Formal Analysis of the Sigmoid Function and Formal Proof of the Universal Approximation Theorem

DUSTIN BRYANT, Independent, United States

JIM WOODCOCK, Southwest University, China, Aarhus University, Denmark, and University of York, United Kingdom

SIMON FOSTER, University of York, United Kingdom

This paper presents a formalized analysis of the sigmoid function and a fully mechanized proof of the Universal Approximation Theorem (UAT) in Isabelle/HOL, a higher-order logic theorem prover. The sigmoid function plays a fundamental role in neural networks; yet, its formal properties—such as differentiability, higher-order derivatives, and limit behavior—have not previously been comprehensively mechanized in a proof assistant. We present a rigorous formalization of the sigmoid function, proving its monotonicity, smoothness, and higher-order derivatives. We provide a constructive proof of the UAT, demonstrating that neural networks with sigmoidal activation functions can approximate any continuous function on a compact interval. Our work identifies and addresses gaps in Isabelle/HOL's formal proof libraries, introducing more simple methods to reason about limits of real functions. By exploiting theorem proving for AI verification, our work enhances trust in neural networks and contributes to the broader goal of verified and trustworthy machine learning.

Additional Key Words and Phrases: Formal verification, Isabelle/HOL, Universal Approximation Theorem, Sigmoid function, Higher-order differentiation, Neural network approximation, Machine-checked proofs, Verified AI, Theorem proving, Real analysis in Isabelle, Trustworthy machine learning

ACM Reference Format:

Dustin Bryant, Jim Woodcock, and Simon Foster. 2025. Formal Analysis of the Sigmoid Function and Formal Proof of the Universal Approximation Theorem. 1, 1 (November 2025), 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Machine learning now underpins safety-critical systems, including autonomous vehicles, medical diagnostics, and robotics. In these domains, informal proofs no longer suffice. A core prerequisite for trustworthy AI is a machine-checked account of expressiveness: can our network architectures approximate the functions we need, and can we trust the proof that says so? The Universal Approximation Theorem (UAT) answers the first question, but its classical proofs are prose-based, often non-constructive (e.g. [7][10][14]) , and challenging to reuse inside verification workflows.

This paper addresses that gap. We provide, to our knowledge, the first fully mechanized, constructive proof of the UAT in a major proof assistant (Isabelle/HOL)[2]. Beyond validating every inference step, our constructive development yields executable witnesses, making the result directly usable within verified AI pipelines. To support this, we build

Authors' addresses: Dustin Bryant, Independent, United States, brydustin@gmail.com; Jim Woodcock, Southwest University, Chongqing, China and Aarhus University, Aarhus, Denmark and University of York, York, United Kingdom, jim.woodcock@york.ac.uk; Simon Foster, University of York, York, United Kingdom, simon.foster@york.ac.uk.

Manuscript submitted to ACM

missing analysis infrastructure in Isabelle to simplify limit proofs, together with a reusable formal theory of the sigmoid function, including monotonicity, smoothness, and a closed-form derivative (via Stirling numbers). The resulting libraries are designed for reuse across mathematics, physics, engineering, and verified machine learning.

1.1 Contributions

(1) **Mechanized UAT (constructive).** A complete, machine-checked proof in Isabelle/HOL that removes hidden assumptions and sets a benchmark for formalizing foundational ML results.
(2) **Sigmoid calculus.** A reusable, verified development of the sigmoid's properties, including a closed-form n-th derivative and smoothness results.
(3) **Real-analysis extensions for Isabelle.** Limit tools that make subsequent formalizations more natural.
(4) **Path to trustworthy neuro-symbolic systems.** By embedding expressivity guarantees inside a logical framework, our results bridge symbolic and subsymbolic AI and provide a baseline for integrating expressivity verification (UAT) with robustness assurance (e.g., adversarial safety, generalization bounds).

1.2 Significance: Why read this paper?

It delivers a first-of-its-kind, machine-checked, constructive UAT; strengthens the foundations of trustworthy AI; extends Isabelle/HOL with analysis tools that others can reuse; and bridges symbolic and subsymbolic AI, so that verification frameworks can rely on formally proved approximation power.

1.3 Paper organization.

Sect. 2 discusses related work: relates the work to existing research by reviewing classical and constructive proofs of the Universal Approximation Theorem, prior analyses of the sigmoid function and its derivatives, and related formal-methods efforts, thereby motivating the need for a fully mechanized, reusable Isabelle/HOL formalization of these foundational results. Sect. 3 formalizes the sigmoid function in Isabelle/HOL, defining it, proving its monotonicity, smoothness, higher-order derivatives (via Stirling numbers), and limit behavior, to establish it as a mathematically sound, sigmoidal activation function suitable for the later mechanized proof of the Universal Approximation Theorem. Sect. 4 extends Isabelle/HOL's analysis libraries by introducing a helpful bridge for ε–N style formulation of limits, creating the formal infrastructure needed for reasoning about smoothness and limits in the mechanized proof of the Universal Approximation Theorem. Sect. 5 presents the mechanized proof of the Universal Approximation Theorem in Isabelle/HOL, showing that any continuous function on a compact interval can be uniformly approximated by a finite linear combination of shifted and scaled sigmoidal functions, thereby establishing the expressive power of single-layer neural networks. That section concludes by illustrating through a worked example how the theorem applies on an actual function. Sect. 6 concludes by highlighting that the mechanization of the sigmoid function and the Universal Approximation Theorem in Isabelle/HOL eliminates informal assumptions, enhances trust in neural network theory, extends Isabelle's real analysis toolkit, and lays a rigorous foundation for future research in verified and trustworthy AI.

