Tackling Neural Network Expressivity via (virtual Newton) Polytopes

Christoph Hertrich

IOL & DISCOGA Research Seminar
May 11, 2021

Objectives of this Talk

Objectives of this Talk

1. Develop a **geometric understanding** of NNs

Objectives of this Talk

1. Develop a **geometric understanding** of NNs
   - [Zhang, Naitzat, Lim: **Tropical Geometry of Deep Neural Networks**, ICML 2018]
   - [Panina, Streinu: **Virtual Polytopes**, Oberwolfach Preprints 2015]

Objectives of this Talk

1. Develop a **geometric understanding** of NNs
   - [Zhang, Naitzat, Lim: **Tropical Geometry of Deep Neural Networks**, ICML 2018]
   - [Panina, Streinu: **Virtual Polytopes**, Oberwolfach Preprints 2015]
2. Translate two open problems into the **polytope world**:

Objectives of this Talk

1. Develop a **geometric understanding** of NNs
   - [Zhang, Naitzat, Lim:
     Tropical Geometry of Deep Neural Networks, ICML 2018]
   - [Panina, Streinu: **Virtual Polytopes**, Oberwolfach Preprints 2015]
2. Translate two open problems into the **polytope world**:
   - Can 3-layer NNs compute the maximum of 5 numbers?

Objectives of this Talk

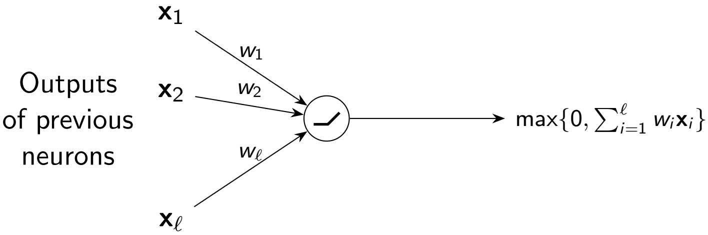
1. Develop a **geometric understanding** of NNs
   - [Zhang, Naitzat, Lim:
     Tropical Geometry of Deep Neural Networks, ICML 2018]
   - [Panina, Streinu: **Virtual Polytopes**, Oberwolfach Preprints 2015]
2. Translate two open problems into the **polytope world**:
   - Can 3-layer NNs compute the maximum of 5 numbers?
   - Can poly-size NNs solve the MST problem?

Objectives of this Talk

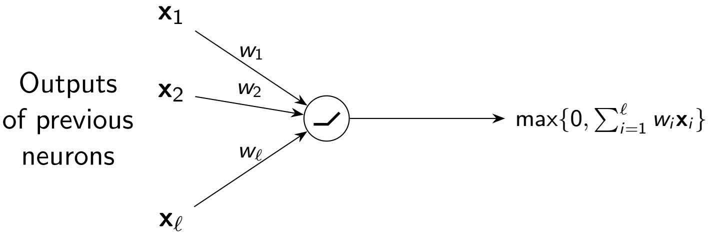
1. Develop a **geometric understanding** of NNs
   - [Zhang, Naitzat, Lim: **Tropical Geometry of Deep Neural Networks**, ICML 2018]
   - [Panina, Streinu: **Virtual Polytopes**, Oberwolfach Preprints 2015]
2. Translate two open problems into the **polytope world**:
   - Can 3-layer NNs compute the maximum of 5 numbers?
   - Can poly-size NNs solve the MST problem?

(both problems have broader context ...)

