Annals of Mathematics and Computer Science ISSN: 2789-7206
Vol 25 (2024) 97-111 https://doi.org/10.56947/amcs.v25.392

A PROOF OF CONVERGENCE AND EQUIVALENCE FOR 1D FINITE ELEMENT METHODS AND RELU NEURAL NETWORKS

RONALD KATENDE[1,2*], HENRY KASUMBA[1], GODWIN KAKUBA[1] AND JOHN MANGO[1]

Abstract. This paper investigates the convergence and equivalence properties of the Finite Element Method (FEM) and Rectified Linear Unit Neural Networks (ReLU NNs) in solving differential equations. We provide a detailed comparison of the two approaches, highlighting their mutual capabilities in function space approximation. Our analysis proves the subset and superset inclusions that establish the equivalence between FEM and ReLU NNs for approximating piecewise linear functions. Furthermore, a comprehensive numerical evaluation is presented, demonstrating the error convergence behavior of ReLU NNs as the number of neurons per basis function varies. Our results show that while increasing the number of neurons improves approximation accuracy, this benefit diminishes beyond a certain threshold. The maximum observed error between FEM and ReLU NNs is 10^{-4}, reflecting excellent accuracy in solving partial differential equations (PDEs). These findings lay the groundwork for integrating FEM and ReLU NNs, with important implications for computational mathematics and engineering applications.

1. Introduction

The equivalence between the Finite Element Method (FEM) and neural networks with Rectified Linear Unit (ReLU) activation functions has gained considerable attention due to its potential for efficiently solving partial differential equations (PDEs) [9, 8, 19, 18, 20]. FEM and ReLU-based neural networks are both widely employed in engineering and applied mathematics for addressing PDEs, yet they differ fundamentally in their approaches. FEM discretizes a continuous domain into finite elements, facilitating accurate numerical solutions but often becoming computationally intensive for high-dimensional problems, which can limit its scalability in large simulations [2]. In contrast, ReLU neural networks offer flexibility in approximating complex functions by leveraging data-driven learning, making them particularly efficient for high-dimensional and nonlinear spaces without the need for traditional meshing techniques [25, 23, 11]. The emergence of Physics-Informed Neural Networks (PINNs) has further bridged

Date: Received: Feb 22, 2024; Accepted: Nov 16, 2024.
* Corresponding author.
2010 *Mathematics Subject Classification.* Primary 65N30; 65N75; Secondary 68T07.
Key words and phrases. Numerical Analysis, Rectified Linear Unit Function, Finite Element Method, Neural Networks, Equivalence.

classical numerical methods and machine learning by embedding physical laws directly into the training process [5]. This innovation allows neural networks to solve both forward and inverse PDE problems with notable accuracy [16, 4]. This paper investigates the conditions under which ReLU neural networks can effectively approximate FEM solutions, focusing on the minimal network configuration necessary to represent a FEM basis function accurately. Additionally, we analyze the convergence properties and function space equivalence between FEM and ReLU-based networks. By establishing their mutual inclusivity in function space approximation, this study aims to lay a foundation for hybrid models that leverage the strengths of both methods, providing scalable, robust solutions for a wide range of PDE-based applications.

2. Preliminaries

Consider the problem of solving the PDE

$$\mathcal{L}u = f, \quad \text{on } \Omega \tag{2.1}$$

over a domain Ω, where $\mathcal{L}$ is a differential operator, u is the unknown solution, and f encodes the data relevant to the PDE, including boundary and initial conditions. We aim to determine the minimal number of neurons necessary for a neural network to approximate a FEM basis (tent) function effectively. Beyond this optimal neuron count, the network output remains largely stable, while configurations with fewer neurons may achieve optimality but at increased computational expense. Previous work has established the theoretical equivalence of FEM and ReLU neural networks, notably by Juncai He et al. [25] and Badia et al. [11].

Theorem 2.1 ([25])**.** *A continuous function $f : \mathbb{R}^d \to \mathbb{R}$, piecewise linear on subdomains, can be represented by a ReLU deep neural network (DNN), where d corresponds to the dimension of the finite element space. The number of hidden layers, N_{layer}, is bounded by $\lceil \log_2(d+1) \rceil$, and the required number of neurons is given by*

$$N_{neurons} = \begin{cases} \mathcal{O}\left(d \cdot 2^{mM+(d+1)(m-d-1)}\right) & \text{if } m \geq d+1, \\ \mathcal{O}\left(d \cdot 2^{mM}\right) & \text{if } m < d+1, \end{cases}$$

where M is such that $m \leq M \leq m!$, representing the number of subdomains.

We extend this foundational work by analyzing the optimal neuron count, as well as convergence and function space equivalence across both FEM and ReLU-based approaches. Theorem 2.1 illustrates that a single hidden layer is sufficient for convergence to the corresponding basis (tent) function in one-dimensional settings. Consequently, we constrain our network to a single hidden layer with k neurons. For a one-layer ReLU network with hidden layer weights $\{w_1^1, w_2^1, \ldots, w_k^1\}$, output weights $\{w_1^2, w_2^2, \ldots, w_k^2\}$, hidden layer neuron values $\{p_1, p_2, \ldots, p_k\}$, hidden layer biases $\{b_1, b_2, \ldots, b_k\}$, and output bias b_0, the architecture is depicted in Figure 1.

