Lower Bounds on Neural Network Depth via Lattice Polytopes

Christian Haase  Christoph Hertrich  Georg Loho

Bielefeld, 6 September 2022
Neural Networks in Action

Neural Networks in Action


Neural Networks in Action


Screenshot deep1.com (Feb 18, 2022)
A Single ReLU Neuron

Outputs of previous neurons

\[ x_1 \]
\[ x_2 \]
\[ x_\ell \]

\[ w_1 \]
\[ w_2 \]
\[ w_\ell \]

\[ \text{max}\{0, \sum_{i=1}^{\ell} w_i x_i\} \]
A Single ReLU Neuron

Outputs of previous neurons

\[ x_1 \]
\[ x_2 \]
\[ x_\ell \]

\[ w_1 \]
\[ w_2 \]
\[ w_\ell \]

max\{0, \sum_{i=1}^{\ell} w_i x_i\}

Rectified linear unit (ReLU): \( \text{relu}(x) = \max\{0, x\} \)
A Single ReLU Neuron
ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons

The diagram shows a ReLU feedforward neural network with two input nodes $x_1$ and $x_2$, and three output nodes $y_1$, $y_2$, and $y_3$. The network computes a function $T_k \circ \text{relu} \circ T_{k-1} \circ \cdots \circ T_2 \circ \text{relu} \circ T_1$ with linear transformations $T_i$.

Example: depth 3 (2 hidden layers).

Usage: Learn weights of $T_i$ from given input-output pairs.
ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons

![Diagram of ReLU feedforward neural network]

- Computes function

\[ T_k \circ \text{relu} \circ T_{k-1} \circ \cdots \circ T_2 \circ \text{relu} \circ T_1 \]

with linear transformations \( T_i \).
ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons

\[ x_1 \rightarrow x_2 \rightarrow y_1 \]
\[ x_1 \rightarrow x_2 \rightarrow y_2 \]
\[ x_1 \rightarrow x_2 \rightarrow y_3 \]

\[ T_1 \quad T_2 \quad T_3 \]

- Computes function

\[ T_k \circ \text{relu} \circ T_{k-1} \circ \cdots \circ T_2 \circ \text{relu} \circ T_1 \]

with linear transformations \( T_i \).

- Example: depth 3 (2 hidden layers).
ReLU Feedforward Neural Networks

▶ Acyclic (layered) digraph of ReLU neurons

$\begin{align*}
x_1 & \rightarrow & & \rightarrow & & T_1 & & y_1 \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
x_2 & \rightarrow & T_2 & \rightarrow & T_3 & \rightarrow & y_2 \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
& & & \downarrow & & \downarrow & & y_3 \\
& & & & & & T_1 & & y_1 \\
& & & & & & T_2 & & y_2 \\
& & & & & & T_3 & & y_3
\end{align*}$

▶ Computes function

$T_k \circ \text{relu} \circ T_{k-1} \circ \cdots \circ T_2 \circ \text{relu} \circ T_1$

with linear transformations $T_i$.

▶ Example: depth 3 (2 hidden layers).

▶ Usage: Learn weights of $T_i$ from given input-output pairs.
What is the class of functions computable by ReLU Neural Networks with a certain depth?
Universal approximation theorems:
One hidden layer enough to **approximate** any continuous function.
Universal approximation theorems:
One hidden layer enough to **approximate** any continuous function.

