



Received: 08 September, 2023
Accepted: 20 September, 2023
Published: 21 September, 2023

***Corresponding author:** Thaddeus JA Kobylarz, PhD, Department of Wireless Technology, Bell Laboratories Retiree, USA, Tel: 973-539-3086, 873-539-3086; E-mail: t.kobylarz@ieee.org

ORCID: <https://orcid.org/0000-0003-4129-5944>

Keywords: Neural networks; Structural neural plasticity; Functional neural plasticity; Associative learning; Inter-association; Intra-association; Memory formation; "Gestalt" phenomenon; Linearly separable; Nonlinearly separable; Excitatory synapses; Inhibitory synapses

Copyright License: © 2023 Kobylarz TJA, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

<https://www.peertechzpublications.org>



Check for updates

Review Article

Neurological Properties to Circumvent AI's Error Reduction Impasse

Thaddeus JA Kobylarz^{1*} and Erik J Kobylarz²

¹Department of Wireless Technology, Bell Laboratories Retiree, USA

²Geisel School of Medicine & Thayer School of Engineering, Dartmouth College, USA

Abstract

Our paper proposes significant changes to AI technology. We believe this is necessary because current implementations have stagnated at average error rates of approximately 8%. Implementers hope that further improvements will lower error rates to 5% by 2025. This would require 10^{28} floating-point operations, which is not possible with today's algorithms and computer technology. Even errors of 5% are excessive for many practical applications.

The current AI implementations have ignominious errors. Near bankruptcy of a prominent real estate corporation, and the obligatory resignation of an elected government official resulted from AI errors. The causation errors were ludicrous and unlikely performed by humans. Applications of AI are therefore limited to those for which errors are nugatory.

In contrast, the human brain's capabilities and efficiency are astonishing. In significant contrast to current AI models, the human brain is impressive in terms of its relatively small size (adult average 79 in³), weight (approximately 4#), and power consumption (nominally 15W). We feel that this implies that AI technology needs to adopt excluded neurological properties.

The current AI neuron model is an overly simplified linear model, which was proposed about 70 years ago. We propose emulating the neurological neuron's nonlinear capabilities. The versatility of the improved AI model would be many orders of magnitude beyond that of the currently implemented linear neuron models.

Also, the proposed neurological properties are of neural plasticity. Specifically, we describe the neurological associative learning aspect of neuroplasticity, partitioning associative plasticity into "inter-association" (neural network structure), and "intra-association" (neuron functioning).



Abbreviations

AI: Artificial Intelligence; CO₂: Carbon Dioxide; LED: Light Emitting Diode; W: Watts; #: Pounds; in: Inches.

Introduction

In this section, we argue for the need to drastically alter current AI technology. The ongoing small improvements will not successfully circumvent the error reduction impasse of today's AI technology. Colossal failures represent a rude awakening that the current AI technology has reached a dead end regarding applications for which errors are intolerable. The frequent hype of being able to arrive at decisions more accurately than humans, because of the ability to digest immense amounts of data, has been shown to be untrue for such applications.

In order to circumvent the error reduction impasse, we are proposing changes that will require a substantial investment in work and expense. There therefore needs to be extensive justification for the substantial investment. The justification is presented by an elaborate discourse identifying reasons for the impasse and other current AI implementation problems.

At first, touted to revolutionize [1] how real estate is bought and sold, this AI application nearly drove Zillow into bankruptcy [2]. Zillow lost \$304 million dollars, its stock dropped 18%, and 2000 employees were laid off in the third quarter of 2021. Needless to say, their AI division was dissolved [3]. Interestingly, the decisions to buy or sell properties were so ridiculous, that most humans would not have made them [3].

The inadequacy of current AI technology also destroyed the political career of a Netherlands prime minister [4]. This leader decided to use AI to determine the qualifications of families for the government's childcare allowance. The AI algorithm developed a pattern of falsely labeling claims as fraudulent, and harried civil servants rubber-stamped the fraud labels. So, for years the tax authority baselessly ordered thousands of families to pay back their claims, pushing many into onerous debt and destroying many lives in the process.

These are only two examples of erroneous conclusions made by the current AI technology. Many more exist. For example, training an AI program to perform mathematics problems [5], using hundreds of thousands of examples with step-by-step solutions, was a disaster. Following this training, the AI program yielded an average accuracy of only 5 percent (95% errors) for high school algebra and trigonometry problems.

Very poor results also exist for examples of AI use in medical applications. It has been reported that 85 percent of studies using machine learning to detect COVID-19 in chest scans failed reproducibility and quality checks. None of the models was deemed ready for use in clinics [6]. In addition, it has been discovered that a nationally deployed healthcare AI algorithm in the United States was racially biased, affecting millions of Americans [7]. The AI algorithm was designed to identify which patients would benefit most from intensive-care

programs, but it routinely enrolled healthier white patients into such programs ahead of black patients who were generally in poorer health.

There is now overwhelming evidence that AI performance can often prove to be unstable [8]. A slight alteration in received data can lead to a wild change in outcomes. For example, it has been demonstrated that changing a single pixel on an image can cause an AI application to consider a horse image to be a frog [9]. In addition, medical images can be modified in a way that is imperceptible to the human eye such that a misdiagnosis of cancer occurs 100 percent of the time [8].

The best AI deep-learning systems designed for recognizing objects currently have an error rate of approximately 8% and are projected to improve only to about 5% by 2025 [10]. It is anticipated that achieving a 5% error rate would require 10¹⁹ billion floating-point operations [10], which is regarded to not be possible with today's AI algorithms and computer technology [10]. Nevertheless, the 5% error rate is unacceptable for many applications, such as AI autonomous vehicles. At a 5% error rate, an error is expected for every 20 times the vehicle is used. Because an error may result in an accident that could include a fatality, the error rate should be no more than the deadly crash rate of 0.00135% [11] per 1000 miles of travel in 2022. We are confident all AI practitioners will agree this error rate is impossible with today's technology. There is no alternative but to make major AI technology changes in order to further reduce error rates.

Another application, which obligates a near 0% error rate, is for an internet search engine. Software programs have been developed by Microsoft (Bing Chat) and Google (Bard) to perform online searches using AI [12]. These programs are intended to allow personalized, conversational search experiences by means of a chat between the user and the search engine. Bing Chat has been found to fabricate sources and facts when none are available in order to support its assertions [12]. "The danger is that, taken at face value and without checking the sources, the user may be misinformed or even misled by the result" [12]. Google's Bard had the same problem having produced factual errors in its demo [12,13]. Even a promoter (Riedl) expressed, "It's difficult to predict if or when conversational AI searches will reach a level of accuracy that's acceptable for users". This suggests that after an AI-guided search is completed, one needs to perform a conventional search on the topic to ascertain AI's search accuracy. Our conclusion is "Why bother with the AI search in the first place?". The anticipated cost to implement an AI search engine using current AI technology is over \$100 billion to develop the new server and new network infrastructure. In addition, new recurring costs for such technology are estimated to be \$36 billion [12]. Therefore, a more efficient and accurate AI technology is required to significantly reduce these astronomical costs.

