Simplicial Neural Networks: First Steps and Future Applications

Anonymous Author(s)
Affiliation
Address
email

Abstract

It is well-known that artificial neural networks are universal approximators. The Universal Approximation Theorem proves that, given a continuous function on a compact set embedded in an n-dimensional space, there exists a one-hidden-layer feed-forward network that approximates the function; however, it does not provide a way of building such a network. In a previous work, the authors presented a constructive approach to tackle this problem for the case of a continuous function on triangulated spaces by connecting the Simplicial Approximation Theorem, a classical result from algebraic topology, and the Universal Approximation Theorem. In this paper, we revisit such a result and propose future applications.

1 Introduction

A classical result in the mathematical theory of neural networks is the *Universal Approximation Theorem* [1, 7]. This result shows that any continuous function on a compact set in $\mathbb{R}^n$ can be approximated by a multi-layer feed-forward network with only one hidden layer and a non-polynomial activation function. However, this result has two important drawbacks for its practical use: firstly, the width of the hidden layer grows exponentially with respect to the accuracy of the neural network, and, secondly, the classical proofs do not provide a practical algorithm for building such a network. In [8], the authors proved the existence of a two-hidden-layer neural network which can approximate any continuous multivariable function with arbitrary precision, and, in [3], a constructive method is provided through a numerical analysis approach. In this work, we revisit a different direction to that presented in [8, 3], and address the aforementioned challenges by connecting the Universal Approximation Theorem with a classical result from algebraic topology known as the *Simplicial Approximation Theorem* [9]. The result is a family of neural networks, called *simplicial neural networks*, such that their architecture and parameters are found by using algebraic topology tools. Namely, an effective method for finding the weights of a two-hidden-layer feed-forward network which approximates a given continuous function between two triangulable metric spaces is provided. This is a restriction from the classical Universal Approximation Theorem that is valid for all compact sets on $\mathbb{R}^n$, but triangulable spaces are common in real-world problems. It is worth mentioning that the method presented in [10] for building simplicial neural networks is constructive and it only depends on the desired level of approximation to the given function. In this paper, we present the basis of such result and propose ideas to further research.

2 Simplicial Neural Networks

In this section, the explicit construction of an artificial neural network that is a universal approximator for continuous functions between triangulable spaces is presented. In general, a neural network

Submitted to the Topological Data Analysis and Beyond Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

can be formalized as a function $\mathcal{N}_{\omega,\Theta} : \mathbb{R}^n \to \mathbb{R}^m$ that depends on a set of weights ω and a set of parameters Θ which involves the description of activation functions, layers, synapses between nodes (neurons), and whatever other consideration in its architecture [6]. Multi-layer feed-forward networks are a particular case of artificial neural networks that can be formalized as follows.

Definition 1 (adapted from [7]) *A multi-layer feed-forward network defined on a real-valued n-dimensional space is a function $\mathcal{N} : \mathbb{R}^n \to \mathbb{R}^m$ such that, for each $x \in \mathbb{R}^n$, $\mathcal{N}(x)$ is the composition of $k+1$ functions $\mathcal{N}(x) = f_{k+1} \circ f_k \circ \cdots \circ f_1(x)$ where $k \in \mathbb{Z}$ is the number of hidden layers, $k \geq 1$, and, for $1 \leq i \leq k+1$, $f_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$ is defined as $f_i(y) = \phi_i(W^{(i)}; y; b_i)$ being $W^{(i)}$ a real-valued $d_{i-1} \times d_i$ matrix (that is, $W^{(i)} \in \mathcal{M}_{d_{i-1} \times d_i}$), $b_i \in \mathbb{R}^{d_i}$ the bias term, and ϕ_i a bounded, continuous, and non-constant function (called activation function). Notice that $d_0 = n$, $d_{k+1} = m$ and $d_i \in \mathbb{Z}$, $1 \leq i \leq k$, is called the width of the i-th hidden layer.*

It is well-known that multi-layer feed-forward networks are universal approximators. The formal details are fixed in the next classical theorem.

