Tackling Neural Network Expressivity via Polytopes

Christoph Hertrich

joint work with

Amitabh Basu   Marco Di Summa   Martin Skutella

(Polytop)ics conference
April 6, 2021

Can **3-layer neural networks** compute the **maximum of 5 numbers**?

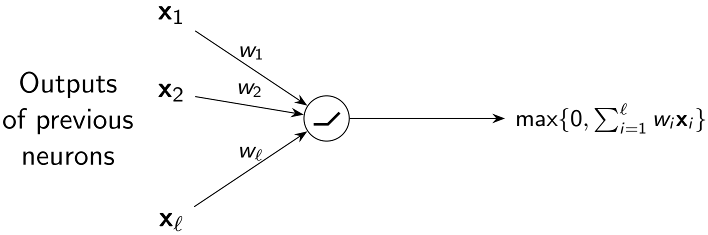
Christoph Hertrich

joint work with

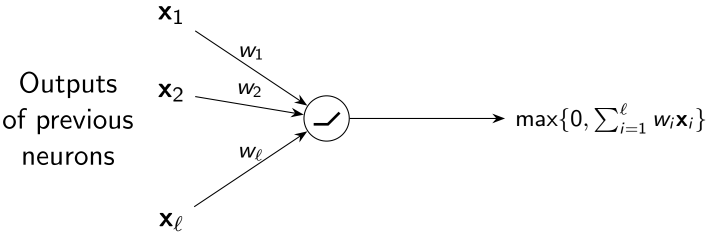
Amitabh Basu   Marco Di Summa   Martin Skutella

(Polytop)ics conference
April 6, 2021

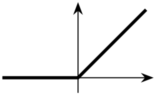
A Single ReLU Neuron



A Single ReLU Neuron

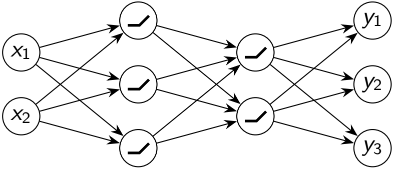


Rectified linear unit (ReLU): $\mathrm{relu}(x) = \max\{0, x\}$



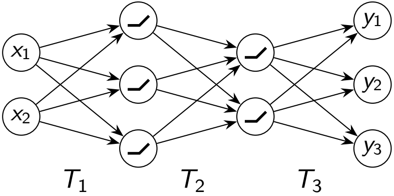
ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons



ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons



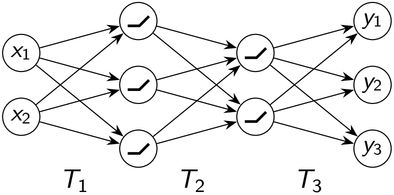
- Computes function

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

with linear transformations T_i.

ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons



- Computes function

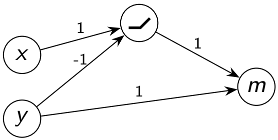
$$T_k \circ \mathrm{relu} \circ T_{k-1} \circ \cdots \circ T_2 \circ \mathrm{relu} \circ T_1$$

with linear transformations T_i.

- Example: depth 3 (2 hidden layers).

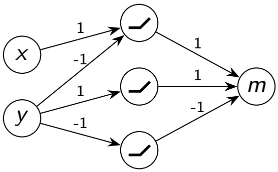
Example: Computing the Maximum of Two Numbers