What about **exact** representability?
Example: Computing the Maximum of Two Numbers

\[
\max\{x, y\} = \max\{x - y, 0\} + y
\]

Diagram: A diagram illustrating the computation of the maximum of two numbers, with nodes labeled 'x', 'y', and 'm', and edges labeled with the values 1, -1, and 1.
Example: Computing the Maximum of Two Numbers

$$\max\{x, y\} = \max\{x - y, 0\} + y$$
Example: Computing the Maximum of Four Numbers

Inductively: Max of $n$ numbers with $\lceil \log_2(n) \rceil$ hidden layers.
Example: Computing the Maximum of Four Numbers

\[ \begin{align*}
   x_1 & \\
   x_2 & \\
   x_3 & \\
   x_4 & \\
\end{align*} \]

\[ m \]

Inductively: Max of \( n \) numbers with \( \lceil \log_2(n) \rceil \) hidden layers.
Representing Arbitrary Piecewise Linear Functions

Observation

*Every function represented by a ReLU NN is continuous and piecewise linear (CPWL).*
Observation

Every function represented by a ReLU NN is continuous and piecewise linear (CPWL).

Theorem (Wang, Sun [WS05])

Any CPWL function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) can be written as

\[
f(x) = \sum_{i=1}^{p} \lambda_i \max\{a_{i,1}^T x, \ldots, a_{i,n+1}^T x\}.
\]
Observation

*Every function represented by a ReLU NN is continuous and piecewise linear (CPWL).*

**Theorem (Wang, Sun [WS05])**

Any CPWL function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ can be written as

$$f(x) = \sum_{i=1}^{p} \lambda_i \max\{a_{i,1}^T x, \ldots, a_{i,n+1}^T x\}.$$  

**Theorem (Arora, Basu, Mianjy, Mukherjee [ABMM18])**

Any CPWL function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ can be represented by a ReLU NN with $\lceil \log_2(n + 1) \rceil$ hidden layers.
Theorem (Arora, Basu, Mianjy, Mukherjee [ABMM18])

Any CPWL function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) can be represented by a ReLU NN with \( \lceil \log_2(n + 1) \rceil \) hidden layers.

▶ Is logarithmic depth best possible?
Conjecture

Yes, there are functions which need $\lceil \log_2(n + 1) \rceil$ hidden layers!
Conjecture

Yes, there are functions which need \( \lceil \log_2(n + 1) \rceil \) hidden layers!

Using [WS05], we show that this is equivalent to:

Conjecture

\[ \max \{0, x_1, \ldots, x_{2^k}\} \text{ cannot be represented with } k \text{ hidden layers.} \]
What lower bounds are known?
What lower bounds are known?

- Mukherjee, Basu (2017):
  \( \max\{0, x_1, x_2\} \) not representable with 1 hidden layer.
What lower bounds are known?

- Mukherjee, Basu (2017):
  \[ \max\{0, x_1, x_2\} \] not representable with 1 hidden layer.

That’s all!
What lower bounds are known?

- Mukherjee, Basu (2017):
  \[ \max\{0, x_1, x_2\} \] not representable with 1 hidden layer.

That’s all!

- No function known that provably needs more than 2 hidden layers \( \mapsto \) gap between 2 and \( \lceil \log_2(n + 1) \rceil \).
What lower bounds are known?

- Mukherjee, Basu (2017):
  \( \max\{0, x_1, x_2\} \) not representable with 1 hidden layer.

That’s all!

- No function known that provably needs more than 2 hidden layers \( \xrightarrow{\sim} \) gap between 2 and \( \lceil \log_2(n + 1) \rceil \).

- Smallest candidate: \( \max\{0, x_1, x_2, x_3, x_4\} \).
Partial Results

- Hertrich, Basu, Di Summa, Skutella (NeurIPS 2021):

  2 hidden layers not enough for \( \max\{0, x_1, x_2, x_3, x_4\} \)

  under an additional assumption on the network.
Partial Results

- Hertrich, Basu, Di Summa, Skutella (NeurIPS 2021):
  2 hidden layers not enough for \( \max\{0, x_1, x_2, x_3, x_4\} \) under an additional assumption on the network.

- Haase, Hertrich, Loho (this talk!):
  Conjecture is true for networks with only integer weights.
How do Polytopes Come into Play?

CPWL function = difference of two convex CPWL functions

= difference of two tropical polynomials

= tropical rational function

⇝

study Newton polytopes
How do Polytopes Come into Play?