Theorem 1 (Universal Approximation Theorem, [7]) *Let A be any compact subset of $\mathbb{R}^n$ and let $C(A)$ be the space of real-valued continuous functions on A. Then, given any $\epsilon > 0$ and any function $g \in C(A)$, there exists a multi-layer feed-forward network $\mathcal{N} : \mathbb{R}^n \to \mathbb{R}$ approximating g, that is, $||g - \mathcal{N}|| < \epsilon$.*

Classical proofs of the Universal Approximation Theorem are not constructive, and finding the correct architecture for a given problem is a challenging task. In [4], a constructive approach to build neural networks based on numerical analysis was presented. In contrast, we tackle the problem by establishing a connection between the Universal Approximation Theorem and the Simplicial Approximation Theorem. Specifically, in [10], a constructive approach to Theorem 1 through a two-hidden-layer feed-forward network for continuous functions between triangulable spaces is provided. Roughly speaking, the approach presented in [10] for building neural networks is based on two observations: (1) triangulable spaces can be *modelled* using simplicial complexes; and (2) a continuous function between two triangulable spaces can be approximated by a simplicial map between simplicial complexes.

We assume that the reader is familiar with basic concepts from algebraic topology and refer to [2, 5, 9] for classical definitions and results. The next proposition shows one of the key ideas of the main result presented in this paper: If a continuous function g is considered between two finitely triangulable metric spaces, then successive barycentric subdivisions[1] can be applied on triangulations of such spaces such that the compositions of such triangulations with an appropriate simplicial map can approximate g as much as desired.

Proposition 1 ([10]) *Let X and Y be two finitely triangulable metric spaces, $g : X \to Y$ a continuous function, and $\epsilon > 0$. Then, there exist two finite triangulations (K, τ_K) and (L, τ_L) of X and Y, respectively, and a simplicial approximation $\varphi_c : |Sd^{t_1} K| \to |Sd^{t_2} L|$ such that $||g - \tilde{\varphi}_c|| \leq \epsilon$ being $\tilde{\varphi}_c = \tau_L^{-1} \circ \varphi_c \circ \tau_K$.*

In order to find a simplicial map that approximates the continuous function g, the Simplicial Approximation Theorem is considered.

Theorem 2 (Simplicial Approximation Theorem [2, p. 56]) *If $g : |K| \to |L|$ is continuous then there is an integer $t > 0$ such that $\varphi_c : |Sd^t K| \to |L|$ is a simplicial approximation of g.*

Now, it remains to show how to compute a two-hidden-layer feed-forward network that models a simplicial map $\varphi_c : |K| \to |L|$ where K and L are finite pure simplicial complexes.

Proposition 2 ([10]) *Let us consider a simplicial map $\varphi_c : |K| \to |L|$ between the underlying space of two finite pure simplicial complexes K and L. Then a two-hidden-layer feed-forward network $\mathcal{N}_\varphi$ such that $\varphi_c(x) = \mathcal{N}_\varphi(x)$ for all $x \in |K|$ can be explicitly defined.*

Proof (hints) Let us assume that $\dim(K) = n$ and $\dim(L) = m$. Let $\{\sigma_1, \dots, \sigma_k\}$ be the maximal n-simplices of K, where $\sigma_i = (v_0^i, \dots, v_n^i)$ for all i; and let $\{\mu_1, \dots, \mu_\ell\}$ be the maximal

[1] The t-th iteration of the barycentric subdivision of a simplicial complex K will be denoted by $Sd^t K$.

