Exact Upper and Lower Bounds for the Output Distribution of Neural Networks with Random Inputs

Andrey Kofnov[1] **Daniel Kapla**[1]

Ezio Bartocci[2] **Efstathia Bura**[1]

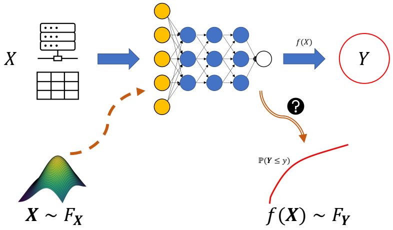
[1]Faculty of Mathematics and Geoinformation, TU Wien, Vienna, Austria

[2]Faculty of Informatics, TU Wien, Vienna, Austria

Vancouver, ICML 2025

Neural Networks under Uncertainty

- Neural networks are typically *deterministic*
- Inputs in real-world are *noisy/uncertain*
- Why characterize output distribution?
  - Risk quantification
  - Robustness
  - Explainability



★ Our Contributions

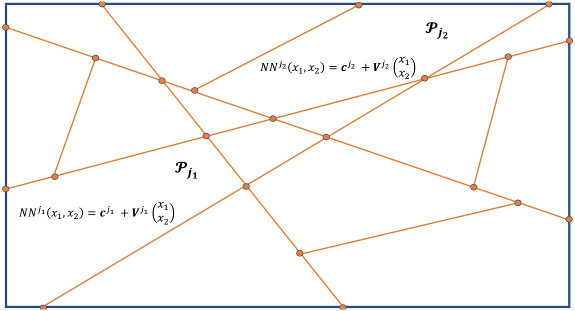
- Exact cdf computation for
  ReLU NNs + piecewise polynomial inputs
- Bounds for general feedforward NNs via ReLU approximation
- New Universal Distribution Approximation Theorem (UDAT):
  Constructive proof

ReLU Split

$\widetilde{\mathbf{Y}} : K \to \mathbb{R}^{n_L}$ is a ReLU neural network with

- $K = \bigcup_{j=1}^{q_Y} \mathcal{P}_j$ - input domain is represented as a union of polytopes,
- $\widetilde{\mathbf{Y}}\big|_{\mathcal{P}_j} = NN^j : \mathcal{P}_j \to \mathbb{R}^{n_L}$ - is an affine transformation

We utilize a GPU-accelerated algorithm from [Berzins, 2023][1].

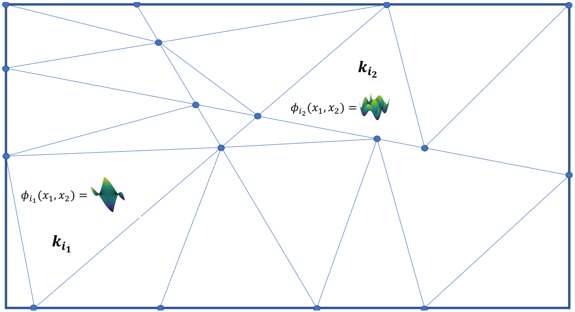


[1] Arturs Berzins. Polyhedral complex extraction from ReLU networks using edge subdivision. In Proceedings of the 40th International Conference on Machine Learning, 2023.

Piecewise polynomial pdf

$\phi : K \to \mathbb{R}$ is a *piecewise polynomial*[2] if

- $K = \bigcup_{i=1}^{q} k_i$ - input domain is represented as a union of simplices,
- $\phi\big|_{k_i^o} : k_i^o \to \mathbb{R}$ - is a polynomial

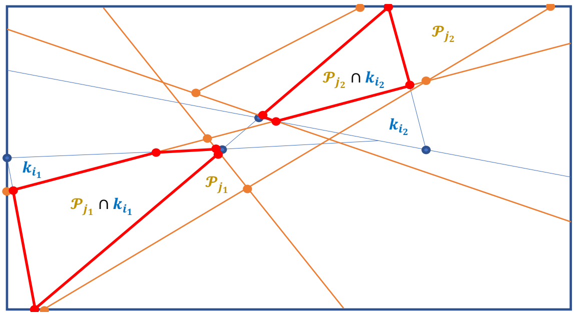


[2]By the Stone–Weierstrass Theorem, any continuous function on a compact hyperrectangle can be approximated arbitrarily well by polynomials