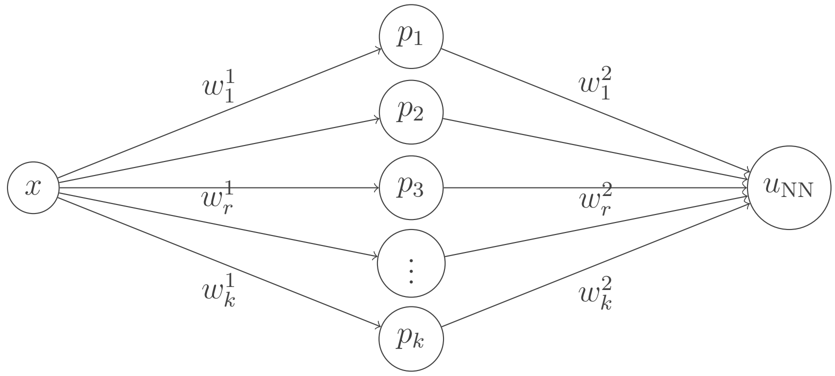

FIGURE 1. A single-layer neural network with k neurons.

The value of the i^{th} neuron in the hidden layer is defined as

$$p_i = w_i^{(1)} x + b_i,$$

where $w_i^{(1)}$ is the weight for the input x associated with the i^{th} neuron, and b_i is the bias term. Using a Rectified Linear Unit (ReLU) activation function, the network output u_{NN} becomes

$$u_{\text{NN}} = \text{ReLU}\left(\sum_{i=1}^{k} w_i^{(2)} \cdot \text{ReLU}(w_i^{(1)} x + b_i) + b_0\right),$$

where $w_i^{(2)}$ represents the weights from the hidden layer to the output layer and b_0 is the output bias. Since the ReLU function is defined as $\text{ReLU}(x) = \max(0, x)$, we can rewrite the expression as

$$\begin{aligned} u_{\text{NN}} &= \max\left(0, \sum_{i=1}^{k} w_i^{(2)} \cdot \max(0, w_i^{(1)} x + b_i) + b_0\right) \\ &= \max\left(0, \left[w_1^{(2)} \max(0, w_1^{(1)} x + b_1) + w_2^{(2)} \max(0, w_2^{(1)} x + b_2) + \cdots + w_k^{(2)} \max(0, w_k^{(1)} x + b_k)\right] + b_0\right). \end{aligned} \tag{2.2}$$

The finite element method (FEM) solution for a one-dimensional problem on n elements is given by

$$u_{\text{FEM}} = \sum_{r=1}^{n} \alpha_r \phi_r(x),$$

where α_r is the coefficient obtained from solving a linear system, and $\phi_r(x)$ is the piecewise linear basis function defined by

$$\phi_r(x) = \begin{cases} \frac{x - x_{r-1}}{x_r - x_{r-1}}, & \text{if } x_{r-1} \le x \le x_r, \\ \frac{x_{r+1} - x}{x_{r+1} - x_r}, & \text{if } x_r \le x \le x_{r+1}, \\ 0, & \text{otherwise.} \end{cases}$$

3. Equivalence Between FEM and Neural Networks

To establish the equivalence between the FEM and neural network solutions, we examine their parameter, convergence, and function space characteristics.

3.1. Equivalence in Parameters. We begin by expressing the FEM basis function ϕ_r in terms of the neural network model. Equating the NN solution in (2.2) with the FEM solution, we get

$$u_{\text{FEM}} = u_{\text{NN}},$$

implying

$$\sum_{r=1}^{n} \alpha_r \phi_r(x) = \max\left(0, \sum_{i=1}^{k} w_i^{(2)} \cdot \max(0, w_i^{(1)} x + b_i) + b_0\right),$$

where $w_i^{(1)}$ are the hidden layer weights, $w_i^{(2)}$ are the output weights, b_i are the hidden layer biases, and b_0 is the output bias. The FEM basis functions ϕ_r can be re-expressed using the ReLU function by the following identities

$$\frac{x - x_{r-1}}{x_r - x_{r-1}} = \frac{\max(0, x - x_{r-1}) - 2\max(0, x - x_r) + \max(0, x_{r+1} - x)}{2h},$$

$$\frac{x_{r+1} - x}{x_{r+1} - x_r} = \frac{\max(0, x - x_{r+1}) - 2\max(0, x - x_r) + \max(0, x_{r+1} - x)}{2h},$$

where $h = x_{r+1} - x_r = x_r - x_{r-1}$. Thus, the FEM solution becomes

$$\begin{aligned} u_{\text{FEM}} = \frac{1}{2h} \sum_{r=1}^{n} \alpha_r \Big[&\max(0, x - x_{r-1}) + \max(0, x_{r-1} - x) - 2\max(0, x - x_r) \\ &+ 2\max(0, x_r - x) + \max(0, x - x_{r+1}) + \max(0, x_{r+1} - x) \Big]. \end{aligned} \tag{3.1}$$

Equating (2.2) and (3.1), we obtain

$$\max\left(0, \sum_{i=1}^{k} w_i^{(2)} \max(0, w_i^{(1)} x + b_i) + b_0\right) = \frac{1}{2h} \sum_{r=1}^{n} \alpha_r \left[\max(0, x - x_{r-1}) + \cdots + \max(0, x_{r+1} - x)\right].$$

Matching terms, we determine parameter relationships

$$b_0 = 0, \quad b_i = \{-x_0, x_0, -x_1, x_1, -x_2, x_2\}, \quad w_i^{(1)} = \{1, -1, 1, -1, 1, -1\},$$

and

$$w_i^{(2)} = \left\{\frac{\alpha_1}{2h}, \frac{\alpha_1}{2h}, -\frac{\alpha_1}{h}, -\frac{\alpha_1}{h}, \frac{\alpha_1}{2h}, \frac{\alpha_1}{2h}\right\}.$$

Therefore, the neural network output for a single basis function is

$$u_{\text{NN}}(x) = \mathbf{w}^{(2)} \cdot \max(0, \mathbf{w}^{(1)} x + \mathbf{b}) = \alpha \mathbf{B}, \tag{3.2}$$

where

$$\alpha = \begin{bmatrix} \frac{\alpha_1}{2h} & \frac{\alpha_1}{2h} & -\frac{\alpha_1}{h} & -\frac{\alpha_1}{h} & \frac{\alpha_1}{2h} & \frac{\alpha_1}{2h} \end{bmatrix},$$

and

$$\mathbf{B} = \begin{bmatrix} \max(0, 1 - x_0) \\ \max(0, -1 + x_0) \\ \max(0, 1 - x_1) \\ \max(0, -1 + x_1) \\ \max(0, 1 - x_2) \\ \max(0, -1 + x_2) \end{bmatrix}.$$

