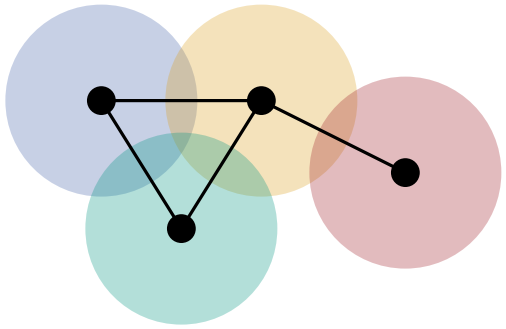
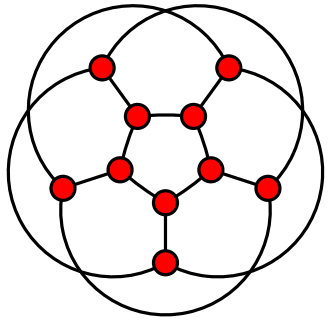
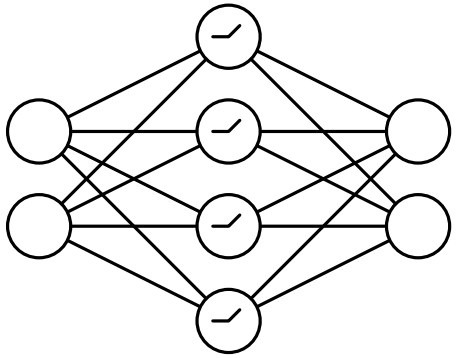
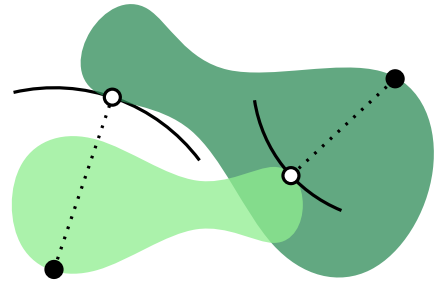
Capturing the Computational Complexity of Geometric Problems by the First-Order Theory of the Reals

Paul Jungeblut
Betreut durch: PD Dr. Torsten Ueckerdt

Karlsruher Institut für Technologie

Mündliche Promotionsprüfung · 26. April 2024

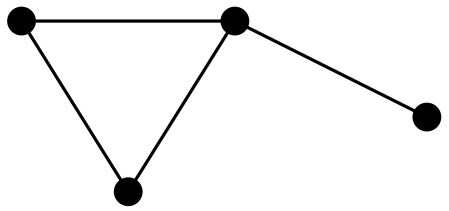
Problem 1: Erkennung von Unit-Disk-Graphen

Problem: Unit-Disk-Graphen erkennen

Eingabe: Graph $G = (V, E)$

Finde: Schnittrepräsentation von G bestehend aus Einheitskreisen.

Beispiel: $G = (V, E)$



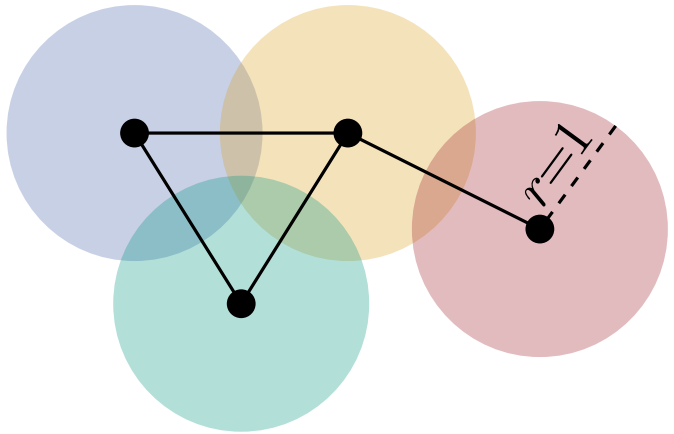
Problem 1: Erkennung von Unit-Disk-Graphen

Problem: Unit-Disk-Graphen erkennen

Eingabe: Graph $G = (V, E)$

Finde: Schnittrepräsentation von G bestehend aus Einheitskreisen.

Beispiel: $G = (V, E)$



Problem 1: Erkennung von Unit-Disk-Graphen

Problem: Unit-Disk-Graphen erkennen

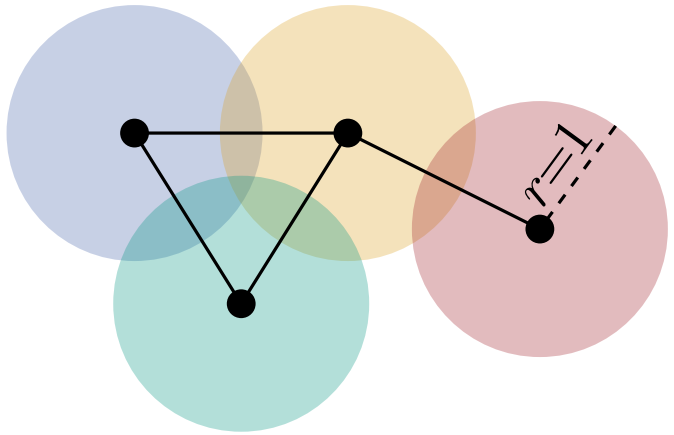
Eingabe: Graph $G = (V, E)$

Finde: Schnittrepräsentation von G bestehend aus Einheitskreisen.

Lösungen:

- Punkt aus $\mathbb{R}^2$ für jeden Knoten
- Kante $\iff$ Abstand ≤ 2

Beispiel: $G = (V, E)$



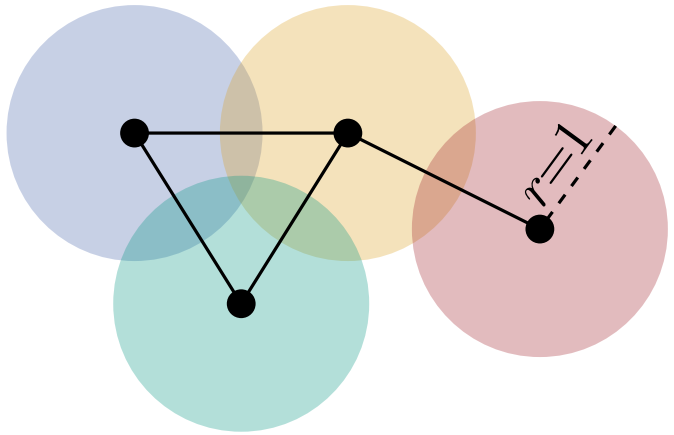
Problem 1: Erkennung von Unit-Disk-Graphen

Problem: Unit-Disk-Graphen erkennen

Eingabe: Graph $G = (V, E)$

Finde: Schnittrepräsentation von G bestehend aus Einheitskreisen.

Beispiel: $G = (V, E)$



Lösungen:

- Punkt aus $\mathbb{R}^2$ für jeden Knoten
- Kante $\iff$ Abstand ≤ 2

Komplexität:

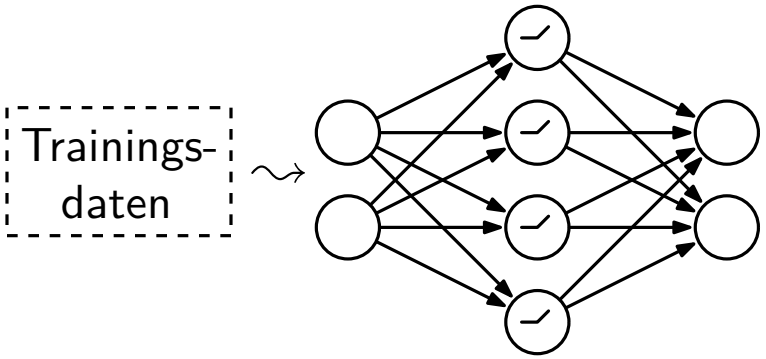
- NP-schwer [Breu, Kirkpatrick 1998]
- Überhaupt berechenbar?
  $\rightsquigarrow$ ja: polynomielles Ungleichungssystem

Problem 2: Training neuronaler Netze

Problem: TrainNN

Eingabe:	Netzwerk + Trainingsdaten
Finde:	Parameter, sodass Trainingsdaten möglichst gut abgebildet werden.

Beispiel:



Problem 2: Training neuronaler Netze

Problem: TrainNN

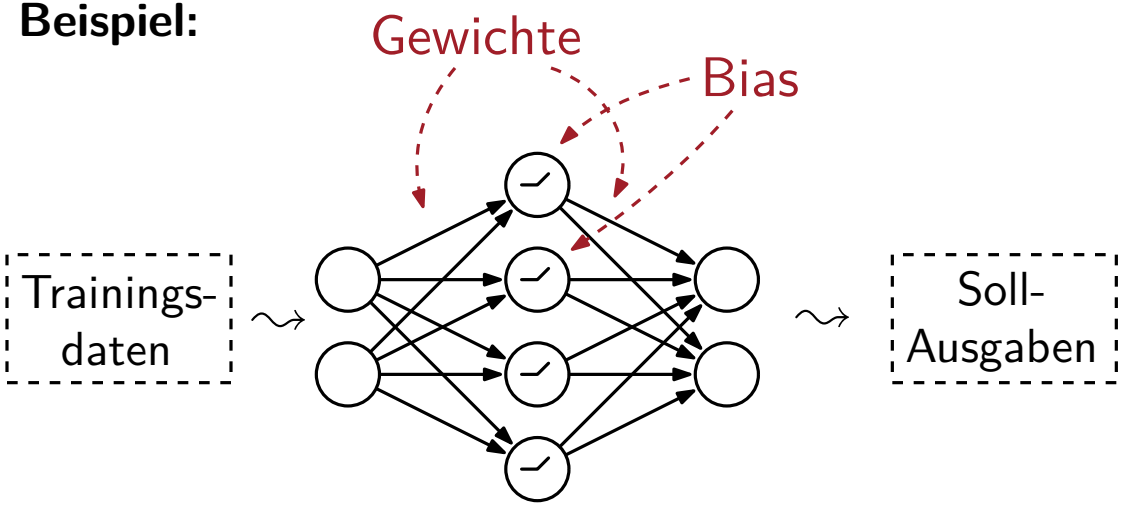
Eingabe: Netzwerk + Trainingsdaten

Finde: Parameter, sodass Trainingsdaten möglichst gut abgebildet werden.

Lösungen:

- Parameter aus $\mathbb{R}$ (Gewichte + Bias)
- jeder Trainingsdatenpunkt wird (exakt) berechnet

Beispiel:



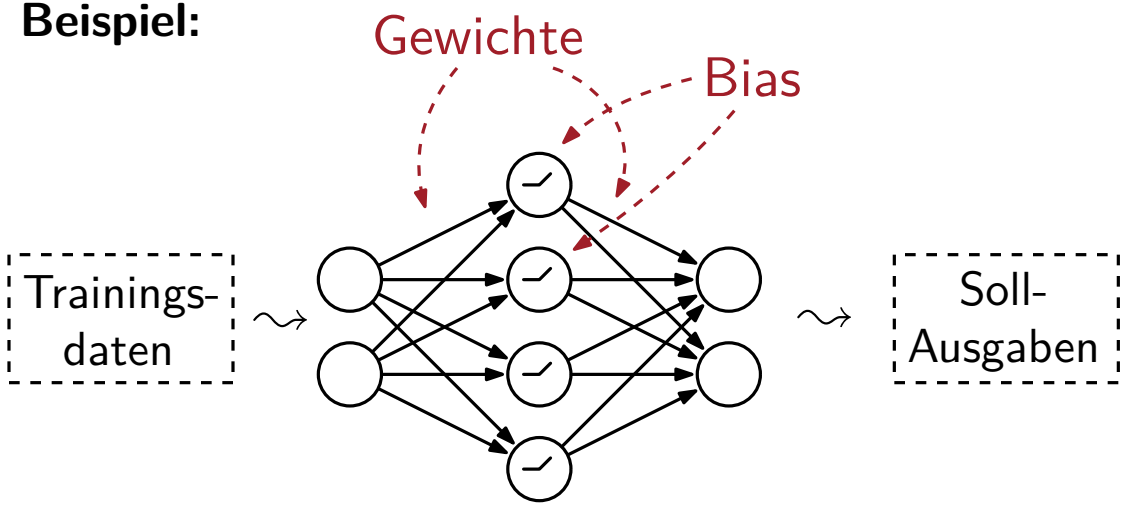
Problem 2: Training neuronaler Netze

Problem: TrainNN

Eingabe: Netzwerk + Trainingsdaten

Finde: Parameter, sodass Trainingsdaten möglichst gut abgebildet werden.

Beispiel:



Lösungen:

- Parameter aus $\mathbb{R}$ (Gewichte + Bias)
- jeder Trainingsdatenpunkt wird (exakt) berechnet

Komplexität:

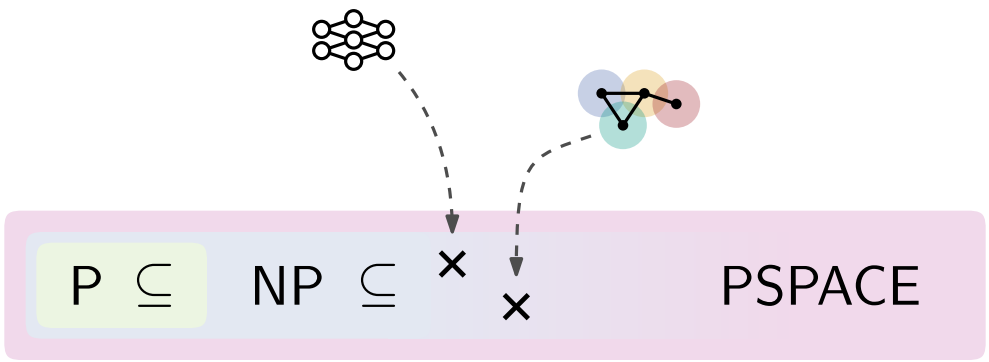
- NP-schwer [Blum, Rivest 1992]
- Überhaupt berechenbar?
  $\rightsquigarrow$ ja: polynomielles Ungleichungssystem

Komplexitätsklasse $\exists\mathbb{R}$

Wie schwierig sind  **und**  **wirklich?**

- beide sind NP-schwer
- beide sind in PSPACE

gleiche Lücke wie bei vielen (geometrischen) Problemen



Komplexitätsklasse $\exists\mathbb{R}$

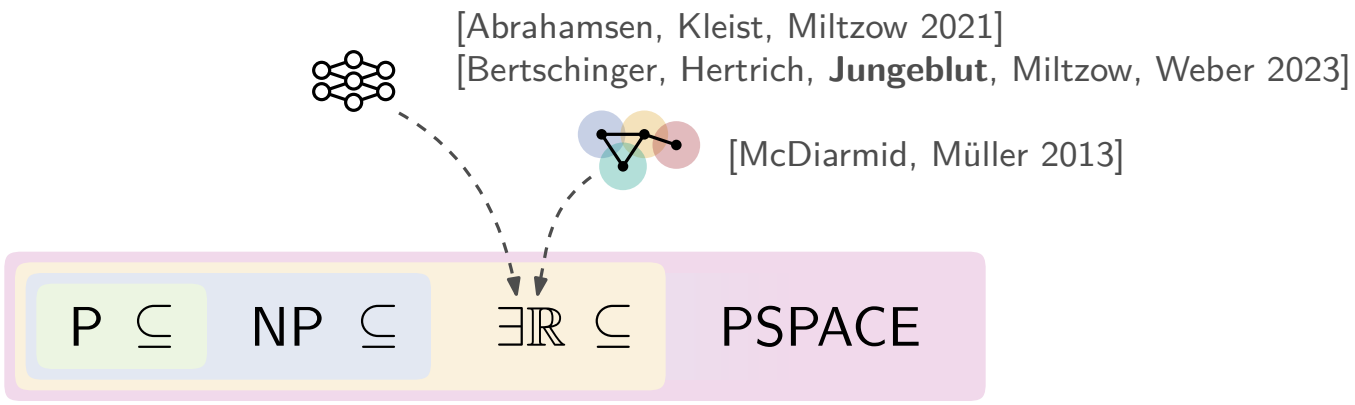
Wie schwierig sind  **und**  **wirklich?**

- beide sind NP-schwer
- beide sind in PSPACE

gleiche Lücke wie bei vielen (geometrischen) Problemen

Beide sind $\exists\mathbb{R}$-vollständig.