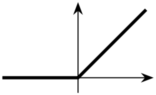
A Single ReLU Neuron



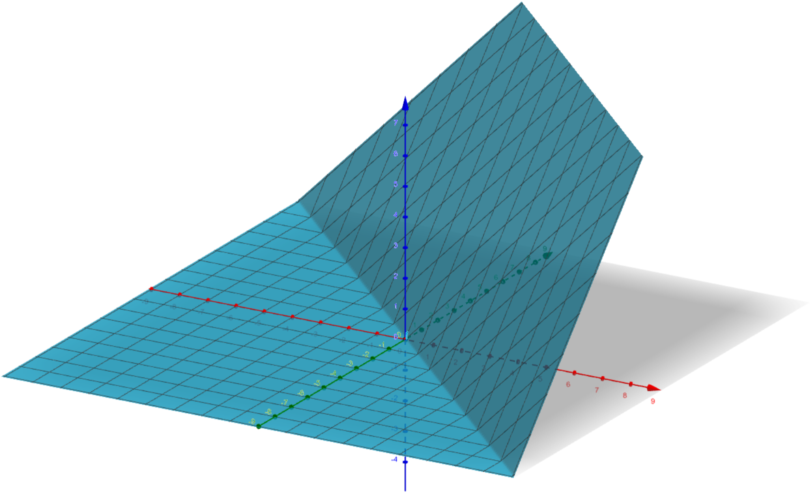
A Single ReLU Neuron



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

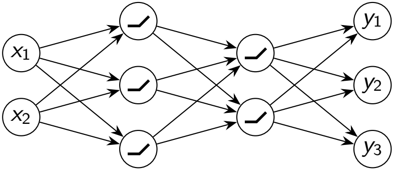


A Single ReLU Neuron



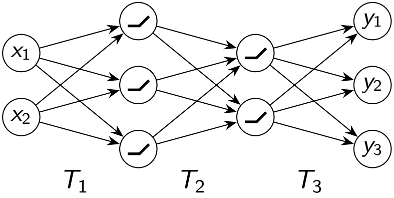
ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons



ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons



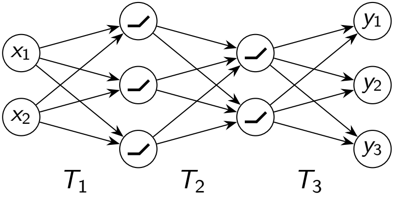
- Computes function

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

with linear transformations T_i.

ReLU Feedforward Neural Networks

- Acyclic (layered) digraph of ReLU neurons



- Computes function

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

with linear transformations T_i.

- Example: depth 3 (2 hidden layers).

NNs and CPWL Functions

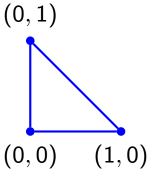
Theorem (Arora, Basu, Mianjy, Mukherjee (ICLR 2018))

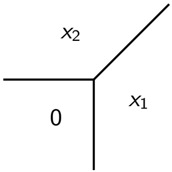
$f : \mathbb{R}^n \to \mathbb{R}$ can be represented by a ReLU NN if and only if f is continuous and piecewise linear (CPWL).

Support Functions and Newton Polytopes

Definition

The *support function* f_P of a polytope $P \subseteq \mathbb{R}^n$ maps a cost vector $x \in \mathbb{R}^n$ to the objective value of the linear program $\max_{v \in P} x^T v$.

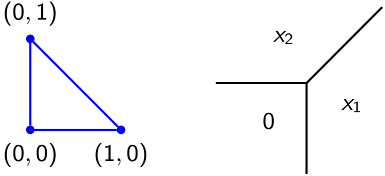




Support Functions and Newton Polytopes

Definition

The *support function* f_P of a polytope $P \subseteq \mathbb{R}^n$ maps a cost vector $x \in \mathbb{R}^n$ to the objective value of the linear program $\max_{v \in P} x^T v$.



Definition

The *Newton polytope* of a convex CPWL function $f(x) = \max\{v_1^T x, v_2^T x, \ldots, v_p^T x\}$ is $P(f) = \text{conv}\{v_1, v_2, \ldots, v_p\}$.

Bijection between Convex CPWL Functions and Polytopes

Have a bijection between these two sets:

- Convex CPWL functions $\mathbb{R}^n \to \mathbb{R}$,
- Polytopes in $\mathbb{R}^n$.

The bijection maps ...

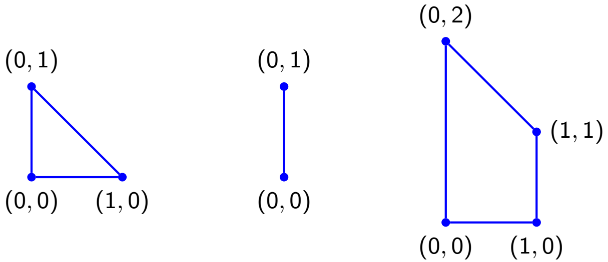
- Functions to their Newton polytope $f \mapsto P_f$,
- Polytopes to their support function $P \mapsto f_P$.