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

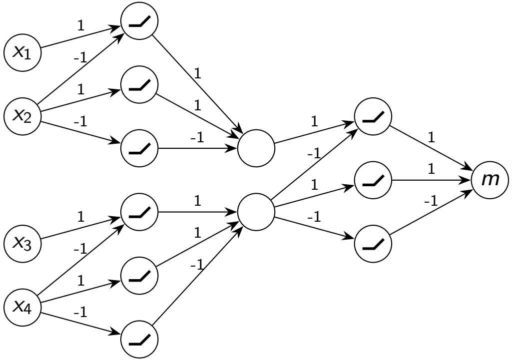


Example: Computing the Maximum of Two Numbers

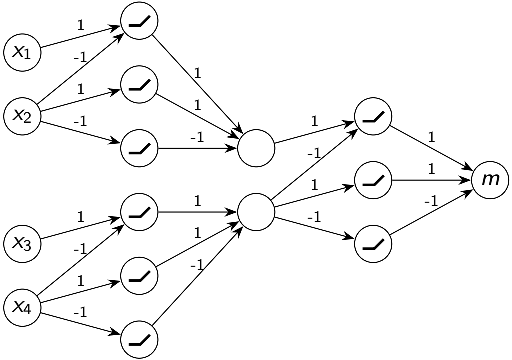
$$\max\{x, y\} = \max\{x - y, 0\} + y$$



Example: Computing the Maximum of Four Numbers

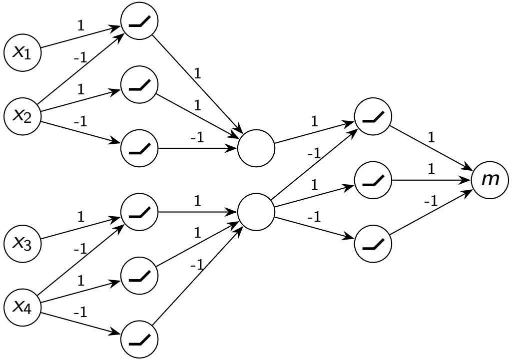


Example: Computing the Maximum of Four Numbers



- Inductively: Maximum of n numbers with depth $\lceil \log_2(n) \rceil + 1$.

Example: Computing the Maximum of Four Numbers



- Inductively: Maximum of n numbers with depth $\lceil \log_2(n) \rceil + 1$.

Question: Is this best possible?

Why is the maximum function so interesting?

Theorem (Arora, Basu, Mianjy, Mukherjee (2018))

$f : \mathbb{R}^n \to \mathbb{R}$ can be represented by a ReLU NN if and only if f is continuous and piecewise linear (CPWL).

Why is the maximum function so interesting?

Theorem (Arora, Basu, Mianjy, Mukherjee (2018))

$f : \mathbb{R}^n \to \mathbb{R}$ can be represented by a ReLU NN if and only if f is continuous and piecewise linear (CPWL).

Theorem (Wang, Sun (2005))

Any (CPWL) function $f : \mathbb{R}^n \to \mathbb{R}$ can be written as linear combination of maxima of $n + 1$ linear terms.

Why is the maximum function so interesting?

Theorem (Arora, Basu, Mianjy, Mukherjee (2018))

$f : \mathbb{R}^n \to \mathbb{R}$ can be represented by a ReLU NN if and only if f is continuous and piecewise linear (CPWL).

Theorem (Wang, Sun (2005))

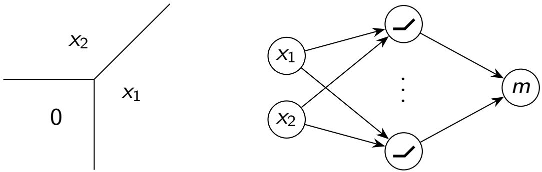
Any (CPWL) function $f : \mathbb{R}^n \to \mathbb{R}$ can be written as linear combination of maxima of $n + 1$ linear terms.

$\Rightarrow$ Everything depends on the maximum function!

What's known?

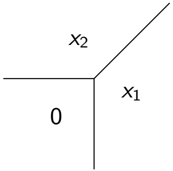
What's known?

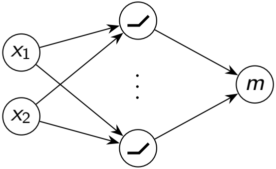
- $\max\{0, x_1, x_2\}$ cannot be computed with 2 layers.



What's known?

- $\max\{0, x_1, x_2\}$ cannot be computed with 2 layers.

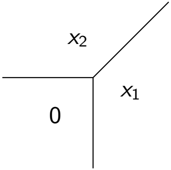


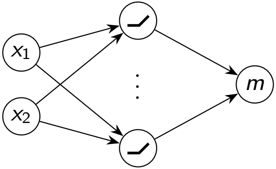


(set of break points must be union of lines)

What's known?

- $\max\{0, x_1, x_2\}$ cannot be computed with 2 layers.



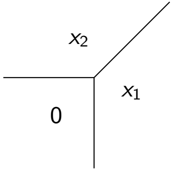


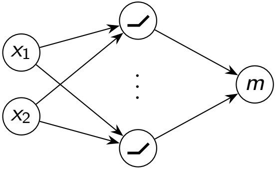
(set of break points must be union of lines)

That's all!