2 RELATED WORK

2.1 Universal Approximation.

Classical results show that shallow feed-forward networks are dense in $C(K)$ under broad conditions: Cybenko established universality for sigmoids [7]; Hornik generalized the picture [13]; and Leshno–Lin–Pinkus–Schocken proved

that any *non-polynomial* activation yields universality [18]; see also Pinkus's survey [24]. Constructive refinements give explicit networks and bounds (e.g., [5, 6]), but these proofs are typically on paper, where subtle assumptions can remain implicit. Our work provides a *machine-checked, constructive* UAT in Isabelle/HOL, turning this cornerstone into reusable, verified mathematics.

2.2 Formal Methods and Verified ML.

Isabelle/HOL offers deep support for analysis (e.g., [23]), yet a convenient package for a compact UAT proof is wanting. In parallel, ML verification has advanced through DNN analyzers (Reluplex [15], Marabou [16, 27, 28]) and formal learning theory in Coq and Lean (e.g., [1, 25, 26]). We extend Isabelle's toolkit with reusable formalizations and alternative limit reasoning that these lines of work can build upon.

2.3 Activation Functions and Verified AI.

While much verification targets piecewise-linear activations (ReLU, Leaky ReLU [11, 19, 22]), the sigmoid remains central in theory, probabilistic modeling[21], and is and remains canonical in formulations of the Universal Approximation Theorem [12]. Comprehensive surveys map a vast design space (e.g., [9, 17]; see also [8] for links to initialization), reinforcing the value of a carefully *formalized* smooth activation. Our development closes this gap by verifying the higher-order properties of the sigmoid function—including a closed-form expression for its n^{th} derivative via Stirling numbers—thereby enabling principled symbolic differentiation and smooth analysis in verified machine learning. Together, these results mark a step toward integrating numerical approximation theorems into the broader ecosystem of formally verified mathematics and machine learning.

3 FORMALIZING THE SIGMOID FUNCTION

3.1 Core Definition and Basic Properties

The *sigmoid function*, $\sigma : \mathbb{R} \to \mathbb{R}$, is defined as follows:

$$\sigma(x) = \frac{e^x}{1 + e^x},$$

```
definition sigmoid :: "real ⇒ real" where
"sigmoid = (λ x::real. exp(x) / (1 + exp(x)))"
```

Listing 1. Definition of Sigmoid

and a simple computation shows that an equivalent representation for the sigmoid is $\sigma(x) = \dfrac{1}{1 + e^{-x}}$.

```
lemma sigmoid_alt_def: "sigmoid x = inverse (1 + exp(-x))"
proof -
  have "sigmoid x = (exp(x) * exp(-x)) / ((1 + exp(x))*exp(-x))"
    unfolding sigmoid_def by simp
  also have "... = 1 / (1*exp(-x) + exp(x)*exp(-x))"
    by (simp add: distrib_right exp_minus_inverse)
  also have "... = inverse (exp(-x) + 1)"
    by (simp add: divide_inverse_commute exp_minus)
  finally show ?thesis
```

```
    by simp
qed
```

Listing 2. Alternate Representation of Sigmoid

From the definition of $\sigma(x)$ it is clear that $0 < \sigma(x) < 1$. The alternate representation shows that σ is increasing and $\sigma(0) = 1/2$.

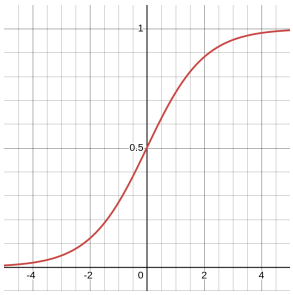

Fig. 1. Sigmoid Function

3.2 Higher-Order Derivatives of σ

Central to the learning aspect of deep learning is computing the derivative of the activation function. In the case of the sigmoid function, we have the following formulae for the first two derivatives:

$$\sigma'(x) = \sigma(x)\,[1 - \sigma(x)]$$
$$\sigma''(x) = \sigma(x)\,[1 - \sigma(x)]\,[1 - 2\sigma(x)]\ .$$

Deriving these equations involves a rather straightforward calculation with the chain rule. Perhaps less well-known is the general representation for the n^{th} derivative of the sigmoid function(see [20]):

$$\sigma^{(n)}(x) \ = \ \sum_{k=1}^{n+1} (-1)^{k+1}\,(k-1)!\,S(n+1,k)\,(\sigma(x))^k$$

where $S(\cdot,\cdot)$ denotes a Stirling number of the second kind, formally defined in Isabelle/HOL as part of `Stirling.thy`[4]. This can easily be proved by induction and amounts to splitting this into two summations, re-indexing one of the summations, and using the fact that

$$k!S(n+1,k) + (k-1)!S(n+1,k-1) = (k-1)!S(n+2,k)\,.$$

We briefly mention the formal statement of this result:

```
theorem nth_derivative_sigmoid:
  "⋀x. Nth_derivative n sigmoid x =
     (Σk = 1..n+1. (-1)^(k+1) * fact (k - 1) *
     Stirling (n+1) k * (sigmoid x)^k)"
```

Listing 3. Derivatives of Sigmoid

We note that we have used the higher-order derivative as defined in [3], which recursively defines the kth derivative. The above fact shows that σ is a smooth function as each of its derivatives is continuous.