Catastrophic forgetting, the tendency of AI implementations to entirely and abruptly forget information it previously knew after acquiring new information is another AI weakness [9]. Due to inefficient memory use, AI programs overwrite past

knowledge with new knowledge. As a consequence, the abysmal memory of artificial neural networks imposes many hours of repeated training.

Errors represent only one aspect of AI's deficiency. Significant energy consumption for training makes it impractical for many applications. Training a single AI model can result in the emission of as much carbon as five cars produce in their lifetimes [14]. While training an AI application to only 13 percent completion, its graphics processing unit emitted almost as much carbon as powering a home for a year in the United States [15]. A study found that training an off-the-shelf AI language-processing system produced 1,400 pounds of carbon emissions. This is approximately the amount produced by flying one-person roundtrip between New York and San Francisco [16]. The full suite of experiments needed to build and train an AI language system from scratch can generate up to 78,000 pounds of CO₂, which is twice as much as the average American exhales over an entire lifetime [16].

By comparison, the human brain, which is less than the size of three softballs, can perform astounding reasoning with very little energy consumption. The total human power rate typically varies between 45 and 85 Watts [17]. Of the total power, 20% to 25% may be consumed by the human brain. This implies human brain consumption is between 9 and 21 Watts (power in the range of a single LED lamp). Obviously, AI researchers have much to learn from the human brain's efficiency. In the sections that follow, neurological properties are described from which significantly more efficacious neural networks will result.

Two significant conclusions are that current AI applications should be limited to those that can justify the pollution resulting from training and have only innocuous errors. When errors of serious consequences may result, then conclusions drawn from AI applications should be considered as suggestions for actions, and humans need to intercede in order to determine the proper actions. A frequent response to AI errors is that larger neural networks and more training would preclude such errors. The problem with this retort is that current AI technology has encountered a size/training limit [10]. In order to improve error rates, the computing resources and energy required to train such a future system would be enormous, leading to the emission of as much carbon dioxide as New York City generates in one month [10].

The neuron model currently being deployed is based on neurological evidence published about 70 years ago [18-20]. For AI to progress and evolve beyond its current limit, more neuroscience principles need to be adopted [21]. We have undertaken the task of introducing, heretofore not considered neurological properties to AI technology in this paper. The neurological aspects of plasticity's associative learning and reasoning are proposed to be the basis of the next generation of AI technology.

Neural plasticity

Neural plasticity, also known as neuroplasticity, is the ability of neural networks in the brain to change through

growth and reorganization [22]. Through this mechanism, the brain is rewired to function in some way that differs from how it previously functioned. Neural plasticity was once thought to occur only during childhood, but research in the latter half of the 20th century showed that many aspects of the brain can be altered (or are "plastic") even through adulthood [23]. It is true that the developing childhood brain exhibits a higher degree of plasticity than the adult brain. However, it should be noted that neuroplasticity can occur not only with brain development, but also in neurologic diseases, such as epilepsy, as well as recovery from neurologic insults, e.g., following a stroke, and brain injury [24].

The associative learning aspect of neuroplasticity is examined in this paper. It is through associative recall that such learning is utilized. We regard associative learning and its corresponding associative recall to be an extremely important, if not the most important cognitive process. To demonstrate associative recall, consider the word "dog". After reading this word, one or more associations come to mind. For example, an image of one's dog may appear, or one may be reminded of an incident with an aggressive dog, etc. Such associations are vital to our thinking process. Recalling procedures, used in solving past problems provide insight to solve new problems. Associations guide us through a day's activities, such as dressing, washing, eating, etc.

Two broad categories of neural plasticity exist [19]: 1) structural neural plasticity (the brain's ability to change its neuronal connections) and 2) functional neural plasticity (the brain's ability to alter and adapt the functional properties of neurons). Virtually all past work has involved functional plasticity and nearly none concerns structural plasticity. However, structural plasticity is indeed very important to learning and takes place throughout a human's lifetime; this suggests the performance of artificial neural networks would benefit from this type of learning. The associative aspect of plasticity will be applied to each of these two categories. For structural neural plasticity, inter-association learning and recall will be described. For functional neural plasticity, a description of intra-association learning and recall is presented.

Structural neural plasticity's inter-association

Inter-association deals with the axonal (output) connection of one neuron that connects to the dendrites (inputs) of another neuron. This is a structural connection that establishes an association between the two neurons.

An example of structural plasticity's inter-associative learning is illustrated in Figure 1 [25]. The growth of a neural circuit can be considered to consist of three steps. Its catalyst is the simultaneous axonal firing of neurons. When pairs of neurons fire together (step 1) they become strongly linked (as with memory development). Their simultaneous firing or "potentiating" may also forge a link (step 2) to a nearby third neuron (network growth).

Through repeated firing, the 3 neurons become strongly

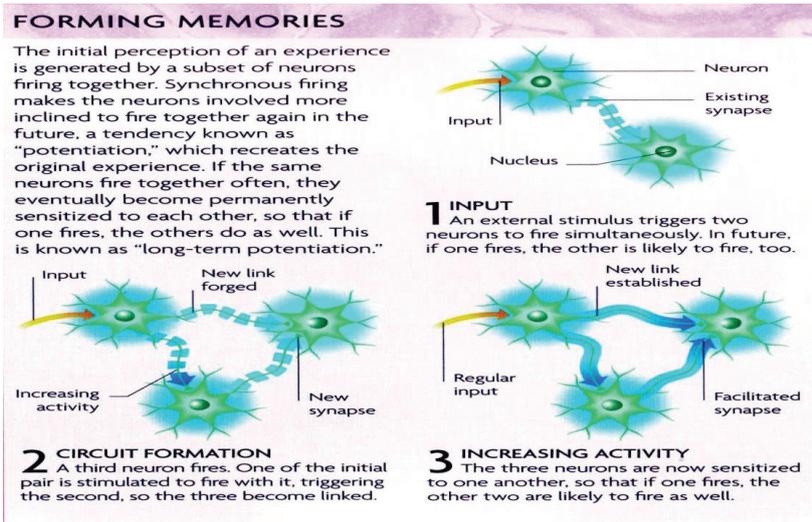


Figure 1: Structural Plasticity Example of Neural Inter-Associative Circuit Growth [25].

linked (step 3). The newly formed neuron interconnections link or inter-associate the neurons. This phenomenon enables the formation of recall memories as a result of neuron network development. The inter-association process of Figure 1 is a potential basis for an artificial neural network growth algorithm.

It is important to underscore the difference between the example just provided and current AI neural networks. The current AI neural networks have fixed configurations and only the weights of connections change, usually by means based on the Hebb/Allport posit [26]. The AI deep-learning models are overparameterized, which is to say they have more parameters than there are data points available for training [10]. These neural networks are overly cumbersome, in that many extraneous interconnections exist.

No superfluous neuron connections occur in the process just described. By employing structural plasticity only necessary connections are forged with inter-associated neuron activity. Thus, the neural network has changed because of an associative need and not with an a priori-defined structure. This results in a more efficient use of the software for the neural network of an application.