Intersection of ReLU-based polytopes with pdf-based simplices

For every ReLU-based polytope $\mathcal{P}_j$ and every pdf-based simplex k_i we compute an intersection $\mathcal{P}_j \cap k_i$ to define an area where

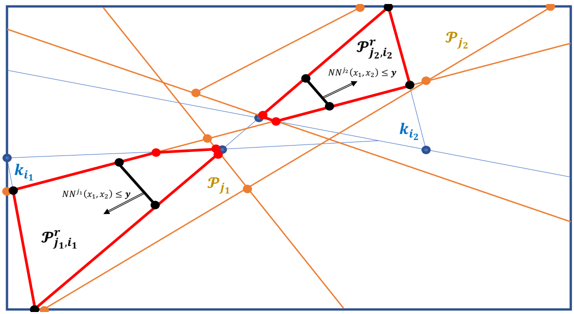
- the Neural Network $\widetilde{Y}$ behaves as an affine transformation
- the input density $\phi(x)$ is a polynomial



What is the reduced polytope?

$$\mathbb{P}(\widetilde{\mathbf{Y}}\big|_{\mathcal{P}_j} \leq \mathbf{y}) = \int\limits_{\{\mathbf{x}\in\mathcal{P}_j\}\cap\{NN^j(\mathbf{x})\leq\mathbf{y}\}} \phi(\mathbf{x})d\mathbf{x} = \sum_i \int\limits_{\mathcal{P}^r_{j,i}} \phi_i(\mathbf{x})d\mathbf{x},$$

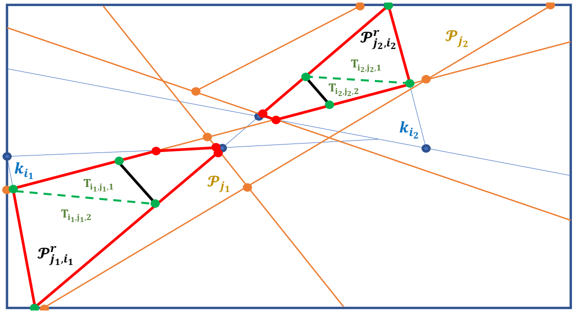
where $\mathcal{P}^r_{j,i} = \mathcal{P}_j \cap k_i \cap \{NN^j(\mathbf{x}) \leq \mathbf{y}\}$ – *reduced polytope*



Delaunay triangulation leads to the union of simplices

Every convex polytope $\mathcal{P}^r_{j,i}$ can be triangulated and represented as a finite union of disjoint simplices

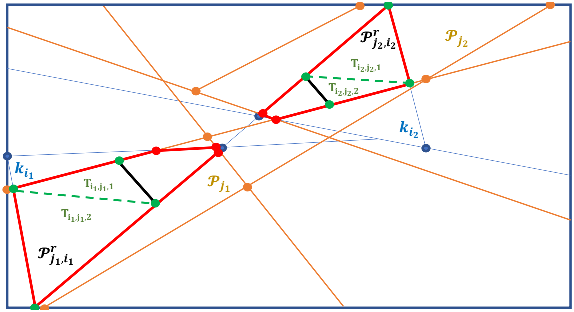
$$\mathcal{P}^r_{j,i} = \bigcup_{s=1}^{S_{i,j}} \mathcal{T}_{i,j,s}$$



Exact CDF: ReLU + Piecewise Polynomial Input

- Compute integrals over the triangulation of reduced simplices
  - Exact integration is made possible by the approach outlined in [Lasserre, 2021][3]
- Produce exact CDF:

$$F_{\widetilde{\mathbf{Y}}}(\mathbf{y}) = \mathbb{P}(\widetilde{\mathbf{Y}} \leq \mathbf{y}) = \sum_j \mathbb{P}(\widetilde{\mathbf{Y}}\big|_{\mathcal{P}_j} \leq \mathbf{y}) = \sum_{i,j,s} \int_{T_{i,j,s}} \phi_i(\mathbf{x})dx$$



[3] Jean B. Lasserre. Simple formula for integration of polynomials on a simplex. BIT Numerical Mathematics, 61(2):523–533, 2021.

