Understanding Neural Network Expressivity via Polyhedral Geometry

Christoph Hertrich

LSE

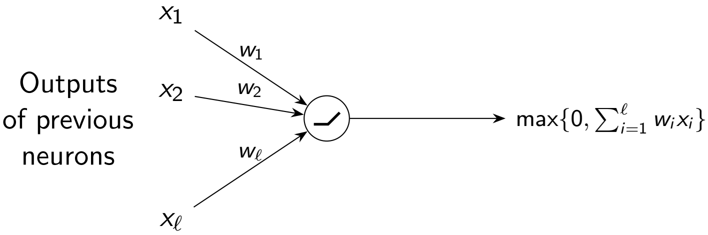
joint works with

Amitabh Basu, Marco Di Summa, Martin Skutella (NeurIPS 2021)

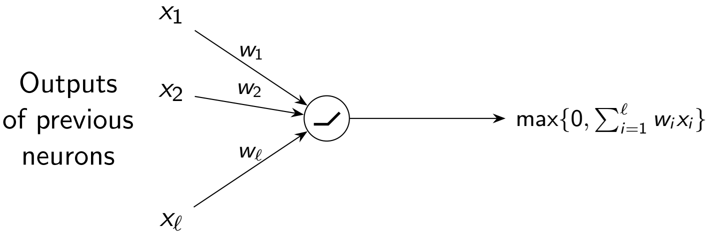
Christian Haase, Georg Loho (ICLR 2023)

Workshop "Trends in Computational Discrete Optimization",
ICERM, Brown University, Providence, RI, USA,
April 26, 2023

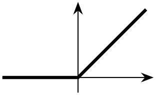
A Single ReLU Neuron



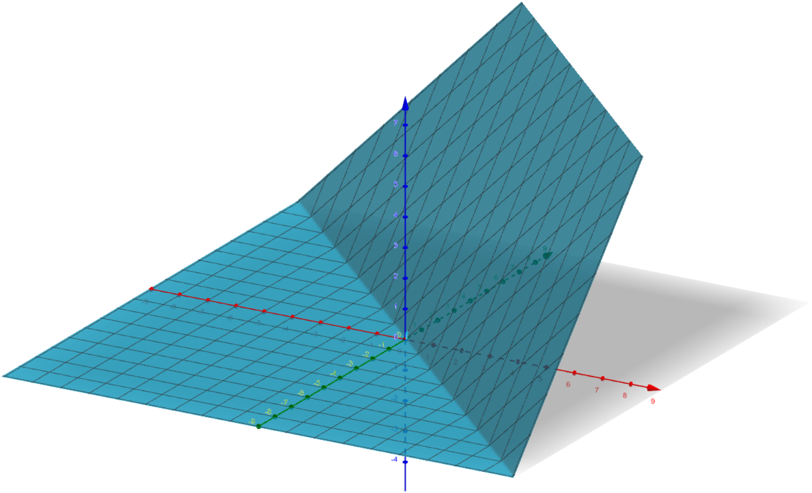
A Single ReLU Neuron



Rectified linear unit (ReLU): $\mathrm{relu}(x) = \max\{0, x\}$

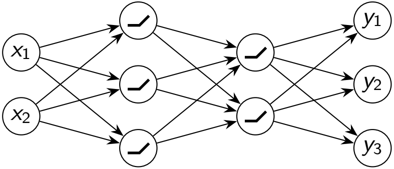


A Single ReLU Neuron



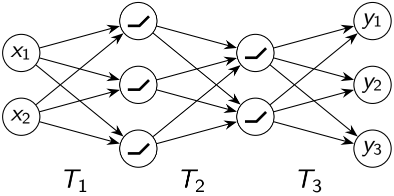
ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons



ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons



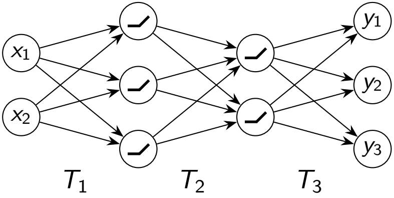
- Computes function

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

with linear transformations T_i.

ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons



- Computes function

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

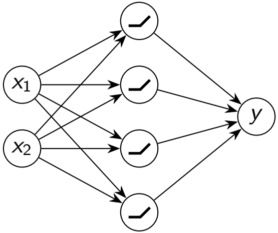
with linear transformations T_i.

- Example: depth 3 (2 hidden layers).

What is the class of functions computable by
ReLU Neural Networks
with a certain depth?