A potential scenario for network growth by inter-association begins with a primitive network, which only contains the minimal connections necessary for growth. Included in the primitive network are peripheral neurons (peripheral nervous system [27]) connected to sensory detectors (afferent division [27]), as for visual, auditory, somatosensory, etc. inputs. A second layer of neurons (central nervous system [27]) exists beyond the peripheral neuron layer. Interconnections between the peripheral and central layers are, at first, minimal. As training ensues, the inter-associative firing of neurons will create more neuronal connections. It is also possible for a primitive network connection to be removed and even some neuronal atrophy may take place during training.

Many neuronal central nervous system layers exist. These

deeper layers can be relegated to specialized applications, such as pattern recognition, memory recall, etc. Interconnections among deeper layers are also minimal, prior to training. Inter-associative learning will result in the growth of more interconnections and inter-associations, which are possible among these application layers. For example, a pattern recognition application could become interconnected with an application that recalls name spelling.

Functional neural plasticity will also develop via intra-association, simultaneously with structural growth as training proceeds. The next section addresses functional neural plasticity. Papers are being prepared that address algorithms for the previous inter-association scenario and a forthcoming intra-association scenario.

Another important neurological structural property to incorporate is the "Gestalt" phenomenon. A simple definition of Gestalt is to observe the whole. Gestalt psychologists emphasize that organisms perceive entire patterns or configurations, not merely individual components [28]. The view is sometimes summarized using the adage, "The whole is more than the sum of its parts". The Gestalt principles; of proximity, similarity, continuity, closure, and connection describe human perception in connection with different objects and environments [29].

The utilization of Gestalt in AI can be demonstrated in Figure 2. Termed "invariance" [30], the same object is concluded in its various orientations. This is because each orientation shows the same salient features to create a holistic image of the object.

The salient features provide the holistic Gestalt view of an image, irrespective of size, orientation, etc. Considering features, rather than the current AI approach of considering pixels, will reduce errors considerably. Mistaking a horse image to be a frog, due to a single pixel change [9], likely evokes the reaction "This is ridiculous, a horse looks nothing like a frog". This reaction implies a human's recognition of a horse by means of holistic features yields no similarity to the features

of a frog. Changing one pixel may distort a feature, but it will not alter the remaining features. This strongly implies that the current AI “pixel only” means for recognition is neurologically inconsistent and Gestalt feature recognition is the proper methodology.

Additionally, the current AI limitation of confidently recognizing only images that have been used in training will not exist if the Gestalt holistic means of recognition is adopted. Being able to recognize an image not used in training is currently an AI matter of luck. Hence, the Gestalt feature recognition process will considerably reduce errors, training time, and memory requirements.

A caveat exists for the Gestalt holistic approach. In Figure 3, a triangle is perceived although no triangle exists [30]. This is because our brains utilize the features of three vertices to identify the image of a triangle. Although a triangle is visualized, the image is of three black globs, each having a “V” cut. A triangle contains three lines with the three possible line pairs connected at three vertices without crossing the lines. For such mistaken conclusions, training must add features, such as a straight line connecting each pair of vertices, to remedy errors.

The illusory triangle is shown in Figure 3 for another purpose. It manifests the point that human image recognition is based on salient features to represent an entire object. Because the three vertices are used to identify the triangle, the visualized triangle can have its shape changed or be rotated and still be identified as a triangle, as long as the features exist. Figure 4 illustrates this phenomenon, showing a rotation and a size/shape change. This illustrates that the adoption of Gestalt feature recognition in AI has the potential to reduce errors, training, and memory requirements; underscoring the value of Gestalt feature recognition.

We envision the transformation from pixels to features to be performed by means of a layered neuron network. The layering

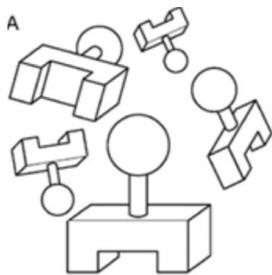


Figure 2: Various Sizes and Orientations of the Same Object [30].

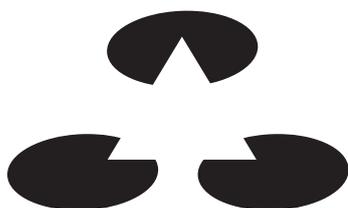


Figure 3: Vertex Features of a Triangle [30].



Figure 4: Illusion of Triangles Irrespective of Orientation or Shape.

will not resemble that currently used for AI deep learning networks. Rather, the first layer will begin with primitive features formed by the pixels, which constitute elementary shapes. By training via inter-association, succeeding layers will make connections that compound features of prior layers to form more complex features. The final layers will combine the most complex features to identify observed objects.

Functional neural plasticity's intra-association

Intra-association deals with the relationship of a neuron's dendrites (inputs). A neuron has many dendrites connected to many neurons. These many dendrites play a functional role in a neuron's operation, termed a functional inter-association of a neuron's plasticity.

A major purpose of this section is to provide sufficient information to write a computer program for a new neuron model. Hence, it contains a mathematical definition, a proof of completeness, a basis for a training algorithm, training examples, and sundry model properties. A casual reader may desire to merely skim through the new model's definition and perhaps read only portions of interest.

Functional neural plasticity involves neuronal properties. Although the currently deployed AI neuron model and the model we propose both deal with weights associated with a neuron's axonal inputs, our model is significantly different.

A major distinction is linearity. The AI-deployed neuron model's linear threshold function severely limits its logic capability to only linearly separable functions [31].

Shown in Table 1, are the amounts of possible logic functions for linearly separable functions $L(n)$ [32] and the total number of logic functions $T(n)$, with respect to the number (n) of axonal inputs.

Where the total number $T(n)$ of logic functions for “ n ” inputs is given by:

$$T(n)=2^{2^n} \quad (1)$$

The exceedingly rapid growth of $T(n)$, as shown in Table 1, illustrates the weakness of the currently deployed linear neuron model. For $n=7$ there are greater than 4.06×10^{28} (41 octillion) times more functions available than those that can be realized by AI linear deployments. Even more startling is that a human neuron receives an average of 10^3 to 10^4 inputs [33] from other neurons, suggesting the percentage of linearly separable functions is essentially zero for physiological neurons.

A partial plot of the ratio versus inputs is shown in Figure



equation represents a template from which pertinent variable product terms are extracted by means of intra-associative learning. Assignments of non-zero weights, by means of a training algorithm, provides the neuron model.

Examples will be given of the intra-associative learning process. But first, a proof of completeness will be provided. The proof is by means of an algorithm that will assign threshold weights for an arbitrary logic function, linear or nonlinear, according to the complete equation. Demonstrating that the algorithm will always provide an assignment of threshold weights such that $F_n(\mathbf{X})$ satisfies an arbitrary $f_n(\mathbf{x})$ of "n" variables represents proof that the equation is complete. The algorithm utilizes a truth table (Table 2) to assign the weights to the complete threshold function, according to any logic function $f_n(\mathbf{x})$.

The truth table (Table 2) has its variables specified in the complete threshold function $F_n(\mathbf{X})$, with columns ordered as (x_n, \dots, x_1) . The rows are arranged according to the computation sequence specified in equation (3).