CPWL function = difference of two convex CPWL functions
= difference of two tropical polynomials
= tropical rational function
How do Polytopes Come into Play?

CPWL function = difference of two convex CPWL functions
  = difference of two tropical polynomials
  = tropical rational function

\[ \Rightarrow \text{study Newton polytopes!} \]
Newton Polytope of a Convex CPWL Function

- \( f(x) = \max\{a_1^T x, \ldots, a_k^T x\} \implies P(f) = \text{conv}\{a_1, \ldots, a_k\} \)
- dual to underlying polyhedral complex of the CPWL function

Example for \( \max\{0, x_1, x_2\} \):
Newton Polytope of a Convex CPWL Function

\[ f(x) = \max \{ a_1^T x, \ldots, a_k^T x \} \quad \leadsto \quad P(f) = \text{conv} \{ a_1, \ldots, a_k \} \]

dual to underlying polyhedral complex of the CPWL function

Example for \( \max \{ 0, x_1, x_2 \} \):

Convex CPWL functions \( \sim \) Newton Polytopes

(positive) scalar multiplication \( \sim \) scaling
addition \( \sim \) Minkowski sum
taking maximum \( \sim \) taking convex hull of union
Newton Polytopes and Neural Networks

Are there polytopes $Q, R \in P_2$ with $Q + \Delta_4 = R$?
Newton Polytopes and Neural Networks

- Polytope $P_2$: Finite Minkowski sum of polytopes in $P_1$.
- Newton polytope of $\max \{0, x_1, x_2, x_3, x_4\}$: 4-dimensional simplex $\Delta^4$.
- Are there polytopes $Q, R \in P_2$ with $Q + \Delta^4 = R$?
Newton Polytopes and Neural Networks

Newton polytope of \( \max\{0, x_1, x_2, x_3, x_4\} \): 4-dim. simplex \( \Delta_4 \).

Are there polytopes \( Q, R \in \mathcal{P}_2 \) with 

\[ Q + \Delta_4 = R. \]
Newton Polytopes and Neural Networks

Let \( P_1 \) and \( P_2 \) be the convex hull of the union of two zonotopes. Then \( P_2 \) is the finite Minkowski sum of polytopes in \( P_1 \).

The Newton polytope of max \( \{0, x_1, x_2, x_3, x_4\} \) is a 4-dimensional simplex \( \Delta_4 \).

Are there polytopes \( Q, R \in P_2 \) such that \( Q + \Delta_4 = R \)?

Points and line segments in the diagram represent elements of the Newton polytope.
Newton Polytopes and Neural Networks

Newton polytope of max \{0, x_1, x_2, x_3, x_4\}: 4-dim. simplex \(\Delta_4\).

Are there polytopes \(Q, R \in P_2\) with \(Q + \Delta_4 = R\)?
Newton Polytopes and Neural Networks

\[ \mathcal{P}_2 = \{ P \text{ polytope} \mid P \text{ convex hull of union of two zonotopes} \} \]
Newton Polytopes and Neural Networks

\[ P_2' = \{ P \text{ polytope} \mid P \text{ convex hull of union of two zonotopes} \} \]

\[ P_2 = \{ P \text{ polytope} \mid P \text{ finite Minkowski sum of polytopes in } P_2' \} \]
Newton Polytopes and Neural Networks

\[ P' = \{ P \text{ polytope} \mid P \text{ convex hull of union of two zonotopes} \} \]

\[ P_2 = \{ P \text{ polytope} \mid P \text{ finite Minkowski sum of polytopes in } P'_2 \} \]