Thus, for each FEM basis function, the neural network requires six neurons, making $k = 6n$. This yields the following relationships for all $r = 1, 2, \ldots, n$

$$w_r^{(1)} = \{1, -1, 1, -1, 1, -1\},$$

$$b_r = \{-x_{r-1}, x_{r-1}, -x_r, x_r, -x_{r+1}, x_{r+1}\},$$

$$w_r^{(2)} = \left\{ \frac{\alpha_r}{2h}, \frac{\alpha_r}{2h}, -\frac{\alpha_r}{h}, -\frac{\alpha_r}{h}, \frac{\alpha_r}{2h}, \frac{\alpha_r}{2h} \right\}.$$

3.2. **Equivalence in Convergence.** To establish convergence of $u_{\text{NN}}(x)$ to $u_{\text{FEM}}(x)$, we examine uniform and pointwise convergence as the number of neurons $k \to \infty$.

3.2.1. *Uniform Convergence.* Uniform convergence is defined as

$$\lim_{k \to \infty} \|u_{\text{FEM}} - u_{\text{NN}}\|_{L^\infty([0,1])} = 0,$$

where

$$u_{\text{FEM}}(x) = \sum_{r=1}^{n} \alpha_r \phi_r(x),$$

with $\phi_r(x)$ as piecewise linear basis functions on subintervals $[x_{r-1}, x_r]$. As $n \to \infty$, these subintervals diminish, and

$$\|u - u_{\text{FEM}}\|_{L^\infty([0,1])} \to 0.$$

The ReLU-NN approximation

$$u_{\text{NN}}(x) = \sum_{i=1}^{k} w_i^{(2)} \text{ReLU}(w_i^{(1)} x + b_i) + b_0$$

can uniformly approximate any continuous piecewise linear function for large k. Hence, for any $\epsilon > 0$, a sufficiently large k exists such that

$$\|u_{\text{FEM}} - u_{\text{NN}}\|_{L^\infty([0,1])} < \epsilon,$$

yielding

$$\lim_{k \to \infty} \|u_{\text{FEM}} - u_{\text{NN}}\|_{L^\infty([0,1])} = 0.$$

3.2.2. *Pointwise Convergence.* Pointwise convergence is verified by

$$\lim_{k\to\infty} |u_{\mathrm{FEM}}(x) - u_{\mathrm{NN}}(x)| = 0, \quad \forall x \in [0,1].$$

Let $u(x)$ be the target function. For $u_{\mathrm{FEM}}(x) = \sum_{r=1}^{n} \alpha_r \phi_r(x)$, we have

$$\lim_{n\to\infty} u_{\mathrm{FEM}}(x) = u(x).$$

Similarly, for the ReLU-NN approximation $u_{\mathrm{NN}}(x)$, each basis function $\phi_r(x)$ can be approximated by ReLU-activated neurons, allowing

$$\lim_{k\to\infty} \left| \sum_{r=1}^{n} \alpha_r \phi_r(x) - u_{\mathrm{NN}}(x) \right| = 0.$$

Thus, we obtain

$$\lim_{k\to\infty} u_{\mathrm{NN}}(x) = \lim_{n\to\infty} u_{\mathrm{FEM}}(x) = u(x).$$

For any fixed $x \in [0,1]$ and any $\delta > 0$, there exists k such that

$$|u_{\mathrm{FEM}}(x) - u_{\mathrm{NN}}(x)| < \delta.$$

Thus, we conclude

$$\lim_{k\to\infty} |u_{\mathrm{FEM}}(x) - u_{\mathrm{NN}}(x)| = 0, \quad \forall x \in [0,1].$$

3.3. **Equivalence in Function Spaces.** We establish the equivalence between the function spaces of ReLU neural networks (NNs) and finite element methods (FEM) by proving mutual inclusion. The FEM approximation space V_{FEM} consists of piecewise linear basis functions $\phi_r(x)$ over a finite partition of $\Omega = [0,1]$

$$V_{\mathrm{FEM}} = \left\{ u \in H_0^1(\Omega) : u(x) = \sum_{r=1}^{n} \alpha_r \phi_r(x),\ \alpha_r \in \mathbb{R} \right\}.$$

Here, $H_0^1(\Omega)$ is the Sobolev space of functions with square-integrable first derivatives that vanish on the boundary, typical for FEM solutions.

For a sufficiently deep and wide ReLU NN, the network can approximate any continuous piecewise-linear function. Define the ReLU NN approximation space V_{NN} as

$$V_{\mathrm{NN}} = \left\{ u : u(x) = \sum_{i=1}^{k} w_i^{(2)} \operatorname{ReLU}(w_i^{(1)} x + b_i) + b_0,\ w_i^{(1)}, w_i^{(2)}, b_i, b_0 \in \mathbb{R} \right\}.$$

To show $V_{\mathrm{FEM}} \subset V_{\mathrm{NN}}$, each FEM basis function $\phi_r(x)$ (continuous piecewise-linear over $[x_{r-1}, x_{r+1}]$) can be represented by a ReLU NN using a finite number of neurons, thus any linear combination in V_{FEM} can be approximated in V_{NN}

$$V_{\mathrm{FEM}} \subseteq V_{\mathrm{NN}}.$$

Conversely, since ReLU NNs approximate continuous piecewise-linear functions, any function $u(x) \in V_{\mathrm{NN}}$ can be expressed as a sum of continuous linear segments. FEM basis functions can similarly approximate any continuous piecewise-linear function over a partition, so $V_{\mathrm{NN}} \subseteq V_{\mathrm{FEM}}$

$$V_{\mathrm{NN}} \subseteq V_{\mathrm{FEM}}.$$

Thus, $V_{\text{FEM}} = V_{\text{NN}}$, implying that ReLU NNs and FEM are equivalent for continuous piecewise-linear function approximation over $H_0^1(\Omega)$.

