information

MDPI

Article

Artificial Neural Network Learning, Attention, and Memory

Vincenzo Manca

Dipartimento di Informatica, University di Verona, 37134 Verona, Italy; vincenzo.manca@univr.it

Abstract: The learning equations of an ANN are presented, giving an extremely concise derivation based on the principle of backpropagation through the descendent gradient. Then, a dual network is outlined acting between synapses of a basic ANN, which controls the learning process and coordinates the subnetworks selected by attention mechanisms toward purposeful behaviors. Mechanisms of memory and their affinity with comprehension are considered, by emphasizing the common role of abstraction and the interplay between *assimilation* and *accommodation*, in the spirit of Piaget's analysis of psychological acquisition and *genetic epistemology*. Learning, comprehension, and knowledge are expressed as different levels of organization of informational processes inside cognitive systems. It is argued that formal analyses of cognitive artificial systems could shed new light on typical mechanisms of "natural intelligence" and, in a specular way, that models of natural cognition processes could promote further developments of ANN models. Finally, new possibilities of chatbot interaction are briefly discussed.

Keywords: artificial neural networks; Machine Learning; artificial intelligence; cognitive systems; back-propagation

Citation: Manca , V. Artificial Neural Network Learning, Attention, and Memory. *Information* **2024**, *15*, 387. https://doi.org/10.3390/info15070387

Academic Editor: Francesco Fontanella

Received: 9 June 2024
Revised: 30 June 2024
Accepted: 1 July 2024
Published: 2 July 2024

1. Computation: Centralized vs. Distributed Models

In 1936, Alan Turing introduced the first mathematical model of a computing machine [1,2]. Such a machine executes a program consisting of instructions that it applies based on its internal state, performing actions that alter a workspace and interact with the external environment. The workspace can be reduced to a linear tape (divided into squares) capable of holding symbols. At the start of a computation, symbols placed on the tape constitute the input for the machine. The execution of an instruction involves reading and replacing symbols, moving to other squares, and changing the internal state. The computation ends when the machine reaches a state where there are no instructions to apply. At that point, the tape's content is the result that the machine provides, corresponding to the input placed on the tape at the beginning of the computation. The calculation of a machine, thus, identifies a function that associates input data with output results. A crucial aspect of such a computation model is its partiality; i.e., there may be cases where a machine continues calculation indefinitely without providing any output, resulting in an indefinite for the provided input. The computation is centralized, and the program (list of instructions) is an expression, in an appropriate language, that describes the calculation realized by the machine.

In the same year as the publication of Turing's model, Church proposed a thesis, known as the Turing–Church thesis, which posits that for every computable function, there is a Turing machine computing it, i.e., such machines define a universal model of computation.

Seven years after Turing's seminal work, an elderly neurologist and a young mathematician, Warren McCulloch and Walter Pitts [1,3], defined artificial neural networks (ANNs) as a computation model inspired by the human brain. Neural networks are a distributed model.

Presently, the more popular notion of a neural network is a directed graph of nodes called neurons and edges (arrows) called synapses. The nodes are labeled with basic functions, called transfer functions, and the edges by numbers called weights, which express the strength with which the synapse transports the output of the neuron to the target neuron that receives it. In such a model, there is no program of instructions, but the behavior of the network emerges from how its parts are connected and from the weights of the synapses. This difference between the two models is profound and, over about half a century of research [4], has gradually matured Machine Learning (ML) as an alternative paradigm to the Computing Machine. A network is not "programmed", but "trained", and training consists of presenting it with pairs $(X, F(X))$ where F is a function to be learned that sends real vectors of n components to real vectors of m components. With each example, the network is "adjusted" by varying the weights to decrease the error committed between the output Y it produces and the $F(X)$ to be learned (the network's weights are initially chosen randomly). In other words, a network "learns" functions from examples. When an ANN has learned a function F, then it computes F on all its input values, with the possibility of error which, under certain assumptions, remains under a fixed threshold. The class of functions computed by ANNs is strictly included in that of functions computed with Turing machines. To achieve complete equivalence, neural networks require features not present in the basic model just outlined [1].

However, a remarkable result holds [5–8] that, for every continuous and bounded real function from n-vectors to m-vectors, and every chosen level of approximation, there is an appropriate artificial neural network that computes the function with an approximation error lower than the fixed level of approximation. Thus, although neural networks are not as universal as Turing machines, they satisfy another type of universality in approximating real functions (of many arguments and results).

The present work suggests how the neural network model can become a formidable tool for analyzing natural intelligence in mathematical terms, providing precise counterparts to concepts difficult to characterize in rigorous terms.

Today, we have machines that speak [9,10], in many ways, indistinguishably from human interlocutors. Tomorrow, we will have machines that approach increasingly powerful human competencies. The development of these machines will shed new light on our minds. In turn, this knowledge will become the basis for new developments in artificial intelligence. Apart from the recent success of evolute chatbots, such as those developed by OpenAI or Google, ANNs are applied in almost all scientific and technological contexts, together with many other fields of artificial intelligence [11].