m-simplices of L, where $\mu_j = \left(u_0^j, \ldots, u_m^j\right)$ for all j. Let us consider a multi-layer feed-forward network $\mathcal{N}_\varphi$ with the following architecture: (1) an input layer composed of $d_0 = n$ neurons; (2) a first hidden layer composed of $d_1 = k \cdot (n+1)$ neurons that correspond to the vertices of the maximal simplices of K; (3) a second hidden layer composed of $d_2 = \ell \cdot (m+1)$ neurons that correspond to the vertices of the maximal simplices of L; and (4) an output layer with $d_3 = m$ neurons. Then, $\mathcal{N}_\varphi = f_3 \circ f_2 \circ f_1$ being $f_i(y) = \phi_i(W^{(i)}; y; b_i)$, $i = 1, 2, 3$, is constructed as follows.
Firstly, a point x in $\mathbb{R}^n$ is transformed into a $k \cdot (n+1)$ vector that can be seen as the juxtaposition of k vectors of dimension $n+1$ (one for each of the k simplices in K), each one representing the barycentric coordinates of x with respect to the corresponding simplex. From the barycentric coordinate relations, we obtain the matrix $W^{(1)} = \begin{pmatrix} W_1^{(1)} \\ \vdots \\ W_k^{(1)} \end{pmatrix}$ and the bias term $b_1 = \begin{pmatrix} B_1 \\ \vdots \\ B_k \end{pmatrix} \in \mathbb{R}^{k(n+1)}$, where $W_i^{(1)} \in \mathcal{M}_{(n+1)\times n}$ and $B_i \in \mathbb{R}^{n+1}$ are $\begin{pmatrix} v_0^i & \cdots & v_n^i \\ 1 & \cdots & 1 \end{pmatrix}^{-1} = \left(W_i^{(1)} \mid B_i \right)$ being $\{v_0^i, \ldots, v_n^i\}$ the set of vertices of the maximal simplex σ_i of K. The function f_1 is then defined as $\phi_1(W^{(1)}; y; b_1) = W^{(1)}y + b_1$.
Secondly, the matrix of weights $W^{(2)} \in \mathcal{M}_{\ell(m+1)\times k(n+1)}$ encodes the vertex map φ and it is composed of values zeros and ones. An element of $W^{(2)}$ has value 1 if the corresponding vertices in K and L are related by the vertex map φ, and it has value 0 otherwise. Then, $W^{(2)} = \left(W^{(2)}_{s_1,s_2}\right)$ where $W^{(2)}_{s_1,s_2} = \begin{cases} 1 \text{ if } \varphi(v_t^i) = u_r^j, \\ 0 \text{ otherwise;} \end{cases}$ being $s_1 = j(r+1)$ and $s_2 = i(t+1)$ for $i = 1, \ldots, k$; $j = 1, \ldots, \ell$; $t = 0, \ldots, n$; and $r = 0, \ldots, m$. The bias term b_2 is the null vector. Then, the function f_2 is defined as $\phi_2(W^{(2)}; y; b_2) = W^{(2)}y$.
The output of the second hidden layer can be seen as the juxtaposition of ℓ vectors of dimension $m+1$, one vector for each simplex in the simplicial complex L. Each of these vectors represents the barycentric coordinates of $\varphi_c(x)$ with respect to the corresponding simplex in L.
Finally, only vectors whose all coordinates are greater than or equal to 0 are considered. This condition encodes the simplices of L to which $\varphi_c(x)$ belongs. Then, $\phi_3(W^{(3)}; y; b_3)$ maps the barycentric coordinates of $\varphi_c(x)$ with respect to each maximal simplex of L to which $\varphi_c(x)$ belongs, to the Cartesian coordinates of $\varphi_c(x)$. Specifically, $W^{(3)} = \left(W_1^{(3)} \;\cdots\; W_\ell^{(3)}\right) \in \mathcal{M}_{m\times \ell(m+1)}$, being $W_j^{(3)} = \left(u_0^j \;\cdots\; u_m^j\right)$; and b_3 is the null vector. Finally, f_3 is defined as $\phi_3(W^{(3)}; y; b_3) = \frac{\sum_{j=1}^{\ell} z^j \psi(y^j)}{\sum_{j=1}^{\ell} \psi(y^j)}$ for $y = \begin{pmatrix} y^1 \\ \vdots \\ y^\ell \end{pmatrix} \in \mathcal{M}^{\ell\cdot(m+1)}$, with $z^j = W_j^{(3)} y^j$ and $\psi(y^j) = 1$ if all the coordinates of y^j are greater than or equal to 0 and $\psi(y^j) = 0$ otherwise.
The particular choice of ϕ_3 and ψ is motivated by the use of the barycentric coordinates that depend on the maximal simplex considered. Besides, maximal simplices can share common vertices. Then, the map ψ is used to determine if a given input is located in a specific simplex. The map ϕ_3 is used to normalize the result in case that a point belongs to more than one simplex. □

Summing up, Proposition 2 establishes that a two-hidden-layer feed-forward network can act equivalently to a simplicial map. The architecture and the specific computation of the parameters of the network are provided in the proof of the theorem. Now, we have all the ingredients to state and prove a constructive version of the Universal Approximation Theorem that approximates any continuous function between triangulable spaces arbitrary close. Figure 1 illustrates the process.