3.4. **Equivalence in Consistency.** To show consistency equivalence, we analyze error convergence for FEM and ReLU NNs as discretization is refined.

3.4.1. *Consistency in Finite Element Method (FEM).* For FEM, let $u_{\text{FEM}} \in V_{\text{FEM}}$ approximate the exact solution $u \in H_0^1(\Omega)$. The FEM consistency error is

$$\|u - u_{\text{FEM}}\|_{H^1(\Omega)} \leq Ch^p,$$

where h is the mesh size, with $h \to 0$ as the partition refines, p is the polynomial degree of FEM basis functions, C depends on u's regularity but not on h. As $h \to 0$, the FEM solution u_{FEM} converges to u in the H^1-norm, demonstrating FEM consistency.

3.4.2. *Consistency in ReLU Neural Networks.* Let u_{NN} approximate u using a ReLU NN over Ω. The NN consistency error is

$$\|u - u_{\text{NN}}\|_{L^2(\Omega)} \leq C'k^{-\alpha},$$

where k is the neuron count in the hidden layer, $\alpha > 0$ depends on network depth, width, and architecture, C' depends on u's regularity and network properties, independent of k. As $k \to \infty$, u_{NN} converges to u in the L^2-norm, implying that increasing network capacity improves consistency.

3.4.3. *Equivalence of Consistency in the Limit.* As $h \to 0$ for FEM and $k \to \infty$ for NNs, both consistency errors vanish

$$\lim_{h\to 0} \|u - u_{\text{FEM}}\|_{H^1(\Omega)} = 0, \quad \text{and} \quad \lim_{k\to\infty} \|u - u_{\text{NN}}\|_{L^2(\Omega)} = 0.$$

This demonstrates that FEM and ReLU NNs achieve convergence to u through mesh refinement and network capacity increase, respectively, affirming that both methods can yield equivalent accuracy in the limit.

3.5. **Equivalence in Robustness.** Robustness measures the sensitivity of Finite Element Method (FEM) and ReLU neural network (NN) solutions to input perturbations. We analyze robustness by examining each method's response to noise.

3.5.1. *Robustness in Finite Element Method (FEM).* For FEM, robustness is assessed by introducing noise η in the force vector f, yielding the perturbed system

$$Ku^{\eta}_{\text{FEM}} = f + \eta,$$

where K is the stiffness matrix and u^{η}_{FEM} is the perturbed solution. Robustness requires

$$\|u^{\eta}_{\text{FEM}} - u_{\text{FEM}}\| \leq C\|\eta\|,$$

where C is a constant. Subtracting the original system $Ku_{\text{FEM}} = f$ gives

$$K(u^{\eta}_{\text{FEM}} - u_{\text{FEM}}) = \eta, \quad \text{or} \quad \delta u_{\text{FEM}} = K^{-1}\eta,$$

where $\delta u_{\text{FEM}} = u^{\eta}_{\text{FEM}} - u_{\text{FEM}}$. Thus,

$$\|\delta u_{\text{FEM}}\| \leq \|K^{-1}\|\|\eta\|,$$

showing that FEM robustness depends on the norm $\|K^{-1}\|$, which controls sensitivity to perturbations.

3.5.2. *Robustness in ReLU Neural Networks.* For ReLU NNs, robustness is determined by the output's sensitivity to input perturbations. Given an input x perturbed by ξ, we have

$$\delta u_{\mathrm{NN}} = u_{\mathrm{NN}}(x + \xi; \theta) - u_{\mathrm{NN}}(x; \theta),$$

with the robustness condition

$$\|\delta u_{\mathrm{NN}}\| \leq L' \|\xi\|,$$

where L' is the Lipschitz constant, depending on network architecture and regularization.

3.5.3. *Regularization Techniques for Robustness in ReLU NNs.* The Lipschitz constant L' is crucial for NN robustness, often controlled by regularization;

(1) Weight Decay: L_2-regularization penalizes large weights, minimizing

$$\mathcal{L}(\theta) = \mathcal{L}_0(\theta) + \frac{\lambda}{2} \sum_{l=1}^{L} \|W_l\|^2,$$

where λ is a regularization parameter and W_l is the weight matrix of layer l. Weight decay reduces L', yielding $L' \leq C\lambda^{-1/2}$ for some constant C.

(2) Dropout: By randomly omitting neurons during training, dropout creates a more distributed representation, enhancing robustness and reducing overfitting. The dropout-modified NN output is

$$\tilde{u}_{\mathrm{NN}}(x; \theta) = W_L \sigma(\tilde{W}_{L-1} \sigma(\cdots \sigma(\tilde{W}_1 x + b_1) \cdots) + b_{L-1}) + b_L,$$

where $\tilde{W}_l$ represents weights after dropout.

3.5.4. *Regularization in FEM: Damping.* In FEM, robustness can be improved by modifying the stiffness matrix with a damping term αI (where $\alpha > 0$), resulting in a modified matrix $K' = K + \alpha I$. The perturbed system

$$K' u^{\eta}_{\mathrm{FEM}} = f + \eta$$

is more robust, as $\|K'^{-1}\| < \|K^{-1}\|$, reducing sensitivity to noise η. Both FEM and ReLU NNs maintain robustness under perturbations: FEM robustness is governed by $\|K^{-1}\|$, while NN robustness is controlled via regularization that bounds the Lipschitz constant L'. These methods achieve solution stability under noisy or irregular input conditions by their respective robustness techniques.

