

NEURAL NETWORK DESIGN

Bui Gia Khanh *

Department of Physics Hanoi University of Science, Vietnam National University

Hanoi, Vietnam

{fujimiyaamane}@outlook.com

ABSTRACT

We revise and reevaluate the theory of artificial neural networks, with the cornerstone notion of *forward process* and *perceptron operations*. This is the framework which gives rise to *neural network architecture* and furthermore, the *deep learning architecture* of multilayer perceptrons (MLP). In particular, we construct a novel structures and analytical component sets, which serves to decompose the neural network architecture to the smallest working component, and within general principle of working.

1 INTRODUCTION

With the development of neurological models since the early 1940s [McCulloch, Pitts, 1943], the works done in 1950s to the ends of 1970s [Hebb, 1949], Rosenblatt (1958) [Bernard, Hoff, 1959] [Ivakhnenko, Lapa, 1965] [Shun'ichi Amari, 1967,1972], [Seppo, 1970] [Kely, 1960] by many people before the period called the AI winter, and progress made of which formed the discipline of *deep learning* up to present, machine learning has been mostly considered, and conducted, within the *neural network architecture*. This architecture is based on its most basic component of an *artificial neuron*, most often described and constructed as a *perceptron*. Many progresses has been made using this architecture involves *forward neural network* (FNN), *backpropagation principle*, *perceptron learning* and *multilayer perceptron* (MLP), *Hebbian learning*, *Hopfield networks*, *recurrent neural network* (RNN) and *long-short term memory* (LSTM) network, *convolutional neural network* (CNN), up to *encoder-decoder*, *belief networks*, and then *transformer* (*sub*)architecture, most prominent in applications of the state-of-the-art operations in the present time.

However, we still do not know entirely how the neural network architecture actually works, even from those successes conceded. The nature of the neural network architecture is partially unknown, typically described as a black box. Its working mechanism is well-known, but the decision making and information processing is untrackable. Works has been conducted to deal with this problem, specifically through viewing the neural network, and its subsequent construction called *deep learning* via theory and traceback to neurological concepts [Samuel et al., 2023] [Daniel, Sho, Boris, 2021v2] [Timothy, Konrad, 2019] [Zhanghao, Ding, 2021], or by practical experiments from different perspective [Jason et al., 2015] [Nguyen et al., 2019]¹. However, those researches conceived gained various insight, not an overall rework of the hypothesis, and most of the time is constrained by the rigid construction already presented. Theories are taken in turns, but some of the works focus on rather the general case, the bounds and conceptual limits, but not the exact phenomenological architecture itself. One example of such is the PAC-learning theory, PAC-Bayesian, VC-Theory, and else. An analysis of the actual optimizer (taken in terms of PAC(-Bayesian) theory *objective*) is very much the main objective of this paper.

Hence, this paper focus, and **goal**, is on the *deconstruction* and a novel construction of the main schema of artificial intelligence researches - the neural network architecture, and presents it in a more conceptual-practical and transparent (if able) cases. This also includes various insights, propositions, conjectures and incorporation of different concepts into the analysis. The obvious goal, *learning*, however, is not in the schedule to be made, or rather, is not considered in such case.

*Undergraduate student

¹We do not specify all papers related to such issues, but those are some highlights.

2 CONSTRUCTIONS

Neural network architecture was based around the construction of the singular processing unit, called *neuron*. This stems from neuroscience researches and knowledge, of which identifies the main processing construction is the *biological neuron* embedded in the development of the brain. McCulloch and Pitts first introduced, the first mathematical model of a biological neuron. This model consists of the classical separation with preactivation and *activation function*, specifically the Thresholding Logic Unit (TLU):

$$y_{in} = \sum_{i=1}^n y_{in_i} \quad \forall x_n \in \{0, 1\} \quad (1)$$

$$f(y_{in}) = \begin{cases} 1 & y_{in} \geq \theta \\ 0 & y_{in} < \theta \end{cases} \quad (2)$$