The following sections aim to develop reflections along this direction, starting with a concise presentation of the equations that govern the learning process of neural networks. Many aspects of the discussion are revisitation of themes already present at the dawn of computing, by the founders of such science who, in an impressively clear manner, had a vision of perspectives of development intrinsic to information processing [12–15].

2. The Structure of ANNs

An ANN is given by a graph of nodes called neurons (identified by indexes) and edges called synapses. Each neuron is associated with a function, called the transfer (or activation) function of the neuron (with the same index as the neuron). Each neuron has afferent and efferent synapses that connect it to other neurons, receive values from the outside, or send values to the outside. The transfer functions are composed according to their links, because they take inputs from afferent synapses coming from other neurons, or outside, and provide results on the efferent synapses going to other neurons, or outside. Figure 1 shows a graphical description of a single neuron and a simple ANN where this neuron is connected with seven other neurons through its synapses. The modern notion of the artificial neural network is an evolution of the initial model by McCulloch and Pitts, emerging since the work [16]. Transfer functions $f_1, f_2, \ldots f_k$ associated with neurons are expressed by mathematical formulas with a single variable as an argument (polynomial, sigmoid, hyperbolic tangent, zero-linear, . . .).

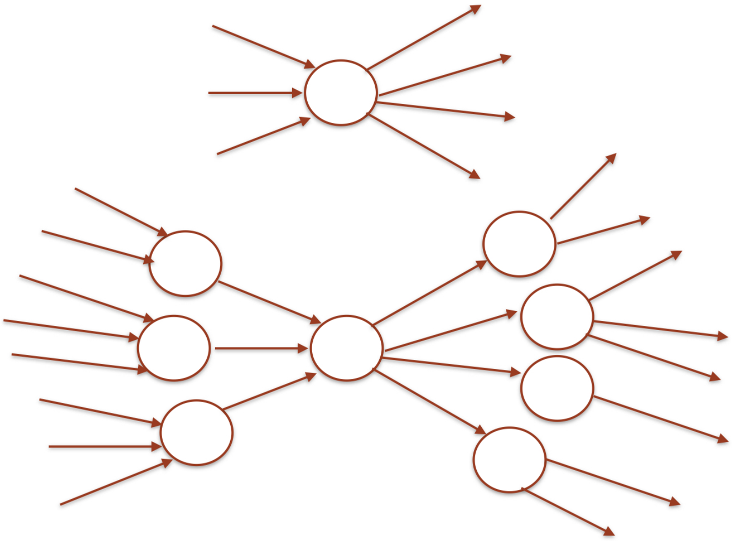

Figure 1. A single neuron (**above**) with 3 afferent synapses and 4 efferent synapses; an ANN (**below**) where the single neuron is connected with 7 neurons, 3 input neurons with 8 synapses, and 4 output neurons with 8 synapses.

Metabolic networks, outlined in [17,18], are a computation model, similar to ANN, inspired by metabolic transformations. Substance nodes contain substances consumed by reaction nodes, producing substances in other substance nodes. The quantities consumed by each reaction node depend on a regulating function taking as arguments the quantities of some substance nodes. In [17], the computational relevance of this model and its relationship with neural networks has been investigated. Interestingly, these networks combine two levels, the transformation and the regulation level, which controls the first one. This situation is very similar to what we describe in Section 4 with the dual network realizing attention mechanisms over a basic neural network. Namely, in [18], machine learning algorithms were given for training metabolic networks to acquire specific biologically relevant competencies.

If $x_1, x_2, \ldots, x_n$ are input arguments and the composition of the transfer functions of an ANN produces output values $y_1, y_2, \ldots, y_m$, then the given artificial neural network computes a function from $\mathbb{R}^n$ to $\mathbb{R}^m$.

Let M be a network and $N(M)$ the set of its neurons. The following notation is used to precisely define the operation of the network by establishing how the transfer functions are composed. We write $i \rightarrow j$ when a synapse from neuron i is afferent to neuron j. Let us denote by $w_{i,j}$ a number associated with the synapse, called its **weight**. When $w_{i,j} = 0$ no connection is present from i to j.

The weights of the synapses of an ANN uniquely identify the function calculated by the network. The sets of input neurons, output neurons, and internal neurons of M are defined by:

$$In(M) = \{j \in N(M) | \ \nexists i \in N(M) : i \rightarrow j\} \tag{1}$$

$$Out(M) = \{j \in N(M) | \ \nexists i \in N(M) : j \rightarrow i\} \tag{2}$$

$$Int(M) = \{j \in N(M) | j \notin In(M), j \notin Out(M)\}. \tag{3}$$

For all $j \in N(M) - In(M)$ we denote by u_j the value given by the transfer function f_j applied to the weighted sum of the values transmitted to the synapses afferent to neuron j. This value is sent from j to all its exit synapses.