The truth table begins with row "0", and thereafter rows are successively numbered. Row "0" has all (x_n, \dots, x_1) components equal to "0" and is the means to assign the threshold " θ_n " value. The row "0" assignment is "1" if the logic function has

a "0" value. Otherwise, the assignment for " θ_n " is "0".

Except for row 0, each successive row is intended to assign a weight for the $F_n(\mathbf{X})$ variable product term associated with the row; i.e., for the variables with a "1" value in $F_n(\mathbf{X})$; where $(X_n, \dots, X_1) = (x_n, \dots, x_1)$. The $F_n(\mathbf{X})$ equation begins with the linear part, which has only one variable of (x_n, \dots, x_1) , per row, equal to "1", and the remaining variables all equal "0". The row variable succession of single "1's" is x_1, x_2, \dots, x_n . The $F_n(\mathbf{X})$ equation then addresses two variables having "1" values. This leads to row $(n^2 + n) / 2$. The process of successively including one more variable continues until the last row, in which all "n" variables have a value of "1".

Although not required, initially all weights of the $F_n(\mathbf{X})$ equation equal "0". Only when necessary are weight changes made for a row. That is, if $F_n(\mathbf{X})$ satisfies the f_n value at a particular row, the weight for its corresponding product of variables remains the same. However, should $f_n = 0$ and $F_n(\mathbf{X}) \geq \theta_n$, a weight, for the corresponding product of variables of the row, is assigned to make $F_n(\mathbf{X}) < \theta_n$. Also, if $f_n = 1$ and $F_n(\mathbf{X}) < \theta_n$, a weight, for the corresponding product of variables of the row, is assigned to make $F_n(\mathbf{X}) \geq \theta_n$. That is, a weight change to a different value is only made if needed to have $F_n(\mathbf{X})$ satisfy $f_n(\mathbf{x})$ for the row being considered.

Table 2: Generalized Truth Table and Threshold Function Value Assignments.

Row	x_n	x_{n-1}	...	x_3	x_2	x_1	f_n Value and Weight Assignment
0	0	0		0	0	0	If $f_n = 0$, then assign $\theta_n \geq 1$; else assign $\theta_n \leq 0$.
1	0	0		0	0	1	$f_n = 0 \wedge F_n(\mathbf{X}) \geq \theta_n \Rightarrow$ assign $W_1 \ominus: F_n(\mathbf{X}) < \theta_n$ $f_n = 1 \wedge F_n(\mathbf{X}) < \theta_n \Rightarrow$ assign $W_1 \oplus: F_n(\mathbf{X}) \geq \theta_n$.
2	0	0		0	1	0	$f_n = 0 \wedge F_n(\mathbf{X}) \geq \theta_n \Rightarrow$ assign $W_2 \ominus: F_n(\mathbf{X}) < \theta_n$ $f_n = 1 \wedge F_n(\mathbf{X}) < \theta_n \Rightarrow$ assign $W_2 \oplus: F_n(\mathbf{X}) \geq \theta_n$.
3	0	0		1	0	0	$f_n = 0 \wedge F_n(\mathbf{X}) \geq \theta_n \Rightarrow$ assign $W_3 \ominus: F_n(\mathbf{X}) < \theta_n$ $f_n = 1 \wedge F_n(\mathbf{X}) < \theta_n \Rightarrow$ assign $W_3 \oplus: F_n(\mathbf{X}) \geq \theta_n$.
.							.
.							.
.							.
n	1	0		0	0	0	$f_n = 0 \wedge F_n(\mathbf{X}) \geq \theta_n \Rightarrow$ assign $W_n \ominus: F_n(\mathbf{X}) < \theta_n$ $f_n = 1 \wedge F_n(\mathbf{X}) < \theta_n \Rightarrow$ assign $W_n \oplus: F_n(\mathbf{X}) \geq \theta_n$.
n+1	0	0		0	1	1	$f_n = 0 \wedge F_n(\mathbf{X}) \geq \theta_n \Rightarrow$ assign $W_{1,2} \ominus: F_n(\mathbf{X}) < \theta_n$ $f_n = 1 \wedge F_n(\mathbf{X}) < \theta_n \Rightarrow$ assign $W_{1,2} \oplus: F_n(\mathbf{X}) \geq \theta_n$.
n+2	0	0		1	0	1	$f_n = 0 \wedge F_n(\mathbf{X}) \geq \theta_n \Rightarrow$ assign $W_{1,3} \ominus: F_n(\mathbf{X}) < \theta_n$ $f_n = 1 \wedge F_n(\mathbf{X}) < \theta_n \Rightarrow$ assign $W_{1,3} \oplus: F_n(\mathbf{X}) \geq \theta_n$.
.							.
.							.
.							.
$(n^2 + n) / 2$	1	1		0	0	0	$f_n = 0 \wedge F_n(\mathbf{X}) \geq \theta_n \Rightarrow$ assign $W_{n-1,n} \ominus: F_n(\mathbf{X}) < \theta_n$ $f_n = 1 \wedge F_n(\mathbf{X}) < \theta_n \Rightarrow$ assign $W_{n-1,n} \oplus: F_n(\mathbf{X}) \geq \theta_n$.
.							.
.							.
.							.
r	r_n	r_{n-1}		r_3	r_2	r_1	$f_n = 0 \wedge F_n(\mathbf{X}) \geq \theta_n \Rightarrow$ assign $W_r \ominus: F_n(\mathbf{X}) < \theta_n$ $f_n = 1 \wedge F_n(\mathbf{X}) < \theta_n \Rightarrow$ assign $W_r \oplus: F_n(\mathbf{X}) \geq \theta_n$.
.							.
.							.
.							.
$2^n - 2$	1	1			1	0	$f_n = 0 \wedge F_n(\mathbf{X}) \geq \theta_n \Rightarrow$ assign $W_p \ominus: F_n(\mathbf{X}) < \theta_n$ $f_n = 1 \wedge F_n(\mathbf{X}) < \theta_n \Rightarrow$ assign $W_p \oplus: F_n(\mathbf{X}) \geq \theta_n$. Where "p" indicates penultimate.
$2^n - 1$	1	1			1	1	$f_n = 0 \wedge F_n(\mathbf{X}) \geq \theta_n \Rightarrow$ assign $W_u \ominus: F_n(\mathbf{X}) < \theta_n$ $f_n = 1 \wedge F_n(\mathbf{X}) < \theta_n \Rightarrow$ assign $W_u \oplus: F_n(\mathbf{X}) \geq \theta_n$. Where "u" indicates ultimate.



Theorem

All switching functions “ f_n ” of “ n ” variables have a realization by means of the complete threshold function “ F_n ”.

Proof

The proof is based on the preceding truth table which represents an algorithm, termed “weight assignment”. Each row of the truth table will imply an independent weight. This independent weight can be assigned any value to satisfy the needed weight summation for that row’s switching function value. This weight independence permits a set of weights to accommodate any switching function. Showing that the weight assignment algorithm assigns the “ F_n ” weights and a “ θ_n ” that satisfies $f_n(x_n, \dots, x_1)$ for all 2^n possible product variable values, represents proof that “ F_n ” is complete.