Theorem 3.1 (Depth-Enhanced Convergence Rate for ReLU Neural Networks in Approximating H^1 Functions)**.** *Let $u \in H^1(\Omega)$ be a target function on $\Omega = [0, 1]$, and let u_{FEM} and u_{NN} represent the FEM and ReLU-NN approximations of u, with a piecewise linear basis for FEM and a ReLU-activated neural network with width k (neurons per layer) and depth d (hidden layers). Suppose that u_{NN} is constructed such that;*

(i) it uses a ReLU activation function, allowing piecewise linear approximation

(ii) it satisfies equivalence in function spaces with FEM, as established in the manuscript.

Then, there exists a constant $C > 0$ and an exponent $\gamma(d) = \gamma_0 + \delta d$, where γ_0 and δ are constants depending on u's regularity and the ReLU-NN structure, such that

$$\|u - u_{NN}\|_{L^2(\Omega)} \leq C(k \cdot d)^{-\gamma(d)}$$

as $k \to \infty$ and $d \to \infty$, where for functions in $H^1(\Omega)$, $\gamma(d)$ achieves up to $\gamma(d) = 1/2 + \delta d$, indicating that increasing depth improves the convergence rate. This rate enhancement applies only for activation functions like ReLU that support piecewise linearity and sufficient neuron allocation in each layer.

Proof. Since $u_{\mathrm{FEM}}, u_{\mathrm{NN}} \in H^1(\Omega)$, we assume both capture piecewise-linear properties in H^1. By equivalence, for sufficiently large k, d, u_{NN} uniformly approximates any continuous piecewise-linear $f \in H^1$, including FEM basis functions. For fixed d and width k, a ReLU-NN approximates any $u \in H^1(\Omega)$ with error $O(k^{-\gamma_0})$, where $\gamma_0 = \frac{1}{2}$. Increasing d refines approximation by allowing layers to segment linearly. Defining u_d as the depth-d approximation, we obtain an error $O((k \cdot d)^{-\gamma(d)})$, where $\gamma(d) = \frac{1}{2} + \delta d$ for $\delta > 0$. To bound the L^2-norm error, we note that pointwise convergence of $u_{\mathrm{NN}} \to u$ as $k, d \to \infty$ and $u \in H^1(\Omega)$ implies L^2-convergence by Sobolev embedding. Thus,

$$\|u - u_{\mathrm{NN}}\|_{L^2(\Omega)} \leq C(k \cdot d)^{-\gamma(d)},$$

where $\gamma(d)$ improves as d increases, leveraging ReLU's piecewise linearity for recursive refinement. Non-piecewise-linear activations lack this rate enhancement. □

3.5.5. *Connection to Equivalence.* This theorem builds on the equivalence established in the manuscript by quantifying the rate at which ReLU-NNs converge to FEM in function approximation, with an enhanced rate through depth. The established equivalence ensures that both FEM and ReLU-NNs operate within the same function space $H^1(\Omega)$. This theorem then extends the equivalence by proving that ReLU-NNs, unlike FEM, can leverage depth to achieve faster convergence rates, offering a computational advantage. This result justifies the use of deeper ReLU networks as a viable alternative to FEM for efficiently approximating solutions to PDEs, especially when higher precision is required. The improved convergence rate suggests that ReLU-NNs can reach FEM-level accuracy with potentially fewer overall neurons, by appropriately leveraging depth. Practically, this highlights ReLU-NNs' scalability and efficiency, particularly for high-dimensional or computationally intensive problems where traditional FEM becomes inefficient due to mesh complexity. This theorem, therefore, not only reinforces the equivalence of ReLU-NNs and FEM in terms of approximation capabilities but also demonstrates how ReLU-NNs can achieve superior efficiency by utilizing network depth, thereby offering a powerful framework for computational mathematics applications where both accuracy and efficiency are critical.

4. Numerical Results

We present numerical results for the FEM-ReLU-NN model with 3, 6, and 9 neurons per basis function to approximate target functions and solve one-dimensional PDEs. The equivalence of FEM and ReLU NN solutions is assessed by comparing absolute errors across various configurations. We first approximate the cosine function over the domain $[0, 1]$ using 10 evenly spaced points. As shown in Table 1, the approximation error is negligible when using more than 6 neurons. Absolute errors are evaluated for different neuron counts per basis function (p) and various values of k.

Table 1. Absolute error for the FEM-ReLU-NN model of architecture $(1, p \cdot k, k, 1)$ in interpolating $\cos(x)$ on $[0, 1]$, with $p = 3, 6$.

Neurons per basis function, p	1	3	6	9
Absolute Error, $k = 2$	1.2416×10^{-2}	2.4653×10^{-3}	1.6539×10^{-4}	1.0216×10^{-4}
Absolute Error, $k = 5$	8.7155×10^{-3}	7.8617×10^{-4}	2.4554×10^{-5}	2.4008×10^{-5}
Absolute Error, $k = 10$	1.7461×10^{-4}	3.6519×10^{-5}	6.4439×10^{-6}	6.4432×10^{-9}

Table 1 indicates a clear trend. That is, as neurons per basis function (p) increase, absolute error decreases across all k values. The most notable error reduction occurs as p increases from 1 to 6, with minimal further gains for $p = 9$, suggesting that 6 neurons per basis function strikes an optimal balance between complexity and accuracy. Beyond this threshold, further increases offer diminishing returns, indicating a saturation in the network's capacity to capture function nuances. Next, we apply the FEM-ReLU-NN model to solve three PDEs, i.e., Burgers' equation, Schrödinger's equation, and the wave equation. Figures 2, 3, and 4 display solutions from both FEM and ReLU-NNs, showing high agreement. In particular, the Schrödinger equation's peak around $x \approx 0.75$ is accurately captured by both methods, validating ReLU-NNs' ability to approximate FEM solutions.