$\rightsquigarrow$ „gleich schwierig“



Komplexitätsklasse $\exists\mathbb{R}$

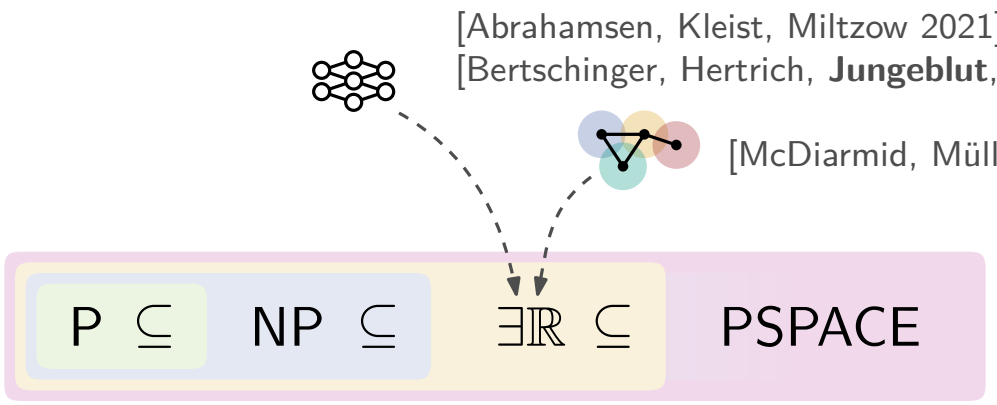
Wie schwierig sind  **und**  **wirklich?**

- beide sind NP-schwer
- beide sind in PSPACE

gleiche Lücke wie bei vielen (geometrischen) Problemen

Beide sind $\exists\mathbb{R}$-vollständig.

$\rightsquigarrow$ „gleich schwierig“



In meiner Arbeit:

- exakte Komplexität bekannter offener Probleme bestimmt
  - zugrundeliegene Geometrie ausgenutzt
  - Wissen über $\exists\mathbb{R}$ bot richtige Perspektive
- neue Beweistechniken entwickelt und bekannte verfeinert
  - wurden bereits in der Literatur angewandt
- natürliche Erweiterungen von $\exists\mathbb{R}$

Existenzielle Theorie der reellen Zahlen & $\exists\mathbb{R}$

Definition: ETR (**E**xistential **T**heory of the **R**eals)

Eingabe: Formel $\exists X, Y, \ldots \in \mathbb{R}\colon \varphi(X, Y, \ldots)$

Polynom(un)gleichungen

Frage: Ist die Formel wahr oder falsch?

Existenzielle Theorie der reellen Zahlen & $\exists\mathbb{R}$

Definition: ETR (**E**xistential **T**heory of the **R**eals)

Eingabe: Formel $\exists X, Y, \ldots \in \mathbb{R}\colon \varphi(X, Y, \ldots)$

Polynom(un)gleichungen

Frage: Ist die Formel wahr oder falsch?

Beispiel 1: $\exists X \in \mathbb{R}\colon X^2 < 0$

Existenzielle Theorie der reellen Zahlen & $\exists\mathbb{R}$

Definition: ETR (**E**xistential **T**heory of the **R**eals)

Eingabe: Formel $\exists X, Y, \ldots \in \mathbb{R}\colon \varphi(X, Y, \ldots)$

Polynom(un)gleichungen

Frage: Ist die Formel wahr oder falsch?

Beispiel 1: $\exists X \in \mathbb{R}\colon X^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Existenzielle Theorie der reellen Zahlen & $\exists\mathbb{R}$

Definition: ETR (**E**xistential **T**heory of the **R**eals)

Eingabe: Formel $\exists X, Y, \ldots \in \mathbb{R}\colon \varphi(X, Y, \ldots)$

Polynom(un)gleichungen

Frage: Ist die Formel wahr oder falsch?

Beispiel 1: $\exists X \in \mathbb{R}\colon X^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2: $\exists X, Y \in \mathbb{R}\colon \varphi(X, Y)$ mit

$$\varphi(X, Y) \equiv X^2 + Y^2 \leq 1 \\ \wedge\ Y \geq 2X^2 - 1$$

Existenzielle Theorie der reellen Zahlen & $\exists\mathbb{R}$

Definition: ETR (**E**xistential **T**heory of the **R**eals)

Eingabe: Formel $\exists X, Y, \ldots \in \mathbb{R}\colon \varphi(X, Y, \ldots)$

Polynom(un)gleichungen

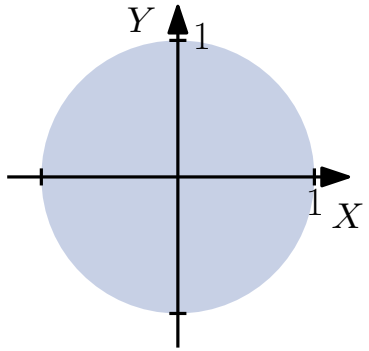
Frage: Ist die Formel wahr oder falsch?

Beispiel 1: $\exists X \in \mathbb{R}\colon X^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2: $\exists X, Y \in \mathbb{R}\colon \varphi(X, Y)$ mit

$$\varphi(X, Y) \equiv X^2 + Y^2 \leq 1 \wedge Y \geq 2X^2 - 1$$



Existenzielle Theorie der reellen Zahlen & $\exists\mathbb{R}$

Definition: ETR (**E**xistential **T**heory of the **R**eals)

Eingabe: Formel $\exists X, Y, \ldots \in \mathbb{R}\colon \varphi(X, Y, \ldots)$

Polynom(un)gleichungen

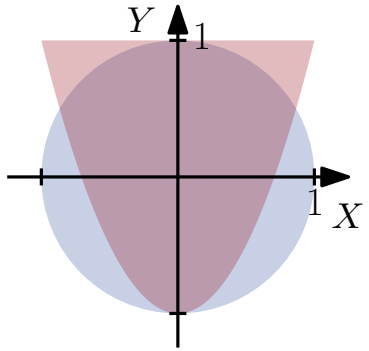
Frage: Ist die Formel wahr oder falsch?

Beispiel 1: $\exists X \in \mathbb{R}\colon X^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2: $\exists X, Y \in \mathbb{R}\colon \varphi(X, Y)$ mit

$$\varphi(X, Y) \equiv X^2 + Y^2 \leq 1 \wedge Y \geq 2X^2 - 1$$



Existenzielle Theorie der reellen Zahlen & $\exists\mathbb{R}$

Definition: ETR (**E**xistential **T**heory of the **R**eals)

Eingabe: Formel $\exists X, Y, \ldots \in \mathbb{R}\colon \varphi(X, Y, \ldots)$

Polynom(un)gleichungen

Frage: Ist die Formel wahr oder falsch?

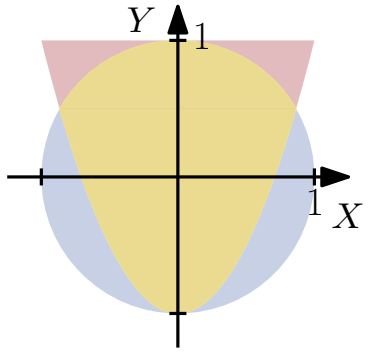
Beispiel 1: $\exists X \in \mathbb{R}\colon X^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2: $\exists X, Y \in \mathbb{R}\colon \varphi(X, Y)$ mit

$\varphi(X, Y) \equiv X^2 + Y^2 \leq 1$

$\wedge\ Y \geq 2X^2 - 1$ ✓



Existenzielle Theorie der reellen Zahlen & $\exists\mathbb{R}$

Definition: ETR (**E**xistential **T**heory of the **R**eals)

Eingabe: Formel $\exists X, Y, \ldots \in \mathbb{R}\colon \varphi(X, Y, \ldots)$

Polynom(un)gleichungen

Frage: Ist die Formel wahr oder falsch?

Wie schwierig ist ETR?

- überabzählbar viele Lösungskandidaten
- entscheidbar [Tarski 1951]

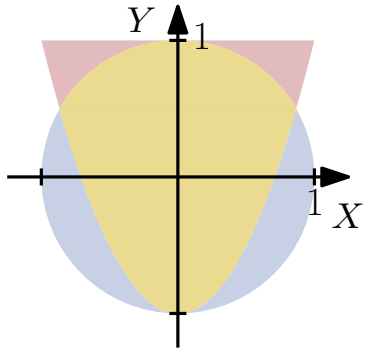
Beispiel 1: $\exists X \in \mathbb{R}\colon X^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2: $\exists X, Y \in \mathbb{R}\colon \varphi(X, Y)$ mit

$\varphi(X, Y) \equiv X^2 + Y^2 \leq 1$

$\wedge\ Y \geq 2X^2 - 1$ ✓



Existenzielle Theorie der reellen Zahlen & $\exists\mathbb{R}$

Definition: ETR (**E**xistential **T**heory of the **R**eals)

Eingabe: Formel $\exists X, Y, \ldots \in \mathbb{R}\colon \varphi(X, Y, \ldots)$

Polynom(un)gleichungen

Frage: Ist die Formel wahr oder falsch?

Wie schwierig ist ETR?

- überabzählbar viele Lösungskandidaten
- entscheidbar [Tarski 1951]
- ETR ist NP-schwer [Shor 1991]
- ETR ist in PSPACE [Canny 1988]

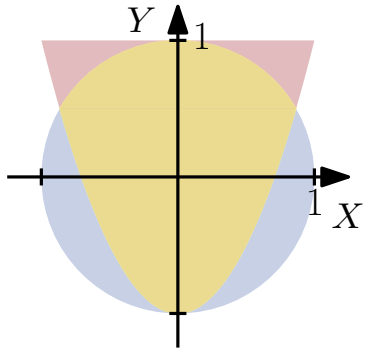
Beispiel 1: $\exists X \in \mathbb{R}\colon X^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2: $\exists X, Y \in \mathbb{R}\colon \varphi(X, Y)$ mit

$\varphi(X, Y) \equiv X^2 + Y^2 \leq 1$

$\wedge\ Y \geq 2X^2 - 1$ ✓



Existenzielle Theorie der reellen Zahlen & $\exists\mathbb{R}$

Definition: ETR (**E**xistential **T**heory of the **R**eals)

Eingabe: Formel $\exists X, Y, \ldots \in \mathbb{R}\colon \varphi(X, Y, \ldots)$

Polynom(un)gleichungen

Frage: Ist die Formel wahr oder falsch?

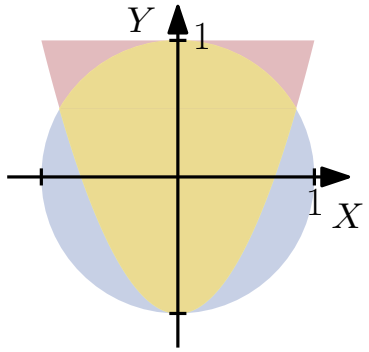
Beispiel 1: $\exists X \in \mathbb{R}\colon X^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2: $\exists X, Y \in \mathbb{R}\colon \varphi(X, Y)$ mit

$\varphi(X, Y) \equiv X^2 + Y^2 \leq 1$

$\wedge\ Y \geq 2X^2 - 1$ ✓



Wie schwierig ist ETR?

- überabzählbar viele Lösungskandidaten
- entscheidbar [Tarski 1951]
- ETR ist NP-schwer [Shor 1991]
- ETR ist in PSPACE [Canny 1988]

Definition:
$\exists\mathbb{R}$ enthält alle Probleme, die in polynomieller Zeit auf ETR reduziert werden können.

$\mathrm{P} \subseteq \mathrm{NP} \subseteq \exists\mathbb{R} \subseteq \mathrm{PSPACE}$

ETR (per Definition)

Bedeutung von $\exists\mathbb{R}$-Vollständigkeit

Wahre Schwierigkeit?

- Schwierigkeit nur relativ zu ETR
- nicht: „irgendwo“ zwischen NP und PSPACE
  sondern: „gleich schwierig“ wie ETR

Bedeutung von $\exists\mathbb{R}$-Vollständigkeit

Wahre Schwierigkeit?

- Schwierigkeit nur relativ zu ETR
- nicht: „irgendwo“ zwischen NP und PSPACE
  sondern: „gleich schwierig“ wie ETR

Perspektivwechsel

- im Kern: reelle algebraische Geometrie
- algorithmische Fortschritte am ehesten
  durch *mathematische* Durchbrüche zu
  erwarten

Bedeutung von $\exists\mathbb{R}$-Vollständigkeit

Wahre Schwierigkeit?

- Schwierigkeit nur relativ zu ETR
- nicht: „irgendwo“ zwischen NP und PSPACE
  sondern: „gleich schwierig“ wie ETR

Genauigkeit

- Lösungen doppelt exponenziell groß
  $\rightsquigarrow$ exponenziell viele Bits
- algebraische/topologische Universalität

Perspektivwechsel

- im Kern: reelle algebraische Geometrie
- algorithmische Fortschritte am ehesten
  durch *mathematische* Durchbrüche zu
  erwarten

Bedeutung von $\exists\mathbb{R}$-Vollständigkeit

Wahre Schwierigkeit?