Adding Two (Convex) CPWL Functions

$$\begin{aligned}\max\{0, x_1, x_2\} + \max\{0, x_2\} &= \max\{0, x_2, x_1, x_1 + x_2, x_2, 2x_2\} \\ &= \max\{0, x_1, 2x_2, x_1 + x_2\}\end{aligned}$$

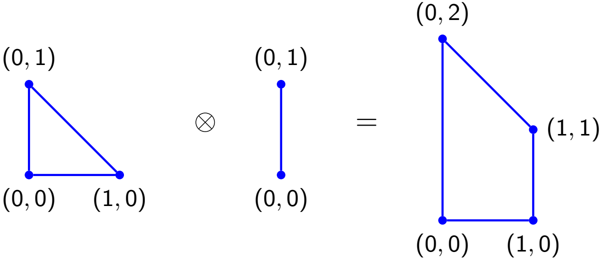
Adding Two (Convex) CPWL Functions

$$\max\{0, x_1, x_2\} + \max\{0, x_2\} = \max\{0, x_2, x_1, x_1 + x_2, x_2, 2x_2\}$$
$$= \max\{0, x_1, 2x_2, x_1 + x_2\}$$



Adding Two (Convex) CPWL Functions

$$\begin{aligned}\max\{0, x_1, x_2\} + \max\{0, x_2\} &= \max\{0, x_2, x_1, x_1 + x_2, x_2, 2x_2\} \\ &= \max\{0, x_1, 2x_2, x_1 + x_2\}\end{aligned}$$



- Minkowski sum: $P \otimes Q := \{p + q \mid p \in P, q \in Q\}$,

The Bijection is a Semigroup Isomorphism

$$P_{f+g} = P_f \otimes P_g,$$
$$f_{P \otimes Q} = f_P + f_Q.$$

(Convex CPWL functions $\mathbb{R}^n \to \mathbb{R}$, $+$)

$\cong$

(Polytopes in $\mathbb{R}^n$, $\otimes$)

But not Every CPWL Function is Convex ... ?

But not Every CPWL Function is Convex ... ?

- (CPWL functions, pointwise addition) is actually a group!

But not Every CPWL Function is Convex ... ?

- (CPWL functions, pointwise addition) is actually a group!
- Polytopes have no inverses w.r.t. Minkowski addition ...

But not Every CPWL Function is Convex ... ?

- (CPWL functions, pointwise addition) is actually a group!
- Polytopes have no inverses w.r.t. Minkowski addition ...
- Can we do something about this?

But not Every CPWL Function is Convex ... ?

- (CPWL functions, pointwise addition) is actually a group!
- Polytopes have no inverses w.r.t. Minkowski addition ...
- Can we do something about this?

YES!

$\rightsquigarrow$ Virtual Polytopes!

Remember When You Learned Mathematics ...

- Semigroup $(\mathbb{N}, +)$

Remember When You Learned Mathematics ...

- Semigroup $(\mathbb{N}, +) \rightsquigarrow$ group $\mathbb{Z} = \{n - m \mid n, m \in \mathbb{N}\}$

Remember When You Learned Mathematics ...

- Semigroup $(\mathbb{N}, +) \rightsquigarrow$ group $\mathbb{Z} = \{n - m \mid n, m \in \mathbb{N}\}$
- Semigroup $(\mathbb{Z} \setminus \{0\}, \cdot)$

Remember When You Learned Mathematics ...

- Semigroup $(\mathbb{N}, +) \rightsquigarrow$ group $\mathbb{Z} = \{n - m \mid n, m \in \mathbb{N}\}$
- Semigroup $(\mathbb{Z} \setminus \{0\}, \cdot) \rightsquigarrow$ group $\mathbb{Q} \setminus \{0\} = \{\frac{a}{b} \mid a, b \in \mathbb{Z} \setminus \{0\}\}$

Remember When You Learned Mathematics ...

- Semigroup $(\mathbb{N}, +) \rightsquigarrow$ group $\mathbb{Z} = \{n - m \mid n, m \in \mathbb{N}\}$
- Semigroup $(\mathbb{Z} \setminus \{0\}, \cdot) \rightsquigarrow$ group $\mathbb{Q} \setminus \{0\} = \{\frac{a}{b} \mid a, b \in \mathbb{Z} \setminus \{0\}\}$

In the same way:

- **Virtual Polytopes** $= \{\frac{P}{Q} = P \oslash Q \mid P, Q \subseteq \mathbb{R}^n$ polytopes$\}$

Remember When You Learned Mathematics ...