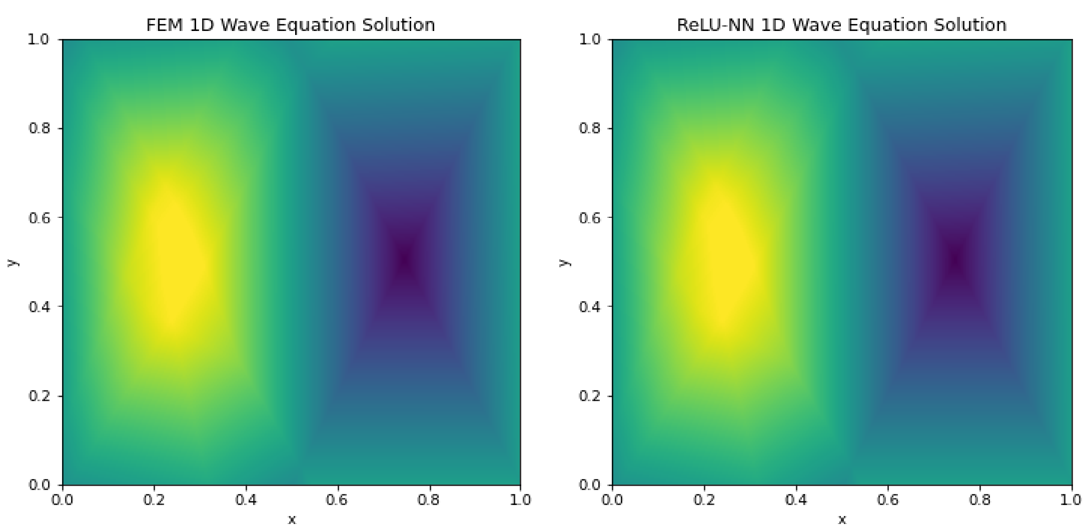

Figure 2. FEM and ReLU-NN solutions for the wave equation.

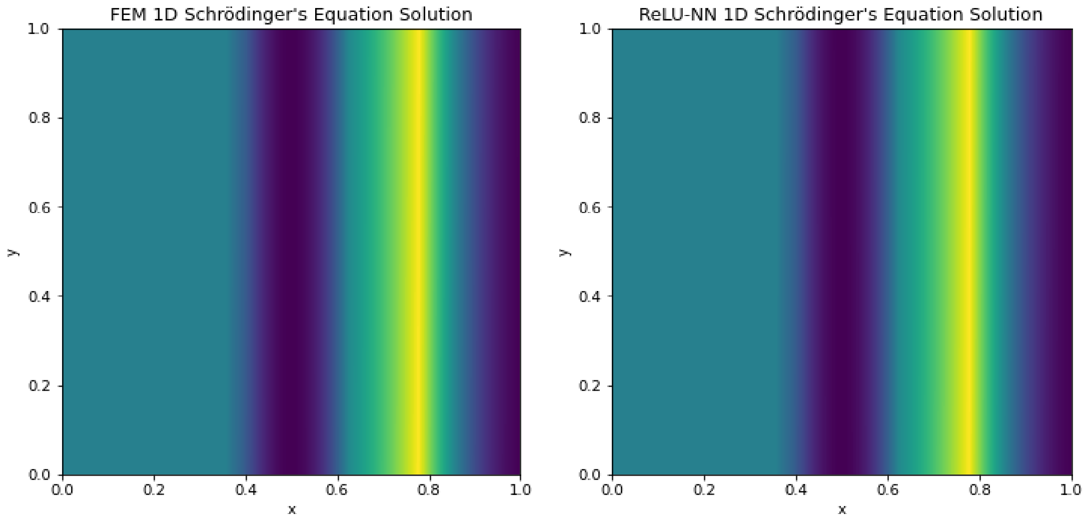

FIGURE 3. FEM and ReLU-NN solutions for the Schrödinger equation.

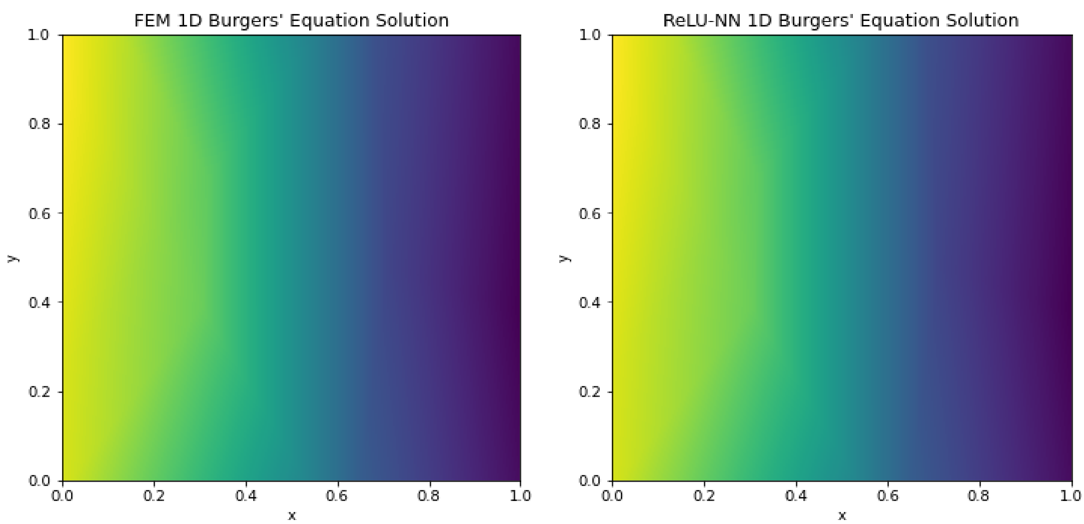

FIGURE 4. FEM and ReLU-NN solutions for the Burgers' equation.