- Schwierigkeit nur relativ zu ETR
- nicht: „irgendwo“ zwischen NP und PSPACE
  sondern: „gleich schwierig“ wie ETR

Genauigkeit

- Lösungen doppelt exponenziell groß
  $\rightsquigarrow$ exponenziell viele Bits
- algebraische/topologische Universalität

Perspektivwechsel

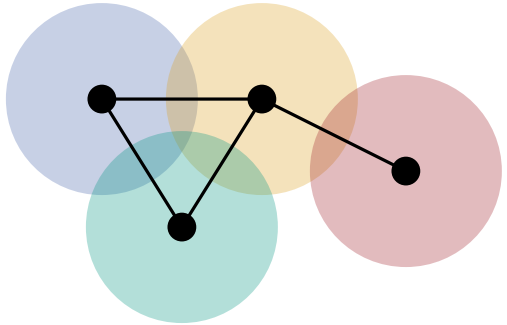
- im Kern: reelle algebraische Geometrie
- algorithmische Fortschritte am ehesten durch *mathematische* Durchbrüche zu erwarten

Lösbarkeit in der Praxis

- NP-vollständig $\rightsquigarrow$ SAT-Solver
- keine solchen Universallöser für ETR
  aber: gute Heuristiken für spezielle Probleme

Beitrag und Ergebnisse

Hyperbolische Geometrie

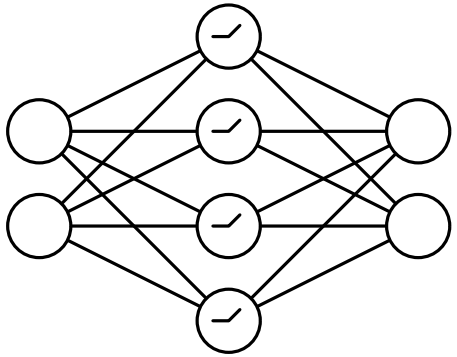


$\exists\mathbb{R}$-Schwere in der hyperbolischen Ebene

[EuroCG 2023]

[Bieker, Bläsius, Dohse, **Jungeblut**]

Training neuronaler Netze

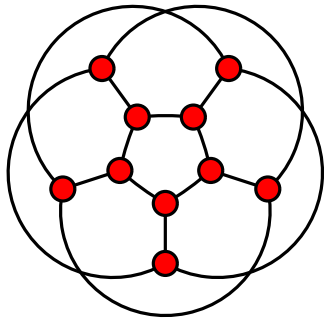


$\exists\mathbb{R}$-vollständig unter realistischen Annahmen

[NeurIPS 2023]

[Bertschinger, Hertrich, **Jungeblut**, Miltzow, Weber]

Lombardi-Zeichnungen

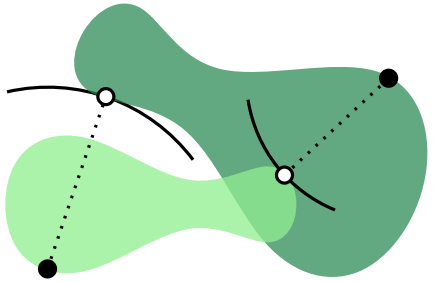


Graphvisualisierung unter geometrischen Nebenbedingungen

[GD 2023]

[**Jungeblut**]

Hausdorff-Distanz



Distanzmaß für Mengen
$\rightsquigarrow \forall\exists\mathbb{R}$-vollständig

[SoCG 2022] [DCG 2024]

[**Jungeblut**, Kleist, Miltzow]

Beitrag und Ergebnisse

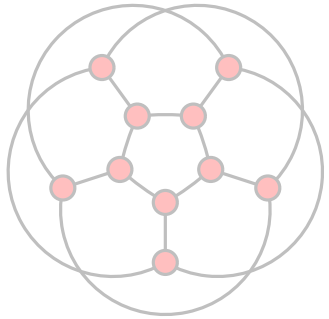
Hyperbolische Geometrie

$\exists\mathbb{R}$-Schwere in der hyperbolischen Ebene

[EuroCG 2023]

[Bieker, Bläsius, Dohse, **Jungeblut**]

Lombardi-Zeichnungen



Graphvisualisierung unter geometrischen Nebenbedingungen

[GD 2023]

[**Jungeblut**]

STRETCHABILITY-Problem:

- $\exists\mathbb{R}$-vollständig in $\mathbb{R}^2$ [Mnëv 1988]
- Startpunkt für viele $\exists\mathbb{R}$-Schwerebeweise

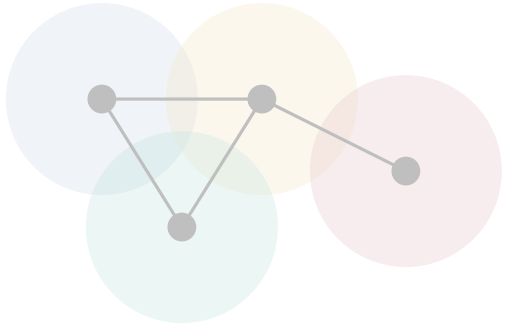
Theorem:
STRETCHABILITY ist $\exists\mathbb{R}$-vollständig in der hyperbolischen Ebene $\mathbb{H}^2$.

Framework:
$\exists\mathbb{R}$-Vollständigkeit von euklidischer in die hyperbolische Ebene übertragen.

$\rightsquigarrow$  $\exists\mathbb{R}$-vollständig in $\mathbb{H}^2$

Beitrag und Ergebnisse

Hyperbolische Geometrie



$\exists\mathbb{R}$-Schwere in der hyperbolischen Ebene

[EuroCG 2023]

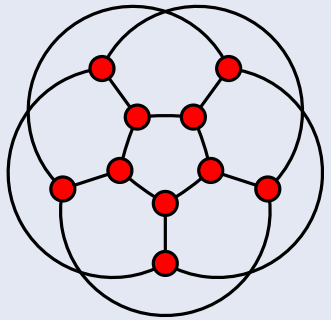
[Bieker, Bläsius, Dohse, **Jungeblut**]

Knoten: Punkte aus $\mathbb{R}^2$

Kanten: Kreisbögen
gleiche Winkel um jeden Knoten

$\rightsquigarrow$ Symmetrien des Graphs gut erkennbar

Lombardi-Zeichnungen



Graphvisualisierung unter geometrischen Nebenbedingungen

[GD 2023]

[**Jungeblut**]

Theorem:
Lombardi-Zeichnen ist $\exists\mathbb{R}$-vollständig.

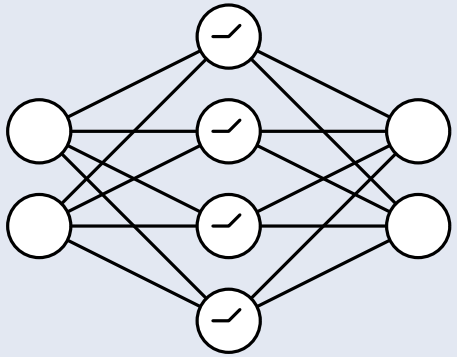
- erstes Komplexitätsresultat
- eleganter Umweg über hyperbolisches Stretchability

Beitrag und Ergebnisse

Theorem:
TrainNN ist $\exists\mathbb{R}$-vollständig.
Selbst unter realistischen Annahmen.

Algebraische Universalität:
Lösungen können beliebig komplizierte irrationale Zahlen enthalten.

Training neuronaler Netze

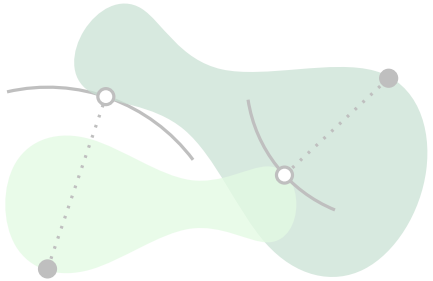


$\exists\mathbb{R}$-vollständig unter realistischen Annahmen

[NeurIPS 2023]

[Bertschinger, Hertrich, **Jungeblut**, Miltzow, Weber]

Hausdorff-Distanz



Distanzmaß für Mengen
$\rightsquigarrow \forall\exists\mathbb{R}$-vollständig

[SoCG 2022] [DCG 2024]

[**Jungeblut**, Kleist, Miltzow]

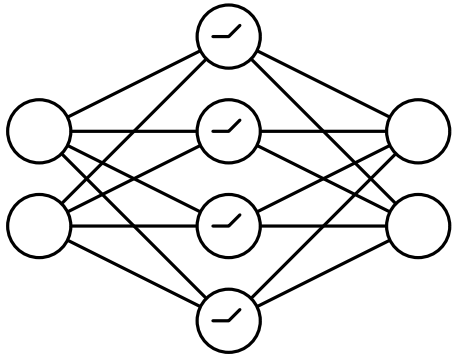
Beitrag und Ergebnisse

Wie „ähnlich“ sind sich zwei Mengen?

Theorem:
Die Hausdorff-Distanz zu berechnen ist $\forall\exists\mathbb{R}$-vollständig.

- $\forall\exists\mathbb{R}$: reelle polynomielle Hierarchie
  $\rightsquigarrow$ noch schwieriger als $\exists\mathbb{R}$
- erster $\forall\exists\mathbb{R}$-Vollständigkeitbeweis für ein natürliches Problem

Training neuronaler Netze

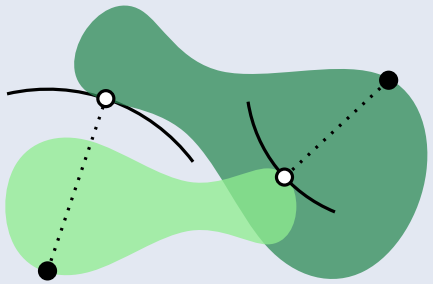


$\exists\mathbb{R}$-vollständig unter realistischen Annahmen

[NeurIPS 2023]

[Bertschinger, Hertrich, **Jungeblut**, Miltzow, Weber]

Hausdorff-Distanz



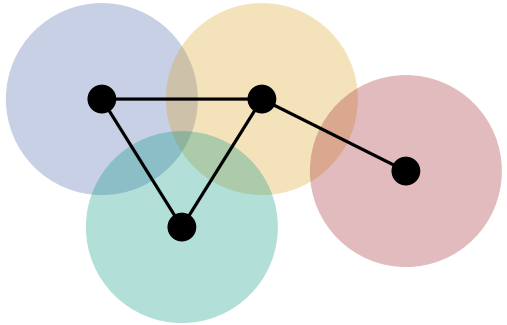
Distanzmaß für Mengen
$\rightsquigarrow$ $\forall\exists\mathbb{R}$-vollständig

[SoCG 2022] [DCG 2024]

[**Jungeblut**, Kleist, Miltzow]

Beitrag und Ergebnisse

Hyperbolische Geometrie

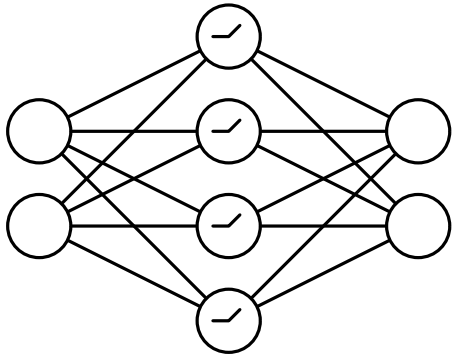


$\exists\mathbb{R}$-Schwere in der hyperbolischen Ebene

[EuroCG 2023]

[Bieker, Bläsius, Dohse, **Jungeblut**]

Training neuronaler Netze

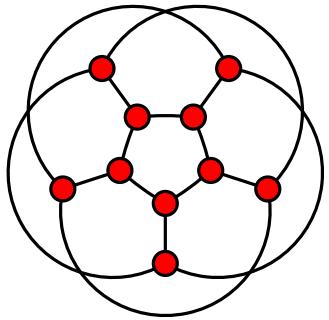


$\exists\mathbb{R}$-vollständig unter realistischen Annahmen

[NeurIPS 2023]

[Bertschinger, Hertrich, **Jungeblut**, Miltzow, Weber]

Lombardi-Zeichnungen

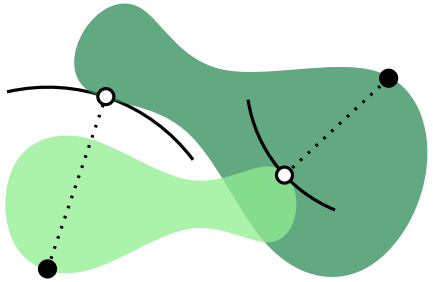


Graphvisualisierung unter geometrischen Nebenbedingungen

[GD 2023]

[**Jungeblut**]

Hausdorff-Distanz

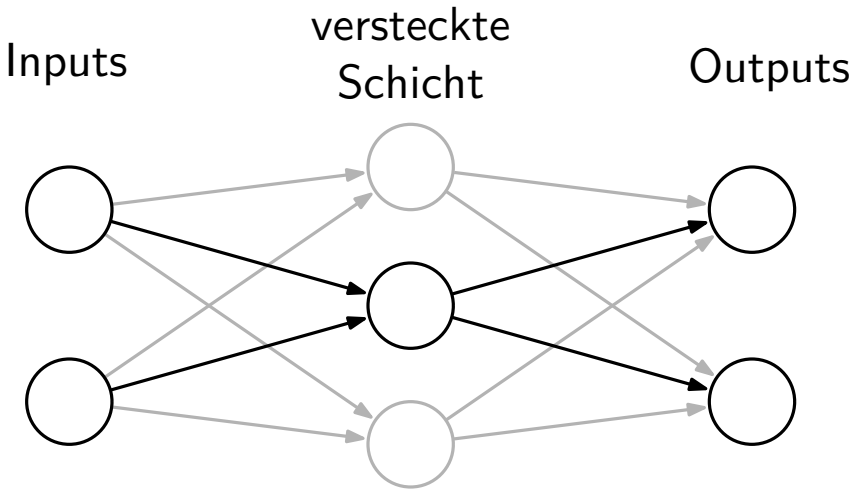


Distanzmaß für Mengen
$\rightsquigarrow \forall\exists\mathbb{R}$-vollständig

[SoCG 2022] [DCG 2024]

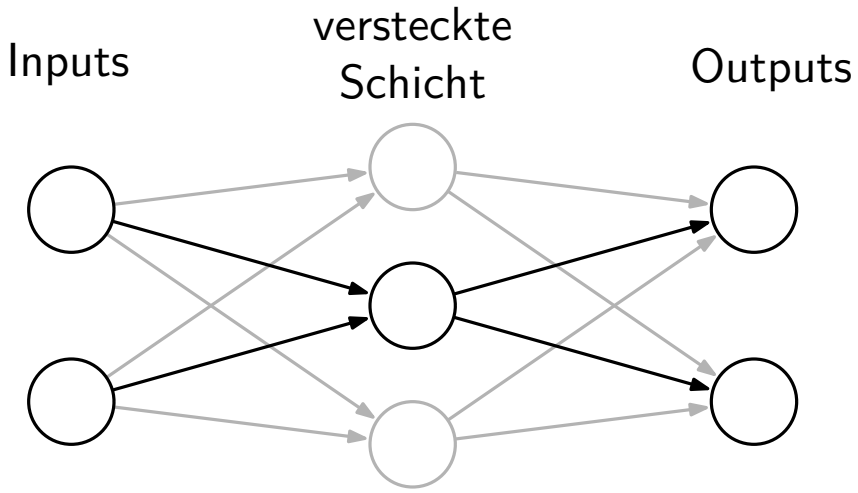
[**Jungeblut**, Kleist, Miltzow]

