$

Stage 2→Is presented the input model x specific to the input level of the neural network, calculates the closest distance of the inputs of the weights for each neuron j:

$$d_j = ||x - w_j|| = \sqrt{\sum_{i=1}^{n}(x_i - w_{ij})^2} \quad (6)$$

Stage 3→ We develop a systemic process of identifying and selecting a neuron positioned at a minimal distance for the "neuron winner" m.

Stage 4→A process of updating the weights that connect the input layer within the neural network with the "neuron winner" and the totality of the neurons positioned in the neighborhood of the neural network according to the learning methodology

$$w_{ji}(t+1) = w_{ji}(t) + c[x_i - w_{ji}(t)] \quad (7)$$

where $c = \alpha(t)\exp\left(-\frac{||r_i - r_m||}{\sigma^2(t)}\right), \forall j \in N_m(t)$, neuron from the network.

Stage 5→ ***Stage 2*** continues for a length of Ω periods, the decreasing dimension of the neighborhood, $\alpha(t)$and $\sigma^2(t)$, repeatable procedure until the weights are stable, update the weights connecting the input layer with the winning neuron and neighboring neurons according to the learning rule.

3. DIFFERENT TYPES OF NEURAL NETWORKS APPLIED TO BUSINESS MODELS

The diversity and complexity of business models used to solve the operational-strategic issues faced by economic organizations are solved with the help of a variety of neural structure alternatives, each of which has peculiarities for a specific applied field.
We mostly identify adaptations of the three main neural network models previously presented.
The applicative potential of these neural models, of business world issues with its specific challenges and constraints, but also to the operational research is practically infinite horizon.
Developmental trends such as modular neural networks, neo-cognitron networks, "brain-state-in-a-box", adaptive resonance networks, radial basis networks, are all generating potential solutions for the future of "smart business."

4. USING NEURAL NETWORK FLEXIBILITY FOR ECONOMIC MODELS

The adaptability, flexibility and high power of real-time computing are the arguments which transform neural networks into effective and useful tools in the service of modern business organizational top management.
The multitude of functions, algorithms and mathematical procedures used in the study and analysis of economic phenomenology find the transposition into operability and the possibility of implementation of informatics with the help of neural-cognitive sciences.
The proposed architecture for this type of problem specific to business models has as foundations V, E, σ fixed, we analyze which functional hypotheses in $H_{V,E,\sigma}$ are implementable as a function in the structure of neural layers with dimensionality V.
We observe that for simulating business models using neural networks computers (machines) are needed, using the storage of numbers from the real number set up to the b bits capacity, calculating the function $f: R^n \to R$ on such smart tool being possible whenever, which is a systemic process equivalent to the calculation of the function $g: \{\pm 1\}^{nb} \to \{\pm 1\}^{nb}$, analyzing what Boolean type functions are operationalized with $H_{V,E,sign}$ on a computer which stores real-numbers using a maximum of b bits capacity.

Observation 1: For $\forall n$, there is a graph (V, E) with depth 2, such that $H_{V,E,sign}$ contains the totality of functions from $\{\pm 1\}^n$ to $\{\pm 1\}$.

Argumentation: We consider the graph with $|V_0| = n + 1, |V_1| = 2^n + 1, |V_2| = 1$, we consider E the totality of all possible edges between adjacent layers, let $f: \{\pm 1\}^n \to \{\pm 1\}$ a set of Boolean functions, we prove that the weights of the neurons are adjustable so that the network implements the f function.
We consider the totality of the vectors of $\{\pm 1\}^n$ as being $u_1, \ldots, u_k$ for which the outputs of f are 1, implicitly result that $\forall i, x \in \{\pm 1\}^n$, if $x <> u_i \to (x, u_i) \leq n - 2, x = u_i \to (x, u_i) = n$, we conclude that the function $g_i(x) = sign\left((x, u_i) - n + 1\right) = 1 < - > x = u_i$.
Continuing this type of reasoning, we identify the possibility of adapting the weights between V_0, V_1, such that $\forall i \in [k]$, neuronal $v_{1,i}$ represents the implementation of the function $g_i(x)$, thus $f(x)$ is the separation of the functions $g_i(x)$, there is relation

$$f(x) = sign\left(\sum_{i=1}^{k} g_i(x) + k - 1\right) \quad (8)$$

which confirms the correctness of the statement within "Observation 1".

The previously developed observation and reasoning demonstrates how a neural network implements any type of boolean function.