where θ is the threshold and y_{in} is the total net input signal received. The construction of this network follows propositional logics, and hence, it is theorized to simulate and construct *boolean logic handler*, for OR, AND, NAND or others. And with the parameter θ , its name as the TLU is common in electronics and computer systems. In principle, in their original paper *A logical calculus of the ideas immanent in nervous activity*, 1943, it is remarked that, their idea is to "[He] therefore attempted to record the behaviour of complicated nets in the notation of symbolic logic of propositions", in which the all-or-none principle of such time infers the absolute certainty of propositional logic. The goal is then apparent - to use symbolic and propositional logic to express the working of every network, and hence applies it to the main structure.

This idea of a computational unit with receiver and activation patterns continues from then. The later idea includes Rosenblatt's perceptron (1957,1958), of which was based on works of McCulloch, the MCP or TLU, Culbertson (1956), Minsky (1956), Hebb (1949), and some of Von Neumann (1951, 1956). In their construction, per example for a photo-perceptron (optical signals), the organization of a perceptron is pretty much complex, based on three components:

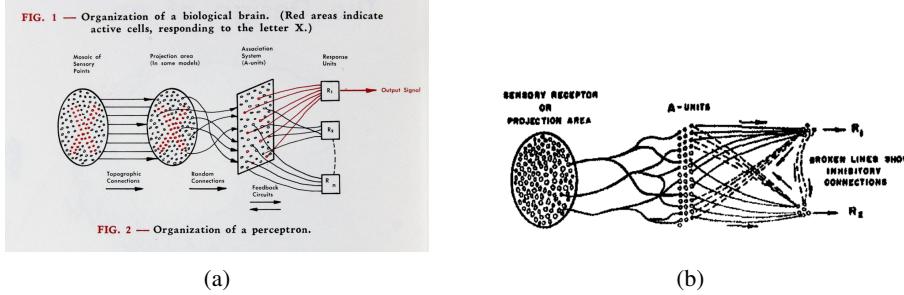


Figure 1: (a) Figure of the original organization of the biological model of the brain functions. (b) Specifically, note that it is specifically for optical case, but can be extended to others type. Furthermore, the layer between last A -unit and the response units, there exists a pattern of feedback loop.

1. The retina with localized connections, containing sensor points, called S -points.
2. The *projection area* A_I of which is described to be connected subsequently to the *sensor area* (this can be omitted, however, in case the next layer is directly connected to the retina), which receives a number of connections from the sensory points, and those formed a set called the *origin points* of such A -unit.
3. Between the projection area and the next layer here, we call as *association area*, connections are assumed to be random, scattered at random throughout the projection area.
4. The responses R_1, R_2, \dots are cells which respond in much the same fashion as the A -unit. Each response has a certain count of origin points located at random (similar to between A_I and A_{II}) in the A_{II} net.

Descriptively, we describe Rosenblatt's perceptron as followed.

Definition 2.1 (Rosenblatt, 1957). A perceptron (or linear classifier) is a function

$$\begin{aligned} \text{lin} : \mathbb{R}^d &\rightarrow \mathbb{R} \\ \mathbf{x} &\rightarrow \mathbf{w} \cdot \mathbf{x} + b \end{aligned} \tag{3}$$

where the parameters to be learned are $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$.

A intricate detail to observe, is the fact that between A_{II} and the receptive field, there exists feedback loops, directly connected back to each other, based on firing principles and thereof. There exists inhibitory and excitatory feedback connections, which separated the propagation phase into the *predominant* and *postdominant* phases. Effectively, this brings the idea of tangent optimization to it for one of the first network to do so. A note on Rosenblatt's perceptron, though, is that there are no *activation function* in the network. Most notably, Rosenblatt's and McCulloch's perceptrons are often deferred to as either linear classifier, or *binary classifier* on linear space [Freund, Schapire, 1999].

Albeit the role of activation function is dubious, it was presented in the very first TLU as a threshold unit, effectively transformed TLU into a 2-partition boundary unit within the parameter space. In one way or another, we can think of activation function belongs to a collection of main operating mechanism of which the neuron is thought to be modelled as.