Consider now row “ r ”. If “ F_n ” for row “ r ” is consistent with the required value for “ f_n ”, the weight “ W_r ” is not to change. Maintaining the value of “ W_r ” will have no impact on any preceding row’s weight. This is because all input variables of preceding rows either have the same quantity of “1” value or fewer “1” values. Hence, at least one of the “ r ” row’s product terms (variable with a “1” value), has a “0” value in a preceding row. This means that row “ r ” has no impact on preceding rows.

There may be an impact on a succeeding row. However, a succeeding row will have an opportunity to adjust its weight, following the consideration of row “ r ”. Just as the weight of row “ r ” had no impact on a preceding row’s weight, the weight of rows, succeeding row “ r ”, will have no impact on row “ r ”.

If “ $W_r = 0$ ” and “ F_n ” is consistent with the required value for “ f_n ”, then the “ F_n ” product term for row “ r ” is unnecessary and should not appear in the neuron network. Eliminating this row will save memory, reduce training, and increase processing speed. If the value “ W_r ” causes “ F_n ” to be inconsistent with the required value for “ f_n ”, then the algorithm instructs “ W_r ” to be changed in a manner to make “ F_n ” and “ f_n ” consistent. As existed for not needing a change to “ W_r ”, changing the value of “ W_r ” will also have no impact on rows preceding “ r ”. Although rows succeeding “ r ” may be impacted, when these rows are considered, their weights are adjusted so that “ F_n ” is consistent with the required value for “ f_n ”.

Hence, execution of the assignment algorithm will result in having all 2^n rows consistent with the defining equation relating “ f_n ” and “ F_n ”. This implies that it is possible to assign weights that provide a threshold function to satisfy an arbitrary logic function, having any number of variables, and the threshold function is complete. **Q.E.D.**

As an example, the weight assignment algorithm will now be used to determine the nonlinear threshold function for the logic function Table 3.

$$f_e = x_1 x_2' x_3' + x_1' x_2 x_3' + x_1 x_2' x_3 + x_1' x_2 x_3 \tag{4}$$

For three variables, the complete threshold function is:

$$F_e = X_1 W_1 + X_2 W_2 + X_3 W_3 + X_1 X_2 W_{12} + X_1 X_3 W_{13} + X_2 X_3 W_{23} + X_1$$

$$X_2 X_3 W_{123} \tag{5}$$

The assignment algorithm truth table is shown in Table 4. Initial weights and “ θ ” are assigned “0”. Only integer values, beginning with “0” will be used.

Observe that “ W_3 ” remains at its initial value of “0”, implying that the input “ x_3 ” is superfluous and can be removed from the input. Setting “ $x_3 = 0$ ” yields the “exclusive or” function:

$$f_e = x_1 x_2' + x_1' x_2 \tag{6}$$

The corresponding threshold function for the example is:

$$F_e = X_1 + X_2 - 2 X_1 X_2; \theta_e = 1. \tag{7}$$

Also observe that for three variables, equation (3) has seven terms ($2^n - 1$). However, “ F_e ” contains only three terms or 42%. More will be discussed later concerning having fewer realization terms than included in the complete equation.

Attempts have been made to determine functions to create multi-dimensional surfaces to separate nonlinear logic functions [32]. For example, threshold functions represented by polynomials and also splines were attempted, but with very limited success. Such functions could only be applied to certain classes of nonlinearly separable logic functions and were never adopted by an AI system. The only other more recent references were abstracts published by the authors of this publication [36,37].

We contend that multi-dimensional hyperbolic functions are capable of separating any nonlinear function. A basic hyperbola, in two dimensions, can be expressed as $X_2 = C_{1,2}/X_1$

Table 3: Truth Table for $f_e = x_1 x_2' x_3' + x_1' x_2 x_3' + x_1 x_2' x_3 + x_1' x_2 x_3$.

x_3	x_2	x_1	f_e
0	0	0	0
0	0	1	1
0	1	0	1
1	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1
1	1	1	0

Table 4: Example of the Weight Assignment Algorithm.

x_3	x_2	x_1	f_e	F_e Before	Algorithm Assignment	F_e After
0	0	0	0	0	$\theta = 1$	0
0	0	1	1	0	$W_1 = 1$	1
0	1	0	1	0	$W_2 = 1$	1
1	0	0	0	0	None	0
0	1	1	0	2	$W_{12} = -2$	0
1	0	1	1	1	None	1
1	1	0	1	1	None	1
1	1	1	0	0	None	0



(where “ $C_{1,2}$ ” is a constant). Alternatively, it can be expressed as $X_1 X_2 = C_{1,2}$. Such products of variables are the main ingredient of the complete threshold function “ F_n ”. It is for this reason that hyperbolic functions are considered to be the means for the binary separation of nonlinearly separable functions.

Consider the “exclusive or” as an example:

$$f_e = x_1 x_2' + x_1' x_2 \tag{8}$$

$$F_e = X_1 + X_2 - 2 X_1 X_2 \theta_e = 1 \tag{9}$$

Manipulation of the equation for $F_e = \theta_e = 1$ results in:

$$X_2 = (1 - X_1) / (1 - 2 X_1) \tag{10}$$

The locus of X_2 is plotted in Figure 6. Because we chose to satisfy the condition $f_n = 1 \Leftrightarrow F_n \geq 0$, by substituting $f_n = 1 \Leftrightarrow F_n = \theta = 1$, the values on the drawn hyperbolic lines represent a logical “1” result.

To suit the multi-dimensional aspect of our switching functions, we extrapolate hyperbolic functions to multi-dimensional hyperspace. For two dimensions a single asymptote, with separation lines, exists. Multiple asymptotes, with separation surfaces, exist in a hyperspace beyond two dimensions. For example, the product of $(X_1 X_2 X_3 W_{123})$, within “ F_n ”, would possess a three-dimensional hyperbolic surface with two asymptotes for the binary separation.

Corollary

All nonlinearly separable functions can be separated by n-dimensional hyperbolic surfaces.

Proof

The previous theorem had proven that the general threshold function “ F_n ” can realize any switching function. The nonlinear part of “ F_n ” contains products of variables in the form $(W_{i,j} X_i \dots X_j)$. For multi-dimensional space, the hyper-dimensional hyperbola is given by a product of two or more variables equal to a constant; i.e., $(X_i \dots X_j = C_{ij})$. This signifies that all nonlinear terms of “ F_n ” correspond to hyperbolic surfaces.

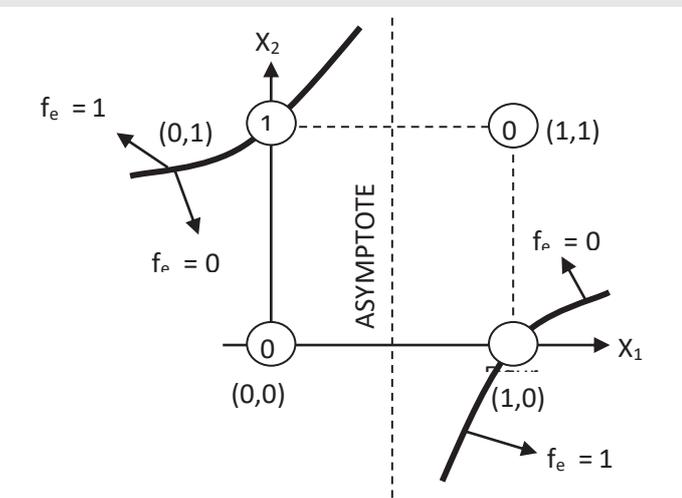


Figure 6: Hyperbolic Separation of “Exclusive Or” Function.