Ultimately, a neural network can abstractly be characterized by a pair (neurons, synapses) of the following type:

$$M = (\{f_j\}_{j \in N(M)}, \{w_{i,j}\}_{i,j \in N(M)}) \tag{4}$$

where $N(M)$ is a set of indices associated with neurons and their corresponding transfer functions, while the synapses are pairs of neuron indices to which the weights are associated. The entire function computed by the network is encoded by the weight matrix $\{w_{i,j}\}_{i,j}$.

For each neuron j that is not an input, we define its **weighted input** z_j by:

$$z_j = \sum_{i:i\to j} w_{i.j} u_i. \tag{5}$$

and its **exit** u_i given by:

$$u_j = f_j(z_j) \tag{6}$$

if j is an **input neuron**, then $x_j = z_j$; if it is an **exit neuron**, then $y_j = u_j$.

Let n be the number of inputs of M, and m the number of its exits. Given a function F from $\mathbb{R}^n$ in $\mathbb{R}^m$, the function $E_M(x_i|i \in In(M))$ of errors committed by M on $(x_i|i \in In(M))$ is usually based on the sum of the squares of differences e_j between the target values t_j provided by F and by M, on the inputs $(x_i|i \in In(M))$:

$$e_j = (t_j - y_j)|i \in Out(M). \tag{7}$$

Backpropagation [7,8,19–25] is a method by which a network learns to compute a given function F through examples.

We can assume that M is associated with a learning network M_l that takes in the errors $(e_j|i \in Out(M))$ and updates the weights by setting:

$$w_{i,j} := w_{i,j} + \Delta w_{i,j} \tag{8}$$

The calculation of $\Delta w_{i,j}$ is performed through the coefficients δ_j associated with each neuron through a backward propagation mechanism described by the learning Equations (9)–(11) of the next section, which reduces the errors e_j committed by M in providing the expected output.

It is demonstrated that, under appropriate assumptions about the structure of M and transfer functions, this procedure converges, meaning that after a sufficient number of examples, the error made by M goes below a predetermined tolerance threshold.

Ultimately, M coupled with M_l, along with several learning instances, acquires weights such that M computes a function that differs from the required function F of an error below the chosen threshold.

3. Learning Equations

The following learning Equations (9)–(11) determine the way M_l operates, where f_j' denotes the derivative of f_j, and η is is a constant less than 1, called the learning rate:

$$\boldsymbol{\Delta w_{i,j} = -\eta \delta_j u_i} \tag{9}$$

$$\boldsymbol{\delta_j = \frac{\partial E}{\partial u_j} f_j'(z_j) \quad \textit{if} \ j \in Out(M)} \tag{10}$$

$$\boldsymbol{\delta_j = f_j'(z_j) \sum_{k:j\to k} \delta_k \, w_{j,k} \quad \textit{if} \ j \in Int(M)} \tag{11}$$

Theorem 1. *[Back-propagation]*
Learning equations derive from the principle of **descendent gradient** *expressed by the following Equation [7]:*

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}} \tag{12}$$

showing that errors diminish when weights change in the direction opposite to the error gradient of weights.

Proof. Namely, let us define:

$$\delta_j = \frac{\partial E}{\partial z_j} \tag{13}$$

then, by using Leibniz's chain rule of derivation, the following equations hold:

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}} = -\eta \frac{\partial E}{\partial u_j} \frac{\partial u_j}{\partial z_j} \frac{\partial z_j}{\partial w_{i,j}}. \tag{14}$$

$$\delta_j = \frac{\partial E}{\partial u_j} \frac{\partial u_j}{\partial z_j} \tag{15}$$

where learning Equation (9) is obtained by replacing (15) in the right member of (14), thus deriving z_j according to Equation (5). Equation (10) follows from (15), deriving u_j according to (6).

Equation (11) comes from Equation (15) when we observe that for an internal neuron, the error derivative is expressed by the chain rule by applying (5) in the calculation of the derivative of z_j concerning $w_{i,j}$:

$$\frac{\partial E}{\partial u_j} = \sum_{i:j \to i} \frac{\partial E}{\partial z_i} \frac{\partial z_i}{\partial u_j} = \sum_{i:j \to i} \frac{\partial E}{\partial z_i} w_{i,j} = \sum_{i:j \to i} \delta_i w_{i,j}. \tag{16}$$

□

Equation (11) is the kernel of the back-propagation of weights updating for reducing the error committed by M in computing F.

Figure 2 illustrates the updating of two weights. The following equations represent this ANN as a system of equations.