Neuronale Netze



Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Neuronale Netze



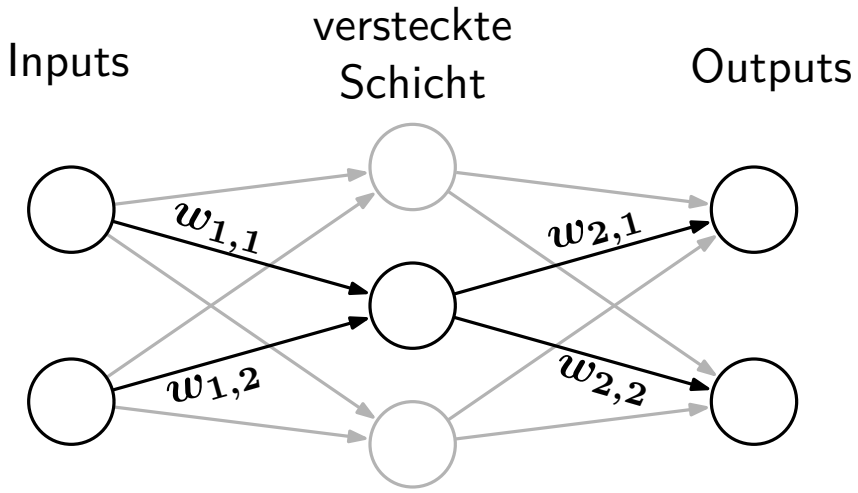
Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Netz berechnet eine Funktion:

$$f_\Theta : \mathbb{R}^2 \to \mathbb{R}^2$$

Θ: Parameter des Netzes

Neuronale Netze



Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

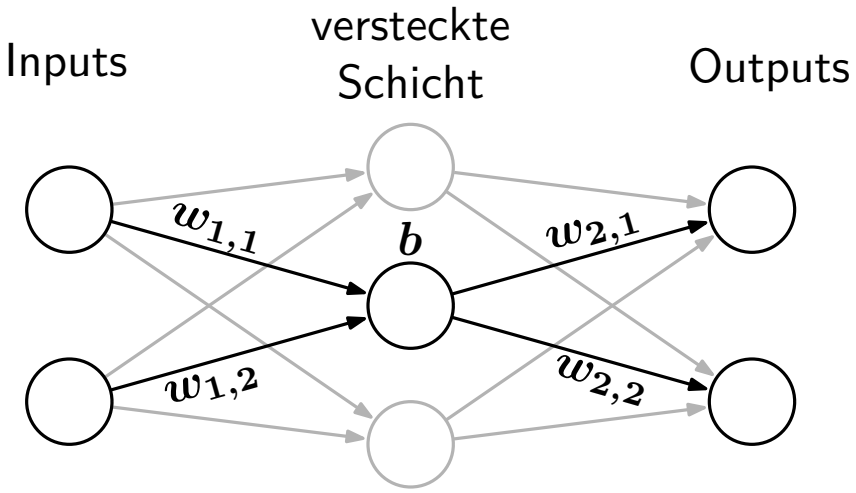
Gewichte: pro Kante

Netz berechnet eine Funktion:

$$f_\Theta : \mathbb{R}^2 \to \mathbb{R}^2$$

Θ: Parameter des Netzes

Neuronale Netze



Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Gewichte: pro Kante

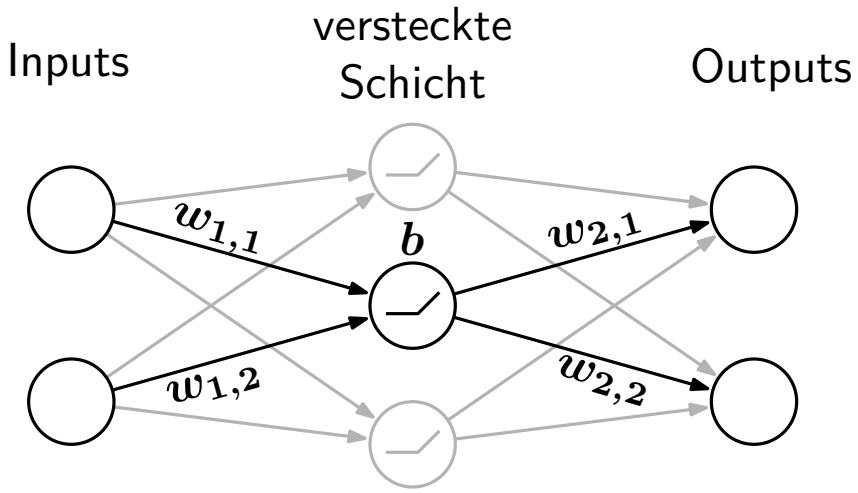
Bias: pro verstecktem Neuron

Netz berechnet eine Funktion:

$$f_\Theta : \mathbb{R}^2 \to \mathbb{R}^2$$

Θ: Parameter des Netzes

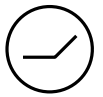
Neuronale Netze



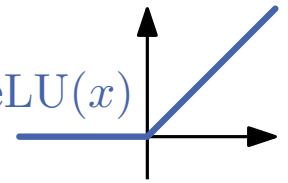
Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Gewichte: pro Kante

Bias: pro verstecktem Neuron

Aktivierungsfunktion: $\mathrm{ReLU} =$ 

$$\mathrm{ReLU}\colon \mathbb{R} \to \mathbb{R}$$
$$x \mapsto \max\{0, x\}$$

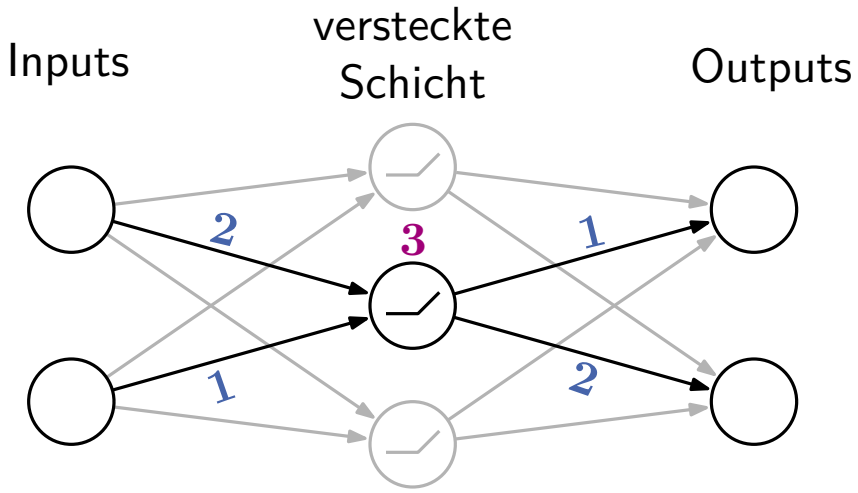


Netz berechnet eine Funktion:

$$f_\Theta\colon \mathbb{R}^2 \to \mathbb{R}^2$$

Θ: Parameter des Netzes

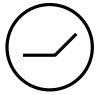
Neuronale Netze



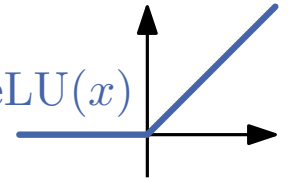
Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Gewichte: pro Kante

Bias: pro verstecktem Neuron

Aktivierungsfunktion: $\mathrm{ReLU} =$ 

$$\mathrm{ReLU}\colon \mathbb{R} \to \mathbb{R}$$
$$x \mapsto \max\{0, x\}$$

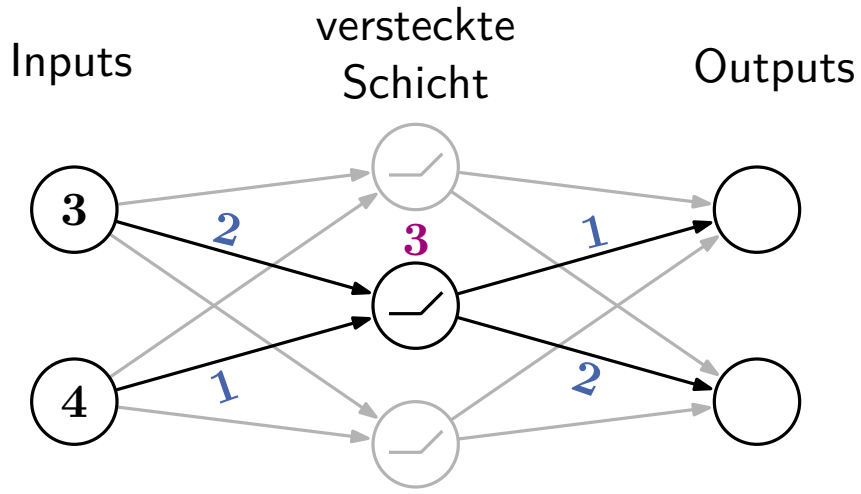


Netz berechnet eine Funktion:

$$f_\Theta\colon \mathbb{R}^2 \to \mathbb{R}^2$$

Θ: Parameter des Netzes

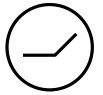
Neuronale Netze



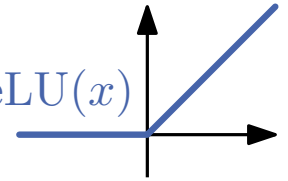
Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Gewichte: pro Kante

Bias: pro verstecktem Neuron

Aktivierungsfunktion: $\mathrm{ReLU} =$ 

$$\mathrm{ReLU}\colon \mathbb{R} \to \mathbb{R}$$
$$x \mapsto \max\{0, x\}$$

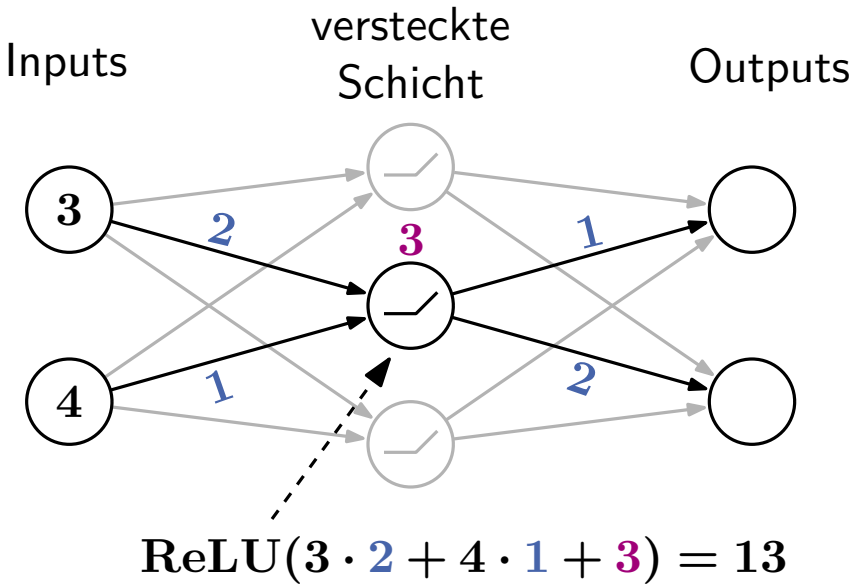


Netz berechnet eine Funktion:

$$f_\Theta\colon \mathbb{R}^2 \to \mathbb{R}^2$$

Θ: Parameter des Netzes

Neuronale Netze



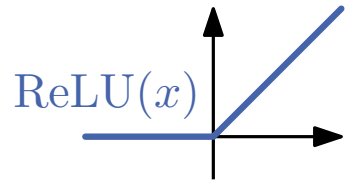
Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Gewichte: pro Kante

Bias: pro verstecktem Neuron

Aktivierungsfunktion: $\mathrm{ReLU} =$

$$\mathrm{ReLU}\colon \mathbb{R} \to \mathbb{R}$$
$$x \mapsto \max\{0, x\}$$

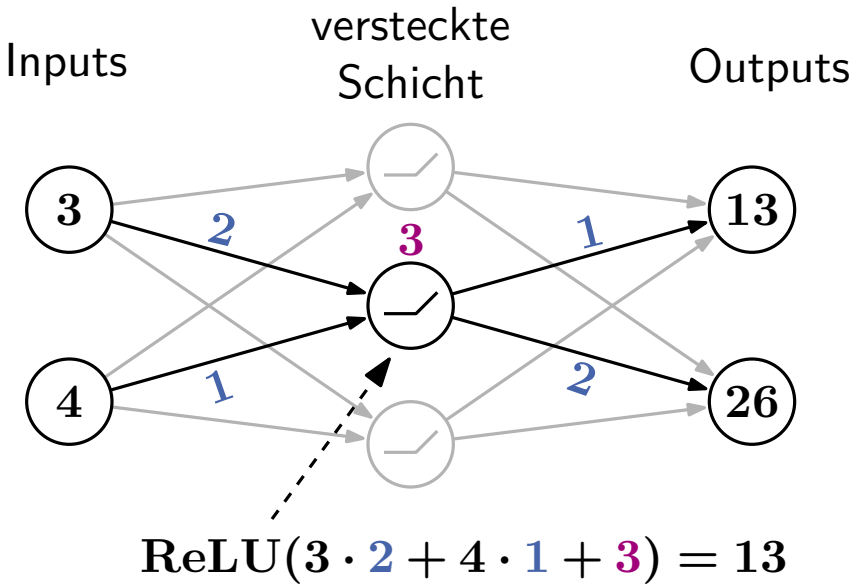


Netz berechnet eine Funktion:

$$f_\Theta\colon \mathbb{R}^2 \to \mathbb{R}^2$$

Θ: Parameter des Netzes

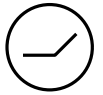
Neuronale Netze



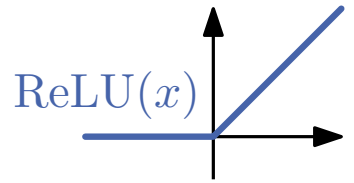
Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Gewichte: pro Kante

Bias: pro verstecktem Neuron

Aktivierungsfunktion: $\mathrm{ReLU} =$ 

$$\mathrm{ReLU}\colon \mathbb{R} \to \mathbb{R}$$
$$x \mapsto \max\{0, x\}$$

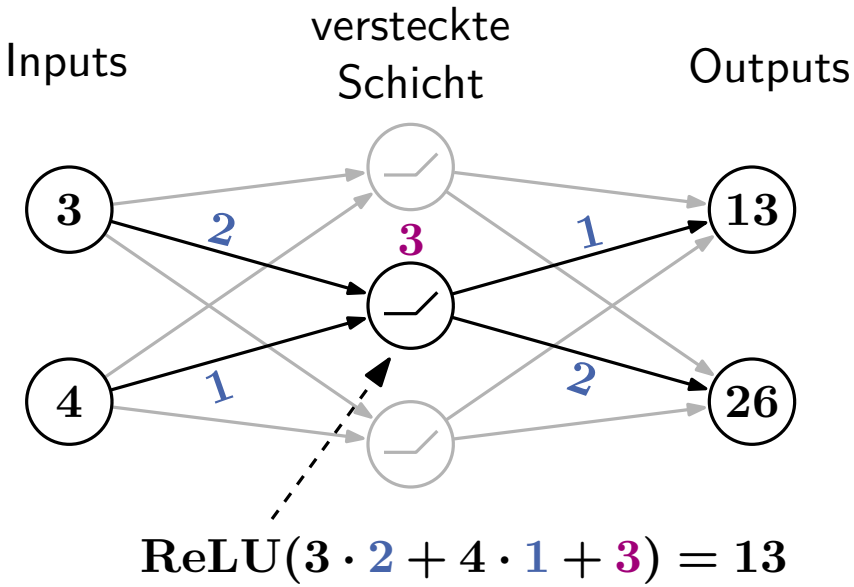


Netz berechnet eine Funktion:

$$f_\Theta\colon \mathbb{R}^2 \to \mathbb{R}^2$$

Θ: Parameter des Netzes

Neuronale Netze



Netz berechnet eine Funktion:

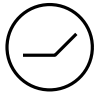
$$f_\Theta : \mathbb{R}^2 \to \mathbb{R}^2$$

Θ: Parameter des Netzes

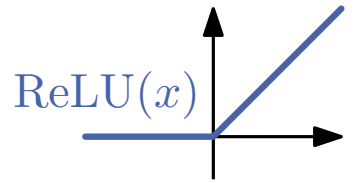
Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Gewichte: pro Kante

Bias: pro verstecktem Neuron

Aktivierungsfunktion: $\mathrm{ReLU} =$ 

$$\mathrm{ReLU} : \mathbb{R} \to \mathbb{R}$$
$$x \mapsto \max\{0, x\}$$



Trainingsproblem: TrainNN

Eingabe: n Datenpunkte: $(x_i, y_i) \in \mathbb{R}^2 \times \mathbb{R}^2$
Eingabe, Soll-Ausgabe

Frage: Existieren Parameter Θ, sodass für alle Datenpunkte gilt: $f_\Theta(x_i) = y_i$

Bisherige Ergebnisse & Beitrag

Literatur: (Auswahl)

TrainNN ist ...

