Informaticae Abstracta. ISSN: 3061-8533
Año 2026, Vol. 4, No. 1, pp. 72-82
ENE | JUN 2026
doi: https://doi.org/10.36677/ia.v4n1.28939

Neural networks as trainable compositional systems

Solís-Winkler, A.*

Información del Artículo

Recibido: 20 de enero, 2026
Aceptado: 25 de abril, 2026

Palabras clave: Neural Networks, Compositional System, Layer Operators, Deep Learning

Resumen

Classical neural network theory is commonly formulated from a neuron-centered perspective, where architectures are described in terms of affine transformations followed by nonlinear activation functions. While this formulation is well suited for multilayer perceptrons, modern neural architectures increasingly rely on heterogeneous operator structures, including convolutional operators, attention mechanisms, residual connections, Kolmogorov–Arnold representations, and explicit interaction models. This work proposes a broader formulation in which neural networks are interpreted as trainable compositional systems of parametrized operators. Under this framework, architecture, parametrization, hypothesis space, learning, and approximation are separated as distinct conceptual levels. This perspective provides a unified language for classical and modern architectures and allows nonlinear expressive behavior to emerge from compositional operator structure rather than exclusively from activation functions.

1. Introduction

Classical neural network theory originated from the work of McCulloch and Pitts [1] where artificial neurons were introduced as simplified models of biological neurons and their interconnection to form complex structures, and later evolved through the perceptron framework introduced by Rosenblatt [2]. Within this tradition, neural architectures are typically described as compositions of affine transformations and nonlinear activation functions. Such formulations have been remarkably successful in the study of multilayer perceptrons and have provided the foundation for fundamental results, including universal approximation theorems [3].

However, modern machine learning architectures increasingly incorporate heterogeneous computational structures that extend beyond the classical neuron-centered paradigm. Convolutional networks [4], attention mechanisms and transformers [5], capsule networks [6], Kolmogorov–Arnold architectures [7], [8], [9], interaction-based models, residual systems [10], and hybrid architectures are often described using different mathematical formalisms despite sharing a common compositional nature.

This observation motivates the search for a more general framework in which neural architectures are understood independently of any specific neuron model, activation function, or optimization procedure. The central thesis of this work is that neural networks can be interpreted as trainable compositional systems whose expressive behavior emerges from the organization of parametrized operators. Under this perspective, neurons, activation functions, convolution kernels, attention mechanisms, Kolmogorov–Arnold representations, and interaction operators become particular instances of a broader compositional framework.

2. Compositional Neural Architectures

Although modern architectures differ substantially in their internal mechanisms, they share a common compositional nature. Complex mappings are constructed through the hie-

rarchical composition of simpler transformations. Under this perspective neurons become particular instances of a broader class of functional operators, while architectural complexity emerges from their composition rather than from any specific internal representation.

In the present work attention is restricted to finite feedforward architectures whose compositional structure is acyclic. Consequently, recurrent architectures and models whose natural graph representations contain directed cycles are left for future extension of the framework.

The following definition formalizes this compositional framework. This perspective is inspired by the compositional interpretation found in modern machine learning literature [11], [12], [13], while extending it beyond activation-centric formulations.

Definition 1 (Compositional Neural Architecture)**.** *Let $n_0, \dots, n_L \in \mathbb{N}$. A compositional neural architecture is an ordered collection of layer operators*

$$\mathcal{A} = (F_1, \dots, F_L)$$

such that

$$F_\ell : \mathbb{R}^{n_{\ell-1}} \to \mathbb{R}^{n_\ell}, \quad \ell = 1, \dots, L.$$

The architecture induces a mapping of the form

$$F = F_L \circ \cdots \circ F_1$$

The integer $L \in \mathbb{N}$ is called the depth of the architecture, while the dimensions $n_0, \dots, n_L$ define its layer widths.