Assume that the product $(W_{i,j} X_i \dots X_j)$ is encountered while deploying the weight assignment algorithm and a positive weight is assigned to “ $W_{i,j}$ ”. Such an assignment is done to have $F_n \geq 0$. It further reveals that no smaller subset of $\{X_1, \dots, X_j\}$, that forms a product, would have resulted in $F_n \geq 0$. Otherwise, a positive weight is not assigned to “ $W_{i,j}$ ”. The hyperspace for $(X_i \dots X_j = C_{ij})$ can therefore be used to form a hyper-dimensional hyperbolic surface for the nonlinear separation. For $F_n \geq 0$, all of the associated variables must satisfy $X_1 = \dots = X_j = 1$. If one or more of these equals “0”, then $F_n < 0$.

Hence, all nonlinearly separable switching functions are separable by a hyperbolic function of two or more dimensions. For the linearly separable part of “ F_n ” a line or a (hyper) plane provides the separation. To create the hyperbolic function, one merely assigns $F_e = \theta_c$ (θ_c is the threshold constant) and algebraically manipulates the equation to have only one dependent variable (the equation’s left side). **Q.E.D.**

It should be noted that, although digital computations may be precise, a neuron contains analog computations having margins of error. Hence, any hyperbolic separation will contain an indeterminate region for which a value is uncertain. This indicates that variables within a threshold function must take on values such that “ F_n ” is always outside of its indeterminate region. To accomplish this, consider an indeterminate width of “ Δ ”. One assigns weights an amount “ Δ ” above that required in the original algorithm.

For the previous example, the threshold function becomes:

$$F_e = (1 + \Delta) X_1 + (1 + \Delta) X_2 - 2 (1 + \Delta) X_1 X_2; \theta_e = 1 \tag{11}$$

Resulting in:

$$X_1 = 1 \wedge X_2 = 0 \Rightarrow F_e = (1 + \Delta) > 0 \tag{12}$$

$$X_1 = 0 \wedge X_2 = 1 \Rightarrow F_e = (1 + \Delta) > 0, \tag{13}$$

$$X_1 = 1 \wedge X_2 = 1 \Rightarrow F_e = 0 < 0 \tag{14}$$

By assigning $F_e = 1$ the equation becomes:

$$(1 + \Delta) X_1 + (1 + \Delta) X_2 - 2 (1 + \Delta) X_1 X_2 = 1 \tag{15}$$

Algebraic manipulation, to determine the hyperbola, yields:

$$X_2 = [1 - (1 + \Delta) X_1] / [(1 + \Delta) (1 - 2 X_1)] \tag{16}$$

Resulting in:

$$X_1 = 0 \Rightarrow X_2 = 1 / (1 + \Delta) \tag{17}$$

(That is, the left line of Figure 6 is lowered for $\Delta > 0$.)

And:

$$X_1 = 1 \Rightarrow X_2 = \Delta / (1 + \Delta) \tag{18}$$

(That is, the right line of Figure 6 is raised for $\Delta > 0$.)

Figure 7 [38] represents a means to posit the neurophysiological process that performs the “exclusive or” logic function. Observe that an axon may have branches that

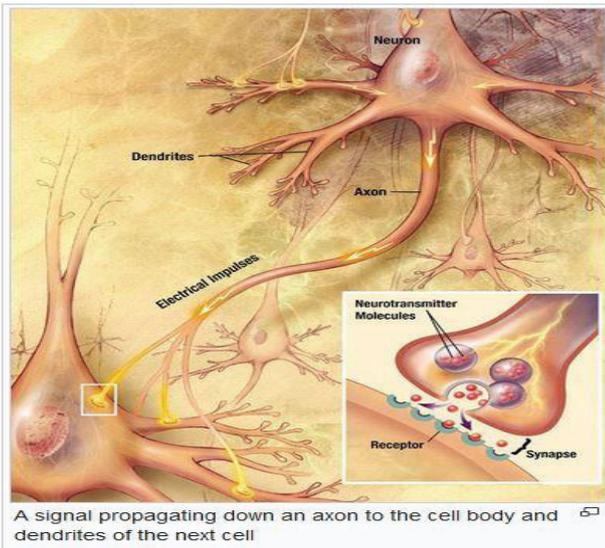


Figure 7: Neuronal Interconnections [38].

form synapses on both the soma and dendrites. The effect on the postsynaptic neuron is determined by the type of receptor that is activated, not by the presynaptic neuron.

Receptors can be either excitatory or inhibitory. This is significant because excitatory (Type 1) synapses are typically located on the shafts or the spines of dendrites; whereas inhibitory (Type II) synapses are typically located on a cell body [39]. The different locations of Type I and Type II synapses divide a neuron into two zones: an excitatory dendritic tree and an inhibitory cell body. This means that the postsynaptic neuron may interpret presynaptic axonal stimuli as excitatory or inhibitory. Whether the presynaptic neuron's axonal signal is excitatory or inhibitory depends upon the specific neurotransmitter released, as well as the training or the learning experience (intra-associative plasticity) of the postsynaptic neuron. That is, the neuron-applied weights of the excitatory and inhibitory synapses determine which of the two possible reactions will predominate.

With respect to the "exclusive or" function ($f_e = x_1 x_2' + x_1' x_2$), when " $x_1 = 1$ " and " $x_2 = 0$ ", the neuron soma training will have its " X_1 " excitatory weight supersede its inhibitory weight. This also exists for " X_2 " when " $x_1 = 0$ " and " $x_2 = 1$ ". However, by intra-association, the weight for the " $X_1 X_2$ " associated product will have a neuron soma-trained weight for which the inhibitory response prevails. Hence, an "exclusive or" function is achieved. An analysis of chemical/electrical neuron properties also reveals the neuron's capability of performing nonlinear logic [40].

As indicated earlier, all terms of equation (3) will not appear in the threshold function realization. The nonlinearly separable logic function $f_4 = x_1 x_2 + x_3 x_4$ provides such an example. By using the assignment algorithm, the threshold function is:

$$F_4 = X_1 X_2 + X_3 X_4; \theta_4 = 1 \tag{19}$$

Equation (3) will contain 15 terms ($2n - 1$). Whereas, " F_4 " has only two product terms (13%). We anticipate that

usually 20% to 40% of the equation (3) terms will appear in a realization.

The AI's McCullough-Pitts neuron model possesses other disparities with the physiological neuron. For instance, an AI "and" logic function requires a large percentage variation of the threshold as the number of inputs increases. A ten input "and" logic function requires a threshold five times as large as a two input "and" logic function. There is no neurological evidence that a neuron's threshold varies to such a degree.