3.3 Limit Behaviors and Other Results

Central to the universal approximation theorem are the so-called sigmoidal functions, $f : \mathbb{R} \to \mathbb{R}$, which satisfy the following property:

$$\lim_{x \to -\infty} f(x) = 0 \quad \text{and} \quad \lim_{x \to +\infty} f(x) = 1.$$

```
definition sigmoidal :: "(real ⇒ real) ⇒ bool" where
  "sigmoidal f ≡ (f ⟶ 1) at_top ∧ (f ⟶ 0) at_bot"
```

Listing 4. Definition of Sigmoidal

Indeed, it is clear that σ is sigmoidal; in the case of mechanizing this in Isabelle, we gave a direct $\varepsilon - N$ style proof. Similarly, we showed that:

$$\lim_{x \to -\infty} \sigma'(x) = 0 \quad \text{and} \quad \lim_{x \to +\infty} \sigma'(x) = 0.$$

4 A FORMAL APPROACH TO LIMITS IN ISABELLE/HOL: EQUIVALENCE WITH THE CLASSICAL ε–N DEFINITION

Working with limits in Isabelle/HOL is formulated via *filters*. For real-valued functions on $\mathbb{R}$, that paradigm is correct but perhaps unfamiliar or unnecessarily abstract. In this section we present a convenient ε–N characterization for the filter `at_top` on $\mathbb{R}$, so that proofs can be conducted in the familiar classical style while remaining definitionally equivalent to the library's filter semantics. Consider the following real example:

```
lemma lim_sigmoid_infinity: "(sigmoid ⟶ 1) at_top"
  unfolding tendsto_def
------------------------------------------------------------------------------------------------------------
Output:
proof (prove)
goal (1 subgoal):
 1. ∀S. open S ⟹ 1 ∈ S ⟹ (∀_F x in at_top. sigmoid x ∈ S)
```

Listing 5. Filter-based goal (unfolded)

`tendsto` corresponds to the arrow denoting the limit while `at_top` specifies that the limit is as x goes to *infinity*. Notice that it is awkward to work with the proof goal produced in the output; and this requires specialized knowledge beyond the usual definitions of limits. Instead, we want something natural to capture the feel of a classical $\varepsilon - N$ style proof from real analysis. We sought a more natural framework that captures the feel of a classical $\varepsilon - N$ proof from real analysis goals, like in the following:

```
lemma lim_sigmoid_infinity: "(sigmoid ⟶ 1) at_top"
proof (subst tendsto_at_top_epsilon_def, clarify)
------------------------------------------------------------------------------------------------------------
Output:
proof (state)
```

```
goal (1 subgoal):
 1. ⋀ ε. 0 < ε ⟹ ∃N. ∀ x ≥ N. ¦sigmoid x - 1¦ < ε
```

Listing 6. ε–N goal obtained by rewriting to the classical form

The intended classical statement for real f is:

$$\left(f \xrightarrow[x \to +\infty]{} L\right) \iff \forall \varepsilon > 0 \ \exists N \in \mathbb{R} \ \forall x \geq N : |\, f(x) - L \,| < \varepsilon.$$

Therefore, we developed a set of lemmas which allowed us to compute limits such as this and others. The limit lemma corresponding to the previously mentioned limit above is `tendsto_at_top_epsilon_def` included formally below.

```
lemma tendsto_at_top_epsilon_def:
  "(f ⟶ L) at_top =
   (∀ ε > 0. ∃N::real. ∀ x ≥ N. ¦(f x::real) - L¦ < ε)"
  by (simp add: Zfun_def tendsto_Zfun_iff eventually_at_top_linorder)
```

Listing 7. ε–N characterization at $+\infty$ on $\mathbb{R}$

With this in hand, standard ε–N estimates discharge goals directly. For the logistic sigmoid $\sigma(x) = \dfrac{1}{1+e^{-x}}$, we have

$$0 < 1 - \sigma(x) = \frac{1}{1+e^{x}} < e^{-x}.$$

Choosing $N = \ln(1/\varepsilon)$ then ensures that

$$|\, \sigma(x) - 1 \,| < \varepsilon \quad \text{for all } x \geq N.$$

Thus, (`sigmoid` $\to 1$) `at_top` follows immediately in the ε–N formulation, while remaining definitionally equivalent to the original filter statement. Thus, one can naturally reason about limits without being familiar with the filters paradigm, `Zfun_def`, or any related theories; this makes research in formal proofs of real-valued functions more accessible.

In summary, our formal development shows that the sigmoid function σ is both smooth and sigmoidal. These properties qualify it as an effective activation function, setting the stage for our formal treatment of the Universal Approximation Theorem.