What's known?

- $\max\{0, x_1, x_2\}$ cannot be computed with 2 layers.





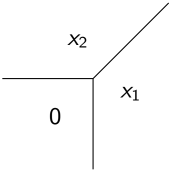
(set of break points must be union of lines)

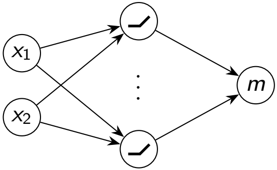
That's all!

- No function known that provably needs more than 3 layers.

What's known?

- $\max\{0, x_1, x_2\}$ cannot be computed with 2 layers.





(set of break points must be union of lines)

That's all!

- No function known that provably needs more than 3 layers.
- Smallest open case:
  Can $\max\{0, x_1, x_2, x_3, x_4\}$ be computed with 3 layers?

In this talk:
Two possible approaches:

In this talk:
Two possible approaches:

1. MIP-based proof that $\max\{0, x_1, x_2, x_3, x_4\}$ cannot be computed with 3 layers

In this talk:
Two possible approaches:

1. MIP-based proof that $\max\{0, x_1, x_2, x_3, x_4\}$ cannot be computed with 3 layers **under an additional assumption**.

In this talk:
Two possible approaches:

1. MIP-based proof that $\max\{0, x_1, x_2, x_3, x_4\}$ cannot be computed with 3 layers **under an additional assumption**.

2. Using Newton polytopes of CPWL functions.

In this talk:
Two possible approaches:

1. MIP-based proof that $\max\{0, x_1, x_2, x_3, x_4\}$ cannot be computed with 3 layers **under an additional assumption**.

2. Using Newton polytopes of CPWL functions.

(for notational purposes: $x_0 := 0$.)

The Assumption

If ... there exists a 3-layer NN computing $\max\{0, x_1, x_2, x_3, x_4\}$,
Then ... also one with the following property:

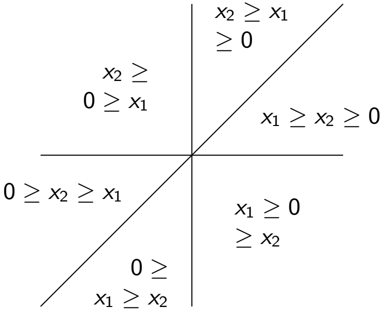
The output of each neuron can only have breakpoints where the relative ordering of the five numbers 0, x_1, ..., x_4 changes.

The Assumption

If ... there exists a 3-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 Rephrased

If ... there exists a 3-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 is linear within each region of the **hyperplane arrangement** given by $\binom{5}{2} = 10$ hyperplanes:
$x_i = x_j$, $0 \leq i < j \leq 4$.

The Assumption Rephrased

If … there exists a 3-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 is linear within each region of the **hyperplane arrangement** given by $\binom{5}{2} = 10$ hyperplanes: $x_i = x_j$, $0 \leq i < j \leq 4$.

- dual to a zonotope, combinatorially the 4-dim. permutahedron

The Assumption Rephrased

If ... there exists a 3-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 is linear within each region of the **hyperplane arrangement** given by $\binom{5}{2} = 10$ hyperplanes: $x_i = x_j$, $0 \leq i < j \leq 4$.

- dual to a zonotope, combinatorially the 4-dim. permutahedron
- $5! = 120$ regions, which are simplicial cones

The Assumption Rephrased

If ... there exists a 3-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 is linear within each region of the **hyperplane arrangement** given by $\binom{5}{2} = 10$ hyperplanes: $x_i = x_j$, $0 \leq i < j \leq 4$.

- dual to a zonotope, combinatorially the 4-dim. permutahedron
- $5! = 120$ regions, which are simplicial cones
- each cone spanned by 4 extreme rays

The Assumption Rephrased

If ... there exists a 3-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 is linear within each region of the **hyperplane arrangement** given by $\binom{5}{2} = 10$ hyperplanes: $x_i = x_j$, $0 \leq i < j \leq 4$.