2.1 MULTILAYER PERCEPTRON (MLP)

Moving on, the limits of perceptron is well-known, at least in its concurrent form. By the standard linear-binary classifier, the "XOR problem" is unsolvable by the typical perceptron system. In fact, any data of which is *linearly inseparable* will render perceptron useless, albeit we still have the perceptron cycling theorem [Block, Levin, 1970]. Thereby, a different architecture is involved, namely the *multilayer perceptron architecture* (MLP).

It is here that we should note that the name multilayer perceptron brings a lot of controversy to the name on itself. The name directly (and incorrectly) infers to the fact that it contains only one more component: a lot of perceptrons in layers. However, the name is talking about the architecture of a single neuron, and we can see the idea of **layers** having introduced earlier on in Rosenblatt's perceptron model. Rather, it is the fact that multilayer perceptron is the architecture that comprises many components into layers, but not every component is a perceptron, and rather, they are in one way or another, *neuron-like* processing unit. Many of today's architecture relies on such interpretation, including the latest structures. Those, for example, Transformer, CNN, and else, can be considered partially heuristic when considered to generality of the neurological processing operations, taking advantage of certain data configurations.

There are no definite definition of a multilayer perceptron. This is mainly because of its configuration and archetypical nature of being the 'way of construction' of meaningful compositions for smaller neural-like units, the name is given toward being an architecture instead on its own, in typical literature. [Ramchoun et al., 2016] [Roberts, Yaida, Hanin, 2021v2]

2.1.1 REMARK

Section 2 introduced the basic ground works for generally, all the currently constructed neural network system, per history and timeline of ideas. It leads to several questions:

1. Activation function is essentially to the formation of MLP. However, their appearance is not generally dated. Before that, MCP and Rosenblatt's perceptron also has activation functions, albeit they are threshold units containing logical syllogism of either 0 or 1, except for ADALINE with linear activation function.

Multilayer perceptron is imperatively different, because it uses *non-linear* activation function. As we recalled, TLU can only solve the binary boundary problem, and while perceptron is better, it is ill-suited for problems that need non-linear solutions. Generally, it is true that linear unit can only interpret linear patterns and situations. Thereby, the non-linear activation function is considered a breakthrough, of which is prominent in the multilayer perceptron system. However, the question remains. What is the potential of an activation function?

2. Early in section 2, we refer to components of multilayer perceptron as neural-like components. But what exactly are them? In a loose sense, they can be considered to be processing unit, with input-output flow of operation, and certain operation mutating the input to certain form - or rather, mapping it from the input space \mathcal{I} to the processor's space \mathcal{P} . But what is the true construction of such neural-like components. And further, what can be their concrete definition and theories?

3. What can be the representative mathematical object and operations that partially represents and interpret a single perceptron unit in actual system?
4. What is the *axiom* or assumption of the multilayer perceptron (for example, this question somewhat refers to Hebb's rule as one of the axiomatic condition of evolution of neural net, which then could be called as Hebbian neural architecture)?
5. What is exactly meant by *neural-like components*? From Rosenblatt to Marvin, their idea of components networks or *Society of Mind* in Marvin's case, but what exactly constitute the neural-like properties, and what constitute those components to works together?

While resourceful, the above analysis of three different models leaves much to desire, as they do not give us reasonable information specifically to act upon. Double descent itself is observed without much problem, so the mechanical details is not fully realized. It is then the time we have to carefully consider the architectural setting and modelling system by itself. We propose a way to conduct this, using the new consideration on both *mathematical modelling* and *neural network-style architecture* considerations, called **Neural Unit Archetype** (NUA).

The main goal of NUA is to express all models as we have seen, in a more generalized template of neural network, not just the usual layer-based architecture, but on the formalization of unit-component analysis. This means modularity and separation of factors, abstractions, compositions and so forth that can be considered of the unit itself. On the mathematical modelling, we use a hybrid of information-theoretic interpretation and classical mathematical modelling system consideration, for example, determining the modified version of the 3-tuple (S, M, Q) Vel (2024). From such, we suggest a possible detail interpretation and analysis on which double descent will be analysed.