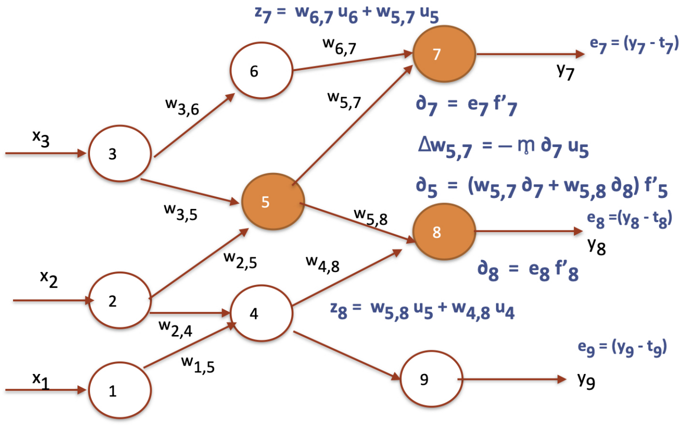

Figure 2. An ANN with 9 neurons updating weights $w_{5,7}$ and $w_{5,8}$ according to la backpropagation (in equation $\Delta w_{5,7} = -\eta\partial_7 u_5$ of the figure a stylized form of η is used). The filled neurons are those involved in the considered weight updating.

$$\begin{aligned}
u_6 &= f_6(w_{3,6}x_3)\\
u_5 &= f_5(w_{3,5}x_3 + w_{2,5}x_2)\\
u_4 &= f_4(w_{2,4}x_2 + w_{1,4}x_1)\\
y_7 &= f_6(w_{6,7}u_6 + w_{5,7}u_5)\\
y_8 &= f_5(w_{5,8}u_5 + w_{4,8}u_4)\\
y_9 &= f_4(w_{4,9}u_4).
\end{aligned}$$

If we replace the first three equations in the last three equations the ANN is expressed as a composition of its transfer functions:

$$y_7 = f_6(w_{6,7} f_6(w_{3,6} x_3) + w_{5,7} f_5(w_{3,5} x_3 + w_{2,5} x_2))$$
$$y_8 = f_5(w_{5,8} f_5(w_{3,5} x_3 + w_{2,5} x_2) + w_{4,8} f_4(w_{2,4} x_2 + w_{1,4} x_1))$$
$$y_9 = f_4(w_{4,9} f_4(w_{2,4} x_2 + w_{1,4} x_1)).$$

4. Attention Mechanisms

The term "Attention" is used widely in cognitive processes, as the capacity of a "cognitive agent" to focus on relevant stimuli and appropriately respond to them. This general meaning is the basis of the following discussion aimed at defining a more specific sense in the context of ANN. To avoid possible misunderstanding, we mention that, in neural networks, the term has a specialist meaning, referred to as the "attention learning" mechanism [26] introduced in sentence translation, but this is not what we are interested in discussing in this paper.

What is the function of a neuron in our brain? With a sort of wordplay, a mathematical function. It calculates when producing signals on efferent synapses, in correspondence to the signals received on its afferent synapses.

This simple consideration tells us that ANN constitutes an abstract model of the brain, as organized in an integrated system of many ANNs. The notion of function, which is the basis of computation, as resulted from Turing's model and McCulloch and Pitt's model, is also the basis of a cognition structure, and it is also the basis of the fundamental mathematical concepts as developed since the new course of modern mathematics. Python language, popular in the AI context, is based on the same notion of function elaborated by Leonhard Euler, and formalized by Alonzo Church [27].

This functional perspective means that each brain competency can be considered a function computed by some neuron network, or by an integrated system of functions computed by subnetworks of the network, possibly organized at various levels. These competencies are formed through a "control mechanism", which applies the learning equations to adapt the network to its usage needs. In this perspective, ANN and ML (Machine Learning) are crucial concepts in AI and analogously in "natural intelligence".

A natural way of realizing a control mechanism for a network is using a **dual network**, also called an **attention network**, whose neurons, or meta neurons, receive values from the first network and input signals and produce outputs that alter the values of the controlled network's synapses. Such a network realizes ANN learning, as depicted in Figure 3, where meta-neurons update weights.

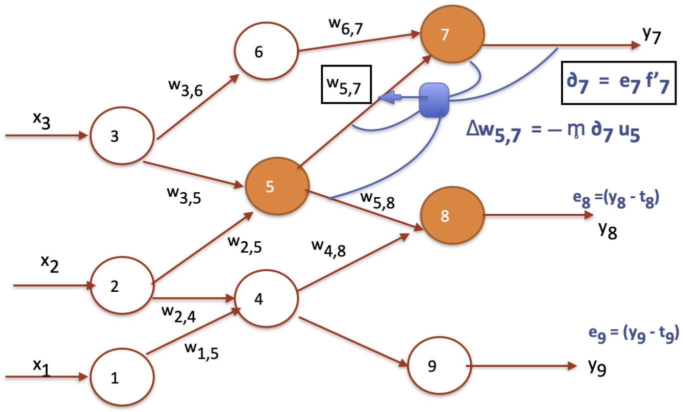

Figure 3. An ANN with 9 neurons and the updating of weight $w_{5,7}$ using a meta-neuron, the squared node (with rounded corners) taking inputs from synapses and neurons and transforming weight $w_{5,7}$.