- NP-schwer [Blum, Rivest 1992]

Bisherige Ergebnisse & Beitrag

Literatur: (Auswahl)

TrainNN ist ...

- NP-schwer [Blum, Rivest 1992]
- in NP für einzelnes Output-Neuron [Arora et al. 2018]

  Frage: Verallgemeinerbar auf kompliziertere Architekturen?

Bisherige Ergebnisse & Beitrag

Literatur: (Auswahl)

TrainNN ist ...

- NP-schwer [Blum, Rivest 1992]
- in NP für einzelnes Output-Neuron [Arora et al. 2018]

  Frage: Verallgemeinerbar auf kompliziertere Architekturen?

- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

  Jedoch unter unrealistischen Annahmen:
  - keine Aktivierungsfunktion
  - speziell gewählte Architekturen, die besonders schwierig zu trainieren sind

Bisherige Ergebnisse & Beitrag

Literatur: (Auswahl)

TrainNN ist ...

- NP-schwer [Blum, Rivest 1992]
- in NP für einzelnes Output-Neuron [Arora et al. 2018]

  Frage: Verallgemeinerbar auf kompliziertere Architekturen?

- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

  Jedoch unter unrealistischen Annahmen:
  - keine Aktivierungsfunktion
  - speziell gewählte Architekturen, die besonders schwierig zu trainieren sind

Theorem: TrainNN ist $\exists\mathbb{R}$-vollständig.

Bisherige Ergebnisse & Beitrag

Literatur: (Auswahl)

TrainNN ist ...

- NP-schwer [Blum, Rivest 1992]
- in NP für einzelnes Output-Neuron [Arora et al. 2018]

  Frage: Verallgemeinerbar auf kompliziertere Architekturen?

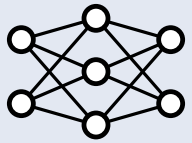
- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

  Jedoch unter unrealistischen Annahmen:
  - keine Aktivierungsfunktion
  - speziell gewählte Architekturen, die besonders schwierig zu trainieren sind

Theorem: TrainNN ist $\exists\mathbb{R}$-vollständig.

Sogar folgender Spezialfall:

- nur eine versteckte Schicht



Bisherige Ergebnisse & Beitrag

Literatur: (Auswahl)

TrainNN ist ...

- NP-schwer [Blum, Rivest 1992]
- in NP für einzelnes Output-Neuron [Arora et al. 2018]

  Frage: Verallgemeinerbar auf kompliziertere Architekturen?

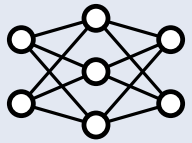
- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

  Jedoch unter unrealistischen Annahmen:
  - keine Aktivierungsfunktion
  - speziell gewählte Architekturen, die besonders schwierig zu trainieren sind

Theorem: TrainNN ist $\exists\mathbb{R}$-vollständig.

Sogar folgender Spezialfall:

- nur eine versteckte Schicht
- nur zwei Inputs und Outputs



Bisherige Ergebnisse & Beitrag

Literatur: (Auswahl)

TrainNN ist ...

- NP-schwer [Blum, Rivest 1992]
- in NP für einzelnes Output-Neuron [Arora et al. 2018]

  Frage: Verallgemeinerbar auf kompliziertere Architekturen?

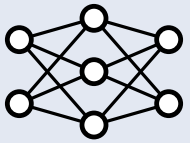
- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

  Jedoch unter unrealistischen Annahmen:
  - keine Aktivierungsfunktion
  - speziell gewählte Architekturen, die besonders schwierig zu trainieren sind

Theorem: TrainNN ist $\exists\mathbb{R}$-vollständig.

Sogar folgender Spezialfall:

- nur eine versteckte Schicht
- nur zwei Inputs und Outputs
- vollständig verbunden



Bisherige Ergebnisse & Beitrag

Literatur: (Auswahl)

TrainNN ist ...

- NP-schwer [Blum, Rivest 1992]
- in NP für einzelnes Output-Neuron [Arora et al. 2018]

  Frage: Verallgemeinerbar auf kompliziertere Architekturen?

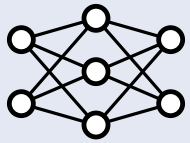
- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

  Jedoch unter unrealistischen Annahmen:
  - keine Aktivierungsfunktion
  - speziell gewählte Architekturen, die besonders schwierig zu trainieren sind

Theorem: TrainNN ist $\exists\mathbb{R}$-vollständig.

Sogar folgender Spezialfall:

- nur eine versteckte Schicht
- nur zwei Inputs und Outputs
- vollständig verbunden
- ...



Bisherige Ergebnisse & Beitrag

Literatur: (Auswahl)

TrainNN ist ...

- NP-schwer [Blum, Rivest 1992]
- in NP für einzelnes Output-Neuron [Arora et al. 2018]

  Frage: Verallgemeinerbar auf kompliziertere Architekturen?

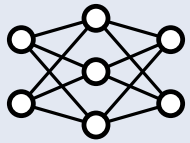
- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

  Jedoch unter unrealistischen Annahmen:
  - keine Aktivierungsfunktion
  - speziell gewählte Architekturen, die besonders schwierig zu trainieren sind

Theorem: TrainNN ist $\exists\mathbb{R}$-vollständig.

Sogar folgender Spezialfall:

- nur eine versteckte Schicht
- nur zwei Inputs und Outputs
- vollständig verbunden
- ...



✗ $\exists\mathbb{R}$-schwer schon bei zwei Outputs

(Annahme: NP $\neq \exists\mathbb{R}$)

Bisherige Ergebnisse & Beitrag

Literatur: (Auswahl)

TrainNN ist ...

- NP-schwer [Blum, Rivest 1992]
- in NP für einzelnes Output-Neuron [Arora et al. 2018]

  Frage: Verallgemeinerbar auf kompliziertere Architekturen?

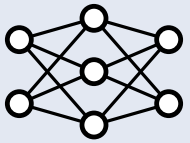
- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

  Jedoch unter unrealistischen Annahmen:
  - keine Aktivierungsfunktion
  - speziell gewählte Architekturen, die besonders schwierig zu trainieren sind

Theorem: TrainNN ist $\exists\mathbb{R}$-vollständig.

Sogar folgender Spezialfall:

- nur eine versteckte Schicht
- nur zwei Inputs und Outputs
- vollständig verbunden
- ...



✗ $\exists\mathbb{R}$-schwer schon bei zwei Outputs (Annahme: NP $\neq \exists\mathbb{R}$)

✓ realistische Annahmen

Beweis (Skizze)

Theorem: $\textsc{TrainNN}$ ist $\exists\mathbb{R}$-vollständig.

Schritt 1: $\textsc{TrainNN}$ ist in $\exists\mathbb{R}$
„höchstens“ so schwierig wie ETR

Schritt 2: $\textsc{TrainNN}$ ist $\exists\mathbb{R}$-schwer
„mindestens“ so schwierig wie ETR

Beweis (Skizze)

Theorem: $\textsc{TrainNN}$ ist $\exists\mathbb{R}$-vollständig.

Schritt 1: $\textsc{TrainNN}$ ist in $\exists\mathbb{R}$
„höchstens“ so schwierig wie ETR

Reduktion: $\textsc{TrainNN} \rightsquigarrow \mathrm{ETR}$

Schritt 2: $\textsc{TrainNN}$ ist $\exists\mathbb{R}$-schwer
„mindestens“ so schwierig wie ETR

Reduktion: $\mathrm{ETR} \rightsquigarrow \textsc{TrainNN}$

Beweis (Skizze)

Theorem: $\textsc{TrainNN}$ ist $\exists\mathbb{R}$-vollständig.

Schritt 1: $\textsc{TrainNN}$ ist in $\exists\mathbb{R}$
„höchstens“ so schwierig wie ETR

Reduktion: $\textsc{TrainNN} \rightsquigarrow \mathrm{ETR}$

Frage: Existieren Parameter Θ, sodass für alle Datenpunkte gilt: $f_\Theta(x_i) = y_i$

Äquivalente ETR-Formel:

$$\exists\, \Theta = (w_1, \ldots, b_1, \ldots) \in \mathbb{R}\colon \bigwedge_{i=1}^{n} f_\Theta(x_i) = y_i$$

als Polynomgleichung formulieren

$\rightsquigarrow \textsc{TrainNN} \in \exists\mathbb{R}$ □

Schritt 2: $\textsc{TrainNN}$ ist $\exists\mathbb{R}$-schwer
„mindestens“ so schwierig wie ETR

Reduktion: $\mathrm{ETR} \rightsquigarrow \textsc{TrainNN}$

Beweis (Skizze)

Theorem: $\textsc{TrainNN}$ ist $\exists\mathbb{R}$-vollständig.

Schritt 1: $\textsc{TrainNN}$ ist in $\exists\mathbb{R}$
„höchstens“ so schwierig wie ETR

Reduktion: $\textsc{TrainNN} \rightsquigarrow \mathrm{ETR}$

Frage: Existieren Parameter Θ, sodass für alle Datenpunkte gilt: $f_\Theta(x_i) = y_i$

Äquivalente ETR-Formel:

$$\exists\, \Theta = (w_1, \ldots, b_1, \ldots) \in \mathbb{R}\colon \bigwedge_{i=1}^{n} f_\Theta(x_i) = y_i$$

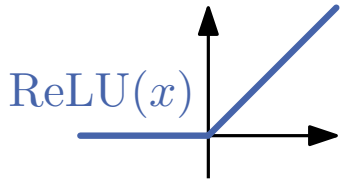
als Polynomgleichung formulieren

$\rightsquigarrow \textsc{TrainNN} \in \exists\mathbb{R}$ □

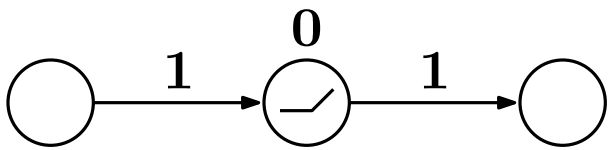
Schritt 2: $\textsc{TrainNN}$ ist $\exists\mathbb{R}$-schwer
„mindestens“ so schwierig wie ETR

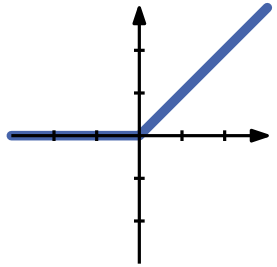
Reduktion: $\mathrm{ETR} \rightsquigarrow \textsc{TrainNN}$

Geometrisch:
Nutzt „Form“ der ReLU-Funktion.