- Semigroup $(\mathbb{N}, +) \rightsquigarrow$ group $\mathbb{Z} = \{n - m \mid n, m \in \mathbb{N}\}$
- Semigroup $(\mathbb{Z} \setminus \{0\}, \cdot) \rightsquigarrow$ group $\mathbb{Q} \setminus \{0\} = \{\frac{a}{b} \mid a, b \in \mathbb{Z} \setminus \{0\}\}$

In the same way:

- **Virtual Polytopes** $= \{\frac{P}{Q} = P \oslash Q \mid P, Q \subseteq \mathbb{R}^n$ polytopes$\}$

(Careful: $\frac{P}{Q} \neq P - Q = \{p - q \mid p \in P, q \in Q\}$)

Remember When You Learned Mathematics ...

- Semigroup $(\mathbb{N}, +) \rightsquigarrow$ group $\mathbb{Z} = \{n - m \mid n, m \in \mathbb{N}\}$
- Semigroup $(\mathbb{Z} \setminus \{0\}, \cdot) \rightsquigarrow$ group $\mathbb{Q} \setminus \{0\} = \{\frac{a}{b} \mid a, b \in \mathbb{Z} \setminus \{0\}\}$

In the same way:

- **Virtual Polytopes** $= \{\frac{P}{Q} = P \oslash Q \mid P, Q \subseteq \mathbb{R}^n$ polytopes$\}$

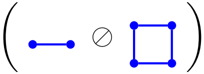
(Careful: $\frac{P}{Q} \neq P - Q = \{p - q \mid p \in P, q \in Q\}$)

- Remember: $\frac{a}{b} = \frac{c}{d} \quad \Leftrightarrow \quad ad = bc$
- In the same way: $\frac{P}{Q} = \frac{R}{S} \quad \Leftrightarrow \quad P \otimes S = Q \otimes R$

(formally via equivalence relations)

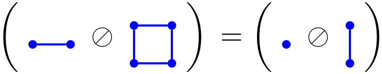
Two Examples

Cancellation Rule:



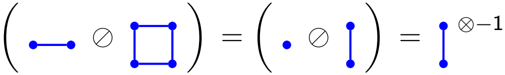
Two Examples

Cancellation Rule:



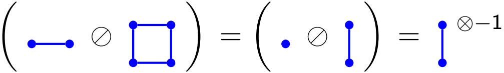
Two Examples

Cancellation Rule:

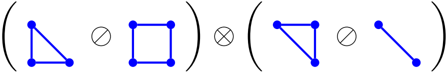


Two Examples

Cancellation Rule:

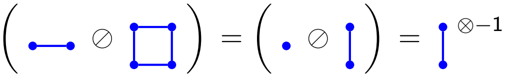


Adding two virtual polytopes:

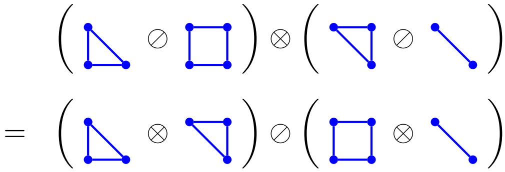


Two Examples

Cancellation Rule:

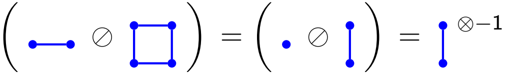


Adding two virtual polytopes:

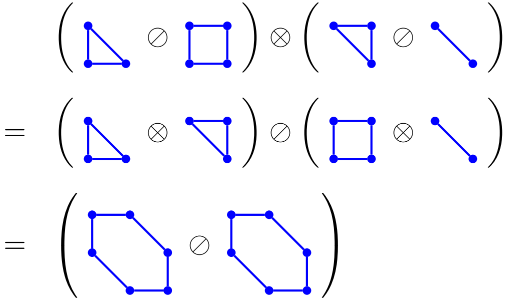


Two Examples

Cancellation Rule:

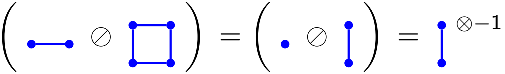


Adding two virtual polytopes:

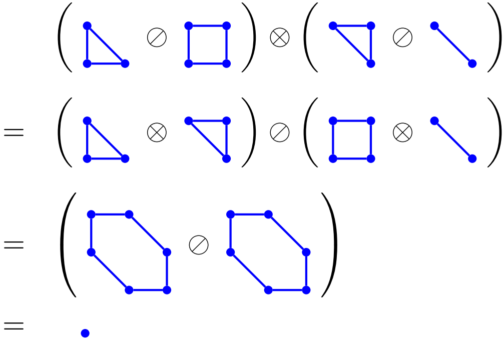


Two Examples

Cancellation Rule:



Adding two virtual polytopes:



We even have a Group Isomorphism!

$$(\text{CPWL functions } \mathbb{R}^n \to \mathbb{R}, +)$$
$$\cong$$
$$(\text{Virtual Polytopes in } \mathbb{R}^n, \otimes)$$

In particular:

Every CPWL function is a difference of two convex CPWL functions.

$$\frac{P}{Q} \mapsto f_P - f_Q$$
$$f - g \mapsto \frac{P_f}{P_g}$$

Virtual Polytopes are Almost Polytopes

- They have a **face lattice**,
  - faces are virtual polytopes.
- They have a **volume**,
  - but volume can be negative.

The General Idea

Find out which **virtual polytopes**

can occur as **Newton polytopes** of **CPWL functions**

computed by **neural networks** of a certain size (or structure).

Which Operations can NNs perform?

- Addition
- Scalar multiplication
- ReLU activations

Which Operations can NNs perform?

- Addition $\quad \leftrightarrow \quad$ Minkowski addition $\quad \checkmark$
- Scalar multiplication
- ReLU activations

Which Operations can NNs perform?

- Addition $\leftrightarrow$ Minkowski addition ✓
- Scalar multiplication ?
- ReLU activations ?

We even have a Vector Space Isomorphism!

$$f = \max_{i=1}^{p}\{v_i^T x\} \quad \Rightarrow \quad P_{\lambda f} = \text{conv}_{i=1}^{p}\{\lambda v_i\} = \lambda P_f$$

(CPWL functions, $+$, scalar multiplication)
$\cong$
(Virtual Polytopes, $\otimes$, scaling)

(for $\lambda < 0$ also consistent with Minkowski difference)

Which Operations can NNs perform?

- Addition $\leftrightarrow$ Minkowski addition ✓
- Scalar multiplication ?
- ReLU activations ?

Which Operations can NNs perform?

- Addition $\leftrightarrow$ Minkowski addition ✓
- Scalar multiplication $\leftrightarrow$ Scaling of (virtual) polytopes ✓
- ReLU activations ?

Which Operations can NNs perform?

- Addition $\leftrightarrow$ Minkowski addition ✓
- Scalar multiplication $\leftrightarrow$ Scaling of (virtual) polytopes ✓
- Taking Maxima ?

“Maximum” for Two Virtual Polytopes

For two convex CPWL functions f and g:

$$P_{\max\{f,g\}} = \operatorname{conv}\{P_f \cup P_g\} =: P_f \oplus P_g.$$

“Maximum” for Two Virtual Polytopes

For two convex CPWL functions f and g:

$$P_{\max\{f,g\}} = \operatorname{conv}\{P_f \cup P_g\} =: P_f \oplus P_g.$$

For $f = f_+ - f_-$ and $g = g_+ - g_-$ (non-convex) CPWL functions.

$$\begin{aligned} \Rightarrow \quad \max\{f,g\} &= \max\{f_+ - f_-, g_+ - g_-\} \\ &= \max\{f_+ + g_-, g_+ + f_-\} - (f_- + g_-). \end{aligned}$$

"Maximum" for Two Virtual Polytopes

For two convex CPWL functions f and g:

$$P_{\max\{f,g\}} = \mathrm{conv}\{P_f \cup P_g\} =: P_f \oplus P_g.$$

For $f = f_+ - f_-$ and $g = g_+ - g_-$ (non-convex) CPWL functions.

$$\Rightarrow \quad \max\{f,g\} = \max\{f_+ - f_-, g_+ - g_-\}$$
$$= \max\{f_+ + g_-, g_+ + f_-\} - (f_- + g_-).$$

Translating to virtual polytopes:

$$P_f \oplus P_g := P_{\max\{f,g\}}$$
$$= \frac{(P_{f_+} \otimes P_{g_-}) \oplus (P_{g_+} \otimes P_{f_-})}{(P_{f_-} \otimes P_{g_-})}$$

“Maximum” for Two Virtual Polytopes

For two convex CPWL functions f and g:

$$P_{\max\{f,g\}} = \operatorname{conv}\{P_f \cup P_g\} =: P_f \oplus P_g.$$

For $f = f_+ - f_-$ and $g = g_+ - g_-$ (non-convex) CPWL functions.

$$\begin{aligned} \Rightarrow \quad \max\{f,g\} &= \max\{f_+ - f_-, g_+ - g_-\} \\ &= \max\{f_+ + g_-, g_+ + f_-\} - (f_- + g_-). \end{aligned}$$