Let us extend the notion of a neuron by admitting multiple outputs, with a specific transfer function for each output. Therefore, we can consider an entire network as a neuron, and for the same reason, a set of networks as a single network. This fact implies a reflexive nature of ANNs, similar to that of a hologram, where a part is an image of the entire structure [28]. This characteristic is the basis of global competencies typical of advanced cognitive systems.

From a strictly mathematical viewpoint, the meta-level of attention oversees weight updates by applying the learning equations.

This schematic description of back-propagation intends only to highlight the essence. Namely, in networks with many composition layers, various problems arise that make the presented scheme inadequate. One of the fundamental problems is the "vanishing gradient", where the values that propagate become increasingly smaller as the propagation moves backward, having no significant effect on the updating of weights [29,30].

5. Memory Mechanisms

The realization of memory within neural networks has been considered in various ways and with different solutions. It is often linked to the possibility of considering cycles that maintain a value always present in re-entering as input to a neuron or a subnetwork that produced it as output [1]. However, it is interesting to consider other possibilities and, especially, to develop mechanisms that can express typical aspects of the memory of natural systems, linking remembering with abstraction. Indeed, in animal cognitive systems, remembering is not just a conservation of data that accumulates, as happens in electronic memories. A memory often extracts certain essential elements, which are transformed into parameters to generate an "image" of the data. Thus, a memory is like a "sketch" of a process or an object in which elements that are not functional to the purposes of the memory are eliminated. In other words, memory is generative and abstract, simultaneously. Networks, by their very nature, allow these features to be realized. Below, we briefly outline some memory mechanisms that align with this intrinsically cognitive perspective of memory.

One realization of neuronal memory consists of translating outputs generated at certain levels of information flows into synaptic weights of appropriate memory neurons that, when activated, produce the right outputs, even without the presence of the inputs that had made them.

In the brain, the realization of a memory neuron requires the formation of new synapses [31]. In ANNs, in terms of what was said in the previous section, an analog can be realized by the meta-level associated with the primary level of an ANN.

To achieve this, a neuron m that memorizes a certain output u_j must be connected with a forward synapse $j \rightarrow m$ and a return synapse $m \rightarrow j$, placing $w_{m,j} = u_j$. In this way, if an activation input arrives at neuron m, for example, from the attention network, then m generates downstream of neuron j, the same information flow that a certain activity had generated in j, even in the absence of the inputs that had originated it.

A memory mechanism that avoids the meta-level is to add a synapse $j \rightarrow m$ and place in m an identity transfer function that sends the received value to its output. This value, through a cyclic synapse with a unit weight (afferent and outgoing from m), constantly maintains the value received from m as its output–input value. Apart from the introduction of cyclic synapses, this mechanism imposes an additional output variable for each memory of values, thus a cost that could become prohibitive. This suggests that such a mechanism could be used only in some special cases (for example, in some kinds of short-memory situations).

The memorization of an event is, in any case, a "trace" that identifies it regardless of its spatio-temporal occurrence. However, such a definition does not consider a crucial aspect of cognitive systems, namely that the memory of an event implies a form of abstraction. Indeed, the trace that a system preserves is not an exact copy of it, describing it totally like an identical replica. The preserved trace is often useful if it abstracts from many details. Thus, the same mechanism that underlies understanding is found in memorization. One understands by "forgetting" what is not functional to a series of logical connections and relations and, although it may seem paradoxical, one remembers by forgetting many details of the contingency within which an event was placed. Then, the mechanism, previously described, of memory neurons that transform output into a weight (with forward and return synapses) is realized by selecting only some outputs of a network, somehow connected to "interpretation" mechanisms that also allow, in addition to a reduction of information, a greater abstraction of the "preserved" elements. Such kinds of memories, which are formed under the control of the attention network and with the support of emotional, discerning, and goal-directed levels, constitute a map of the history of the entire cognitive system and, thus, constitute the most significant aspect of its identity.

6. Comprehension and Knowledge

Besides learning, the attention network can perform functions of stimuli selection, addressing them to subnetworks, coordination, and storing information fluxes.

Combinations of an ANN with the associated meta-network results in a qualitative leap in the functional and evolutionary organization of ANNs. Indeed, this achieves a transition comparable to that from simple Turing machines, each calculating a specific function, to a universal Turing machine within which all functions computed by individual machines can be realized. In more pragmatic terms, the meta-level of an attention network performs functions similar to those of a computer's operating system. Indeed, an operating system coordinates programs that perform specific tasks and manage the machine's resources. Furthermore, if the operating system includes interpreters or compilers of programming languages, it can enhance the machine's functionalities through new programs written in the hosted programming languages.