Observation 2: For $\forall n$, or $s(n)$ the minimal integer for which there exists a graph, (V, E) with $|V| = s(n)$ such that the set of hypotheses $H_{V,E,sign}$ contains the totality of functions from $\{0,1\}^n$ to $\{0, 1\}$, then $s(n)$ is exponential type after n, analog, similar results are generated for $H_{V,E,\sigma}$ a sigmoid function.

Argumentation: We make the working hypothesis that for some (V, E) the set $H_{V,E,sign}$ contains the totality of the functions from $\{0, 1\}^n$ to $\{0, 1\}$, therefore it is possible that the vectors set with $m = 2^n$ of $\{0, 1\}^n$ to be lost (compromised), therefore we conclude that the dimensionality of $V.C.$ with $H_{V,E,sign}$ is 2^n, the size of $V.C.$ of $H_{V,E,sign}$ is bounded by the value provided by the mathematical formalism $O(|E| \log(|E|)) \leq O|V|^3$, which is the subject of a separate demonstration, results that $|V| \geq \Omega(2^{\frac{n}{3}})$, which confirms our hypothesis and our conclusions for activating the sign function, the analogue debate and demonstration are extrapolable for sigmoid function.

Observation 3: Theoretically we identify the possibility of constructing a similar reasoning for $H_{V,E,\sigma}, \forall \sigma$, weights in the neural network are restricted such that they are expressed by using a certain number of bits bounded by a constant value universally, we also take into account the sets (classes) of hypotheses for which different neurons use different activation functions, this approach is allowed as long as the number of activation functions is a defined sequence.

The previous rationality developed and presented demonstrates the impossibility of expressing the totality of the boolean typology functions by means of a polynomial dimensional network, but it is perfectly possible for the boolean function to be calculable during $O(T(n))$ expressible by a neural network of dimensionality $O(T(n)^2)$.

Observation 4*:* Consider $T: N \to N, \forall n$, or F_n is the set of functions that are implementable using a Turing machine with a working time of no more than $T(n)$, then $\exists\, b, c \in R_-$, constant, thus that $\forall n$, there is a graph (V_n, E_n) with the maximum size $cT(n)^2 + b$, with the condition that $H_{V_n,E_n,sign}$ contains F_n.

Argumentation: We identify within the previous mathematical observation the relation between the complexity of the time running the structured programs with the help of neural networks for the business models and the complexity of their circuits.

The boolean circuits are the neural operational networks, at the level of each entity are implemented conjunctions, disjunctions, negations of the input information (inputs), the complexity of the circuits quantifies the dimensionality of the boolean circuits required for the function calculation.

In order to analyze and understand the relation between the complexity of the time of program operationalization and the complexity of the circuits, it is necessary to model each stage of the execution of the software solutions adopted and implemented as the operation itself in the state assigned to the related hardware structure, allocated memory, thus, the neurons positioned in every layer of the neural network structure shows the state of the computer memory for the afferent time, the process of transition to the other layer of the network involves the calculation of the achievable network operations.

In the case of neural networks that contain boolean circuits with sign activation functions, the implementation of logical operations becomes essential for their correct operationality, it is possible that using the sign function activated a process of implementing the logical negation to be operationalized, we will then analyze the implementation of the conjunction and logical disjunction within neural networks, fundamental logical operations in the building and understanding the functioning of business models for modern economic organizations.

Observation 5: We consider a neuron v, we later implement the sign activation function, it has k edges, input sides, connected to the neurons that supplies the values by output $\{\pm 1\}$, generating a process of adding another neuron-binding side constant of v, adjusting the weights placed on the margins at v, the outputs of v offer the possibility of implementing the logic conjunction and logic disjunction from its inputs.

Argumentation: If $f:\{\pm 1\}^k \rightarrow \{\pm 1\}$ is the conjunction function, $f(x) = \wedge_i x_i$, is exppresible as $f(x) = sign(1 - k + \sum_{i=1}^{k} x_i)$, analog, the disjunction function $f(x) = \vee_i x_i$ is expressible as $f(x) = sign(k - 1 + \sum_{i=1}^{k} x_i)$.

In the framework of the business models operationalization, the neural networks are universal approximators, consequently for a fixed parameter, selected with a certain degree of precision, we consider $c > 0$, for each function of the Lipschitz type, $f:[-1,1]^n \rightarrow [-1,1]$ it is possible to elaborate a network for each input $x \in [-1,1]^n$, the network outputs are numeric values, positioned within the boundary range of $(f(x) - c), (f(x) + c)$, in the case of boolean functions, the dimensionality of the neural network can not be polynomial in n, which implicitly leads to a new observation.