- dual to a zonotope, combinatorially the 4-dim. permutahedron
- $5! = 120$ regions, which are simplicial cones
- each cone spanned by 4 extreme rays
- $2^5 - 2 = 30$ extreme rays in total

The Assumption Rephrased

If ... there exists a 3-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 is linear within each region of the **hyperplane arrangement** given by $\binom{5}{2} = 10$ hyperplanes: $x_i = x_j$, $0 \leq i < j \leq 4$.

- dual to a zonotope, combinatorially the 4-dim. permutahedron
- $5! = 120$ regions, which are simplicial cones
- each cone spanned by 4 extreme rays
- $2^5 - 2 = 30$ 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 continuous variables (lin. combination of 1st-layer outputs)

Can we leave the 29-dimensional subspace?

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

- 14 continuous variables (lin. combination of 1st-layer outputs)
- 30 binary variables (sign of input value at each extreme ray)

Can we leave the 29-dimensional subspace?

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

- 14 continuous variables (lin. combination of 1st-layer outputs)
- 30 binary variables (sign of input value at each extreme ray)
- 30 continuous variables (function values at each extreme ray)

Can we leave the 29-dimensional subspace?

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

- 14 continuous variables (lin. combination of 1st-layer outputs)
- 30 binary variables (sign of input value at each extreme ray)
- 30 continuous variables (function values at each extreme ray)
- a few hundred constraints (e.g., to ensure assumption)

Can we leave the 29-dimensional subspace?

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

- 14 continuous variables (lin. combination of 1st-layer outputs)
- 30 binary variables (sign of input value at each extreme ray)
- 30 continuous variables (function values at each extreme ray)
- a few hundred constraints (e.g., to ensure assumption)
- 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 continuous variables (lin. combination of 1st-layer outputs)
- 30 binary variables (sign of input value at each extreme ray)
- 30 continuous variables (function values at each extreme ray)
- a few hundred constraints (e.g., to ensure assumption)
- objective orthogonal to 29-dim. subspace

$\Rightarrow$ Solver: Objective value zero

Can we leave the 29-dimensional subspace?

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

- 14 continuous variables (lin. combination of 1st-layer outputs)
- 30 binary variables (sign of input value at each extreme ray)
- 30 continuous variables (function values at each extreme ray)
- a few hundred constraints (e.g., to ensure assumption)
- objective orthogonal to 29-dim. subspace

$\Rightarrow$ Solver: Objective value zero

No!

In this talk:
Two possible approaches:

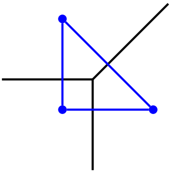
1. MIP-based proof that $\max\{0, x_1, x_2, x_3, x_4\}$ cannot be computed with 3 layers **under an additional assumption**.

2. Using Newton polytopes of CPWL functions.

Newton Polytope of a Convex CPWL Function

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

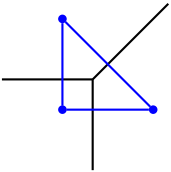
Example for $\max\{0, x_1, x_2\}$:



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

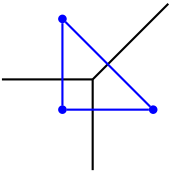


Convex CPWL functions	**Newton Polytopes**
(positive) scalar multiplication	scaling
addition	Minkowski sum
taking maximum	taking joint convex hull

Newton Polytope of a Convex CPWL Function

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

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



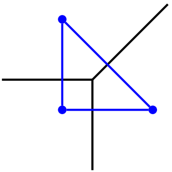
Convex CPWL functions	Newton Polytopes
(positive) scalar multiplication	scaling
addition	Minkowski sum
taking maximum	taking joint convex hull

Problem: Not every CPWL function is convex ...

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

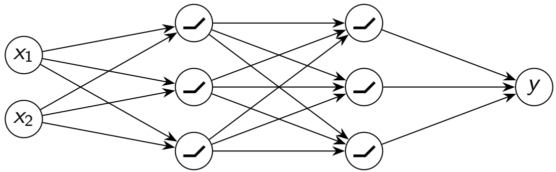


Convex CPWL functions	**Newton Polytopes**
(positive) scalar multiplication	scaling
addition	Minkowski sum
taking maximum	taking joint convex hull

Problem: Not every CPWL function is convex ...
But: Can represent them as difference of two convex ones!