Remark 1.*The compositional representation is fundamental. Graph-theoretic representations introduced later provide an equivalent structural description of the same compositional system and allow the treatment of architectures with branching, residual, and heterogeneous connectivity patterns.*

The previous definition introduces neural architectures as compositional families of mappings. The fundamental building blocks of these compositions are layer operators, which act as transformations between intermediate representation spaces.

Definition 2 (Layer Operator)**.** *Let X and Y be finite-dimensional vector spaces. A layer operator is a mapping*

$$F : X \to Y.$$

The spaces X and Y are called the input and output representation spaces of the operator.

Unlike classical neuron-centered formulations, no assumption is made regarding the internal structure of the operators F. Consequently, different operator classes may coexist within the same architecture. Since layer operators are vector-valued mappings, they admit a coordinate decomposition into scalar-valued component functions.

Proposition 2 (Coordinate representation). *Let $F_\ell : \mathbb{R}^{n_{\ell-1}} \to \mathbb{R}^{n_\ell}$ be a layer operator, then F_ℓ admits a coordinate representation*

$$F_\ell(x) = (f_{\ell,1}(x), \dots, f_{\ell,n_\ell}(x)),$$

where each coordinate mapping

$$f_{\ell,j} : \mathbb{R}^{n_{\ell-1}} \to \mathbb{R}$$

denotes the j-th scalar component of the vector valued layer operator.

Demostración. Since $F_\ell : \mathbb{R}^{n_{\ell-1}} \to \mathbb{R}^{n_\ell}$, each component of the image vector defines a scalar-valued mapping $f_{\ell,j}$.

Remark 3.*In classical perceptron architectures, the coordinate mappings $f_{\ell,j}$ correspond to individual neurons. Within the present framework, however, the coordinate mappings are not required to possess any particular structure and may arise from arbitrary parametrized operator classes. Thus, a layer is fundamentally interpreted as a vector-valued operator whose coordinates are component mappings rather than as a collection of neurons. Classical perceptrons arise as a special case in which the coordinate mappings take the form*

$$f_{\ell,j}(x) = \sigma(w_{\ell,j}^\top x + b_{\ell,j}).$$

More generally, layer operators may correspond to arbitrary parametrized operators. The framework does not impose any restriction on the internal structure of layer operators. Consequently, multiple operator classes can be accommodated within the same formalism. Subsection 2.1 provides examples of layer operators for the most common architectures.

To obtain trainable architectures, it is necessary to introduce trainable parameters governing the behavior of the layer operators, following the standard paradigm of parametrized hypothesis classes in statistical learning and neural network theory [13], [14].

Definition 3 (Operator Parametrization)**.** *Let $F : X \to Y$ be a layer operator. A parametrization of F is a family of mappings*

$$\Theta \ni \theta \mapsto F^\theta : X \to Y.$$

Definition 4 (Parameterized Architecture)**.** *Let $\mathcal{A} = (F_1, \dots, F_L)$ be a compositional neural architecture. For each layer operator $F_\ell : X_\ell \to Y_\ell$, let Θ_ℓ be a parameter space and let*

$$\Theta_\ell \ni \theta_\ell \mapsto F_\ell^{\theta_\ell}$$

be a parametrization of the operator. The global parameter space of the architecture is defined as

$$\Theta = \Theta_1 \times \Theta_2 \times \cdots \times \Theta_L.$$

The family of realizations

$$F_\theta = F_L^{\theta_L} \circ F_{L-1}^{\theta_{L-1}} \circ \cdots \circ F_1^{\theta_1}, \quad \theta \in \Theta,$$

is called the parametrized architecture associated with $\mathcal{A}$.

For example, perceptron architectures use affine coefficients, convolutional architectures use kernel parameters, attention architectures use projection operators, Kolmogorov–Arnold architectures employ parameters describing univariate functions, and interaction-based architectures may include interaction coefficients and functional embeddings.