2.2 MOTIVATION

It is clear that we do not have a great picture at the problem ourselves. One of the main approach that is proposed to solve double descent is to look into its complexity notion, aside from the error measure. This comes into two important parts - the sample complexity which is easier to gauge, and the model complexity, which contains both structural or representation complexity, and effective operational complexity for such model in practice. It is of such knowledge being that parameters $\theta \in \Theta$ are not equal to each other, such is to also say of the polynomial's monomial basis. Current structural analysis can still be considered very specific, in such there exists no fundamental structure to gauge them. On the other hand, neural network formalism opened up such prospect, yet is poorly developed in said structural sense, and is still often focused on the mathematical analytics of it by itself. Thereby, we envision such novel unit archetypical architecture in this section.

Basically, we want to construct an *elementary construction*, where simple, monomial components are constructed from larger one, in which analysis can be made clear. For this, we borrow the idea of neural network. The class of elementary and classical experimental setting that would be considered is the class of typical neural network, or in general, Q -width L -depth neural network of general activation type, without specific engineering or modifications. For completeness, we refer to ? for a more mathematical analysis, and our network will be formulated the same.

Definition 2.2 (Standard multilayer network, Zhang et al. (2023)). *We define a K -layer fully-connected deep neural network with real-valued output. Let $m^{(0)} = d$ and $m^{(K)} = 1$ for m the width of the network, and d the shape of the input vector. We then recursively define:*

$$x_j^{(0)} = x_j \quad (j = 1, \dots, m^{(0)}), \tag{4}$$

$$x_j^{(k)} = h \left(\sum_{j'=1}^{m^{(k-1)}} \theta_{j,j'}^{(k)} x_{j'}^{(k-1)} + b_j^{(k)} \right) \quad (j = 1, \dots, m^{(k)}), \quad k = 1, 2, \dots, K-1 \tag{5}$$

$$f(x) = x_1^{(K)} = \sum_{j=1}^{m^{(K-1)}} u_j x_j^{(K-1)} \tag{6}$$

where the model parameters can be represented by $w = \{[u_j, \theta_{j,j'}^{(k)}, b_j^{(k)}] : j, j', k\}$ with $m^{(k)}$ being the number of hidden units at layer k ; $\theta \in \mathbb{R}^m$ the weight of the neuron.

However, such structure is imperfect, as it is reduced to a functional concept. To truly understand what is happening inside a particular structure as neural network, it requires more than analytical

function representation itself, especially after the success of deep learning despite classical learning method. Our goal is then to generalize structures, including classical models, and neural network alike, to a new, modular and general structural expression.

2.3 STRUCTURE

The structure of our theory on the Neural unit Archetype is influenced by the object-abstracted treatment of mathematically embedded structures, and the unit-wise principle of particular neuron. Before we meet ourselves into the notion of epistemic circularity² problem, we might as well clarify a few prerequisites for such structure to exhibit.

First, we indict on the fundamental encoding environment that any object can take. The main point of any structure here is that there exists fundamentally the encoding space of two types. First is the object's cardinality space, denoted $\Gamma = (\mathbb{N}, F)$ for any given categorization F . Second is the encoding primitive of the field \mathbb{R} for generality - in general any field is alright, and they define the analytic structure of the system itself. Any extension, for example, the \mathbb{R} -algebra of complex number \mathbb{C} is then the primitive field's extension.³ Any type of data or system can then be decomposed to such, with additional structure on top of such primitive. Such is then called the *primitive framework*.