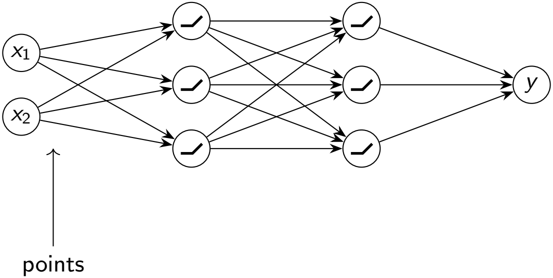
Newton Polytopes and Neural Networks

[Zhang, Naitzat, Lim: Tropical Geometry of Deep Neural Networks. ICML 2018]



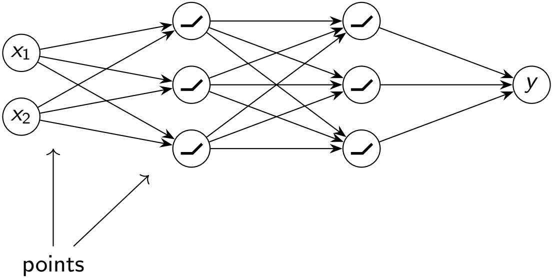
Newton Polytopes and Neural Networks

[Zhang, Naitzat, Lim: Tropical Geometry of Deep Neural Networks. ICML 2018]



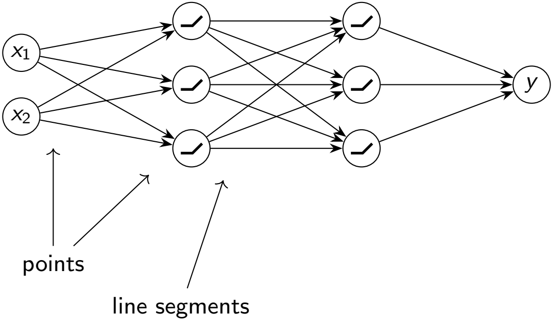
Newton Polytopes and Neural Networks

[Zhang, Naitzat, Lim: Tropical Geometry of Deep Neural Networks. ICML 2018]



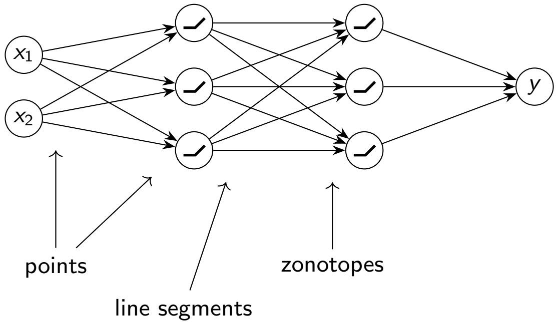
Newton Polytopes and Neural Networks

[Zhang, Naitzat, Lim: Tropical Geometry of Deep Neural Networks. ICML 2018]



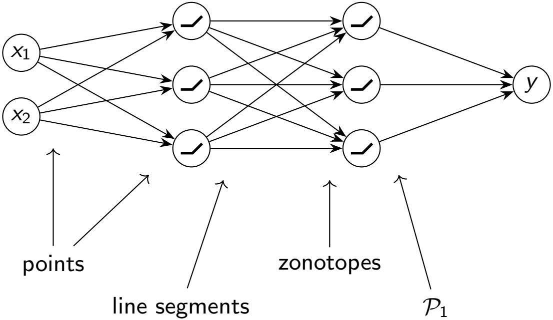
Newton Polytopes and Neural Networks

[Zhang, Naitzat, Lim: Tropical Geometry of Deep Neural Networks. ICML 2018]



Newton Polytopes and Neural Networks

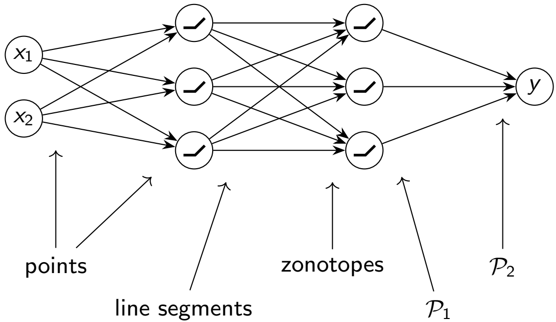
[Zhang, Naitzat, Lim: Tropical Geometry of Deep Neural Networks. ICML 2018]



$$\mathcal{P}_1 = \{P \text{ polytope} \mid P \text{ joint convex hull of two zonotopes}\}$$