Translating to virtual polytopes:

$$\begin{aligned} P_f \oplus P_g &:= P_{\max\{f,g\}} \\ &= \frac{(P_{f_+} \otimes P_{g_-}) \oplus (P_{g_+} \otimes P_{f_-})}{(P_{f_-} \otimes P_{g_-})} \end{aligned}$$

Does this remind you of something? $\quad \rightsquigarrow \quad \frac{a}{b} + \frac{c}{d} = \frac{ad+bc}{bd}$

We also have a Semiring Isomorphism!

$$(\text{CPWL functions}, \max, +)$$
$$\cong$$
$$(\text{Virtual Polytopes}, \oplus, \otimes)$$

(here you see the “tropical” world!)

Which Operations can NNs perform?

- Addition $\leftrightarrow$ Minkowski addition ✓
- Scalar multiplication $\leftrightarrow$ Scaling of (virtual) polytopes ✓
- Taking Maxima ?

Which Operations can NNs perform?

- Addition $\leftrightarrow$ Minkowski addition ✓
- Scalar multiplication $\leftrightarrow$ Scaling of (virtual) polytopes ✓
- Taking Maxima $\leftrightarrow$ convex hull of the union ✓

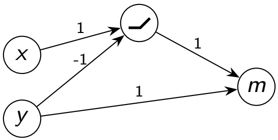
Take a Breath!

Now to the two open problems:

- Can 3-layer NNs compute the maximum of 5 numbers?
- Can poly-size NNs solve the MST problem?

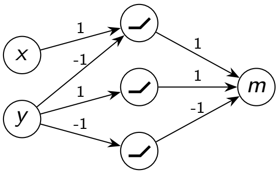
Computing the Maximum of Two Numbers

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

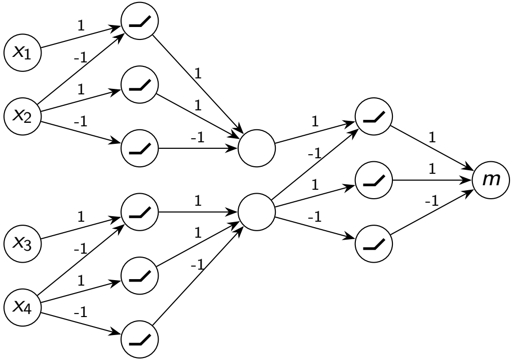


Computing the Maximum of Two Numbers

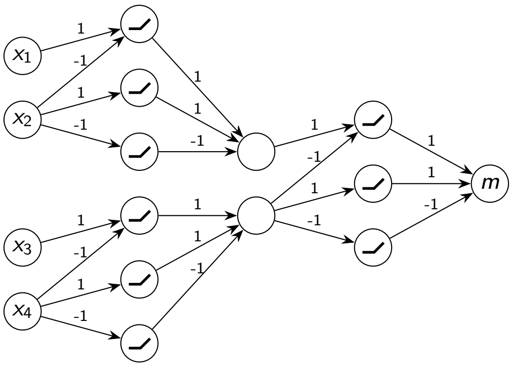
$$\max\{x, y\} = \max\{x - y, 0\} + y$$



Computing the Maximum of Four Numbers

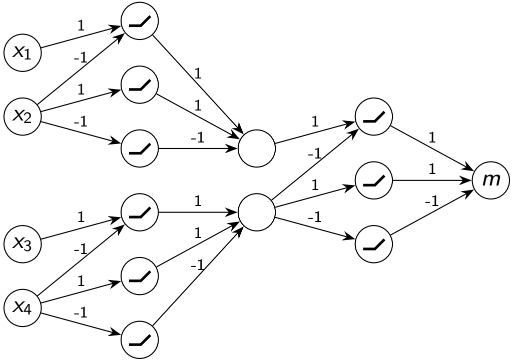


Computing the Maximum of Four Numbers



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

Computing the Maximum of Four Numbers



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

Question: Is this best possible?

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.

That's all!

What's known?

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

That's all!

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

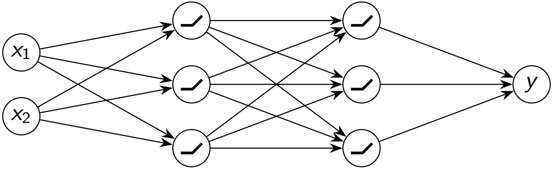
What's known?