We believe that the physiological neuron does not vary its threshold to accommodate "and" logic functions. Although some threshold variation has been observed, "Actual threshold variance is relatively low" [41]. An in vivo threshold, measured at a soma, varies between -52.1 mV and -42.2 mV [42]. By using -70 mV as a resting potential, one can show that the variation is only about 40%; which cannot even support an "and" function of two variables, since a two variable "and" function requires a threshold doubling. This indicates that the McCullough-Pitts model is neurologically inconsistent for "and" function realizations.

Explanations of the threshold variation included neuron accommodation of synaptic ionic densities, not for logic. Stated was that "two known ionic mechanisms were found to make the threshold adapt to the membrane potential, thus providing the cell with a form of gain control [42]".

Our weight assignment algorithm results in a per unit threshold value of "1", when not at "0"; which is per unit consistent with neurological evidence of only a per unit threshold value that equals 1. For this evidence, a McCullough-Pitts model becomes incapable of performing "and" logic. (A threshold of "0" is indicative of an unstable neuron, as action potentials occur continually when there is no stimulus to terminate the instability. This may suggest a means to terminate an epileptic seizure without medication.)

Another neurological disparity exists for the AI McCullough-Pitts model's use of "and" logic threshold variation. Unsustainably tight analog tolerances would exist for the analog portion of a physiological neuron, should a physiological neuron be only linear.

The relationship between the number of "and" logic inputs and acceptable tolerance for a linear threshold function, can be established by considering the following worst case scenario:

"Consider an "n" input "and" function such that all but one input has a "1" logic value."

The single nonconforming input does not occur and therefore has a "0" value. The following would therefore represent the threshold function and its threshold relation.

$$F_n(\mathbf{X}) = \sum_{i=1}^{n-1} W_i X_i < \theta \tag{20}$$

Representing " Δ " as the percentage variation of the variables results in the worst case equation:



$$F_n(\mathbf{X}) = \sum_{i=1}^{n-1} [(1 + \Delta) W_i X_i] < \theta, \Delta > 0 \quad (21)$$

Substituting per unit values of the threshold function permits the simplification:

$$F_n(\mathbf{X}) = \sum_{i=1}^{n-1} [(1 + \Delta) < n \quad (22)$$

Manipulation results in:

$$n < (1/\Delta) + 1 \quad (23)$$

Considering that analog systems have difficulty achieving more than a 1% ($\Delta = 0.01$) accuracy, had current AI systems incorporated analog threshold functions, “and” logic functions would not be error free with more than 100 inputs. Since a physiological neuron possesses thousands of inputs resulting in an infinitesimal tolerance, this precludes having an analog linear threshold function calculation for “and” logic. It is possible to simulate on, digital computers, such analog operations. However, in order to avail the large number of “and” logic inputs, this model is very wasteful of memory, training, and all computer processing.

The threshold tolerance problem is avoided by intra-association within the Kobylarz model. As Mother Nature’s infinite wisdom would conclude, rather than a sum, a product of 1’s and 0’s are utilized to evaluate an “and” function. The intra-associated product terms in the complete threshold function “ F_n ” are the analog portion representation of an “and” function. If any input component of the “and” logical function does not exist, its value of “0” is used in the product. Consequently, no threshold tolerance problem exists.

Conclusion

The current AI implementations have an average error rate of approximately 8%. It is projected that a 5% error will not be achieved until 2025, and at the enormous cost of 10^{19} billion floating-point operations. Computational cost is estimated to grow to at least the fourth power for improvement. A 10-fold improvement, for example, would require at least a 10,000-fold increase in computation. In practice, the actual requirements have been scaled to at least the *ninth* power. This means that to halve the error rate, one can expect to need more than 500 times the computational resources which is a devastatingly high price.

These statistics indicate that the current AI methodology has reached an impasse to error reduction that is essential for its progress. Because of the human brain’s efficiency, the obvious strategy is to incorporate more of the brain’s behavior into AI technology. We believe that the incorporation of plasticity into the structure and function of AI will yield a dramatic, versatile performance improvement. This will allow for the building of novel neural networks that evade the current error reduction impasse experienced by today’s AI technology.

The inter-association of structural plasticity, incorporating Gestalt principles, will reduce network size and training extent. During training, networks will be created that only have the

number of neuron models and their interconnections needed for an application. Functional plasticity and its intra-association training, using the Kobylarz neuron model template, will create networks of neuron models many orders of magnitude more powerful than the current AI neuron model. This vast efficiency improvement will establish a new generation of AI with a performance capability heretofore unavailable.

Development of algorithms represents future work. We believe that the presented basic principles, presented for the structural plasticity’s inter-associative property, can lead to an algorithm to grow a neural network from training. Because it is evolved by training, only relevant neurons having relevant connections will result. Unlike the currently a’ priori defined AI networks that possess extraneous network connections and superfluous neurons. The description of functional inter-association includes a mathematical definition, proof of completeness, and a weight assignment procedure. This information should be sufficient to compose an algorithm, because the Kobylarz neuron model is vastly more versatile than the currently used *McCulloch & Pitts model*, *the size of neural networks and their efficacy will be substantially improved. Hence, errors will be reduced.*

Acknowledgement

I, Thaddeus Kobylarz, acknowledge the assistance of Dr George Stibitz and Dr. Preston Clement to my portion of this paper. My mentor and introducer to digital theory, Dr, Stibitz advised me during my Master’s thesis research and before I left Vermont in 1960, I asked him to suggest a dissertation topic. He provided me with a paper describing Rosenblatt’s Perceptron and stated that this could be a kernel for future digital processing. I eventually built a neuron model for my doctoral dissertation research. It consisted of an analog computer with interfacing digital circuitry. Being a linear neuron model, it became immediately apparent to me that no threshold weight convergence existed for certain switching functions.

While a young professor at the Stevens Institute of Technology, I was able to persuade my Electrical Engineering Department Chairman, Dr. Clement, to request Stevens Administration’s support for performing research on neuron modeling during the summer of 1966. The purpose was to conceive a neuron model that did not have the switching function limitation of linear neuron models and afterwards to solicit funding for implementation of this model in a neural network. The outcome was the neuron model template described in this paper. Unfortunately, additional funding for this endeavor was denied by the National Science Foundation and no further AI work was done.