Newton Polytopes and Neural Networks

[Zhang, Naitzat, Lim: Tropical Geometry of Deep Neural Networks. ICML 2018]



$\mathcal{P}_1 = \{P \text{ polytope} \mid P \text{ joint convex hull of two zonotopes}\}$

$\mathcal{P}_2 = \{P \text{ polytope} \mid P \text{ finite Minkowski sum of polytopes in } \mathcal{P}_1\}$

Translating the Problem into the Polytope World

If ... there is a 3-layer NN computing $f(x) = \max\{0, x_1, x_2, x_3, x_4\}$,
Then ... there are polytopes $Q, R \in \mathcal{P}_2$ with $Q + \Delta^4 = R$.

Translating the Problem into the Polytope World

If ... there is a 3-layer NN computing $f(x) = \max\{0, x_1, x_2, x_3, x_4\}$,
Then ... there are polytopes $Q, R \in \mathcal{P}_2$ with $Q + \Delta^4 = R$.

Sketch of Proof.
From NN we get convex CPWL functions g and h with ...

- $P(h), P(g) \in \mathcal{P}_2$,
- $f = g - h$, and hence $f + h = g$,
- $P(f) + P(h) = P(g)$. $\square$

Translating the Problem into the Polytope World

If ... there is a 3-layer NN computing $f(x) = \max\{0, x_1, x_2, x_3, x_4\}$,
Then ... there are polytopes $Q, R \in \mathcal{P}_2$ with $Q + \Delta^4 = R$.

Sketch of Proof.
From NN we get convex CPWL functions g and h with ...

- $P(h), P(g) \in \mathcal{P}_2$,
- $f = g - h$, and hence $f + h = g$,
- $P(f) + P(h) = P(g)$. □

The key is to ...

- Understand $\mathcal{P}_2$,

Translating the Problem into the Polytope World

If ... there is a 3-layer NN computing $f(x) = \max\{0, x_1, x_2, x_3, x_4\}$,
Then ... there are polytopes $Q, R \in \mathcal{P}_2$ with $Q + \Delta^4 = R$.

Sketch of Proof.

From NN we get convex CPWL functions g and h with ...

- $P(h), P(g) \in \mathcal{P}_2$,
- $f = g - h$, and hence $f + h = g$,
- $P(f) + P(h) = P(g)$. $\square$

The key is to ...

- Understand $\mathcal{P}_2$,
- Understand $\mathcal{P}_1$,

Translating the Problem into the Polytope World

If ... there is a 3-layer NN computing $f(x) = \max\{0, x_1, x_2, x_3, x_4\}$,
Then ... there are polytopes $Q, R \in \mathcal{P}_2$ with $Q + \Delta^4 = R$.

Sketch of Proof.
From NN we get convex CPWL functions g and h with ...

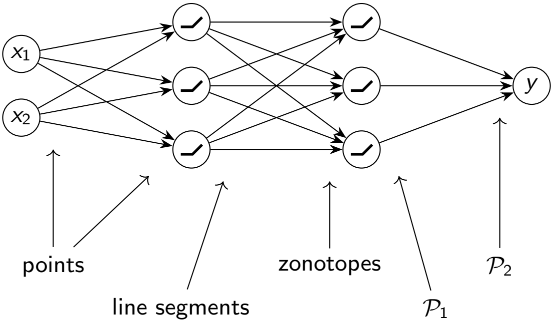
- $P(h), P(g) \in \mathcal{P}_2$,
- $f = g - h$, and hence $f + h = g$,
- $P(f) + P(h) = P(g)$. $\square$

The key is to ...

- Understand $\mathcal{P}_2$,
- Understand $\mathcal{P}_1$,
- Find characterizations for joint convex hulls of two zonotopes!

Thanks!

Questions? Ideas?



$$\mathcal{P}_1 = \{P \text{ polytope} \mid P \text{ joint convex hull of two zonotopes}\}$$

$$\mathcal{P}_2 = \{P \text{ polytope} \mid P \text{ finite Minkowski sum of polytopes in } \mathcal{P}_1\}$$

Are there polytopes $Q, R \in \mathcal{P}_2$ with $Q + \Delta^4 = R$?