The absolute errors between FEM and ReLU-NN solutions for these PDEs are presented in Table 2. For $k = 2$, the wave equation exhibits the highest error, with errors increasing slightly at $k = 5$, but remaining stable, indicating robustness in both methods. Errors decrease again at $k = 10$, showing that larger networks improve accuracy, though with diminishing returns at higher neuron counts.

Table 2 shows that the wave equation has the highest error consistently, possibly due to the challenges in modeling wave phenomena. However, errors are within acceptable limits for both FEM and ReLU-NNs. As k increases, errors reduce, confirming that larger networks enhance accuracy but with diminishing improvements, consistent with trends seen in function approximation. These results demonstrate the equivalence between FEM and ReLU-NNs in solving PDEs.

TABLE 2. Absolute errors between FEM and ReLU-NN solutions for Burgers', Schrödinger, and wave equations.

Errors	Burgers'	Schrödinger	Wave
$k = 2$	5.00533×10^{-5}	1.76518×10^{-4}	4.32171×10^{-4}
$k = 5$	3.21527×10^{-4}	3.28062×10^{-4}	3.37295×10^{-4}
$k = 10$	3.11879×10^{-4}	3.05617×10^{-4}	3.14139×10^{-4}

ReLU-NNs achieve high accuracy in approximating FEM solutions, proving effective for complex PDEs and capable of capturing detailed physical behaviors, such as wave propagation and quantum effects. Thus, ReLU-NNs offer scalability for higher-dimensional and nonlinear problems while matching FEM's accuracy. This equivalence supports the development of hybrid models that harness ReLU-NN computational efficiency with FEM robustness and precision, advancing computational PDE solving in scenarios where both efficiency and accuracy are crucial.

5. Conclusion

This study illustrates the robust potential of ReLU-based neural networks (ReLU-NNs) in accurately approximating solutions to one-dimensional partial differential equations (PDEs), achieving results comparable to those produced by the Finite Element Method (FEM). Our findings confirm that both FEM and ReLU-NNs provide similar error bounds when approximating continuous piecewise linear functions, establishing a practical equivalence between the two methods. Notably, six neurons per basis function were found to offer an optimal balance between computational efficiency and accuracy, suggesting an effective configuration for ReLU-NNs in PDE approximation. These results point to the feasibility of a hybrid approach that leverages the strengths of both FEM and ReLU-NNs, creating a robust framework for solving complex PDEs where both precision and computational efficiency are essential. The practical implications are substantial, especially in computational fields requiring high accuracy and efficiency, such as aerospace, structural engineering, and physics simulations. In these domains, integrating ReLU-NNs within traditional FEM workflows could enhance performance, potentially reducing computational costs while preserving accuracy. Additionally, ReLU-NNs provide scalability, making them well-suited for managing problems of increasing dimensionality and complexity. This hybrid methodology offers potential benefits such as faster prototyping, detailed simulations, and the capability to handle real-time computations, which are particularly valuable in high-stakes applications like material design and fluid dynamics. Future research should focus on extending the FEM-ReLU-NN equivalence to higher-dimensional problems, where PDEs and their solutions become increasingly complex. Addressing higher-dimensional PDEs introduces challenges such as the curse of dimensionality, which can exponentially raise computational demands and memory usage. Additionally, more sophisticated neural network designs are required to manage large-scale data and intricate function approximations while ensuring numerical stability. Potential strategies to address these challenges include dimensionality reduction, sparse representations, and adaptive mesh refinement. Hybrid models could also capitalize on FEM's ability to handle complex geometries and boundary conditions, while utilizing ReLU-NNs for efficient, data-driven function approximation. By tackling these challenges, the integration of FEM and ReLU-NNs in higher dimensions could yield scalable and efficient solutions for a broad range of real-world problems in computational mathematics and beyond.

Acknowledgement. This work has been supported by the Mathematics for Sustainable Development (MATH4SDG) project, a research and development project running in the period 2021-2026 at Makerere University-Uganda, University of Dar es Salaam-Tanzania, and the University of Bergen-Norway.

References

1. Mitusch, Sebastian K. and Funke, Simon W. and Kuchta, Miroslav, Hybrid FEM-NN models: Combining artificial neural networks with the finite element method, Journal of Computational Physics, vol. 446, pp. 110651, 2021. DOI: https://doi.org/10.1016/j.jcp.2021.110651
2. Feng, Xiaobing and Prohl, Andreas, Error analysis of a mixed finite element method for the Cahn-Hilliard equation, Numerische Mathematik, vol. 99, pp. 47-84, 2004, https://doi.org/10.1007/s00211-004-0546-5
3. Tadmor, Eitan, A review of numerical methods for nonlinear partial differential equations, Bulletin of the American Mathematical Society, vol. 49, pp. 507-554, 2012.
4. Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L., "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422-440, 2021. https://doi.org/10.1038/s42254-021-00314-5.
5. Raissi, M., Perdikaris, P., Karniadakis, G. E., "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," Journal of Computational Physics, vol. 378, pp. 686-707, 2019. doi: https://doi.org/10.1016/j.jcp.2018.10.045
6. Umbrello, D. and Ambrogio, G. and Filice, L. and Shivpuri, R., A hybrid finite element method–artificial neural network approach for predicting residual stresses and the optimal cutting conditions during hard turning of AISI 52100 bearing steel, Materials & Design, vol. 29, no. 4, pp. 873-883, 2008. DOI: https://doi.org/10.1016/j.matdes.2007.03.004
7. Zhang, Zhizhou and Jin, Zeqing and Gu, Grace X., Efficient pneumatic actuation modeling using hybrid physics-based and data-driven framework, Cell Reports Physical Science, vol. 3, no. 4, pp. 100842, 2022. DOI: https://doi.org/10.1016/j.xcrp.2022.100842
8. Ghoreishian Amiri, S.A. and Taheri, E. and Lavasan, A.A. Lavasan, A hybrid finite element model for non-isothermal two-phase flow in deformable porous media, Computers and Geotechnics, vol. 135, pp. 104199, 2021. DOI: https://doi.org/10.1016/j.compgeo.2021.104199
9. Amoiralis, Eleftherios I. and Georgilakis, Pavlos S. and Kefalas, Themistoklis D. and Tsili, Marina A. and Kladas, Antonios G., Artificial Intelligence Combined with Hybrid FEM-BE Techniques for Global Transformer Optimization, IEEE Transactions on Magnetics, vol. 43, no. 4, pp. 1633-1636, 2007. DOI: 10.1109/TMAG.2006.892258
10. Qin, Haibo and Huang, Wei and Song, Binbin and Chen, Shengyong, Hybrid Method for Inverse Design of Orbital Angular Momentum Transmission Fiber Based on Neural Network and Optimization Algorithms, Journal of Lightwave Technology, vol. 40, no. 17, pp. 5974-5985, 2022. DOI: 10.1109/JLT.2022.3185059