Newton polytope of \( \max\{0, x_1, x_2, x_3, x_4\} \): 4-dim. simplex \( \Delta^4 \).
Are there polytopes \( Q, R \in P_2 \) with \( Q + \Delta^4 = R \)?
Polytopal Reformulation of the Conjecture

\[ P_0 := \{\text{points}\} \]
\[ P_1 := \{\text{zonotopes}\} \]
\[ P_k := \left\{ \sum_{i=1}^{m} \text{conv}(P_i, Q_i) \bigg| P_i, Q_i \in P_{k-1}, m \in \mathbb{N} \right\} \]
\[ \Delta^n := \text{conv}\{0, e_1, e_2, \ldots, e_n\} \]
Polytopal Reformulation of the Conjecture

\[ P_0 := \{ \text{points} \} \]
\[ P_1 := \{ \text{zonotopes} \} \]
\[ P_k := \left\{ \sum_{i=1}^{m} \text{conv}(P_i, Q_i) \mid P_i, Q_i \in P_{k-1}, m \in \mathbb{N} \right\} \]
\[ \Delta^n := \text{conv}\{0, e_1, e_2, \ldots, e_n\} \]

**Conjecture**

*There is no pair of polytopes \( P, Q \in P_k \) such that \( P + \Delta^{2^k} = Q \).*
From now on: all weights integer!

Note: logarithmic upper bound only uses integer weights
From now on: all weights integer!

Note: logarithmic upper bound only uses integer weights

⇒ All Newton polytopes are lattice polytopes
From now on: all weights integer!

Note: logarithmic upper bound only uses integer weights

⇒ All Newton polytopes are lattice polytopes

Let $\mathcal{P}_k^\mathbb{Z} := \mathcal{P}_k \cap \{\text{lattice polytopes}\}$.
Results for the Integer Case
Haase, Hertrich, Loho (work in progress)

**From now on:** all weights integer!

**Note:** logarithmic upper bound only uses integer weights

⇒ All Newton polytopes are lattice polytopes

Let $\mathcal{P}^\mathbb{Z}_k := \mathcal{P}_k \cap \{\text{lattice polytopes}\}$.

**Theorem**

*There is no pair of polytopes $P, Q \in \mathcal{P}^\mathbb{Z}_k$ such that $P + \Delta^{2^k} = Q$.*
Results for the Integer Case
Haase, Hertrich, Loho (work in progress)

**From now on:** all weights integer!

**Note:** logarithmic upper bound only uses integer weights

⇒ All Newton polytopes are lattice polytopes

Let $\mathcal{P}_k^\mathbb{Z} := \mathcal{P}_k \cap \{\text{lattice polytopes}\}$.

**Theorem**

*There is no pair of polytopes $P, Q \in \mathcal{P}_k^\mathbb{Z}$ such that $P + \Delta^{2^k} = Q$.*

**Corollary**

*The minimum number of hidden layers to represent* \(\max\{0, x_1, \ldots, x_{2^k}\}\) *with integer weights is precisely* $k + 1$. 
Proof Idea

- Consider normalized volume of lattice polytopes \((\in \mathbb{Z})\)
- By carefully subdividing Minkowski sums and convex hulls:

**Lemma**

A \(2^k\)-dimensional polytope in \(\mathcal{P}_k^{\mathbb{Z}}\) has even normalized volume.
Proof Idea

- Consider normalized volume of lattice polytopes ($\in \mathbb{Z}$)
- By carefully subdividing Minkowski sums and convex hulls:

**Lemma**

A $2^k$-dimensional polytope in $\mathcal{P}_k^\mathbb{Z}$ has even normalized volume.

- Theorem follows because $\Delta^{2^k}$ has normalized volume one.
- Example in 2D:
Outlook

- Volume unsuitable for non-integer case
- Find different way to separate $P_k$ from each other!
Outlook

- Volume unsuitable for non-integer case
- Find different way to separate $\mathcal{P}_k$ from each other!

Conjecture

There is no pair of polytopes $P, Q \in \mathcal{P}_k$ such that $P + \Delta^{2^k} = Q$. 
Outlook

- Volume unsuitable for non-integer case
- Find different way to separate $P_k$ from each other!

Conjecture

There is no pair of polytopes $P, Q \in P_k$ such that $P + \Delta^{2^k} = Q.$