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

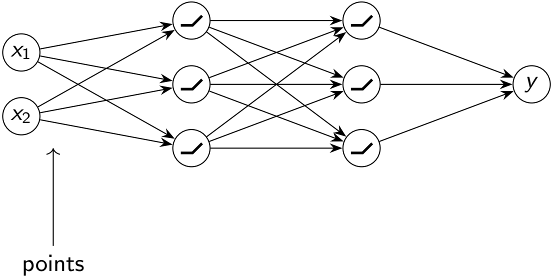
That's all!

- 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?

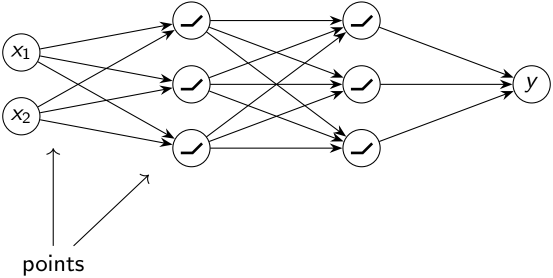
Virtual Newton Polytopes of Neural Networks



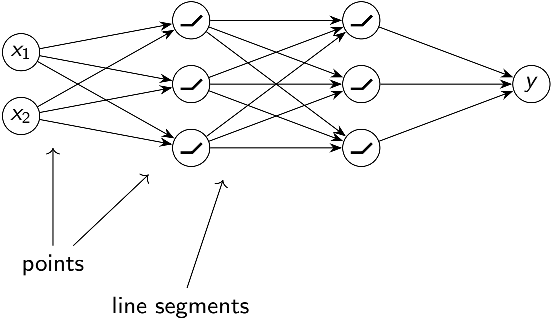
Virtual Newton Polytopes of Neural Networks



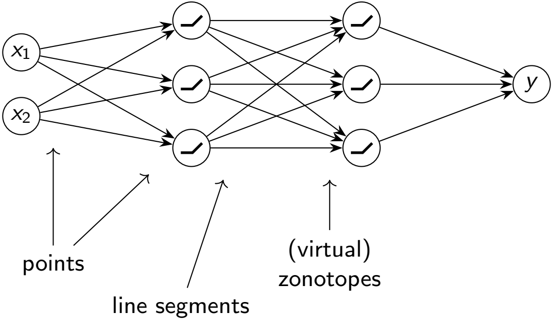
Virtual Newton Polytopes of Neural Networks



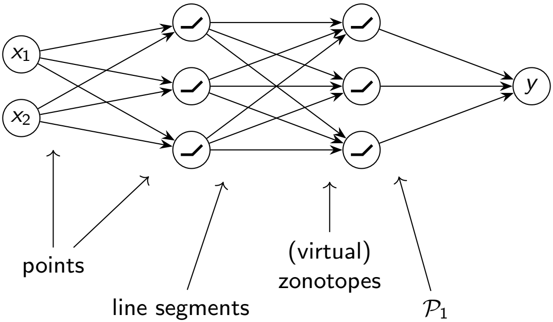
Virtual Newton Polytopes of Neural Networks



Virtual Newton Polytopes of Neural Networks



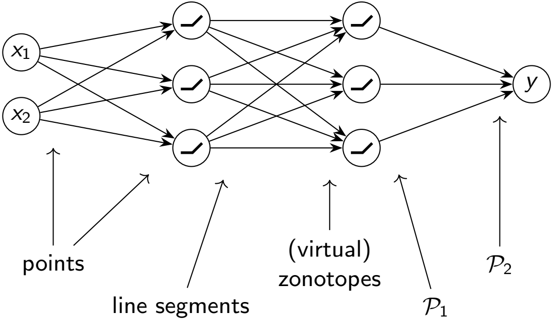
Virtual Newton Polytopes of Neural Networks



$$\mathcal{P}_1 = \{ \; Y \oplus Z \mid Y, \, Z \text{ (virtual) zonotopes } \}$$

(convex hull of the union of two zonotopes)