5 THE UNIVERSAL APPROXIMATION THEOREM

The proof of the UAT is central to using sigmoidal functions as approximators. Sigmoidal functions behave similarly to the Heaviside function. To motivate this idea, we briefly consider σ as an approximator. We can parametrize it by introducing a *weight* factor,w, which controls how rapidly the function transitions from ≈ 0 to ≈ 1, and we can parametrize its center by x_k. More concretely, if we parametrize the sigmoid as

$$\sigma_{w,x_k}(x) = \frac{1}{1+e^{-w(x-x_k)}},$$

where $w > 0$ is the weight parameter and x_k is a translation parameter, increasing w stretches the function so that its transition from 0 to 1 becomes sharper. This controlled steepness is key to constructing approximations by combining appropriately weighted and shifted sigmoidal functions. Now we make precise in what sense sigmoidal functions approximate Heaviside functions:

Lemma 5.1. *Sigmoidal Uniform Approximation [Costarelli and Spigler]*

Let $x_0, x_1, \ldots, x_N \in \mathbb{R}$, $N \in \mathbb{N}^+$, be fixed. For every $\varepsilon, h > 0$, there exists $\overline{w} := \overline{w}(\varepsilon, h) > 0$ such that for every $w \geq \overline{w}$ and $k = 0, 1, \ldots, N$, we have

(1) $|\sigma(w(x - x_k)) - 1| < \varepsilon$, *for every $x \in \mathbb{R}$ such that $x - x_k \geq h$;*

(2) $|\sigma(w(x - x_k))| < \varepsilon$, *for every $x \in \mathbb{R}$ such that $x - x_k \leq -h$.*

```
lemma sigmoidal_uniform_approximation:
  assumes "sigmoidal σ"
  assumes "ε > 0" and "h > 0"
  shows "∃ ω >0. ∀ w ≥ ω. ∀ k<length xs.
    (∀x. x - xs!k ≥ h  ⟶ ¦ σ (w * (x - xs!k)) - 1¦ < ε) ∧
    (∀x. x - xs!k ≤ -h ⟶ ¦ σ (w * (x - xs!k))¦ < ε)"
```

Listing 8. Sigmoidal Uniform Approximation

The carefully constructed lemmas from the previous section made this proof nearly trivial. We are almost in a position to state the UAT, but first, we must define some of the last few definitions used in Isabelle. Recall that a function, $f : \mathbb{R} \to \mathbb{R}$, is *bounded* provided there exists $M > 0$ such that $f(\mathbb{R}) \subset [-M, M]$. We formally defined this in Isabelle as:

```
definition bounded_function :: "(real ⇒ real) ⇒ bool" where
  "bounded_function f ⟷ bdd_above (range (λx. ¦f x¦))"
```

Listing 9. Bounded Function Definition

Next, given any interval $[a, b]$, we need a way to partition the interval easily widened a little on the left side into $N + 1$ even subintervals each of length $\frac{b-a}{N}$, that is we need to partition $[a - \frac{b-a}{N}, b]$ uniformly. We defined it in Isabelle as follows:

```
definition unif_part :: "real ⇒ real ⇒ nat ⇒ real list" where
  "unif_part a b N =
     map (λk. a + (real k -1 ) * ((b - a) / real N )) [0..<N+2]"

value "unif_part (0::real) 1 4"
(* Output: [-.25, 0,  0.25, 0.5, 0.75, 1] :: real list *)
```

Listing 10. Uniform Partition Function with Example

This definition may seem somewhat arbitrary, but it is precisely what is required to approximate a continuous function f over $[a, b]$ using sigmoidal functions, which is the UAT. Finally, we are in a position to state our main result.

Theorem 5.2 (Uniform Approximation by Sigmoidal Functions [6]). *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a bounded, sigmoidal function, and let f be a continuous function on the interval $[a, b]$ with $a < b$. Then for every $\varepsilon > 0$, there exists a positive integer N and a real $w > 0$ such that:*

$$\left| \sum_{k=2}^{N+1} \left[f(x_k) - f(x_{k-1}) \right] \sigma\big(w\,(x - x_k)\big) \;+\; f(a)\,\sigma\big(w\,(x - x_0)\big) \;-\; f(x) \right| \;<\; \varepsilon$$

for all $x \in [a, b]$, where $\{x_0, x_1, \ldots, x_{N+1}\}$ is a uniform partition of the interval $\left[a - \frac{b-a}{N},\ b\right]$. That is, f can be approximated uniformly on $[a, b]$ by a finite linear combination of translates and scalings of the sigmoidal function σ.

Remark: This theorem shows that f can be *learned* by a neural network consisting of a single layer with $N + 1$ neurons that use a sigmoidal activation function σ.

```
theorem sigmoidal_approximation_theorem:
  assumes sigmoidal_function: "sigmoidal σ"
  assumes bounded_sigmoidal: "bounded_function σ"
  assumes a_lt_b: "a < b"
  assumes contin_f: "continuous_on {a..b} f"
  assumes eps_pos: "0 < ε"
  defines "xs N ≡ unif_part a b N"
  shows "∃N::nat. ∃(w::real) > 0.(N > 0) ∧
    (∀x ∈ {a..b}. ¦(Σk∈{2..N+1}.
    (f(xs N!k) - f(xs N!(k - 1))) * σ(w * (x - xsN!k)))
    + f(a) * σ(w * (x - xs N!0)) - f x¦ < ε)"
```