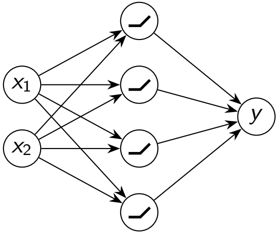
Universal approximation theorems:

One hidden layer enough to **approximate** any continuous function.



Universal approximation theorems:

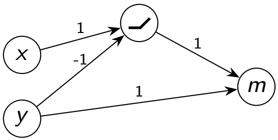
One hidden layer enough to **approximate** any continuous function.



What about **exact** representability?

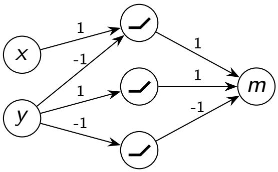
Example: Computing the Maximum of Two Numbers

$$\max\{x, y\} = \max\{x - y, 0\} + y$$

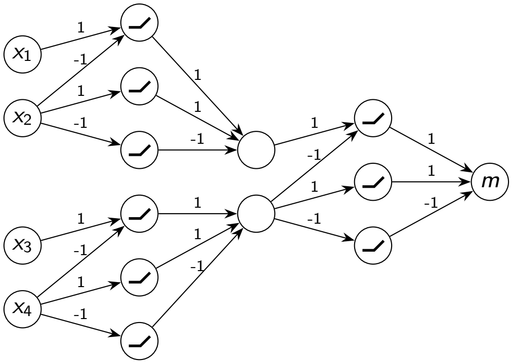


Example: Computing the Maximum of Two Numbers

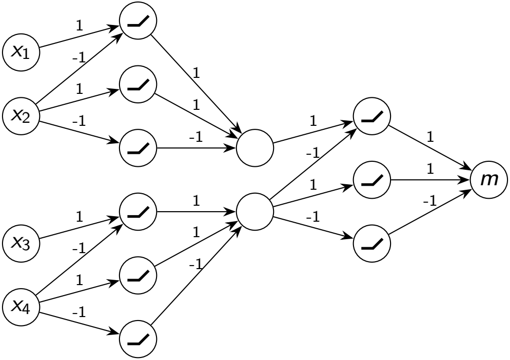
$$\max\{x, y\} = \max\{x - y, 0\} + y$$



Example: Computing the Maximum of Four Numbers



Example: Computing the Maximum of Four Numbers



- 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).

Representing Arbitrary Piecewise Linear Functions

Observation

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

Theorem (Wang, Sun [WS05])

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

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

Representing Arbitrary Piecewise Linear Functions

Observation

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

Theorem (Wang, Sun [WS05])

Every CPWL function $f : \mathbb{R}^n \to \mathbb{R}$ 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])

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

Natural Question

Theorem (Arora, Basu, Mianjy, Mukherjee [ABMM18])
Every CPWL function $f : \mathbb{R}^n \to \mathbb{R}$ 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:

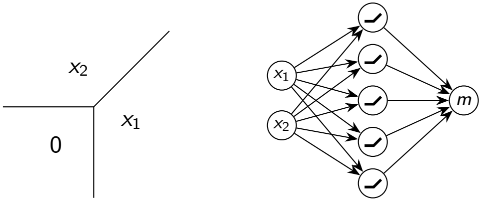
Conjecture

$\max\{0, x_1, \ldots, x_{2^k}\}$ *cannot be represented with* k *hidden layers.*

What is known?

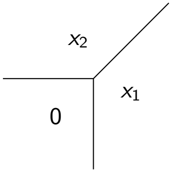
What is known?

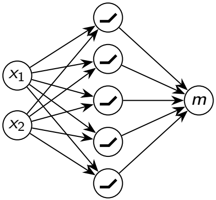
- Mukherjee, Basu (2017):
  $\max\{0, x_1, x_2\}$ not representable with 1 hidden layer:



What is known?

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

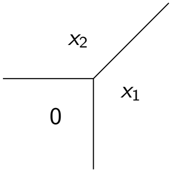


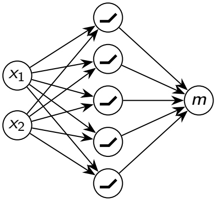


That's all!

What is known?

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



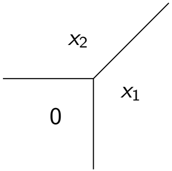


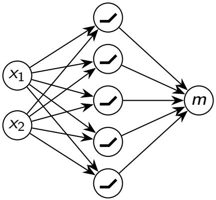
That's all!

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

What is 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 $\rightsquigarrow$ gap between 2 and $\lceil \log_2(n+1) \rceil$.
- Smallest candidate: $\max\{0, x_1, x_2, x_3, x_4\}$.