Bounding Output of General Networks

- Approximate continuous monotonic *piecewise twice continuously differentiable* activations (e.g., tanh) using piecewise linear bounds.
- Construct upper and lower ReLU networks by propagating piecewise linear bounds through each node at every layer.:

$$\underline{f}_n(x) \leq f(x) \leq \bar{f}_n(x)$$

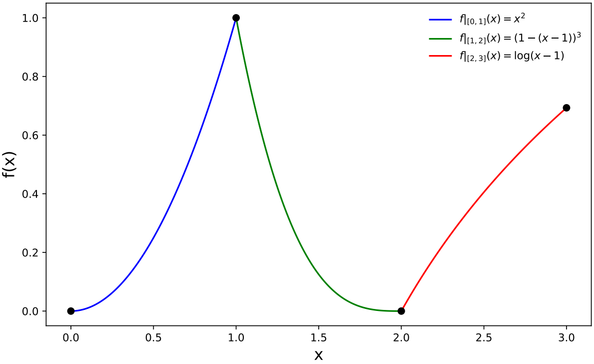
- Converges uniformly and monotonically to true NN.

Piecewise twice continuously differentiable activations

Let $f : [\underline{a}, \overline{a}] \to \mathbb{R}$ be a continuous function, where

$$[\underline{a}, \overline{a}] = \bigcup_{i=1}^{n} [a_i, a_{i+1}], \quad \underline{a} = a_1 < a_2 < \cdots < a_{n+1} = \overline{a},$$

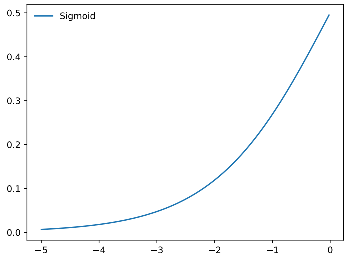
and assume that $f|_{[a_i, a_{i+1}]} \in \mathcal{C}^2([a_i, a_{i+1}]) \quad$ for each $i = 1, \ldots, n$.



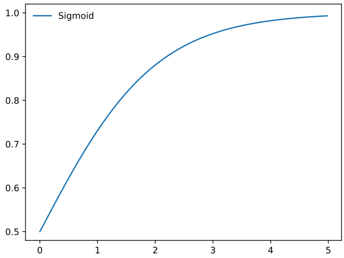
Piecewise Twice Continuously Differentiable Function on $[0, 3]$

Linear interpolation

- Works for the upper approximation on a convex segment and the lower approximation on a concave segment



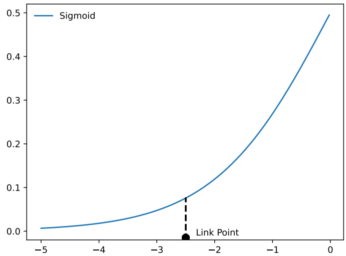
Convex segment



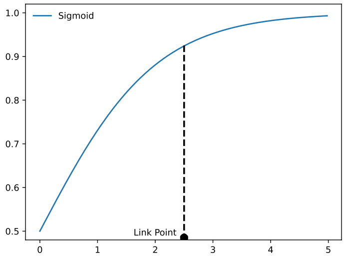
Concave segment

Linear interpolation

- Works for the upper approximation on a convex segment and the lower approximation on a concave segment
- Choose a linking point in the middle of the interval, i.e., $a_{k'} = (a_k + a_{k+1})/2$