Observation 6: Choosing certain $c \in (0,1)$, for $\forall n$, let $s(n)$ the minimal integer for which there exists a graph (graph), (V, E) with $|V| = s(n)$ for the set of assumptions (hypotheses), $H_{V,E,\sigma}$, σ is a sigmoid function, is approximated inside with a degree of precision c, each function "$1 -$Lipschitz", $f:[-1,1]^n \rightarrow [-1,1]$, it results that $s(n)$ is exponential after n.

5. ASSIGNING TIME TO OPERATIONALIZE THE LEARNING PROCESS, NEURAL NETWORKS

According to previous scientific considerations and analyzes, the sets of neural networks used to improve the operationality of business models, containing an underlying graph of polynomial dimensionality, have the ability to express the totality of the implementable functions, the dimensionality of the neural network subsequently determines the complexity of the sample, of the selected model.

We introduce the concept of complexity class "N.P." ("nedeterminist polynomial time"), in this notion are included the decisional problems executed during polynomial time by a non-deterministic Turning machine, the number of steps from the initial state is initial.

Also, the concept of "empirical risk minimization", "E.R.M." (Empirical Risk Minimization) as a principle of the static learning theory for neural networks which defines a family of procedures, operational learning algorithms that delimit the theoretical performances of this type of process.

Taking into account the situation where several supervised learning processes are approached, we identify two spaces of objects, X, Y, or the learning function $h: X \rightarrow Y, x \in X, y \in Y$, called hypothesis, which at the output displays an object $y \in Y$ for $x \in X$.

We approach a lot of training with m examples $(x_1, y_1), \ldots, (x_m, y_m), x_i \in X, y_i \in Y$, x_i is the input to the neural network, y_i the answer through the function $h(x_i)$.

Suppose that the common probability value is $P(x, y)$ over X, Y with m states $(x_1, y_1), \ldots, (x_m, y_m)$, determined by $P(x, y)$.

We mention that the distribution hypothesis of the common probability allows to use a uncertainty model in case of predictions (forecast), inducing a certain noise in the processed data, "noise in data", since y is a function of x, but a random variable with the conditional distribution $P(y|x)$, for a fixed x, selected.

Suppose there is a non-negative value of the "loss function", $L(y^{\hat{}}, y)$ which represents the quantifiable difference between the $y^{\hat{}}$ prediction and the concrete output y, the risk associated with the hypothesis $h(x)$ is defined as the expectation of the loss function:

$$R(h) = E[L(h(x), y)] = \int L(h(x), y)dP(x, y) \quad (9)$$

The loss function is located in the numerical range $[0, 1]$, $L(y^{\hat{}}, y) = I(y^{\hat{}} <> y)$ is a notation indicator, the ultimate purpose of the learning algorithm is to identify a hypothesis h' valid for a fixed set of functions H, for which the risk $R(h)$ is minimizable, becomes minimal, $h' = argmin_{h\in H} R(h)$.

Observation 6: We consider $k > 3, \forall n$, ∀n, let (V, E) a multistage layered graph with n input nodes, $(k+1)$ nodes for each hidden layer, in which one of them is a constant load neuron and a single output node, N.P. it has difficulty operationalize the approach, the rule, E.R.M., concerning $H_{V,E,sign}$.

The identifiable difficulty in the previous observation is induced by the purpose of learning, which requires the identification of a predictor, $h \in H$, with reduced empirical error, not an precisely E.R.M., the central task of finding the weights that have as a result the minimization of empirical errors is almost inoperable due to the dimensionality of the computational algorithms which must be operationalized.

Modify the architecture of the neural network such that to bypass the learning process induced difficulties in the original structure of the E.R.M. network, is difficult to quantify-evaluate by calculation, a possible implementation-operationalization becomes possible with the modification of the architectural arrangement of the studied network.

A highly effective alternative to this type of problem is to use as sigmoid type activation functions or any other type of relevant function in these types of systemic challenge.

The transposition into operational practice of a multitude of business models of this type of approach does not lead to significant, satisfactory results for the top organizational decision-management, the problem of half-space learning processes is one to avoid for the modern and efficient approach of the economic issues, because of the very high degree of difficulty, even in seemingly simple cases, of the independent model of representation of processes of knowledge-learning acquisition.

The use of the "Stochastic Gradient Descent" (S.G.D.) is a guarantee of success for learning processes adapted to modern business models, if the identified loss function is convex, within the neural networks this function (loss function) is usually non - convex, the algorithm and the methods that underlying diverts from S.G.D. allow the development of a search strategy for a reasonable and acceptable solution, which has identified operational practice on several concrete case studies.