Thus, parametrization is interpreted independently of any specific operator class. Once a parameterization is introduced, the architecture induces a family of realizable mappings. This family corresponds to the hypothesis space associated with the architecture. The notion of hypothesis space plays a central role in statistical learning theory [14].

Definition 5 (Hypothesis Space)**.** *Let $\mathcal{A}_\Theta$ be a parametrized architecture. The induced hypothesis space is defined as*

$$\mathcal{H}(\mathcal{A}_\Theta) = \{F_\theta : \theta \in \Theta\}.$$

Remark 4 (Conceptual hierarchy). *The proposed framework separates four conceptual levels that are frequently intertwined in classical neural network formulations. A compositional architecture specifies the structural organization of a family of mappings through the composition of layer operators. Layer operators define the admissible classes of transformations acting between intermediate representation spaces. A parametrization introduces trainable degrees of freedom governing the behavior of these operators. Finally, the induced hypothesis space consists of all realizable mappings obtained by varying the parameters within their admissible domains. Consequently, the following conceptual hierarchy emerges naturally:*

$$\textit{Architecture} \longrightarrow \textit{Layer Operators} \longrightarrow \textit{Parametrization} \longrightarrow \textit{Hypothesis Space}.$$

This separation allows architectural structure, functional representation, and learning mechanisms to be studied independently while remaining part of a unified compositional framework.

The previous definitions describe neural architectures through compositions of parametrized operators. While this functional representation is fundamental, it is often convenient to introduce an equivalent structural description that explicitly captures connectivity patterns between intermediate representation spaces. Such a description becomes particularly useful for architectures involving residual connections, parallel branches, encoder–decoder structures, attention pathways, and other heterogeneous connectivity patterns that extend beyond purely sequential compositions.

For this reason, compositional architectures may alternatively be represented through directed graphs, where nodes correspond to intermediate representation spaces and edges correspond to parametrized operators. Graph-based descriptions of neural architectures have appeared in various forms in the literature, including computational graph formulations and learning-theoretic representations [12].

Definition 6 (Architectural Graph Representation)**.** *Let $\mathcal{G} = (V, E)$ be a directed acyclic graph, and let each edge $e = (u, v) \in E$ be associated with a class of parametrized operators*

$$\mathcal{O}_e = \{F_e^\theta : X_u \to X_v\}.$$

An architectural graph representation of a compositional neural architecture is the collection

$$\mathcal{A} = (\mathcal{G}, \{\mathcal{O}_e\}_{e \in E}),$$

consisting of a compositional graph together with a family of admissible operator classes associated with its edges.

Sequential feedforward architectures correspond to compositions along a directed path. More generally, modern architectures may involve residual connections, parallel branches, encoder–decoder structures, and heterogeneous connectivity patterns that cannot be characterized through uniform activation-centric formalism using the layer widths and the selected activation function.

The global mapping associated with the architecture emerges from the composition of operators along admissible directed paths in the graph, whose nodes represent intermediate representation spaces and whose edges represent parametrized operators.

Proposition 5.*Every sequential compositional architecture admits a graph representation as a directed path.*

Demostración. Let $\mathcal{A} = (F_1, \dots, F_L)$. Define the graph $V = \{v_0, \dots, v_L\}$ and $E = \{(v_{\ell-1}, v_\ell) : \ell = 1, \dots, L\}$. Associate operator F_ℓ with edge $(v_{\ell-1}, v_\ell)$. Then the resulting graph is a directed path whose associated composition is precisely

$$F_L \circ \cdots \circ F_1.$$

Sequential feedforward networks appear as a particular case of a more general compositional architecture in which the graph forms a directed path

$$v_0 \to v_1 \to \cdots \to v_L.$$

In this case, the global mapping reduces to the sequential composition

$$F_\theta = F_L^{\theta_L} \circ F_{L-1}^{\theta_{L-1}} \circ \cdots \circ F_1^{\theta_1}.$$

Thus, the standard multilayer perceptron architecture appears naturally as a special case of the more general compositional graph formulation.