Listing 11. Uniform Approximation Theorem

The approximant

$$G_{N,f}(x) = f(a)\,\sigma\big(w(x - x_0)\big) + \sum_{k=2}^{N+1} \big(f(x_k) - f(x_{k-1})\big)\,\sigma\big(w(x - x_k)\big).$$

is a one-hidden-layer network where all units share the same slope w and only their shifts x_k differ. The output weights are just forward differences of f on a uniform grid. Two error terms arise in the proof, I_1 and I_2; I_1 measures how well each sigmoid acts like a step function away from its center (controlled by making w large) and I_2 measures how much the piecewise-constant "finite-difference" reconstruction deviates from f locally (controlled by making the mesh h small and hence N large). More precisely, the proof introduces a local surrogate of $G_{N,f}$, namely L_i, that pretends distant sigmoids are already saturated (exactly 1 on the left, 0 on the right), but keeps the two boundary sigmoids at x_i and x_{i+1} in their nonsaturated true form. Concretely, in our mechanization we fix $\varepsilon > 0$ then we define $\eta = \frac{\varepsilon}{\left(\mathrm{Sup}_{x\in[a,b]}|f(x)|\right)+2\left(\mathrm{Sup}_{x\in\mathbb{R}}|\sigma(x)|\right)+2}$, and we obtain δ so that whenever $|\,x-y\,| < \delta$ we have $|\,f(x)-f(y)\,| < \eta$. Finally, we require at least N neurons with $N = \left\lfloor \max\left\{3,\ \frac{2(b-a)}{\delta},\ \frac{1}{\eta}\right\}\right\rfloor + 1$. With this in place we let $\{x_0, x_1, \ldots, x_{N+1}\}$ be the uniform partition of the interval $[a - \frac{b-a}{N}, b]$ and note that the distance between adjacent x_k is $|\,x_k - x_{k-1}\,| = \frac{b-a}{N} < \delta/2$ from our choice of N. Let us consider an arbitrary $x \in [a, b]$, we define $i = i(x) = \max\{i \in \{1, \ldots, N\} : x_i \le x\}$ so that $x \in [x_i, x_{i+1}]$ then for $i \ge 3$:

$$L_i(x) = \underbrace{f(a) + \sum_{k=2}^{i-1} \big(f(x_k) - f(x_{k-1})\big)}_{\text{“all steps left of } x \text{ are on”}} + \underbrace{\big(f(x_i) - f(x_{i-1})\big)\,\sigma\big(w(x - x_i)\big)}_{\text{left boundary still transitioning}}$$

$$+ \underbrace{\big(f(x_{i+1}) - f(x_i)\big)\,\sigma\big(w(x - x_{i+1})\big)}_{\text{right boundary still transitioning}}.$$

Now we measure the error of our approximant against the objective function f and split this into two terms:

$$\mid G_{N,f}(x) - f(x) \mid \leq \underbrace{\mid G_{N,f}(x) - L_i(x) \mid}_{I_1} + \underbrace{\mid L_i(x) - f(x) \mid}_{I_2}.$$

We will make it explicit that I_1 represents the cumulative error from all the “far” sigmoids—that is, those which should already be saturated to 0 or 1. After a careful rearrangement of $G_{N,f}(x) - L_i(x)$, we see that:

$$\begin{aligned} I_1(i,x) &= \Big| f(a)\big(\sigma(w(x - x_0)) - 1\big) + \sum_{k=2}^{i-1} \big(f(x_k) - f(x_{k-1})\big)\big(\sigma(w(x - x_k)) - 1\big) \\ &\quad + \sum_{k=i+1}^{N+1} \big(f(x_k) - f(x_{k-1})\big)\,\sigma\big(w(x - x_k)\big)\Big| \\ &\overset{\triangle}{\leq} \mid f(a) \mid \big|\sigma\big(w(x - x_0)\big) - 1\big| + \sum_{k=2}^{i-1} \big|f(x_k) - f(x_{k-1})\big|\,\big|\sigma\big(w(x - x_k)\big) - 1\big| \\ &\quad + \sum_{k=i+1}^{N+1} \big|f(x_k) - f(x_{k-1})\big|\,\big|\sigma\big(w(x - x_k)\big)\big|. \quad \dagger \end{aligned}$$

The next step is to bound each term using the uniform-continuity and sigmoidal estimates. Specifically, we aim to establish

$$I_1(i,x) < \mid f(a) \mid\mid \sigma(w(x - x_0)) - 1 \mid + \sum_{k=2}^{i-1} \eta\,\tfrac{1}{N} + \sum_{k=i+2}^{N+1} \eta\,\tfrac{1}{N}.$$

From here, the formalization compels us to analyze a number of side cases. First, we consider the case when $i \geq 3$. If $\sigma(w(x - x_k)) = 0$ for every $k \in \{2, \ldots, i-1\}$ then † above greatly simplifies. We further consider the case by when $i = N$. If $i = N$ then the result follows immediately; otherwise, we next consider when the second summation's summands of † are all identically 0. In the edge case when all the summands are zero, we get the strict inequality for free; otherwise, we collect the terms where the summand is nonzero and use the fact that $\mid f(x_k) - f(x_{k-1}) \mid < \eta$ by construction of the mesh of points. Moreover, when $k < i$ we have that $\mid \sigma(w(x - x_k)) - 1 \mid < \frac{1}{N}$ but when $k > i$ we get that $\mid \sigma(w(x - x_k)) \mid < \frac{1}{N}$. Thus, we can see that when we approximate the neural layer G with terms that are mostly in their limit, we obtain a very good estimate. We further note that checking all of these edge cases is not merely an exercise in logic; with $\sigma(x)$ equal to a Heaviside function, it is easy to see that one of the summands could be entirely composed of zeros. Observe that we required $N > 3$ as this made it possible for there to exist an i such that $\{2, \ldots, i-1\} \neq \emptyset$ and $\{i+2, \ldots, N+1\} \neq \emptyset$ whenever $i \geq 3$ in the main case so that the summations are nonempty.