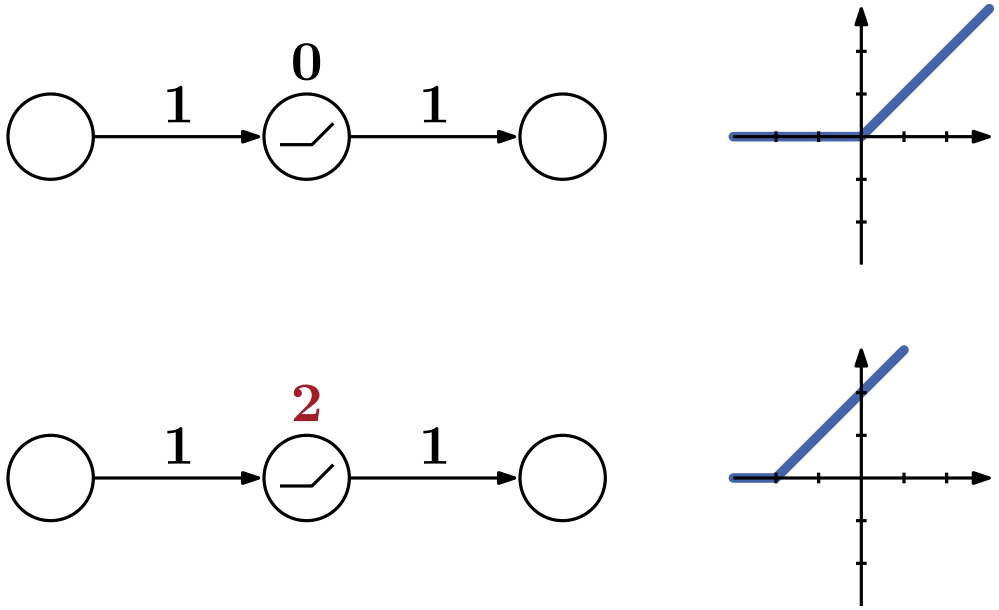


Geometrie von ReLU-Netzwerken

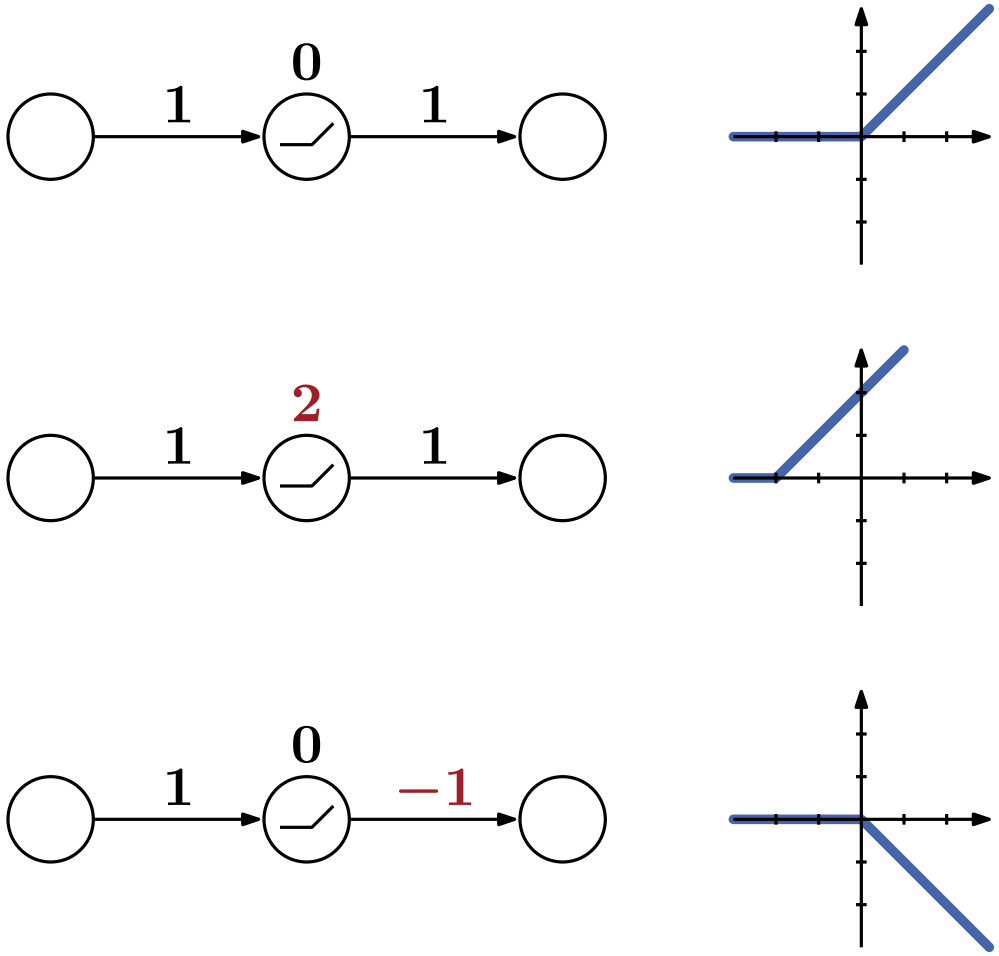




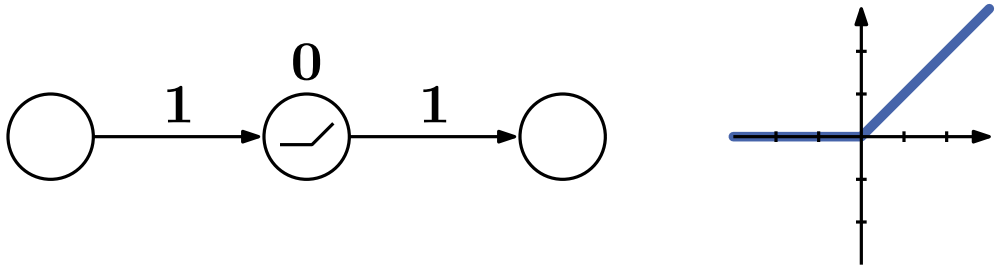
Geometrie von ReLU-Netzwerken

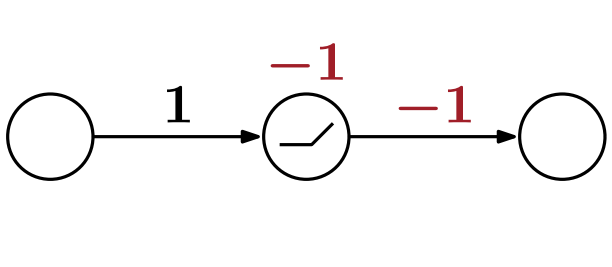


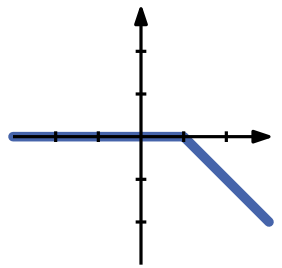
Geometrie von ReLU-Netzwerken

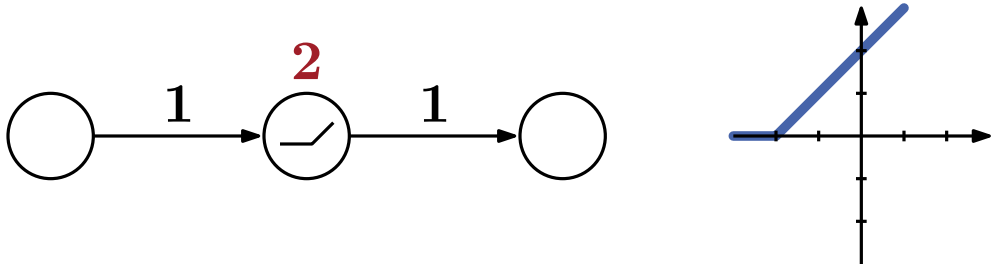


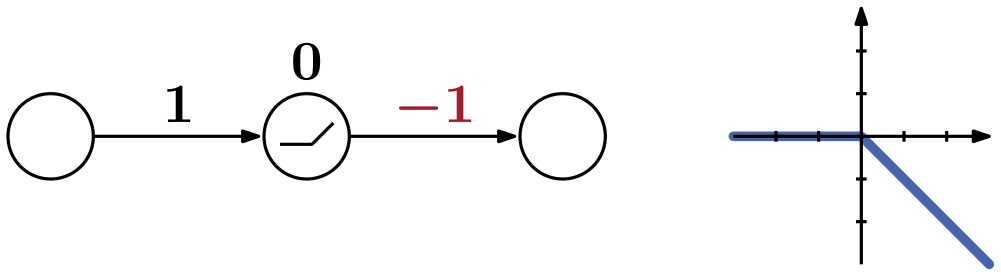
Geometrie von ReLU-Netzwerken



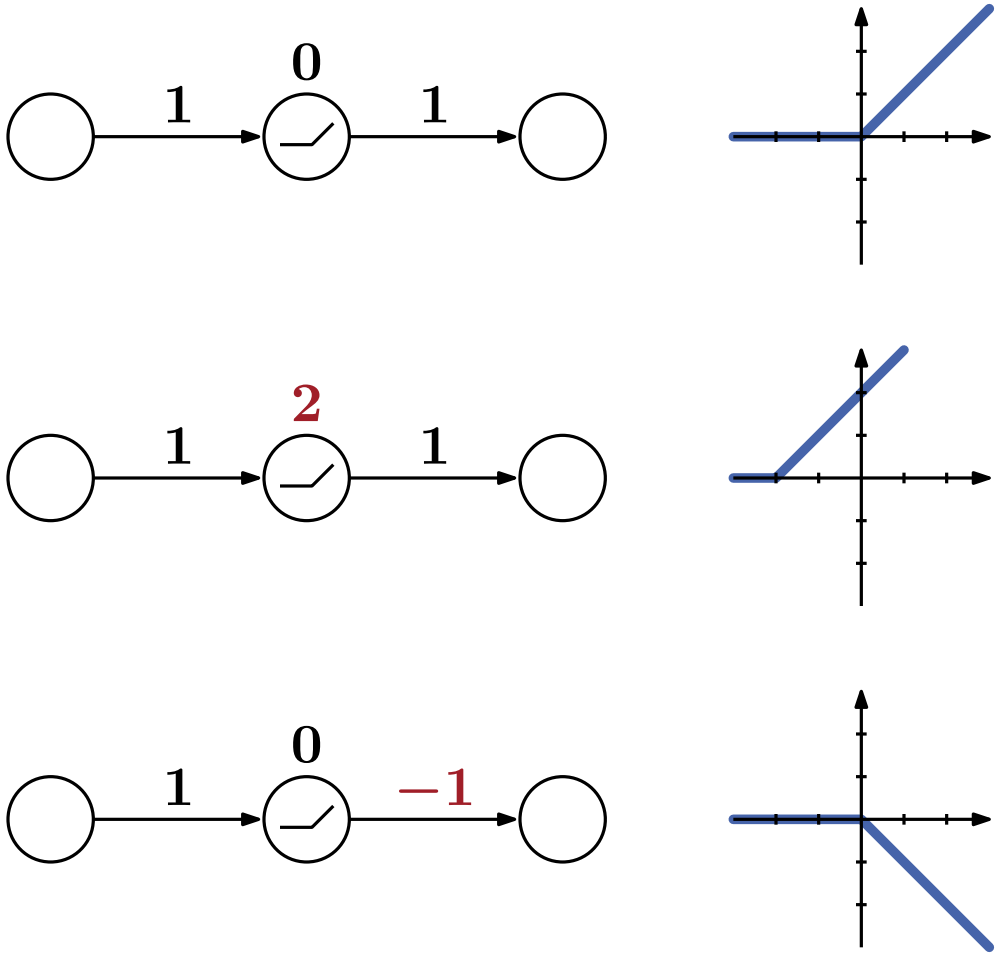


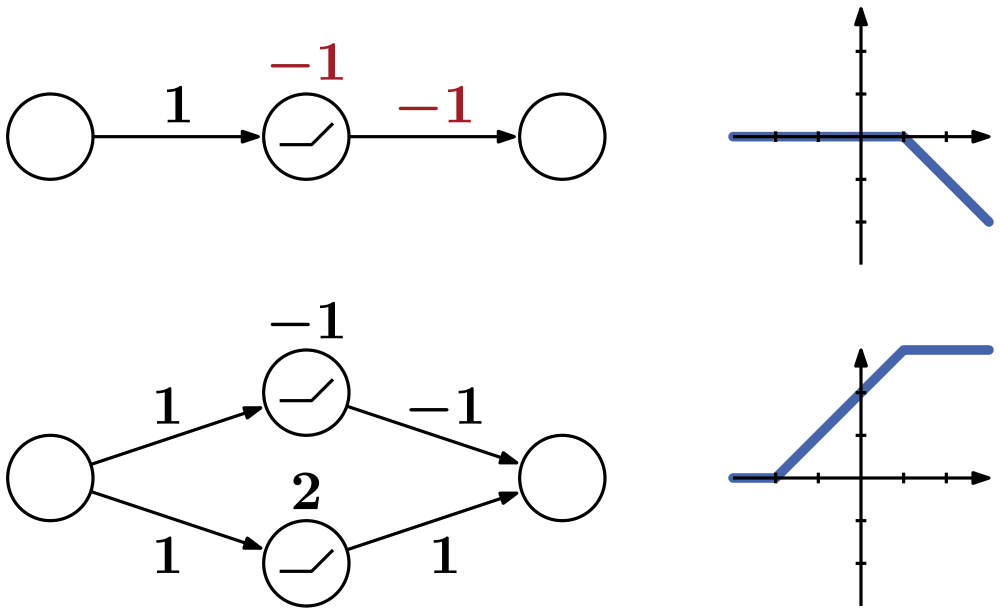




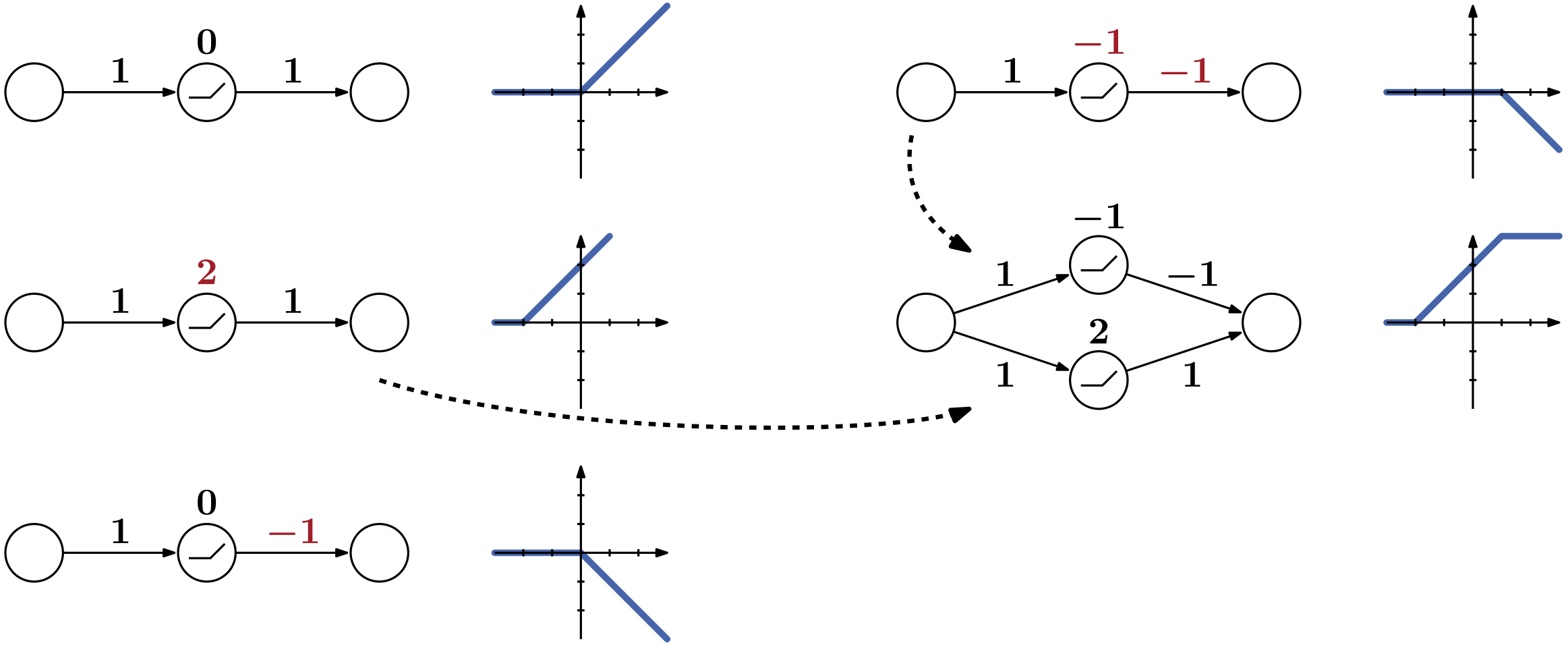


Geometrie von ReLU-Netzwerken

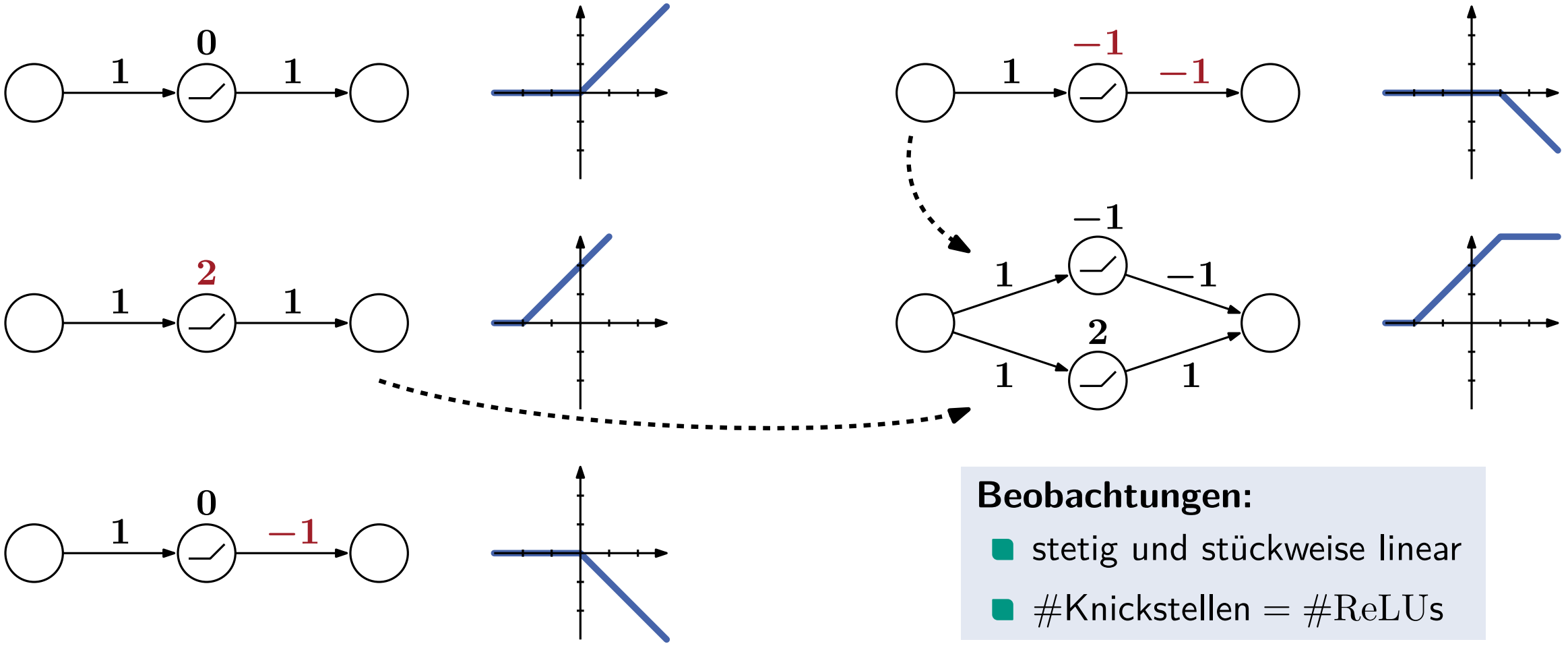




Geometrie von ReLU-Netzwerken



Geometrie von ReLU-Netzwerken



Beobachtungen:

- stetig und stückweise linear
- #Knickstellen = #ReLUs

Reduktion

Ziel: Polynomielle Transformation $\exists\mathbb{R} \rightsquigarrow \textsc{TrainNN}$

Gegeben:

Instanz von $\exists\mathbb{R}$

- Variablen: $X, Y, Z, \ldots \in \mathbb{R}$
- lineare Terme: $X + Y = Z$
- nichtlineare Terme: $X \cdot Y = 1$

Zu konstruieren:

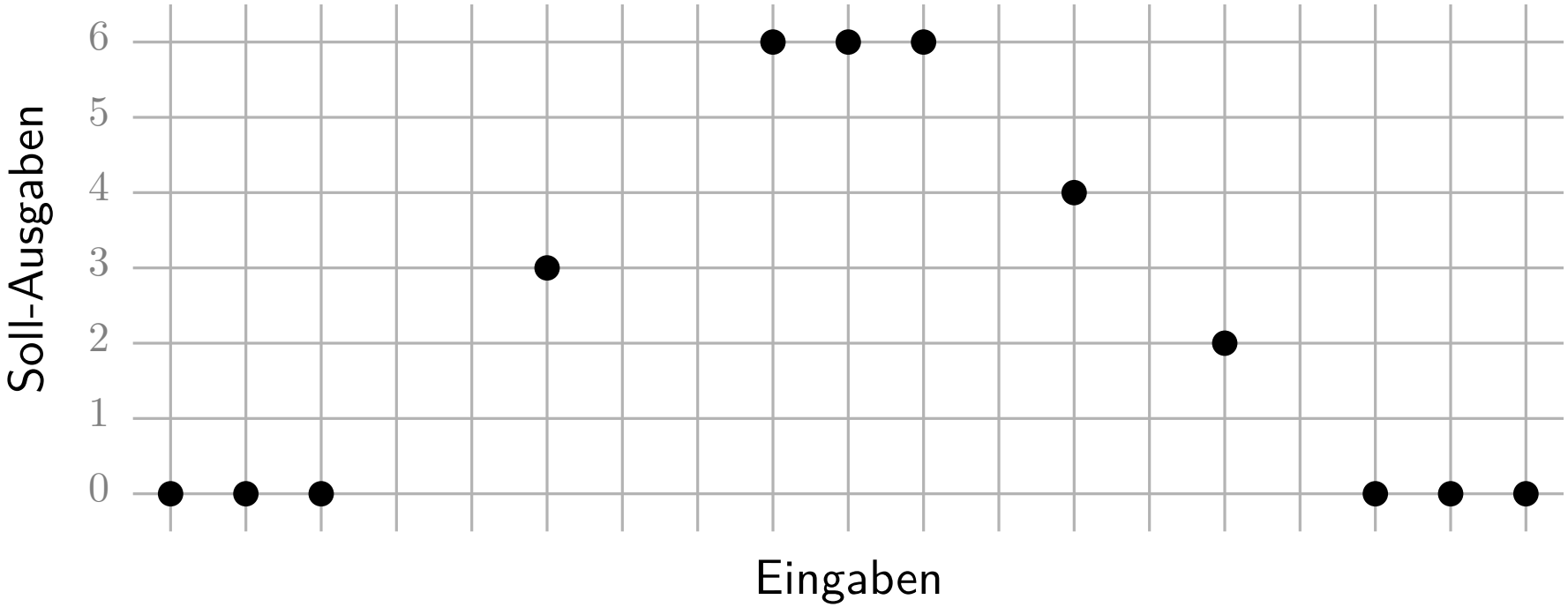