Convex segment



Concave segment

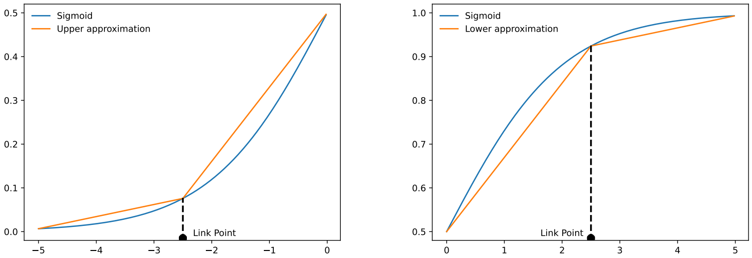
Linear interpolation

- Perform linear interpolation using the linking point:

$$\kappa_1 = \frac{f(a_{k'}) - f(a_k)}{a_{k'} - a_k}, \qquad \kappa_2 = \frac{f(a_{k+1}) - f(a_{k'})}{a_{k+1} - a_{k'}}$$

$$\widetilde{f}(\tau) = f(a_k) + (\tau - a_k)\kappa_1, \qquad \tau \in [a_k, a_{k'}]$$

$$\widetilde{f}(\tau) = f(a_{k'}) + (\tau - a_{k'})\kappa_2, \quad \tau \in [a_{k'}, a_{k+1}]$$

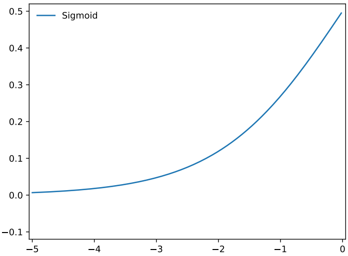


Convex segment

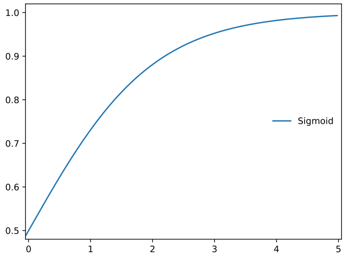
Concave segment

Piecewise tangent

- Works for the upper approximation on a concave segment and the lower approximation on a convex segment



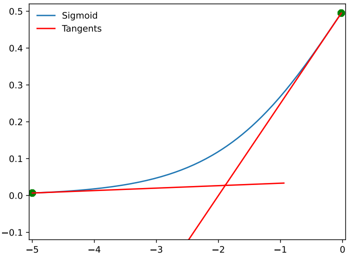
Convex segment



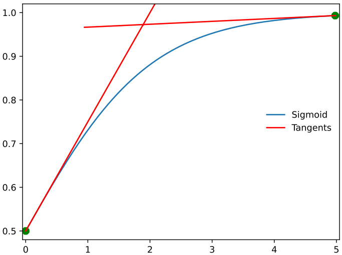
Concave segment

Piecewise tangent

- Works for the upper approximation on a concave segment and the lower approximation on a convex segment
- Compute the tangents of the function at the segment boundaries



Convex segment



Concave segment

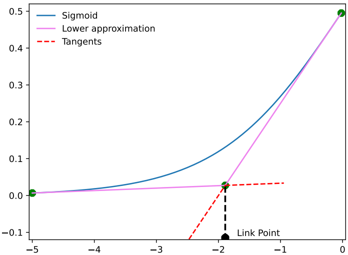
Piecewise tangent

- Define the approximation as a piecewise tangent function, with the linking point at the intersection

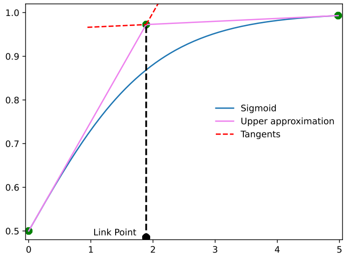
$$a_{k'} = \frac{f(a_k) - f(a_{k+1}) - (f'_+(a_k)a_k - f'_-(a_{k+1})a_{k+1})}{f'_-(a_{k+1}) - f'_+(a_k)}$$

$$\widetilde{f}(\tau) = f(a_k) + f'_+(a_k)(\tau - a_k), \qquad \tau \in [a_k, a_{k'}]$$

$$\widetilde{f}(\tau) = f(a_{k+1}) + f'_-(a_{k+1})(\tau - a_{k+1}), \ \tau \in [a_{k'}, a_{k+1}],$$