Definition 2.3 (Primitive framework). *Let us define the primitive framework \mathcal{P}_0 of the dual (Γ, \mathbb{R}) where $\Gamma = (\mathbb{N}, F)$ is the cardinality encoding space, and \mathbb{R} is the base field primitive of the analytical encoding. An object $X \in \mathcal{P}_0$ admits a dual representation,*

$$X \mapsto (\gamma(X), \rho(X)) \quad (7)$$

for $\gamma : \text{Obj} \rightarrow \Gamma$ of cardinality encoding, and $\rho : \text{Obj} \rightarrow \mathcal{E}(\mathbb{R})$ for the field extension of \mathbb{R} category, or the category of all \mathbb{R} -algebras. For $\mathcal{E}(\mathbb{R})$ without extension, then $\mathcal{E}(\mathbb{R}) \cong \{\mathbb{R}\}$ of all \mathbb{R} -algebra.

For a structure that is supposed to be unit-wise constructed like neural network and the like, one of the main principle is the principle of abstraction. With this, come the idea of layer. Specifically, a *layer* separates abstraction in terms of subspace. Let us take an example of such kind for clarification. Let us define the primitive framework \mathcal{P}_0 as now the layer L_0 of this layering scheme. Then, we define the layer L_1 of all unit-wised neuron-like units taking over the representation scheme on L_0 . We then define the structure of the neuron class \mathcal{N}_1 upon such as followed.

Definition 2.4 (Base neuron class). *We define the base neuron class $\mathcal{N}_1 \equiv L_1$ as followed. For any $\mathcal{U} \in L_1$ for \mathcal{U} as unit-wise construction over L_0 , then every \mathcal{U} satisfies the input signature $\mathcal{I}_{sig} : \mathcal{E}'(\mathbb{R}) \rightarrow L_0$, output signature $\mathcal{O}_{sig} : L_0 \rightarrow \mathcal{E}'(\mathbb{R})$, and \mathcal{C}_{ext} as extensible construction over L_0 , for $\mathcal{E}'(\mathbb{R})$ particular extension on analytical encoding of L_0 . The type template of a neural unit $\mathcal{U} \in L_0 \equiv \mathcal{N}_1$ is then defined as $\mathcal{U} = \langle \mathcal{I}_{sig}, \mathcal{C}_{ext}, \mathcal{O}_{sig} \rangle$ where \mathcal{I}_{sig} and \mathcal{O}_{sig} are invariant as type.*

Those definitions directly link it to type theory, while such development is perhaps more complex. Let's take an example on it.

²Also called bootstrapping, where to understand or define certain notion, requires the knowledge of the object wishes to be defined itself - for example, defining the size of a finite natural number set, using the elements of the number set itself.

³This corresponds to a variety of idea. For example, see Cartuyvels et al. (2021) for the idea of discrete and continuous representations and processing, and Müller et al. (2022) for the same idea, but used on computer graphic process where it is constructed as discrete indexing structure (hash tables) with continuous neural fields. Cardinality space takes inspiration from combinatorial species, sheaf theory of global-local set, and Grothendieck's approach to algebraic topology. Extension is the idea from field extension itself, for structures that can be extended. The general idea of this section relies on interpretation of category theory.

REFERENCES

- Mathematical Modeling and Simulation: Introduction for Scientists and Engineers, 2nd Edition | Wiley, 2024. URL <https://www.wiley.com/en-us/Mathematical+Modeling+and+Simulation%3A+Introduction+for+Scientists+and+Engineers%2C+2nd+Edition-p-9783527839407>.
- Ruben Cartuyvels, Graham Spinks, and Marie-Francine Moens. Discrete and continuous representations and processing in deep learning: Looking forward. *AI Open*, 2:143–159, 2021. ISSN 2666-6510. doi: 10.1016/j.aiopen.2021.07.002. URL <http://dx.doi.org/10.1016/j.aiopen.2021.07.002>.
- Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, July 2022. ISSN 1557-7368. doi: 10.1145/3528223.3530127. URL <http://dx.doi.org/10.1145/3528223.3530127>.
- Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958. URL <https://api.semanticscholar.org/CorpusID:12781225>.
- Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning, 2023. URL <https://arxiv.org/abs/2106.11342>.

A APPENDIX

You may include other additional sections here.