Äquivalente Instanz von $\textsc{TrainNN}$

- Architektur
  - zwei Input- und zwei Output-Neuronen
  - eine verstecke Schicht
  - fest vorgegebene Anzahl an ReLUs
- Trainingsdaten
  - Eingaben
  - Soll-Ausgaben

Konstruktion reellwertiger Variablen

Ziel: ReLU-Netzwerk, dass sich wie eine reellwertige Variable verhält

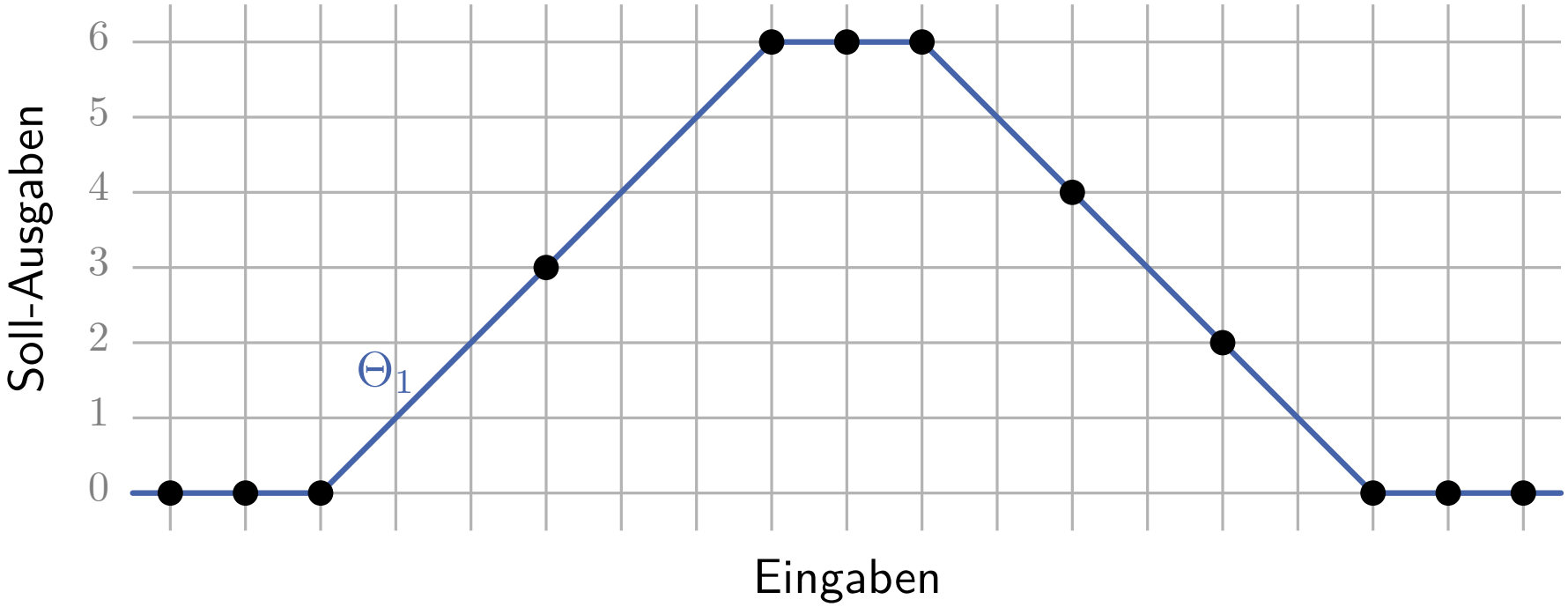
- wir konstruieren 12 Datenpunkte und wählen Architektiur mit 4 ReLU-Neuronen
- Training: Finden von Parametern Θ, sodass f_Θ alle Datenpunkte trifft



Konstruktion reellwertiger Variablen

Ziel: ReLU-Netzwerk, dass sich wie eine reellwertige Variable verhält

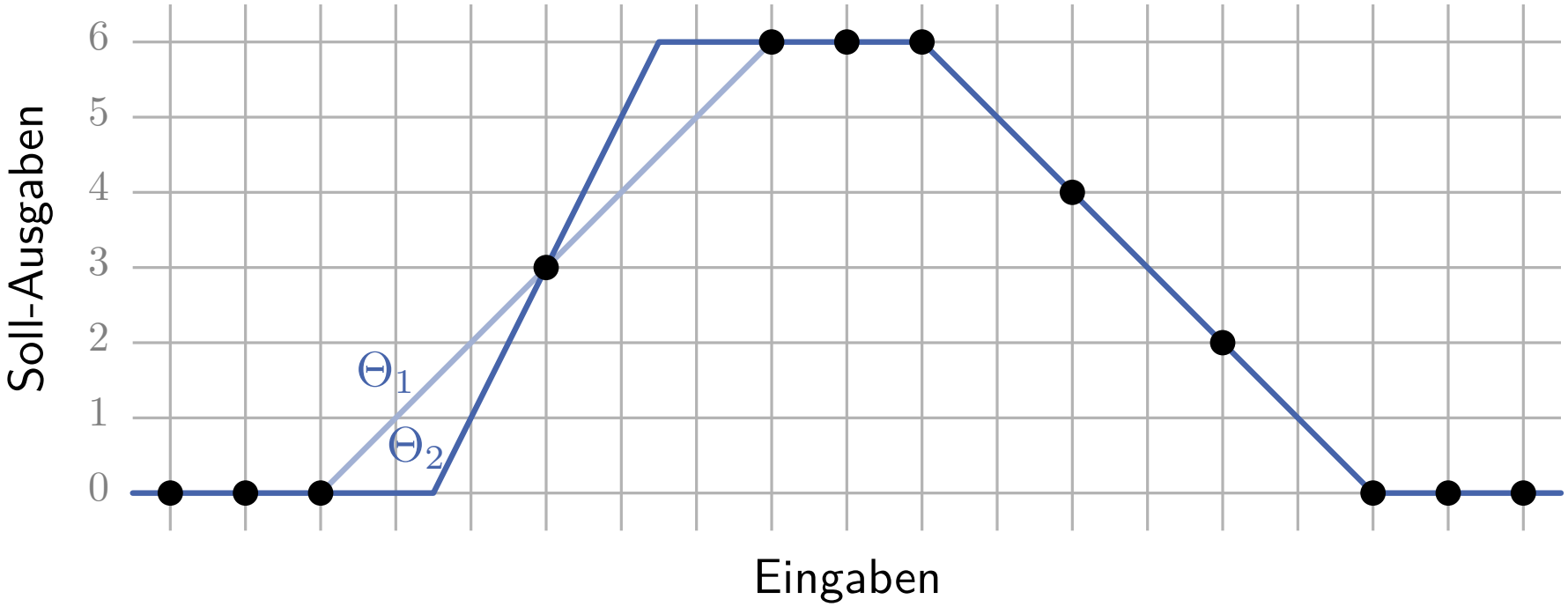
- wir konstruieren 12 Datenpunkte und wählen Architektiur mit 4 ReLU-Neuronen
- Training: Finden von Parametern Θ, sodass f_Θ alle Datenpunkte trifft



Konstruktion reellwertiger Variablen

Ziel: ReLU-Netzwerk, dass sich wie eine reellwertige Variable verhält

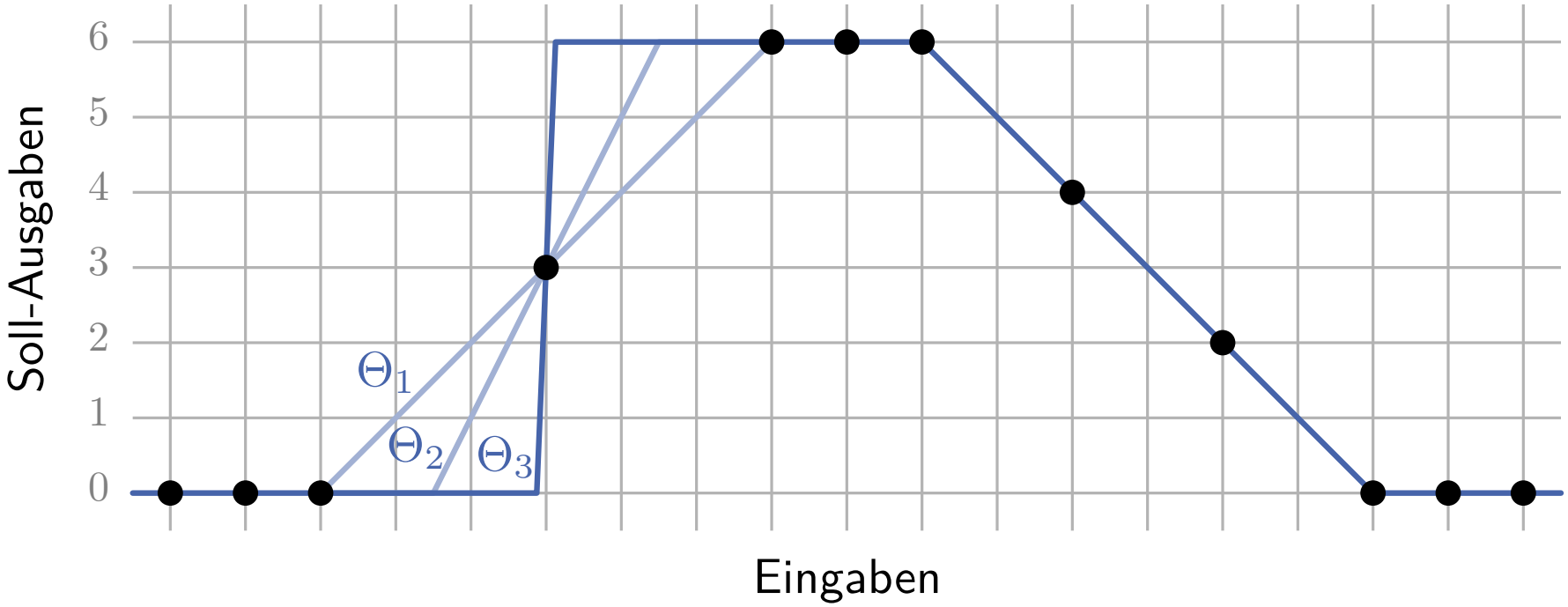
- wir konstruieren 12 Datenpunkte und wählen Architektiur mit 4 ReLU-Neuronen
- Training: Finden von Parametern Θ, sodass f_Θ alle Datenpunkte trifft