THE REVERSE-BACK PROPAGATION PROCESS AND THE IMPLEMENTATION OF S.G.D.

At this point of the scientific analysis it is necessary to find a hypothesis belonging to $H_{V,E,\sigma}$ with a minimized degree of risk, in which the loads exceeding the margins of the model are weighted, the development of a heuristic approach for identifying the correct load using the concepts, paradigms and algorithms specific to the S.G.D., we assume that σ is a sigmoid type function, computable by $\sigma(a) = \frac{1}{(1-e^{-a})}$, with derivatives valid for any scalar differentiable function.

Considering that E is a finite set, we assume that the weight function is a vector $w \in R^{|E|}$, we assume that the studied neural network has n input neurons and k output neurons, noting $h_w: R^n \rightarrow R^k$, this function calculates if within the network it is defined with the hel of weights w, introducing the predicted loss by $h_w(x)$ with the notation $\Delta(h_w(w), y)$ in case in which the target is $y \in Y$, if we note with Δ the square loss, $\Delta(h_w(x), y) = \frac{1}{2}\|h_w(x) - y\|^2$ derived is easy to obtain using each differentiable function.

By choosing the distribution function D from the business models positioned within the $R^n \times R^k$ domain, we consider $L_D(w)$ is the risk function of the neural network, defined by the mathematical relation, $L_D(w) = E_{(x,y)\sim D}[\Delta(h_w(x), y)]$.

We introduce the S.G.D. algorithm. to minimize the $L_D(w)$ risk function within the neural network used within the business models.

The vector w is initiated with 0, or with a randomly selected structure close to this value, we make this choice that drives the totality of the hidden neurons to have the same weight in the case of a multiple-layer neural network, repeating the S.G.D. procedure several times, the process having as initiator a new random vector, the identified values tend towards a local minimum.

If we choose a fixed η step size this leads to significant results for convex problems, and there is also the possibility of a variable size of steps η_t, in the case of the non-convex type loss function, the second choice of the steps which must be taken is a lot more accurate having a large spreading area for test patterns and error calculation.

S.G.D. used for Neural Networks

parameters:

the number of iterations τ

sequence of step size $\eta_1, \ldots, \eta_\tau$

regulation of parameter $\lambda > 0$

input:

layered graph (V, E)

the differentiable function of activation $\sigma: R \rightarrow R$

initialization:

we choose $w^1 \in R^{|E|}$ *randomly*

from the distribution provided that $w^{(1)}$ *tends to 0*

for $i = 1, \ldots, \tau$

we consider the sample $(x, y) \sim D$

calculate the gradient $v_i = retropropagation(backpropagation)(x, y, w, (V, E), \sigma)$

updating $w^{(i-1)} = w^{(i)} - \eta_i(v_i + \lambda w^{(i)}$

output: w^{-} *is the best performance of* $w^{(i)}$ *on the validated set*

Figure 1.

Source: adapted from Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning. Cambridge University Press, 7th printing 2017, p.237

Implementation and calculation of Backpropagation

input:

$we\ choose\ (x, y), vector\ weightw, layered\ graph\ (V, E)$

$activation\ function\ \sigma: R \rightarrow R$

$initialization: description\ of\ the\ layers\ of\ the\ graph\ V_0, \ldots, V_T, V_t = \{v_{t,1}, \ldots, v_{t,k_t}\}$

$we\ define\ W_{t,i,j}\ as\ weight\ of\ (v_{t,j}, v_{t+1,i})$

$we\ set\ W_t, i, j = 0\ if\ (v_{t,j}, v_{t-1,i})\ does\ not\ belong\ to\ E$

$forwarding: choice\ O_0 = x$

$for\ t = 1, \ldots, T$

$for\ i = 1, \ldots, k_t$

$$we\ choose\ a_{t,i} = \sum_{j=1}^{k_{t-1}} W_{t-1,i,j} O_{t-1,j}$$

$we\ choose\ O_{t,i} = \sigma(a_{t,i})$

$backward: we\ choose\ \delta_T = O_T - y$

$for\ t = T-1, T-2, \ldots, 1$

$for\ i = 1, \ldots, k_t$

$$\delta_{t,i} = \sum_{j=1}^{k_{t+1}} W_{t,j,i} \delta_{t-1,j} \sigma'(a_{t-1,j})$$

$output: for\ each\ margin\ (v_{t-1}, v_{t,i}) \in E$

$we\ choose\ partial\ derivatives\ \delta_{t,i} \sigma'(a_{t,i}) O_{t-1,j}$

Figure 2.
Source: adapted from Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning.Cambridge University Press, 7th printing 2017, p.237

6. CONCLUSIONS

Addressing business model issues with the help of neural networks is a way to help top decision management in the process of developing operational-strategic decision-making, with a lesser degree of error and a higher capacity of generating added value.