Proposition 6.*Every finite acyclic compositional architecture admits a directed graph representation.*

Demostración. Let $\mathcal{A}$ be a finite compositional architecture formed by a finite collection of layer operators $\{F_1, \dots, F_L\}$. Each operator maps an input representation space into an output representation space. Define a directed graph $\mathcal{G} = (V, E)$ as follows.

For each intermediate representation space appearing in the architecture, introduce a node $v \in V$. For each operator $F_\ell : X_u \to X_v$, introduce a directed edge $e_\ell = (u, v) \in E$ from the node associated with its input space to the node associated with its output space, and label this edge by F_ℓ.

Since the architecture is finite, the sets V and E are finite. Since the architecture is compositional and acyclic, the dependencies among operators define a directed acyclic graph. The composition of operators along directed paths in $\mathcal{G}$ recovers the functional transformations specified by the original architecture. Therefore, $\mathcal{A}$ admits a directed graph representation.

Examples of Layer Operators

The proposed framework is intentionally independent of any particular operator class. Different neural architectures can therefore be interpreted as particular realizations of layer operators within the same compositional formalism.

Activation-Based Operators In classical multilayer perceptrons, a layer is commonly written as

$$F_\ell(\mathbf{x}) = \sigma(W\mathbf{x} + b).$$

From the compositional perspective, this layer can be interpreted as

$$F_\ell = \sigma \circ T,$$

where

$$T(\mathbf{x}) = W\mathbf{x} + b$$

is an affine transformation and σ acts coordinate-wise.

Convolutional Operators A convolutional layer is typically expressed as

$$F_\ell(\mathbf{x}) = K * \mathbf{x} + b,$$

where K denotes a family of convolution kernels. Within the proposed framework, the same layer may be interpreted as

$$F_\ell = \rho \circ K,$$

where K is the convolution operator induced by the kernel family and ρ represents optional post-processing operations such as normalization, pooling, or nonlinear transformations.

Attention-Based Operators Transformer architectures commonly employ attention mechanisms of the form

$$F_\ell(\mathbf{x}) = \text{Attention}(Q, K, V),$$

where Q, K, and V denote query, key, and value representations. From the compositional viewpoint,

$$F_\ell = \mathcal{A} \circ (Q, K, V),$$

where

$$(Q, K, V) : \mathbb{R}^n \to \mathbb{R}^{d_q} \times \mathbb{R}^{d_k} \times \mathbb{R}^{d_v}$$

generates the intermediate representations and $\mathcal{A}$ denotes the attention operator.

Kolmogorov–Arnold Operators Kolmogorov–Arnold networks are commonly written as

$$f_k(\mathbf{x}) = \sum_q \Phi_{kq} \left(\sum_p \phi_{kq,p}(x_p) \right).$$

Within the compositional framework, each coordinate mapping may be expressed as

$$\zeta_k = \zeta \circ \Phi_{kq} \circ \zeta \circ \phi_{kq} : \mathbb{R}^n \to \mathbb{R}.$$

Vector-valued outputs are obtained by collecting several coordinate mappings into a single layer operator.

Interaction-Based Operators Interaction-based architectures are commonly represented through explicit interaction terms between transformed variables. In the IKAM formulation, a layer may be expressed as

$$F_\ell = \mathcal{A} \circ \mathcal{I} \circ \mathcal{S} \circ \zeta_{\text{in}},$$

where the constituent operators perform feature embedding, interaction generation, aggregation, and output transformation.

In coordinate form, the corresponding outputs may be written as

$$f_i(\mathbf{x}) = \sum_q \Phi_{iq}(s_q(\mathbf{x}) + I_q(\mathbf{x})),$$

where s_q denotes additive components and I_q denotes explicit interaction terms.

The examples above illustrate that widely different neural architectures can be interpreted as particular realizations of layer operators. Consequently, the distinction between architectures arises primarily from the choice of operator classes and their composition, rather than from the existence of neurons or activation functions alone.