In the case of ANNs with a meta-level of attention, the notion of programming is replaced by that of training. This means that the training mechanism that realizes the network, once internalized in the machine (with the attention network), allows it to develop new functionalities. When the attention network connects to networks of discernment and will, supported by stimuli of emotions, motivations, and purposes, it leads to further levels of consciousness and self-control that can not only increase the complexity of the entire system but can also serve as models for complex functionalities of the human mind.

What is the relationship between attention and understanding?

Generally, a system that acquires data realizes mechanisms of understanding when acquiring that data, it integrates them within the system's structure by reducing them in terms of data already internally available.

In understanding, two opposing and cooperating mechanisms intervene in **assimilation** and **accommodation** [32]. The acquired data are assimilated into the internal structure, trying to keep it as unchanged as possible. At the same time, the internal structure "accommodates" to more completely embrace the acquired data.

In this sense, adding data to a list is the most rudimentary form of understanding, because what is added changes nothing of the previous structure, and the data undergoes no transformation that considers the structure that acquires it.

Conversely, if the received data has aspects in common with data already present in the system, the more common aspects there are, the less the system has to add to assimilate the data internally.

Thus, understanding involves analyzing the data and searching for common aspects by abstracting from particular features of single data. In other words, true understanding requires an abstraction that brings out salient aspects while neglecting others that are not interesting to its assimilation. This filter reduces the costs of accommodation of the system in assimilating what is understood.

The assimilation–accommodation mechanism naturally connects to interpretation, which seeks the data elements known to the system. In finding known elements, there is an act of recognition peculiar to the interpretative mechanism.

Knowledge has a broader significance than understanding and interpreting. It can involve the "invention" of new elements, restructuring internal data according to a deeper logic that connects them.

This situation occurs in science when new theories allow for common explanations of old theories that are not reducible to one another. Knowledge expands horizons with new elements that enable a better understanding of the old ones. In this mechanism, ideal elements [33] are crucial.

An ideation process consists of creating an element whose presence improves knowledge. The typical example made by Hilbert is that of imaginary numbers. The number $i = \sqrt{-1}$ cannot be equal to any real number, because its square is -1, while the square of a real number is always positive. Thus, i is a new type of number, precisely ideal, whose presence allows for the natural calculations performed to solve algebraic equations.

Similarly, a physical particle is something that, in certain experiments, produces the measurement of certain parameters. When certain phenomena are clearer and better explained by assuming a particle (electron, proton, neutron, . . .), then we have "discovered" that particle.

The only things a physicist measures directly are displacements of points, so any physical process could be entirely described in terms of sequences of displacements (even times are sequences of displacements).

In general, something exists mathematically or physically if its existence produces descriptions more understandable than those that would be made without its existence.

The same applies to notions definable in cognitive systems. We use the term "training" to denote a process by which a system acquires a competency. However, it is a function in a neural system, or better, a function that produces functions.

All the language of science is based on ideations, and natural language provides linguistic mechanisms to shape concepts generated by ideal elements.

Training is a particular case of understanding, because it allows for acquiring a functional correspondence with which cases different from the examples are assimilated at almost no cost of accommodation.

Meaning is often taken as a guarantee of understanding. It is a typical example of an ideal element, postulated as responsible for successful communication. Its presence is only internal to the system that understands, but it is often difficult to define outside of the dynamics in which it is recognized.

For example, in an automatic dialogue system [9,10], the meanings of words are represented as numeric vectors of thousands of components (4096 in ChatGPT4). However, such a system never externally provides these representations. In other words, meanings as vectors oversee the proper functioning of the dialogue, but remain internal and inaccessible entities. This situation highlights the intrinsic value of ideal elements and the complex relationship between abstraction and levels of existence. Something that exists at a certain level of observation may not exist at another level, but more importantly, not all levels of existence are accessible or fully accessible.

It is impressive, but in line with Piaget's speculation [32], that general principles, such as **abstraction, ideation,** and **reflexion**, individuated in metamathematics, are also acting in the organization of cognition structures: abstraction (Russell [34]), ideation (Hilbert [33]), reflexion (De Giorgi [35]: "All the mathematics developed up a given time can become object of the following mathematics").

Surprisingly, in his 1950 essay [14], Alan Turing, discussing machines with conversational abilities comparable with humans, considered the aspect of an intrinsic ignorance of the trainer about the processes that oversee the learned behavior. This aspect is partially connected to the fact that complex behavior requires the presence of randomness, to confer flexibly to the situations in their specific contexts.

Claude Shannon also sensed the possibility of machines that learn and outlined various possibilities with which a machine can realize this competency [13]; among these, the general scheme that has emerged in Machine Learning: examples, errors, approval, disapproval. In any case, the profound difference between programming and training is in the autonomy of the latter, which is an internal process escaping any complete control from external. In this intrinsic autonomy perhaps lies the basis of "intelligence".