References

1. Huang A. Zillow utilizes explainer AI, data to revolutionize how people sell houses. The Machine, Making sense of AI. Venture Beat. 2021 July 14. <https://venturebeat.com/ai/zillow-utilizes-ai-data-to-revolutionize-how-people-sell-houses/>
2. Soper T. Zillow to shutter home buying business and lay off 2,000 employees as its big real estate bet falters. 2021 November 2. GeekWire. <https://www.geekwire.com/2021/Zillow-shutter-home-buying-business-lay-off-2k-employees-big-real-estate-bet-falters/>



3. Cook J. Why the iBuying algorithms failed Zillow, and what it says about the business world's love affair with AI. GeekWire. 2021 Nov 3. <https://www.geekwire.com/2021/iBuying-algorithms-failed-Zillow-says-business-worlds-love-affair-ai/>
4. Rahul R, The Dutch tax authority was felled by AI - What comes next? European regulation hopes to rein in ill-behaving algorithms. IEEE Spectrum. 2022 May 9.
5. Hendrycks D. Measuring mathematical problem solving with the MATH Dataset. Computer Science. Machine Learning, Cornell University. 2021 Nov 8.
6. Roberts M. Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans. Nat Mach Intell. 2021; 3:199-217. doi: 10.1038/s42256-021-00307-0
7. Strickland E. Racial bias found in algorithms that determine health care for millions of patients. Researchers argue for audit systems to catch cases of algorithmic bias. IEEE Spectrum. 2019.
8. Choi C. Some AI Systems may be impossible to compute. New research suggests there are limitations to what deep neural networks can do. IEEE Spectrum. 2022.
9. Choi C. 7 revealing ways AIs fail: neural networks can be disastrously brittle, forgetful, and surprisingly bad at math. IEEE Spectrum. 2021 Oct;58(10): 42-47. doi: 10.1109/MSPEC.2021.9563958. .
10. Thompson N. Deep learning's diminishing returns: the cost of improvement is becoming unsustainable. IEEE Spectrum. 2021 Oct 8 58(10):50-55. doi: 10.1109/MSPEC.2021.9563954
11. National Highway Traffic Safety Administration. NHTSA Estimates for 2022 Show Roadway Fatalities Remained Flat After Two Years of Dramatic Increases., United States Department of Transportation Report, 2023 April 20. <https://www.nhtsa.gov/press-releases/traffic-crash-death-estimates-2022>
12. Smith MS. Google and Microsoft race to unveil AI-based search. But both Bard and Bing Chat stumble out of the gate. IEEE Spectrum. 2023.
13. Olson E. Google shares drop \$100 billion after its new AI chatbot makes a mistake. NPR Technology, 2023 Feb 9 2023. <https://www.npr.org/2023/02/09/1155650909/google-chatbot-error-bard-shares>
14. Hao K. Training a single AI model can emit as much carbon as five cars in their lifetimes. MIT Technology Review. 2019 Jun 6. <https://www.technologyreview.com/2019/06/06/239031/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/>
15. Hutson M. Measuring AI's Carbon Footprint. New tools track and reduce emissions from machine learning. IEEE Spectrum. 2022.
16. Edmund L. AI's carbon footprint problem. Stanford Engineering. 2020.
17. Henry CJ. Basal metabolic rate studies in humans: measurement and development of new equations. Public Health Nutr. 2005 Oct;8(7A):1133-52. doi: 10.1079/phn2005801. PMID: 16277825.
18. McCulloch WS, Pitts W. A logical calculus of ideas immanent in nervous activity. Bull Math Biophys. 1943; 5(4):115-133. doi:10.1007/BF02478259.
19. Hebb D. The Organization of Behavior: A Neuropsychological Theory. New York: Wiley and Sons; 1949.
20. ROSENBLATT F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev. 1958 Nov;65(6):386-408. doi: 10.1037/h0042519. PMID: 13602029.
21. Mok K. Why Artificial Intelligence Needs Neuroscience for Inspiration. The New Stack. 2017 Aug 14. <https://thenewstack.io/stronger-artificial-intelligence-needs-neuroscience-inspiration/>
22. Costandi M Neuroplasticity, MIT Press: Cambridge, MA. 2016.
23. Hensch TK, Bilimoria PM. Re-opening Windows: Manipulating Critical Periods for Brain Development. Cerebrum. 2012 Jul;2012:11. Epub 2012 Aug 29. PMID: 23447797; PMCID: PMC3574806.
24. Nigam VP, Graupe D. A neural-network-based detection of epilepsy. Neurol Res. 2004 Jan;26(1):55-60. doi: 10.1179/016164104773026534. PMID: 14977058.
25. Carter R. The Human Brain Book. New York: DK Publishing; 2009; 70, 156, & 193.
26. Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci U S A. 1982 Apr;79(8):2554-8. doi: 10.1073/pnas.79.8.2554. PMID: 6953413; PMCID: PMC346238.
27. Kandel ER, Schwartz JH, Jessell TM, Siegelbaum SA, Hudspeth AJ, Mack S (Eds.). Principles of Neural Science. 5th Ed. New York: McGraw Hill; 2013.
28. Sternberg RJ, Sternberg K. Cognitive Psychology. 6th Ed. Belmont, CA: Wadsworth/Cengage Learning. 2012.
29. Craighhead WE, Nemeroff CB. The Concise Corsini Encyclopedia of Psychology and Behavioral Science. New York: John Wiley & Sons. 2004 Apr 19. Gestalt Psychology. 401-404.
30. Lehar S. The World in your Head: A Gestalt View of the Mechanism of Conscious. 1st Ed. Psychology Press: London. 2003.
31. Minsky M, Papert S. Perceptrons: An Introduction to Computational Geometry. 2nd Ed. MIT Press: Cambridge MA; 1972.
32. Gruzling N. Linear separability of the vertices of an n-dimensional hypercube. M.Sc. Thesis. University of Northern British Columbia. 2001.
33. Kriesel DA Brief Introduction to Neural Networks. Published online Bonn, Germany. 2007. https://dkriesel.com/en/science/neuroal_networks
34. Kobylarz TJ, Bradley W. Adaptation in linear and non-linear threshold models of neurons. IEEE Transactions on Information Theory and the International Symposium on Information Theory, San Remo, Italy. 1968 and 1967.
35. Rojas R. Neural Networks, A systematic introduction. Berlin: Springer-Verlag, Berlin. 1996.
36. Kobylarz TJA, Kobylarz EJ. A general nonlinear neuron model. Eastern Association of Electroencephalographers, 74th Annual Meeting. 2020 February 14-15. Clin Neurophysiol. 2021; 132(9): e1. doi: 10.1016/j.clinph.2021.03.026
37. Kobylarz TJA, Kobylarz EJ. Potential means to circumvent a neural network performance impasse. Eastern Association of Electroencephalographers, 76th Annual Meeting. Clin Neurophysiol. 2022.
38. Kanold P. Study identifies first brain cells that respond to sound. Science News. 2017 Dec 21. <https://www.sci.news/othersciences/neuroscience/subplate-neurons-sound-05555.html>.
39. Shier D. Hole's Human Anatomy & Physiology. 14th Ed. New York: McGraw-Hill Education. 2016.
40. Gidon A, Zolnik TA, Fidzinski P, Bolduan F, Papoutsi A, Poirazi P, Holtkamp M, Vida I, Larkum ME. Dendritic action potentials and computation in human layer 2/3 cortical neurons. Science. 2020 Jan 3;367(6473):83-87. doi: 10.1126/science.aax6239. PMID: 31896716.
41. Yu Y, Shu Y, McCormick DA. Cortical action potential backpropagation explains spike threshold variability and rapid-onset kinetics. J Neurosci. 2008 Jul 16;28(29):7260-72. doi: 10.1523/JNEUROSCI.1613-08.2008. PMID: 18632930; PMCID: PMC2664555.
42. Platkiewicz J, Brette R. A threshold equation for action potential initiation. PLoS Comput Biol. 2010 Jul 8;6(7):e1000850. doi: 10.1371/journal.pcbi.1000850. PMID: 20628619; PMCID: PMC2900290.