3. Learning as Functional Approximation

The purpose of a compositional neural architecture is to define an hypothesis space of realizable mappings from which suitable functional representation may be selected through learning. Once an architecture and its associated parametrization have been specified, learning can be interpreted as the process of selecting a mapping from the induced hypothesis space that best explains the observed data.

Let $X \subseteq \mathbb{R}^{n_0}$, $Y \subseteq \mathbb{R}^{n_L}$, and let

$$f^* : X \to Y$$

denote unknown target function generating observed data. Assume that only a finite sample

$$S = \{(x_i, y_i)\}_{i=1}^m \subseteq X \times Y$$

is available, in the deterministic case, one may write

$$y_i = f^*(\mathbf{x}_i)$$

whereas more generally, the observations may be regarded as samples drawn from an unknown probability distribution associated with the target process.

The learning problem consists of identifying a mapping

$$F_\theta \in \mathcal{H}(\mathcal{A}_\Theta)$$

that approximates the target function from the available observed examples. Under this interpretation, learning becomes a problem of functional approximation over a compositional hypothesis space, by adjusting the parameters governing the functional behavior of the architecture in order to approximate an unknown target mapping from observed data.

Empirical Risk Minimization

Let $\ell : Y \times Y \to \mathbb{R}$ be a loss function. The empirical risk associated with F_θ is defined as

$$L_S(F_\theta) = \frac{1}{m} \sum_{i=1}^{m} \ell(F_\theta(x_i), y_i).$$

Under the empirical risk minimization principle, learning consists in solving an approximation problem of the form [14]

$$\theta^* = \arg \min_{\theta \in \Theta} L_S(F_\theta).$$

Importantly, the optimization procedure is conceptually independent of the architectural definition. Gradient-based optimization, second-order methods, evolutionary algorithms, constructive procedures, and operator-specific optimization schemes all fit naturally within the proposed framework. Consequently, neural architectures should not be identified with any particular training mechanism.

Learning Algorithms

The optimization problem defining learning is conceptually distinct from the architecture itself. The architecture determines the hypothesis space, while the learning algorithm determines how candidate solutions are explored within that space.

Consequently, different optimization strategies may be applied to the same parametrized architecture. Examples include gradient-based methods, second-order optimization procedures, evolutionary algorithms, constructive methods, operator-specific optimization schemes, and hybrid approaches.

From the present perspective, learning algorithms should be regarded as search procedures acting on the parameter space rather than as defining components of the architecture.

Equivalent Functional Representation

An important consequence of parametrized compositional architectures is that different parameter configurations may induce identical or functionally equivalent mappings. This observation is closely related to the non-uniqueness of neural representations frequently encountered in overparameterized architectures. That is, there may exist

$$\theta_1, \theta_2 \in \Theta$$

such that

$$F_{\theta_1}(x) = F_{\theta_2}(x) \quad \forall x \in X.$$

This property appears naturally in many neural architectures due to symmetries, redundancies, over-parametrization, or equivalent compositional decompositions.

Consequently, learning should not be interpreted as the identification of a unique parameter configuration, but rather as the search for suitable functional representations within the hypothesis space.

4. Approximation Properties of Compositional Architectures

The expressive power of a neural architecture is determined by the richness of its induced hypothesis space. Given a target function $f^* : X \to Y$, the approximation problem consists in determining whether there exists

$$F_\theta \in \mathcal{F}_\Theta$$

such that

$$F_\theta \approx f^*$$

under an appropriate notion of approximation.

Within classical perceptron theory, approximation capability is commonly attributed to nonlinear activation functions. The present framework adopts a broader perspective: approximation capability may emerge from hierarchical composition, functional superposition, explicit interaction structures, contextual transformations, routing mechanisms, and adaptive connectivity patterns.

Consequently, activation functions should be interpreted as one particular source of nonlinear expressive behavior rather than as universally required architectural components.