Theorem 3 ([10]) *Given a continuous function $g : X \to Y$ between two finitely triangulable metric spaces X and Y and finite triangulations (K, τ_K) and (L, τ_L) of, respectively, X and Y, a two-hidden-layer feed-forward network $\mathcal{N}$ such that $||g - \tilde{\mathcal{N}}|| \le \epsilon$, being $\tilde{\mathcal{N}} = \tau_L^{-1} \circ \mathcal{N} \circ \tau_K$, can be explicitly defined.*

Proof By Proposition 1, there exists a simplicial approximation φ_c such that $||g - \tilde{\varphi_c}|| \le \epsilon$. Finally, by Proposition 2, there exists $\mathcal{N}$ such that $\mathcal{N} = \varphi_c$ in all the domain. □

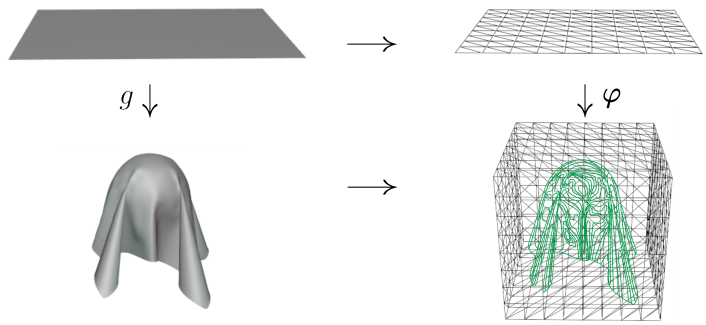

Figure 1: From left to right: a continuous function that deforms a plane into a *cloth-like* folding and a simplicial map between triangulations of both spaces. Then, the triangulations would be subdivided until reaching a desired proximity between the continuous and the simplicial version. Finally, that simplicial map would be used to define a two-hidden neural network.

3 Conclusions and Further Work

The main contribution presented here is a *constructive* method to find the *exact* weights of a two-hidden-layer feed-forward network without the need for a training process, which approximates a continuous function between two finitely triangulable metric spaces. Such technical restriction has no influence on real-world problems, where non-finitely triangulable metric spaces are very odd.

Currently, most of the real-world applications of neural networks try to find a set of weights via a local search method (mainly based on gradient of a loss function) on a large amount of hidden layers. The present alternative method can be a good approach for computing boundaries of complex spaces in classification problems or volumes in medical applications.

As further work, we plan to use simplicial neural networks to address one of the main weakness of deep learning methods: adversarial examples [11]. Namely, we will study how simplicial neural networks can be constructed by maximizing the margin of the models to improve their robustness.

References

[1] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control Signal and Systems*, 2(4):303–314, 1989.

[2] H. Edelsbrunner and J. Harer. *Computational Topology - an Introduction.* American Mathematical Society, 2010.

[3] N. J. Guliyev and V. E. Ismailov. Approximation capability of two hidden layer feedforward neural networks with fixed weights. *Neurocomputing*, 316:262 – 269, 2018.

[4] N. J. Guliyev and V. E. Ismailov. Approximation capability of two hidden layer feedforward neural networks with fixed weights. *Neurocomputing*, 316:262 – 269, 2018.

[5] A. Hatcher. *Algebraic topology*. Cambridge University Press, Cambridge, 2002.

[6] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[7] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, Mar. 1991.

[8] V. E. Ismailov. On the approximation by neural networks with bounded number of neurons in hidden layers. *Journal of Mathematical Analysis and Applications*, 417(2):963 – 969, 2014.

[9] J. R. Munkres. *Elements of algebraic topology.* Addison-Wesley, 1984.

[10] E. Paluzo-Hidalgo, R. Gonzalez-Diaz, and M. A. Gutiérrez-Naranjo. Two-hidden-layer feed-forward networks are universal approximators: A constructive approach. *Neural Networks*, 131:29–36, Nov 2020.

[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks, 2013.