Concepts and paradigms specific to the neural network models are a particularly valuable tool in a wide range of economic environmental segments, with all the characteristics to which it is subjected by the broadest and most complex phenomenon that influences human civilization in its entirety, globalization.

This new type of approach to complex business issues puts into a new perspective the relationship between information-data, immediate operationality, developed strategic processes.

Flexibility, transparency, efficiency, simplicity, these characteristics of the neural networks help to identify patterns and data structures in their entirety.

The name of the data used for these types of models is that of training data, learning processes having a central role in this systemic phenomenology.

The approach offers the opportunity for the learning process to make predictions for future events based on the patterns observed within the training, the classification of data that is not perceptible on the basis of the training characteristics, the organization of the group training data based on the characteristics observed during the training processes.

The classification of the approached neural network types in multiple layers feedforward networks, Hopfield networks, organized networks, provides an overview of the theoretical amplitude of neuromimetic approach, as well as a strategic trend to improve and address the challenges of economic competition.

We consider that the main task of the top organizational management, that of decision elaboration under uncertainty conditions is supported and deeply and rigorously scientifically grounded in the implementation and use of neuro-mimic models with all the resulting consequences.

REFERENCES

Anton, H., Rorres, C. (2010). *Elementary Linear Algebra: Applications Version*. John Wiley & Sons.

Ayres, I. (2008). Super Crunchers:*Why Thinking-By-Numbers is the New Way to Be Smart*. Bantam.

Barber, D. (2012). *Bayesian reasoning and machine learning.* Cambridge University Press.

Bengio, Y. (2009). *Learning deep architectures for ai. Foundation and trends in Machine Learning 2(1), 1-127.*

Bishop, C. (2006). *Pattern recognition and machine learning.* Springer.

Bishop, C., M. (1996). *Neural Networks for Pattern Recognition.* Oxford University Press.

Breiman, L. (2001). *Random forests. Machine learning 45(1), 5-32.*

Cover, T., Thomas, J. (1991). *Elements of information theory.* Wiley New York.

Daelemans, W., van den Bosch, A. (2005). *Memory based language processing. Studies in natural language processing.* Cambridge University Press.

Dalgaard, P. (2008). *Introductory Statistics with R*. Springer.

Davenport, T., H., Kim, J. (2013). Keeping Up with Quants: *Your guide to Understanding and Using Analytics.* Harvard Business Press Books.

Eco, U. (1999). *Kant and the platypus.* Vintage U.K. Random House.

Frank, E. (2000). *Pruning decision trees and lists.* Ph.D. thesis, Departament of Computer Science, The University of Waikato.

Franklin, J. (2009). *Mapping Species Distributions: Spatial Inference and Prediction (Ecology, Biodiversity and Conservation).* Cambridge University Press.

Gadenfors, P. (2004). *Conceptual Spaces: The geometry of throught.* MIT Press.

Gleick, J. (2011). *The information: A history, a theory, a flood.* HarperCollins U.K.

Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning.* Springer, New York.

Kollar, D., Friedman, N. (2009). *Probabilistic graphical models: principles and techniques.* The MIT Press.

MacKay, D., J. (2003). *Information theory, inference and learning algorithms.*Cambridge University Press.

Montgomery, D., C., Runger, G., C. (2010). *Applied statistics and probability for engineers.* Wiley.com

Shai, S., S., Shai, B., D. (2017). *Understanding Machine Learning.* Cambridge University Press, 7th.

Shannon, C., E., Weaver, W. (1949). *The mathematical theory of communication.* Urbana: University of Illinois Press.

Silver, N. (2012). *The Signal and the Noise: Why So Many Predictions Fail-but Some Don't.* The Penguin Press.

Stewart, J. (2012). *Calculus (7e ed.).* Cengage Learning.

Tufte, E., R. (2001). *The Visual Display of Quantitative Information.* Graphics Press.

Tukey, J., W. (1997). *Explorarory Data Analysis.*Addison-Wesley.

Vapnik, V. (2000). *The Nature of Statistical Theory*. Springer.