Our 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.

Our 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 (ICLR 2023):

  Depth $\mathcal{O}(\log n)$ tight for networks with only integer weights.

Our 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 (ICLR 2023):

  Depth $\mathcal{O}(\log n)$ tight for networks with only integer weights.

The Assumption

If ... there is a 2-hidden-layer NN computing $\max\{0, x_1, x_2, x_3, x_4\}$,
Then ... also one with the following property:

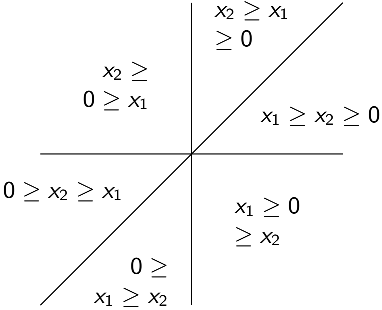
The output of each neuron can only have breakpoints where the relative ordering of the five numbers 0, x_1, ..., x_4 changes.

The Assumption

If ... there is a 2-hidden-layer NN computing $\max\{0, x_1, x_2, x_3, x_4\}$,
Then ... also one with the following property:

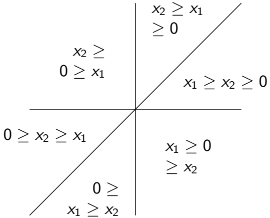
The output of each neuron can only have breakpoints where the relative ordering of the five numbers 0, x_1, ..., x_4 changes.

Example for $\max\{0, x_1, x_2\}$:



The Assumption

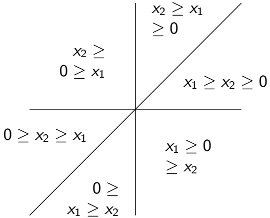
Example for $\max\{0, x_1, x_2\}$:



- $\binom{5}{2} = 10$ hyperplanes ...
- ... divide the input space into $5! = 120$ simplicial cones.

The Assumption

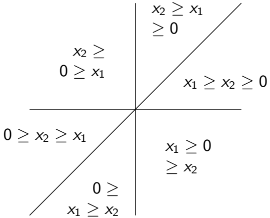
Example for $\max\{0, x_1, x_2\}$:



- $\binom{5}{2} = 10$ hyperplanes ...
- ... divide the input space into $5! = 120$ simplicial cones.
- Each cone spanned by 4 extreme rays.

The Assumption

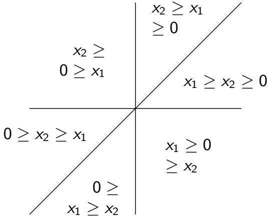
Example for $\max\{0, x_1, x_2\}$:



- $\binom{5}{2} = 10$ hyperplanes ...
- ... divide the input space into $5! = 120$ simplicial cones.
- Each cone spanned by 4 extreme rays.
- Within each cone everything is linear.

The Assumption

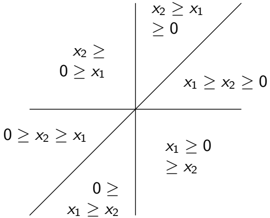
Example for $\max\{0, x_1, x_2\}$:



- $\binom{5}{2} = 10$ hyperplanes ...
- ... divide the input space into $5! = 120$ simplicial cones.
- Each cone spanned by 4 extreme rays.
- Within each cone everything is linear.
- 30 extreme rays in total.

The Assumption

Example for $\max\{0, x_1, x_2\}$:



- $\binom{5}{2} = 10$ hyperplanes ...
- ... divide the input space into $5! = 120$ simplicial cones.
- Each cone spanned by 4 extreme rays.
- Within each cone everything is linear.
- 30 extreme rays in total.

$\Rightarrow$ Vector space of possible CPWL functions is 30-dimensional!

Basic Linear Algebra Shows ...

- ... after 1 hidden layer:
  exactly 14 of 30 dimensions can be reached.

Basic Linear Algebra Shows ...

- ... after 1 hidden layer:
  exactly 14 of 30 dimensions can be reached.
- ... after 2 hidden layers:
  at least 29 of 30 dimensions can be reached.

Basic Linear Algebra Shows ...

- ... after 1 hidden layer:
  exactly 14 of 30 dimensions can be reached.
- ... after 2 hidden layers:
  at least 29 of 30 dimensions can be reached.

$$\max\{0, x_1, x_2, x_3, x_4\}$$

is not contained in the 29-dimensional subspace!

Can we leave the 29-dimensional subspace?

Can we leave the 29-dimensional subspace?