Virtual Newton Polytopes of Neural Networks

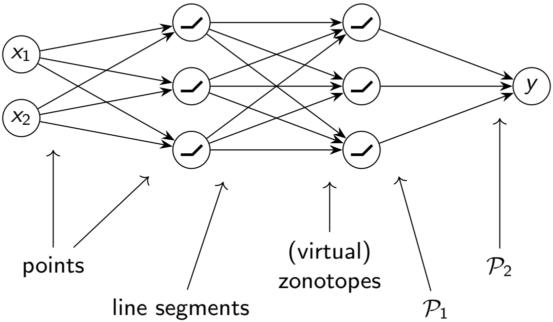


$$\mathcal{P}_1 = \{ \ Y \oplus Z \mid Y, \, Z \text{ (virtual) zonotopes } \}$$

(convex hull of the union of two zonotopes)

$$\mathcal{P}_2 = \{ \text{ finite Minkowski sums of (virtual) polytopes in } \mathcal{P}_1 \ \}$$

Virtual Newton Polytopes of Neural Networks



$$\mathcal{P}_1 = \{\ Y \oplus Z \mid Y,\ Z \text{ (virtual) zonotopes } \}$$

(convex hull of the union of two zonotopes)

$$\mathcal{P}_2 = \{\ \text{finite Minkowski sums of (virtual) polytopes in } \mathcal{P}_1\ \}$$

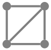
$f(x) = \max\{0, x_1, x_2, x_3, x_4\}$ can be computed by 3-layer NN

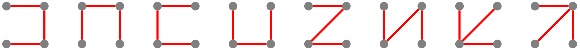
$$\Leftrightarrow \quad \text{4-simplex } \Delta^4 = P_f \in \mathcal{P}_2.$$

The Two Open Problems

- Can 3-layer NNs compute the maximum of 5 numbers?
- Can poly-size NNs solve the MST problem?

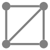
The Minimum Spanning Tree Problem

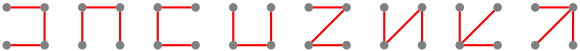
$G =$ 

$\mathcal{X} = \{$  $\}$

Let $\mathcal{X} \subseteq \{0,1\}^E$ be the set of characteristic vectors of spanning trees in a fixed graph $G = (V, E)$.

The Minimum Spanning Tree Problem

$G =$ 

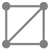
$\mathcal{X} = \{$  $\}$

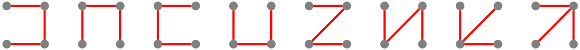
Let $\mathcal{X} \subseteq \{0,1\}^E$ be the set of characteristic vectors of spanning trees in a fixed graph $G = (V, E)$.

MST problem: Compute the CPWL function

$$c \mapsto \min_{x \in \mathcal{X}} c^T x.$$

The Minimum Spanning Tree Problem

$G =$ 

$\mathcal{X} = \Big\{$  $\Big\}$

Let $\mathcal{X} \subseteq \{0,1\}^E$ be the set of characteristic vectors of spanning trees in a fixed graph $G = (V, E)$.

MST problem: Compute the CPWL function

$$c \mapsto \min_{x \in \mathcal{X}} c^T x.$$

Question: Is there a poly-size NN computing this function?

Is there a poly-size NN to solve MST?

$$f(c) = \min_{x \in \mathcal{X}} x^T c = -\max_{x \in \mathcal{X}} (-x)^T c$$

$$\Rightarrow \qquad P_f = \text{conv}\{-x \mid x \in \mathcal{X}\}^{\otimes -1} = (-\text{MST-Polytope})^{\otimes -1}$$

Is there a poly-size NN to solve MST?

$$
\begin{aligned}
& f(c) = \min_{x \in \mathcal{X}} x^T c = -\max_{x \in \mathcal{X}} (-x)^T c \\
\Rightarrow \qquad & P_f = \operatorname{conv}\{-x \mid x \in \mathcal{X}\}^{\otimes -1} = (-\text{MST-Polytope})^{\otimes -1}
\end{aligned}
$$

Proposition

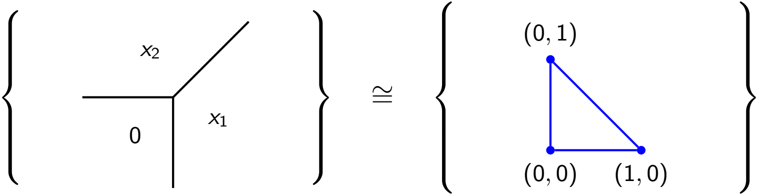
There exists a poly-size NN computing f

$\Leftrightarrow$

The MST-Polytope can be (virtually) generated from points via polynomially many operations of the form

- *Minkowski addition,*
- *Convex hull of the union,*
- *Scaling (with possibly inverting).*

Thank you!



Questions? Ideas?