When $i \leq 2$, the argument essentially repeats itself, but we use a different definition for L_i. Finally, since $\{2, \ldots, i-1, i+2, \ldots, N+1\} \subset \{2, \ldots, N+1\}$ we get that the last inequality simplifies to

$$\begin{aligned} I_1(i,x) &< |\, f(a)\,| \cdot \left|\sigma\big(w(x-x_0)\big) - 1\right| + \eta \\ &\leq |\, f(a)\,| \cdot \frac{1}{N} + \eta \\ &< (1 + |\, f(a)\,|)\,\eta \\ &\leq \Big(1 + \sup_{t\in[a,b]} |\, f(t)\,|\Big)\,\eta, \end{aligned}$$

Next, we need to handle the bounding of I_2:

$$\begin{aligned} I_2(i,x) &= \Big| \sum_{k=2}^{i-1} \big(f(x_k) - f(x_{k-1})\big) + f(a) + \big(f(x_i) - f(x_{i-1})\big)\,\sigma\big(w(x-x_i)\big) \\ &\qquad + \big(f(x_{i+1}) - f(x_i)\big)\,\sigma\big(w(x-x_{i+1})\big) - f(x)\Big|. \\ &= \Big| f(x_{i-1}) - f(x) + \big(f(x_i) - f(x_{i-1})\big)\,\sigma\big(w(x-x_i)\big) + \big(f(x_{i+1}) - f(x_i)\big)\,\sigma\big(w(x-x_{i+1})\big)\Big|. \\ &\leq |\, f(x_{i-1}) - f(x)\,| + \big|f(x_i) - f(x_{i-1})\big|\,\big|\sigma\big(w(x-x_i)\big)\big| + \big|f(x_{i+1}) - f(x_i)\big|\,\big|\sigma\big(w(x-x_{i+1})\big)\big|. \\ &< \big(1 + |\,\sigma\big(w(x-x_i)\big)\,| + |\,\sigma\big(w(x-x_{i+1})\big)\,|\big)\,\eta. \\ &\leq \big(2\,\mathrm{Sup}_{x\in\mathbb{R}}\,|\,\sigma(x)\,| + 1\big)\,\eta. \end{aligned}$$

Thus, our total error is given by

$$\begin{aligned} |\, G_{N,f}(x) - f(x)\,| &\leq \Big(1 + \sup_{t\in[a,b]} |\, f(t)\,|\Big)\,\eta + \big(2\,\mathrm{Sup}_{x\in\mathbb{R}}\,|\,\sigma(x)\,| + 1\big)\,\eta \\ &= \Big(\sup_{t\in[a,b]} |\, f(t)\,| + 2\,\mathrm{Sup}_{x\in\mathbb{R}}\,|\,\sigma(x)\,| + 2\Big)\,\eta. \\ &< \varepsilon. \end{aligned}$$

The Isabelle proof of the UAT is around 1,350 lines of code and approximately two-thirds of it is devoted to proving the bounds on I_1 and I_2 in the various previously mentioned cases. This is done, in particular, because `sum_strict_mono` was used to establish the inequality specifically for the terms when $\sigma(w(x-x_k)) \neq 0$ but establishing the sets where this is the case lead to the proof being longer than anticipated.

It is often casually stated that "the universal approximation theorem is *merely* an existential claim, it does not tell you how many neurons are necessary to estimate your function". While this is true, it is somewhat misleading. Cybenko's classic, nonconstructive proof [7] gave no bound on the number of neurons necessary; however, Costarelli's constructive proof shows us how many neurons are sufficient, provided certain terms can be determined. In general, it might be possible to use fewer neurons than implied by the constructive proof, but the proof essentially gives us an recipe for a sufficient number of neurons. Pick conservative bounds $M_f \geq \sup_{t\in[a,b]} |\, f(t)\,|$ and $M_\sigma \geq \sup_{x\in\mathbb{R}} |\,\sigma(x)\,|$, for example $M_\sigma = 1$ for sigmoid, and let δ be such that $|x-y| < \delta$ implies $|f(x) - f(y)| < \eta$ on $[a,b]$, if f is L-Lipschitz then $\delta = \eta/L$ works. Finally, pick suffiicently large w, if σ is the sigmoid then $w = \frac{\ln(N-1)}{h}$ suffices. With these choices the definition given for N above will suffice for the number of neurons such that the universal approximation theorem is satisfied for f.

6 CONCLUSION

We addressed several fundamental questions: How can the sigmoid function and its higher-order derivatives be formally defined and analyzed? How does the UAT ensure that neural networks with sigmoidal activation functions can approximate any continuous function on a compact interval? How can these results be mechanized in Isabelle/HOL to provide a fully verified, constructive proof of function approximation? In answering these questions, we uncovered gaps in Isabelle's formal proof libraries, particularly in limit reasoning. This formality led us to introduce alternative limit formulations.

Isabelle proved particularly insightful in this endeavor, as it enforced a mathematical rigor that traditional proofs often lack. It eliminated informal reasoning, ensured the correctness of complex derivations (such as the n^{th} derivative of the sigmoid function using Stirling numbers), and provided a constructive proof of the UAT, making it directly applicable to verified AI. Furthermore, our work enhanced Isabelle's real analysis toolkit, improving its usability for future machine learning and mathematical analysis formalizations.