Mixed-Integer Linear Program to model a neuron in 2nd layer:

Can we leave the 29-dimensional subspace?

Mixed-Integer Linear Program to model a neuron in 2nd layer:

- $14 + 30 = 44$ continuous variables
- 30 binary variables
- a few hundred constraints

Can we leave the 29-dimensional subspace?

Mixed-Integer Linear Program to model a neuron in 2nd layer:

- $14 + 30 = 44$ continuous variables
- 30 binary variables
- a few hundred constraints
- objective orthogonal to 29-dim. subspace

Can we leave the 29-dimensional subspace?

Mixed-Integer Linear Program to model a neuron in 2nd layer:

- $14 + 30 = 44$ continuous variables
- 30 binary variables
- a few hundred constraints
- objective orthogonal to 29-dim. subspace

$\Rightarrow$ Solver (with exact arithmetic): Objective value zero

Can we leave the 29-dimensional subspace?

Mixed-Integer Linear Program to model a neuron in 2nd layer:

- $14 + 30 = 44$ continuous variables
- 30 binary variables
- a few hundred constraints
- objective orthogonal to 29-dim. subspace

$\Rightarrow$ Solver (with exact arithmetic): Objective value zero

No!

Can we leave the 29-dimensional subspace?

Mixed-Integer Linear Program to model a neuron in 2nd layer:

- $14 + 30 = 44$ continuous variables
- 30 binary variables
- a few hundred constraints
- objective orthogonal to 29-dim. subspace

$\Rightarrow$ Solver (with exact arithmetic): Objective value zero

No!

Theorem

A neural network satisfying our assumption needs 3 hidden layers to compute $\max\{0, x_1, x_2, x_3, x_4\}$.

Our 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 (ICLR 2023):

  Depth $\mathcal{O}(\log n)$ tight for networks with only integer weights.

Our 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 (ICLR 2023):

  Depth $\mathcal{O}(\log n)$ tight for networks with only integer weights.

Newton Polytope of a Convex CPWL Function

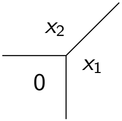
- $f(x) = \max\{a_1^T x, \ldots, a_k^T x\} \quad \rightsquigarrow \quad P(f) = \mathrm{conv}\{a_1, \ldots, a_k\}$
- dual to underlying polyhedral complex of the CPWL function

Example for $\max\{0, x_1, x_2\}$:

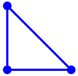
Newton Polytope of a Convex CPWL Function

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

Example for $\max\{0, x_1, x_2\}$:

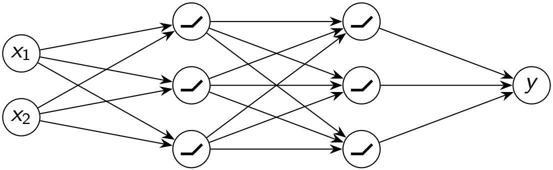


$\rightsquigarrow$

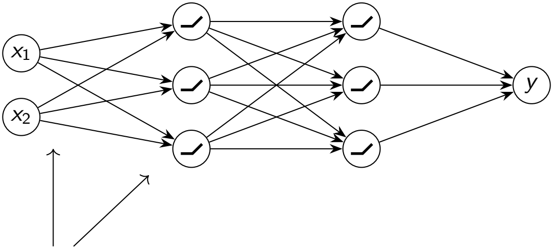


Convex CPWL functions	$\cong$	**Newton Polytopes**
(positive) scalar multiplication		scaling
addition		Minkowski sum
taking maximum		taking convex hull of union