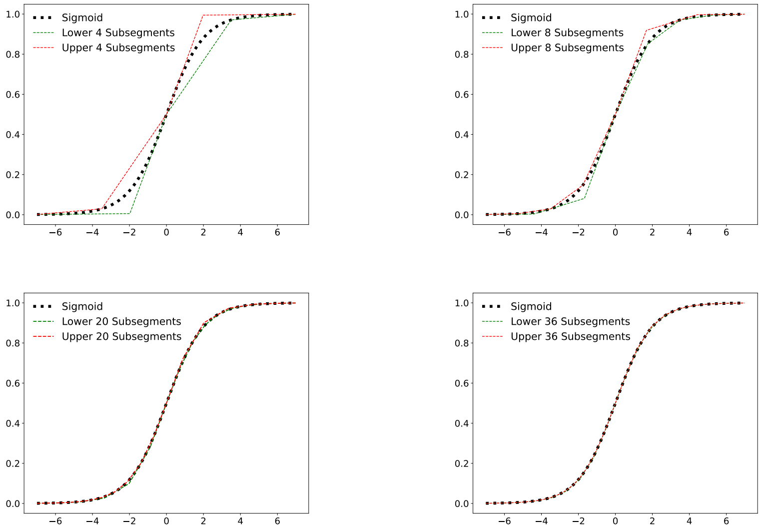
Convex segment



Concave segment

Combination of all segments

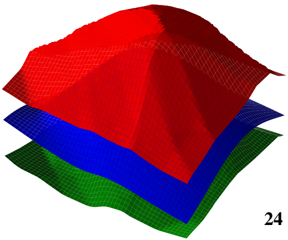
- Refining the grid results in both bounds converging monotonically to the true function within the domain.



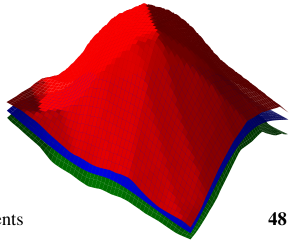
Propagation of bounds

- Propagation of bounds through the entire network generates accurate bounds on the network output.