Although pioneers like Turing and Shannon had intuited the possibility of machines capable of learning, this paradigm emerged gradually and through contributions across classical computability and many other fields [16,36,37]. Backpropagation links learning to classical mathematical themes that date back to Legendre, Gauss, and Cauchy, problems that require the baggage of mathematical analysis (see the proof of the theorem in Section 3), back to Newton and Leibniz, with Leibniz being the same thinker who had envisaged the possibility of a "Calculus Ratiocinator".

In any case, current experience in Machine Learning confirms that in cognitive systems, when certain levels of complexity are exceeded, there are levels the system cannot access. This is a necessary price to reach advanced levels of complexity and to develop reflective mechanisms that can lead to forms of consciousness. In summary, consciousness must be based on an "unconscious" support from which it emerges.

7. Conclusions

The latest significant advancement in artificial intelligence and machine learning has been the development of chatbots with conversational abilities comparable to humans. This development has highlighted another aspect already present in Turing's 1950 article [14], demonstrating its impressive visionary nature. Dialogic competence provides access to the rational and intellectual capabilities of humans. It is no coincidence that the Greeks used the same term (Logos) to denote both language and reason. Furthermore, it is no coincidence that the logical predicative structure of Aristotle's syllogism is based on a wholly general grammatical schema. The endeavor that led to the 2022 launch of the public platform Generative Pre-trained Transformer (ChatGPT3.5) by Open AI is undoubtedly a historic scientific and technological enterprise.

Many factors contributed to the success of ChatGPT, including immense computing power and a vast availability of data for training development. From a scientific viewpoint, a fundamental element was the development of Large Language Model (LLM) systems [9,10] with the introduction of the transformer-based perspective. Before achieving dialogic competence, these systems learned to represent the meaning of words and phrases by transforming them into vectors of thousands of dimensions over rational numbers. This competence provides the abstraction capacity on which a genuine construction of meaning is based. Such meanings could be different from those of a human mind; however, it is entirely reasonable to assimilate them to abstract representational structures that emerge through acquiring appropriate linguistic competencies (comparisons, analogies, differences, similarities, etc.).

In [38], a logical representation system for texts was defined and described to ChatGPT3.5 through various conversations, and the system was questioned, revealing a satisfactory ability to acquire and correctly use the method. The experiment suggests a possibility that can help to investigate applications of chatbots in contexts of complex interactions and in developing deep forms of conceptual understanding.

A detailed analysis of intellectually rich dialogues will be a type of activity of great interest shortly. In the context of dialogues with ChatGPT4 (an advanced version of ChatGPT3.5, accessible with a payment for the service), a possibility has emerged that is interesting to report as a reflection on possible extensions of the competencies of these artificial systems.

Even though the system can track the developed discourse and the concepts acquired during its course, at the end of it, there is no stable trace of the meanings of built-in interaction with the human interlocutor. In other words, the understanding acquired for dialogue does not leave a trace in the semantic network that the chatbot possesses. This observation could suggest mechanisms that, when appropriately filtered to ensure security, confidentiality, and adherence to ethical behavior principles, could adapt to individual users and import conceptual patterns and typical aspects in a stable manner that could make the cognitive development of these systems more fluid and personalized. Similarly to the algorithms that operate on many current platforms, which learn users' tastes and preferences to better adapt to their needs, one could think of forms of "loyalty" that would constitute an advantage for the interlocutors, but especially mechanisms of "induced" training useful for the cognitive development of the systems.

Continuing to develop this perspective, one would obtain a population of variants of the same chatbot that express different modes and levels of development corresponding to the reference interlocutors. The variants of higher intellectual levels would exhibit certain competencies more significantly. In an experimental phase, a restricted group of enabled interlocutors could be observed, and the trainers could control the phenomenon's effects and draw from it useful elements to understand the logics that develop and the potentials that open up. The results obtained in this way are comparable to diversified training for the same type of system. It would be as if different instances of the same chatbot were trained on the texts of the conversations with their respective interlocutors.

Suppose a certain system is exposed to a course in Quantum Physics, or perhaps to more courses in Physics, or another specific discipline. Training of this type is much more complex than that based on the assimilation of texts and supervised training developed by even very experienced trainers. To achieve interesting results, the system should have forms of "gratified attention" more complex than simple approval or disapproval, but the form of learning that would be obtained could be so profound as to produce specialists in scientific research.

As evidence of this idea, we can report that the formulation of Learning Equations, given in the previous section, was obtained by starting from the description of back-propagation given in [7] by a long conversation with ChatGPT, which helped the author in deriving the equational synthesis expressed by (9)–(11).

In conclusion, it seems appropriate to recall Turing's concluding remarks in their visionary essay [14] on thinking machines: "We can only see a short distance ahead, but we can see plenty there that needs to be done".