This paper demonstrates the power of theorem proving in AI verification, reinforcing the trustworthiness of neural network models. By bridging formal methods, real analysis, and deep learning, our work lays the groundwork for future research in verified machine learning and trustworthy AI systems.

REFERENCES

[1] Alexander Bagnall and Gordon Stewart. Certifying the true error: Machine learning in Coq with verified generalization guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2662–2669, 2019. doi:10.1609/aaai.v33i01.33012662.

[2] Dustin Bryant, Jim Woodcock, and Simon Foster. The sigmoid function and the universal approximation theorem. *Archive of Formal Proofs*, May 2025. Formal proof development, https://isa-afp.org/entries/Sigmoid_Universal_Approximation.html.

[3] Dustin Bryant, Jonathan Julián Huerta y Munive, and Simon Foster. Verifying numerical methods with Isabelle/HOL. 2025.

[4] Amine Chaieb, Florian Haftmann, Lukas Bulwahn, and Manuel Eberl. Stirling.thy. Isabelle/HOL distribution, theory HOL/HOL-Combinatorics/Stirling, 2025. Isabelle2025 release. URL: https://isabelle.in.tum.de/library/HOL/HOL-Combinatorics/Stirling.html.

[5] Tianping Chen, Hong Chen, and Ruey wen Liu. A constructive proof and an extension of Cybenko's approximation theorem. In Connie Page and Raoul LePage, editors, *Computing Science and Statistics*, pages 163–168, New York, NY, 1992. Springer New York. doi:10.1007/978-1-4612-2856-1_21.

[6] Danilo Costarelli and Renato Spigler. Constructive approximation by superposition of sigmoidal functions. *Analysis in Theory and Applications*, 29(2):169–196, 2013. doi:10.4208/ata.2013.v29.n2.8.

[7] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control Signal Systems 2, 303–314*, https://doi.org/10.1007/BF02551274, 1989.

[8] Leonid Datta. A survey on activation functions and their relation with Xavier and He normal initialization. *CoRR*, abs/2004.06632, 2020. arXiv preprint. URL: https://arxiv.org/abs/2004.06632.

[9] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, 2022. doi:10.1016/j.neucom.2022.06.111.

[10] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192, 1989. doi:10.1016/0893-6080(89)90003-8.

[11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, volume 15 of *JMLR Proceedings*, pages 315–323, 2011. URL: http://proceedings.mlr.press/v15/glorot11a.html.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: https://www.deeplearningbook.org/.

[13] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. doi:10.1016/0893-6080(91)90009-T.

[14] Kurt Hornik, Maxwell B. Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. doi:10.1016/0893-6080(89)90020-8.

[15] Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Computer Aided Verification (CAV 2017)*, volume 10426 of *Lecture Notes in Computer Science*, pages 97–117. Springer, 2017. doi:10.1007/978-3-319-63387-9_5.

[16] Guy Katz, Derek A. Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljic, David L. Dill, Mykel J. Kochenderfer, and Clark Barrett. The Marabou framework for verification and analysis of deep neural networks. In

Computer Aided Verification (CAV 2019), pages 443–452. Springer, 2019. doi:10.1007/978-3-030-25540-4_26.

[17] Vladimír Kunc and Jiří Kléma. Three decades of activations: A comprehensive survey of 400 activation functions for neural networks. *CoRR*, abs/2402.09092, 2024. doi:10.48550/arXiv.2402.09092.

[18] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a non-polynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993. doi:10.1016/S0893-6080(05)80131-8.

[19] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML 2013 Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013. Introduces Leaky ReLU variant. URL: https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf.

[20] Ali A. Minai and Ronald D. Williams. On the derivatives of the sigmoid. *Neural Networks*, 6(6):845–853, 1993. URL: https://www.sciencedirect.com/science/article/pii/S0893608005801297, doi:10.1016/S0893-6080(05)80129-7.

[21] Kevin P. Murphy. *Probabilistic Machine Learning: An Introduction*. MIT Press, 2022. URL: https://probml.github.io/pml-book/book1.html.

[22] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, pages 807–814. Omnipress, 2010. URL: https://icml.cc/Conferences/2010/papers/432.pdf.

[23] Tobias Nipkow, Markus Wenzel, and Lawrence C. Paulson. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 2002. doi:10.1007/3-540-45949-9.

[24] Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195, 1999. doi:10.1017/S0962492900002919.

[25] Joseph Tassarotti, Koundinya Vajjha, Anindya Banerjee, and Jean-Baptiste Tristan. A formal proof of PAC learnability for decision stumps. In *Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2021)*, 2021. Formalization in Lean; arXiv:1911.00385. doi:10.1145/3437992.3439917.

[26] Koundinya Vajjha, Joseph Tassarotti, Anindya Banerjee, and Jean-Baptiste Tristan. CertRL: Formalizing convergence proofs for value and policy iteration. In *Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2021)*, 2021. doi:10.1145/3437992.3439927.