11. Badia, Santiago and Li, Wei and Martín, Alberto F., Finite element interpolated neural networks for solving forward and inverse problems, Computer Methods in Applied Mechanics and Engineering, vol. 418, pp. 116505, Jan. 2024. DOI: 10.1016/j.cma.2023.116505
12. Nakayama, T., & Washizu, K. (1981). The boundary element method applied to the analysis of two-dimensional nonlinear sloshing problems. *International Journal for Numerical Methods in Engineering*, 17(11), 1631–1646. https://doi.org/10.1002/nme.1620171105.
13. Rezaei, S., Harandi, A., Moeineddin, A., Xu, B.-X., & Reese, S. (2022). A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: Comparison with finite element method. *Computer Methods in Applied Mechanics and Engineering*, 401, 115616. https://doi.org/10.1016/j.cma.2022.115616.
14. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260. https://doi.org/10.1126/science.aaa8415.
15. Barrett, J. W., Blowey, J. F., & Garcke, H. (1999). Finite element approximation of the Cahn-Hilliard equation with degenerate mobility. *SIAM Journal on Numerical Analysis*, 37(1), 286–318. https://doi.org/10.1137/S0036142997325005.
16. Ainsworth, Mark and Dong, Justin, Galerkin neural network approximation of singularly-perturbed elliptic systems, Computer Methods in Applied Mechanics and Engineering, vol. 402, pp. 115-169, 2022. DOI: https://doi.org/10.1016/j.cma.2022.115169
17. de Bezenac, Emmanuel, "Deep learning for physical processes: incorporating prior scientific knowledge," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 124009, pp. 43-44, 2019. DOI: https://doi.org/10.1088/1742-5468/ab319510.1088/1742-5468/ab3195.
18. Hambli, Ridha and Katerchi, Houda and Benhamou, Claude-Laurent, Multiscale methodology for bone remodelling simulation using coupled finite element and neural network computation, Biomechanics and Modeling in Mechanobiology, vol. 10, pp. 133-145, 2011. DOI: https://doi.org/10.1007/s11440-023-01980-8
19. Ashouri, Majid and Khaleghian, Seyedmeysam and Emami, Anahita, Reduced-order modeling of conductive polymer pressure sensors using finite element simulations and deep neural networks, Structural and Multidisciplinary Optimization, vol. 65, no. 146, 2022. DOI: https://doi.org/10.1007/s11440-023-01980-8
20. Malik, Alexander and Abendroth, Martin and Hütter, Geralf and Kiefer, Bjoern, A hybrid approach employing neural networks to simulate elasto plastic deformation behavior of 3D-form structures, Advanced Engineering Materials, vol. 24, no. 2100641, pp. 1-12, 2022. DOI: 10.1002/adem.202100641
21. Longo, Marcello and Opschoor, Joost A.A. and Disch, Nico and Schwab, Christoph and Zech, Jakob, De Rham compatible Deep Neural Network FEM, Neural Networks, vol. 165, pp. 721–739, Aug. 2023. DOI: 10.1016/j.neunet.2023.06.008
22. Liu, Min and Cai, Zhiqiang, Adaptive two-layer ReLU neural network: II. Ritz approximation to elliptic PDEs, Computers & Mathematics with Applications, vol. 113, pp. 103-116, 2022. DOI: https://doi.org/10.1016/j.camwa.2022.03.010
23. Guan, Shaoheng and Feng, Y. T. and Ma, Gang and Qu, Tongming and Wang, Mengqi and Zhou, Wei, An explicit FEM-NN framework and the analysis of error caused by NN-predicted stress, Acta Geotechnica, 2023. DOI: https://doi.org/10.1007/s11440-023-01980-8
24. Liu, Yingjian and Park, Chanwook and Lu, Ye and Mojumder, Satyajit and Liu, Wing Kam and Qian, Dong, HiDeNN-FEM: a seamless machine learning approach to nonlinear finite element analysis, Computational Mechanics, vol. 72, pp. 173-194, 2023. DOI: https://doi.org/10.1007/s00466-023-02293-z
25. He, Juncai and Li, Lin and Xu, Jinchao and Zheng, Chunyue, Relu Deep Neural Networks and Linear Finite Elements, Journal of Computational Mathematics, vol. 38, no. 3, pp. 502-527, 2020. DOI: https://doi.org/10.4208/jcm.1912-m2019-0186

26. Sun, Shuheng and Zeng, Peng, Finite Element Approximation of Multi-term Time-fractional Diffusion Equations, Journal of Computational Mathematics, vol. 38, no. 3, pp. 399-428, 2020. DOI: https://doi.org/10.4208/jcm.1905-m2019-0084

[1] Department of Mathematics, Makerere University, Kampala, Uganda
Email address: rkatende92@gmail.com

[2] Department of Mathematics, Kabale University, Kikungiri Hill, Kabale, Uganda