Newton Polytopes and Neural Networks

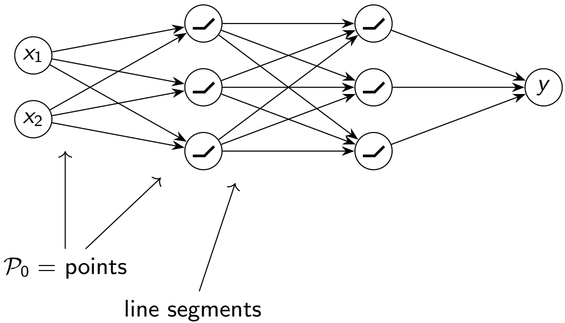


Newton Polytopes and Neural Networks

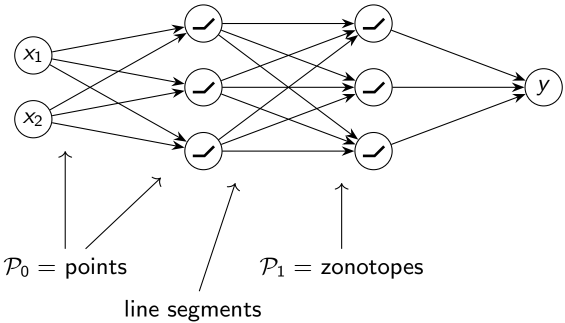


$\mathcal{P}_0 =$ points

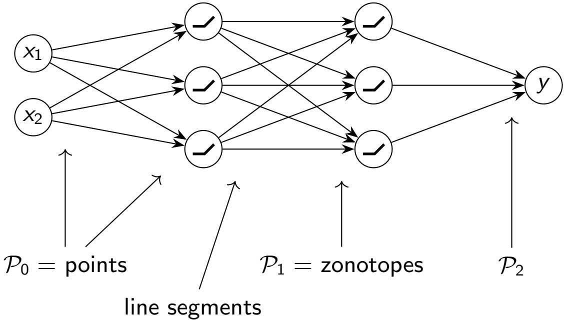
Newton Polytopes and Neural Networks



Newton Polytopes and Neural Networks



Newton Polytopes and Neural Networks



$$\mathcal{P}_k := \left\{ \sum_{i=1}^{m} \operatorname{conv}(P_i, Q_i) \;\middle|\; P_i, Q_i \in \mathcal{P}_{k-1} \right\}$$

Results for the Integer Case

Theorem

With only integer weights, k hidden layers are not enough to compute $\max\{0, x_1, \ldots, x_{2^k}\}$.

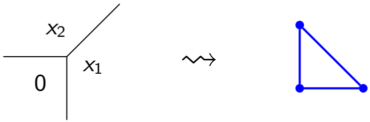
Results for the Integer Case

Theorem

With only integer weights, k hidden layers are not enough to compute $\max\{0, x_1, \ldots, x_{2^k}\}$.

- Use tropical geometry to represent NNs as lattice polytopes.

Example for $\max\{0, x_1, x_2\}$:



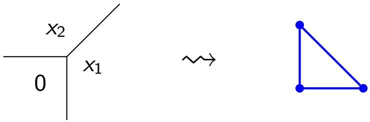
Results for the Integer Case

Theorem

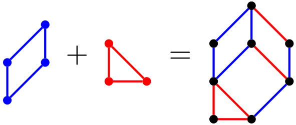
With only integer weights, k hidden layers are not enough to compute $\max\{0, x_1, \ldots, x_{2^k}\}$.

- Use tropical geometry to represent NNs as lattice polytopes.

Example for $\max\{0, x_1, x_2\}$:



- Subdivide polytopes "layer by layer" into "easier" polytopes.



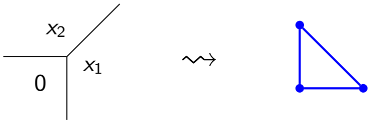
Results for the Integer Case

Theorem

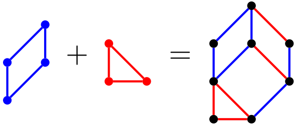
With only integer weights, k hidden layers are not enough to compute $\max\{0, x_1, \ldots, x_{2^k}\}$.

- Use tropical geometry to represent NNs as lattice polytopes.

Example for $\max\{0, x_1, x_2\}$:



- Subdivide polytopes "layer by layer" into "easier" polytopes.



- Separate via parity of the normalized volume.

What's Next?

- 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 (ICLR 2023):

  Depth $\mathcal{O}(\log n)$ tight for networks with only integer weights.

What's Next?

- 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.

  - Getting rid of assumption.
  - Getting rid of MIP.
  - Sharpen MIP to tackle 3-hidden-layer case.

- Haase, Hertrich, Loho (ICLR 2023):

  Depth $\mathcal{O}(\log n)$ tight for networks with only integer weights.

What's Next?

- 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.

  - Getting rid of assumption.
  - Getting rid of MIP.
  - Sharpen MIP to tackle 3-hidden-layer case.

- Haase, Hertrich, Loho (ICLR 2023):

  Depth $\mathcal{O}(\log n)$ tight for networks with only integer weights.

  - For general case: Polytopes and subdivisions seem promising.
  - Replace volume argument by different separation.

What's Next?

- 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.

  - Getting rid of assumption.
  - Getting rid of MIP.
  - Sharpen MIP to tackle 3-hidden-layer case.

- Haase, Hertrich, Loho (ICLR 2023):

  Depth $\mathcal{O}(\log n)$ tight for networks with only integer weights.

  - For general case: Polytopes and subdivisions seem promising.
  - Replace volume argument by different separation.

Thank you!