Compositional Approximation

The Kolmogorov–Arnold representation theorem provides a canonical example in which expressive multivariate representations emerge through structured compositions of univariate functions. Similarly, interaction-based architectures generate nonlinear representations through explicit interaction operators acting on lower-dimensional functional components.

These examples illustrate that nonlinear approximation capability may emerge directly from compositional organization rather than exclusively from activation operators.

Universal Approximation Revisited

From the perspective developed here, classical universal approximation theorems can be interpreted as particular instances of a broader theory of compositional approximation. Under this interpretation, expressive power is not tied to a unique architectural mechanism. Instead, it emerges from the structure of the hypothesis space induced by the architecture and its parametrization.

Different architectures may therefore achieve expressive approximation behavior through fundamentally different compositional principles while remaining instances of the same general framework.

5. Conclusions

This work proposed a unified interpretation of neural architectures as trainable compositional systems. The framework separates five fundamental concepts:

$$\text{Architecture} \longrightarrow \text{Layer Operator} \longrightarrow \text{Parametrization} \longrightarrow \text{Hypothesis Space} \longrightarrow \text{Learning}.$$

Within this formulation, neural architectures are viewed as compositional organizations of parametrized operators, while learning corresponds to the selection of suitable functional representations from the induced hypothesis space.

This perspective generalizes classical neuron-centered formulations and naturally accommodates heterogeneous architectures including perceptron-based, convolutional, attention-based, Kolmogorov–Arnold, and interaction-based systems.

More broadly, the proposed framework suggests that the expressive capabilities of modern neural architectures should be understood as emerging from compositional operator structures rather than exclusively from activation-based neuron models.

Direcciones de correo de los autores

Solís-Winkler, A.*: Facultad de Ingeniería, Universidad Autónoma del Estado de México, Toluca, Estado de México, México
asolisw@uaemex.mx

Referencias

[1] W. S. McCulloch y W. Pitts, «A Logical Calculus of the Ideas Immanent in Nervous Activity,» *Bulletin of Mathematical Biophysics*, vol. 5, págs. 115-133, 1943.

[2] F. Rosenblatt, «The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain,» *Psychological Review*, vol. 65, n.º 6, págs. 19-27, 1958.

[3] G. Cybenko, «Approximation by superpositions of a sigmoidal function,» *Mathematics of Control, Signals and Systems*, vol. 2, n.º 4, págs. 303-314, 1989.

[4] Y. LeCun et al., «Handwritten Digit Recognition with a Back-Propagation Network,» *Advances in Neural Information Processing*, 1989.

[5] A. Vaswani et al., «Attention Is All You Need,» 2017.

[6] G. Hinton, S. Sabour y N. Frosst, «Matrix Capsules with EM Routing,» Google Brain Toronto.

[7] R. Hecht-Nielsen, «Kolmogorov's Mapping Neural Network Existence Theorem,» en *Proceedings of the IEEE First International Conference on Neural Networks*, Picataway, NJ, 1987, III:11-13.

[8] Z. Liu et al., «KAN: Kolmogorov-Arnold Networks,» 2024.

[9] A. Solis-Winkler, J. R. Marcial-Romero y J. A. Hernandez-Servin, «Symmetry in the Algebra of Learning: Dual Numbers and the Jacobian in K-Nets,» *Symmetry 2025*, vol. 17, n.º 8, pág. 1293, 2025.

[10] K. He, X. Zhang, S. Ren y J. Sun, «Deep Residual Learning for Image Recognition,» 2015.

[11] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. MIT Press, 2016.

[12] C. Aggarwal, *Neural Networks and Deep Learning. A textbook*. Springer International Publishing AG, 2018.

[13] P. Petersen y J. Zech, «Mathematical theory of deep learning,» 2024.

[14] S. Shalev-Shwartz y S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, first. New York: Cambridge University Press, 2014.