[27] Haoze Wu, Omri Isac, Aleksandar Zeljić, Teruhiro Tagomori, Matthew Daggitt, Wen Kokke, Idan Refaeli, Guy Amir, Kyle Julian, Shahaf Bassan, Pei Huang, Ori Lahav, Min Wu, Min Zhang, Ekaterina Komendantskaya, Guy Katz, and Clark Barrett. Marabou 2.0: A versatile formal analyzer of neural networks. In *Proceedings of the 36th International Conference on Computer Aided Verification (CAV 2024)*, volume 14682 of *Lecture Notes in Computer Science*, pages 249–264, Montreal, QC, Canada, July 2024. Springer. doi:10.1007/978-3-031-65630-9_13.

[28] Haoze Wu, Guy Katz, Clark Barrett, and Mykel J. Kochenderfer. Marabou: A framework for verification and analysis of deep neural networks. In *Proceedings of the 22nd International Symposium on Formal Methods (FM 2020)*, volume 12232 of *Lecture Notes in Computer Science*, pages 443–451, Porto, Portugal, 2020. Springer. doi:10.1007/978-3-030-61045-3_25.

APPENDIX I: A WORKED EXAMPLE BOUNDING THE NEURONS NEEDED

Now, let us consider the following real example. On $[0, 1]$, let

$$f(x) \;=\; |x - 0.3| \;+\; 0.3\sin(6\pi x) \;+\; 0.2\,x\,(1 - x)$$

and we wish to model this with the sigmoid function. We have $M_f \le 0.7 + 0.3 + 0.05 = 1.05$ and $M_\sigma = 1$. For $\varepsilon = 10^{-2}$,

$$\eta = \frac{0.01}{(1 + 1.05) + (2 \cdot 1 + 1)} = \frac{0.01}{5.05} \approx 1.98 \times 10^{-3}.$$

A Lipschitz bound is $L := 1 + 0.3 \cdot 6\pi + 0.2 \approx 6.855$, hence $\delta = \eta/L \approx 2.89 \times 10^{-4}$. Choose

$$N = \left\lfloor \max\{\,3,\; 2/\delta,\; 1/\eta\,\} \right\rfloor + 1 = \left\lfloor \max\{3,\; 2/(2.89 \times 10^{-4}),\; 505.05\} \right\rfloor + 1 = 6,925$$

so $h = 1/N \approx 1.444 \times 10^{-4}$ and $1/N \le \eta$. Finally,

$$w \;\ge\; \frac{\ln(N-1)}{h} \;=\; \frac{\ln 6924}{1/6925} \;\approx\; 61,237.$$

With these choices, $\sup_{x\in[0,1]} \mid G_{N,f}(x) - f(x) \mid < 10^{-2}$.

APPENDIX II: PROVING THE RECIPE BOUND FOR N

We use the sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$. and want $|\sigma(wt) - 1| \le \frac{1}{N}$ for all $t \ge h$. Since $\sigma(wt) \le 1$, this is equivalent to $1 - \sigma(wt) \le \frac{1}{N}$.

For any $t \geq h$,

$$\begin{aligned} 1 - \sigma(wt) &= 1 - \frac{1}{1 + e^{-wt}} \\ &= \frac{1 + e^{-wt}}{1 + e^{-wt}} - \frac{1}{1 + e^{-wt}} \\ &= \frac{e^{-wt}}{1 + e^{-wt}} \\ &= \frac{1}{1 + e^{wt}}. \end{aligned}$$

Thus the condition $1 - \sigma(wt) \leq 1/N$ is

$$\frac{1}{1 + e^{wt}} \leq \frac{1}{N}.$$

Because $N > 0$ and $e^{wt} > 0$, this is equivalent to

$$\begin{aligned} N &\leq 1 + e^{wt}, \\ N - 1 &\leq e^{wt}, \\ \ln(N - 1) &\leq wt, \\ w &\geq \frac{1}{t} \ln(N - 1). \end{aligned}$$

So for fixed N,

$$1 - \sigma(wt) \leq \frac{1}{N} \quad \Longleftrightarrow \quad w \geq \frac{1}{t} \ln(N - 1).$$

To make this hold for all $t \geq h$, we require

$$w \geq \sup_{t \geq h} \frac{1}{t} \ln(N - 1).$$

Since $\frac{1}{t} \ln(N - 1)$ is decreasing in $t > 0$, the supremum over $t \geq h$ is attained at $t = h$, hence

$$\sup_{t \geq h} \frac{1}{t} \ln(N - 1) = \frac{1}{h} \ln(N - 1).$$

Therefore it suffices to choose

$$w = \frac{1}{h} \ln(N - 1).$$

Now we use the Lipschitz continuity of f to obtain an explicit modulus of continuity. Suppose that f is L-Lipschitz on $[a, b]$, i.e.

$$| f(x) - f(y) | \leq L \mid x - y \mid \qquad \text{for all } x, y \in [a, b].$$

Given any $\eta > 0$, we seek $\delta > 0$ such that

$$| x - y | < \delta \implies | f(x) - f(y) | < \eta \qquad \text{for all } x, y \in [a, b].$$

Using the Lipschitz condition, whenever $| x - y | < \delta$ we have

$$| f(x) - f(y) | \leq L \mid x - y | < L\delta.$$

Thus it suffices to choose δ so that $L\delta \leq \eta$, and one convenient choice is

$$\delta = \frac{\eta}{L}.$$

With this choice, for all $x, y \in [a, b]$ satisfying $\mid x - y \mid < \delta$ we obtain

$$\mid f(x) - f(y) \mid \leq L \mid x - y \mid < L\left(\frac{\eta}{L}\right) = \eta,$$

which is exactly the uniform continuity condition required in the construction.