Funding: This research received no external funding

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Minsky, M. *Computation. Finite and Infinite Machines*; Prentice-Hall Inc.: Hoboken, NJ, USA, 1967.
2. Turing, A.M. On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **1936**, *58*, 230–265.
3. McCulloch, W.; Pitts, W. A Logical Calculus of Ideas Immanent in Nervous Activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
4. Gelb, W.; Kirsch, B. *The Evolution of Artificial Intelligence: From Turing to Modern Chatbots*; Tulane University Archives: New Orleans, LA, USA, 2024. Available online: https://aiinnovatorsarchive.tulane.edu/2024/ (accessed on 8 June 2024).
5. Cybenko, G. Approximation by superposition of a sigmoid function. *Math. Control Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
6. Hornick, K.; Stinchcombe, M.; White, M. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
7. Mitchell, T. *Machine Learning*; McGraw Hill: New York, NY, USA, 1997.
8. Nielsen, M. *Neural Networks and Deep Learning*; Determination Press: San Francisco, CA, USA, 2013.
9. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *NEURIPS* **2020**, *33*, 1877–1901.
10. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling Laws for Neural Language Models. *arXiv* **2020**, arXiv:2001.08361.
11. Suszyński, M.; Peta, K.; Černohlávek, V.; Svoboda, M. Mechanical Assembly Sequence Determination Using Artificial Neural Networks Based on Selected DFA Rating Factors. *Symmetry* **2022**, *14*, 1013. [CrossRef]
12. von Neumann, J. *The Computer and the Brain*; Yale University Press: New Haven, CO, USA, 1958.
13. Shannon, C.E. Computers and Automata. *Proc. Inst. Radio Eng.* **1953**, *41*, 1234–1241. [CrossRef]
14. Turing, A.M. Computing Machinery and Intelligence. *Mind* **1950**, *59*, 433–460. [CrossRef]

15. Wiener, N. *Science and Society*; Methodos: Milan, Italy, 1961.
16. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **1958**, *65*, 386–408. [CrossRef] [PubMed]
17. Manca, V.; Bonnici, V. Life Intelligence. In *Infogenomics*; Springer: Berlin/Heidelberg, Germany, 2023; Chapter 6.
18. Manca, V. *Infobiotics: Information in Biotic Systems*; Springer: Berlin/Heidelberg, Germany, 2013.
19. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
20. Kelley, H.J. Gradient Theory of Optimal Flight Paths. In Proceedings of the ARS Semi-Annual Meeting, Los Angeles, CA, USA, 9–12 May 1960.
21. Le Cun, Y. Une Procédure d'Apprentissage pour Réseau à Seuil Asymétrique. In Proceedings of the Cognitiva 85: A la Frontiere de l'Intelligence Artificielle des Sciences de la Conaissance des Neurosciences, Paris, France, 4–7 June 1985; pp. 599–604.
22. Parker, D.B. Learning logic. In *Technical Report TR-47. Center for Computational Research in Economics and Management Science*; MIT: Cambridge, MA, USA, 1985.
23. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *329*, 533–536. [CrossRef]
24. Werbos, P. Beyond Regression: New Tools for Prediction and Analysis in Behavior Sciences. Ph.D. Thesis, Harvard University, Cambridge, MA, USA, 1974.
25. Werbos, P. Backpropagation Through Time: What It Does and How to Do It. *Proc. IEEE* **1990**, *78*, 1550–1560. [CrossRef]
26. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
27. Church, A. A note on the Entscheidungsproblem. *J. Symb. Log.* **1936**, *1*, 40–41. [CrossRef]
28. Awret, U. Holographic Duality and the Physics of Consciousness. *Front Syst Neurosci.* **2022**, *15*, 685699. [CrossRef] [PubMed]
29. Basodi, S.; Ji, C.; Zhang, H.; Pan, Y. Gradient Amplification: An Efficient Way to Train Deep Neural Networks. *Big Data Min. Anal.* **2020**, *3*, 3. [CrossRef]
30. Hinton, G.E.; Osindero, S.; Teh, Y. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]
31. Kandel, E.R. Search of Memory. In *The Emergence of a New Science of Mind*; W. W. Norton & Company, Inc.: New York, NY, USA, 2006.
32. Piaget, J. *L'epistemologie Génétique*; Presses Universitaires de France: Paris, France, 1970.
33. Hilbert, D. Über das Unendliche. *Math. Ann.* **1926**, *95*, 161–190. [CrossRef]
34. Russell, B.; Whitehead, A.N. *Principia Mathematica*; Cambridge University Press: Cambridge, UK, 1910.
35. De Giorgi, E. *Selected Papers*; Dal Maso, G., Forti, M., Miranda, M., Spagnolo, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2006.
36. Hebb, O. *Organization of Behaviour*; Science Editions: New York, NY, USA, 1961.
37. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558. [CrossRef]
38. Manca, V. Agile Logical Semantics for Natural Languages. *Information* **2024**, *15*, 64. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.