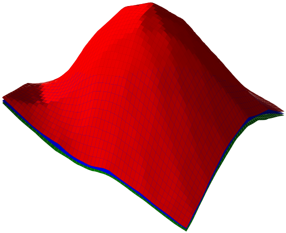
4 subsegments



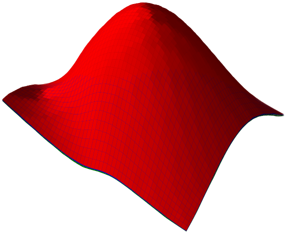
12 subsegments



24 subsegments

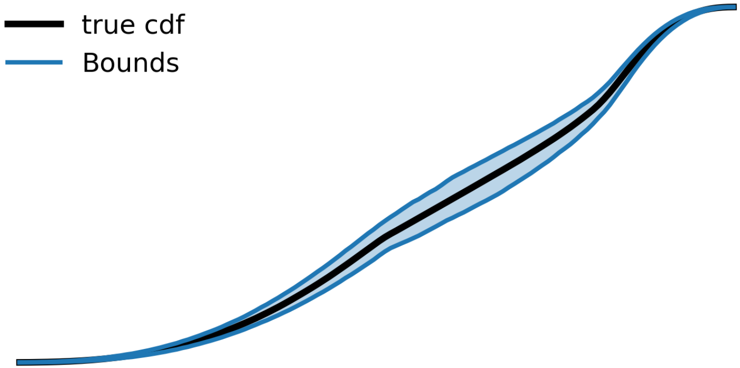


48 subsegments



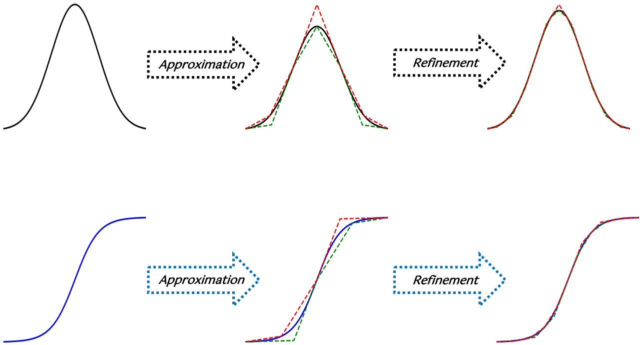
Universal Distribution Approximation Theorem (UDAT)

- The theorem states that the cdf of a continuous function of a random vector can be closely approximated from above and below using ReLU neural networks.



Practical Use of the UDAT

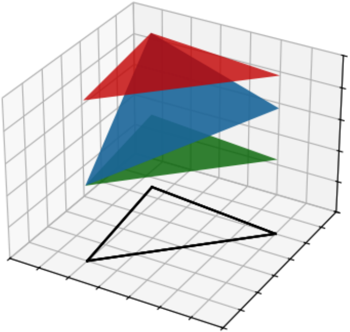
- According to the Universal Approximation Theorem [Hornik et al., 1989][4], any continuous function on a compact domain can be approximated (and bounded) using a ReLU network.
- This applies to any continuous pdf and to any neural network with continuous activation functions



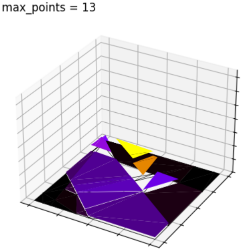
[4] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural Networks, 2(5):359–366, 1989.

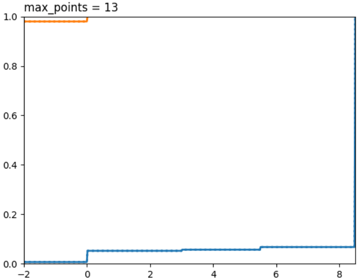
Local Bounds on the input pdf

- The pdf is bounded using ReLU neural networks
- There exists a collection of simplices with local affine bounds
- On a simplex, every affine transformation can be bounded
- This results in piecewise polynomial bounds of the input pdf

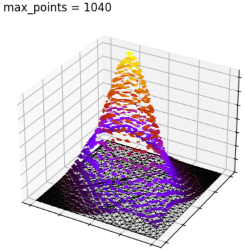


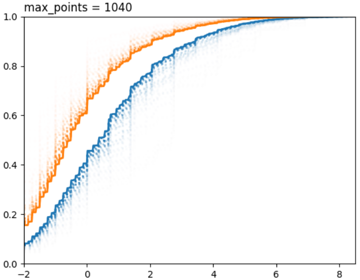
Calculation of upper and lower bounds



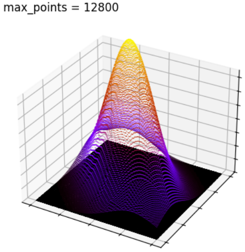


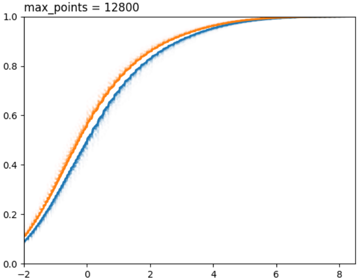
Calculation of upper and lower bounds





Calculation of upper and lower bounds





Thank you!

References I

Arturs Berzins. Polyhedral complex extraction from ReLU networks using edge subdivision. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2234–2244. PMLR, 23–29 Jul 2023.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Jean B. Lasserre. Simple formula for integration of polynomials on a simplex. *BIT Numerical Mathematics*, 61(2):523–533, 2021.