Konstruktion reellwertiger Variablen

Ziel: ReLU-Netzwerk, dass sich wie eine reellwertige Variable verhält

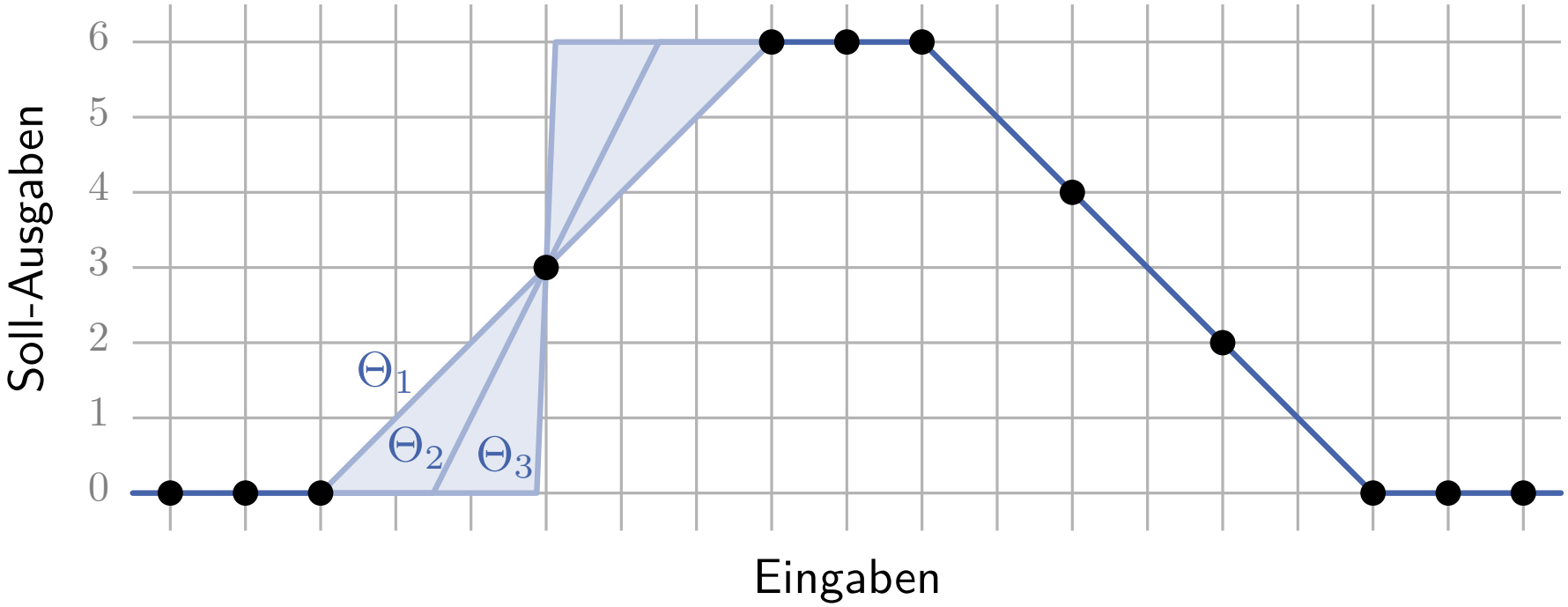
- wir konstruieren 12 Datenpunkte und wählen Architektiur mit 4 ReLU-Neuronen
- Training: Finden von Parametern Θ, sodass f_Θ alle Datenpunkte trifft



Konstruktion reellwertiger Variablen

Ziel: ReLU-Netzwerk, dass sich wie eine reellwertige Variable verhält

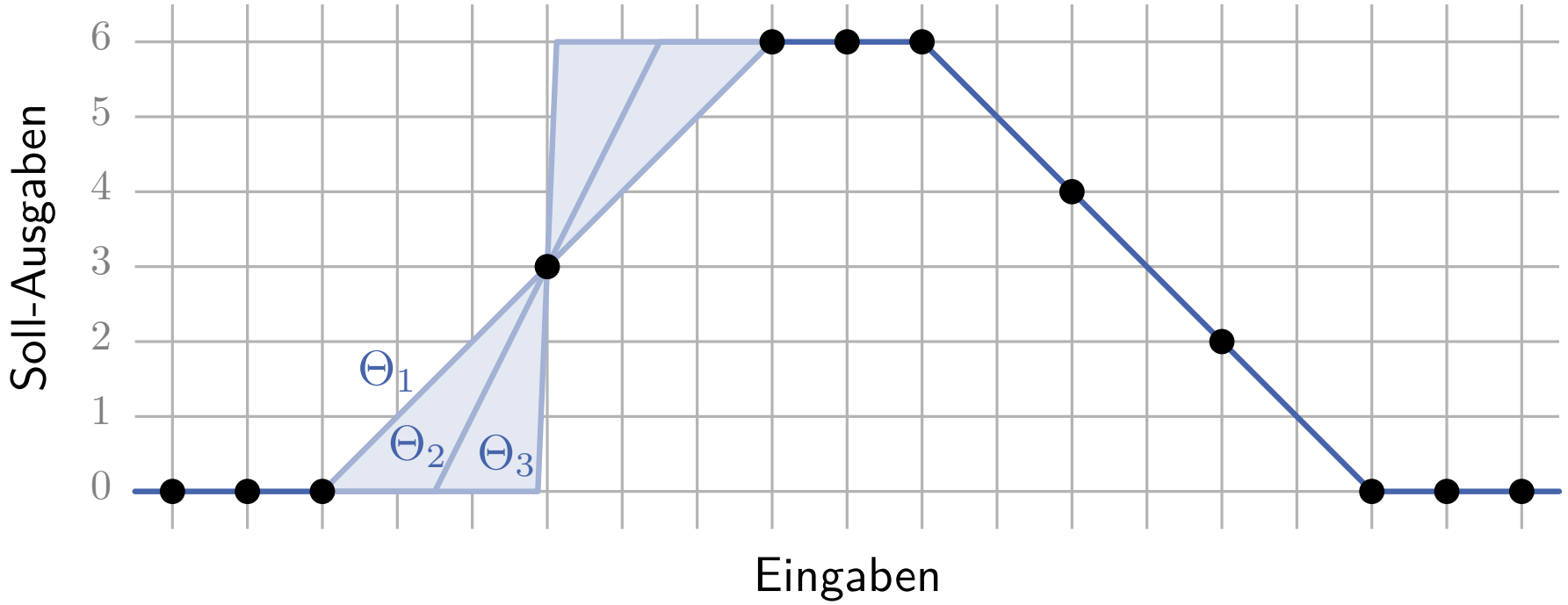
- wir konstruieren 12 Datenpunkte und wählen Architektiur mit 4 ReLU-Neuronen
- Training: Finden von Parametern Θ, sodass f_Θ alle Datenpunkte trifft



Konstruktion reellwertiger Variablen

Ziel: ReLU-Netzwerk, dass sich wie eine reellwertige Variable verhält

- wir konstruieren 12 Datenpunkte und wählen Architektiur mit 4 ReLU-Neuronen
- Training: Finden von Parametern Θ, sodass f_Θ alle Datenpunkte trifft

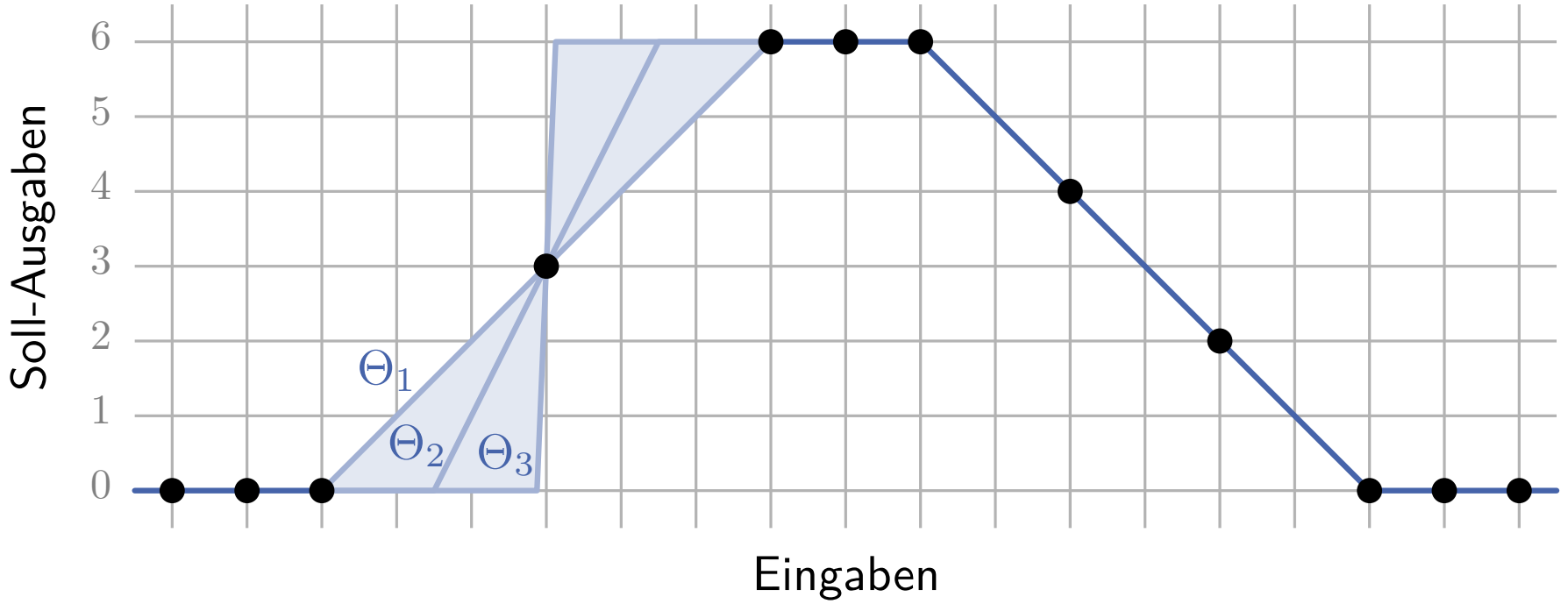


Beobachtungen:

Konstruktion reellwertiger Variablen

Ziel: ReLU-Netzwerk, dass sich wie eine reellwertige Variable verhält

- wir konstruieren 12 Datenpunkte und wählen Architektiur mit 4 ReLU-Neuronen
- Training: Finden von Parametern Θ, sodass f_Θ alle Datenpunkte trifft



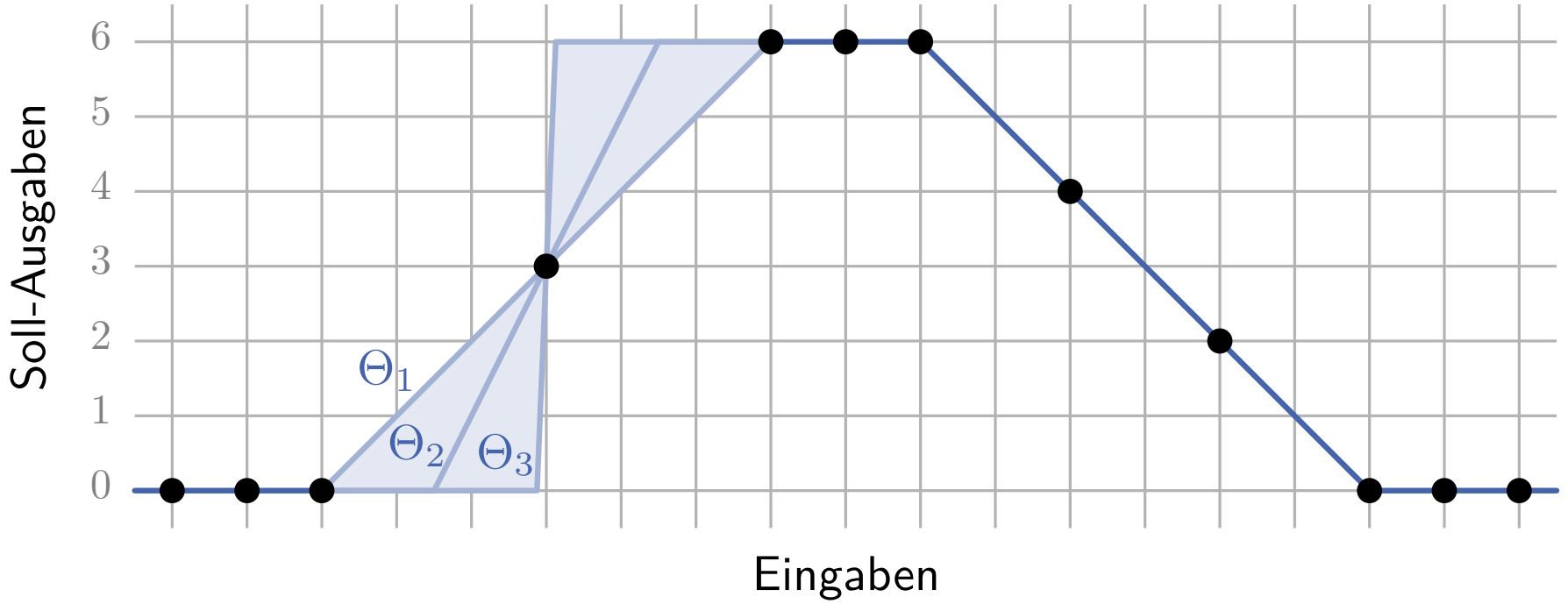
Beobachtungen:

- unendlich viele gültige Parameter Θ

Konstruktion reellwertiger Variablen

Ziel: ReLU-Netzwerk, dass sich wie eine reellwertige Variable verhält

- wir konstruieren 12 Datenpunkte und wählen Architektiur mit 4 ReLU-Neuronen
- Training: Finden von Parametern Θ, sodass f_Θ alle Datenpunkte trifft



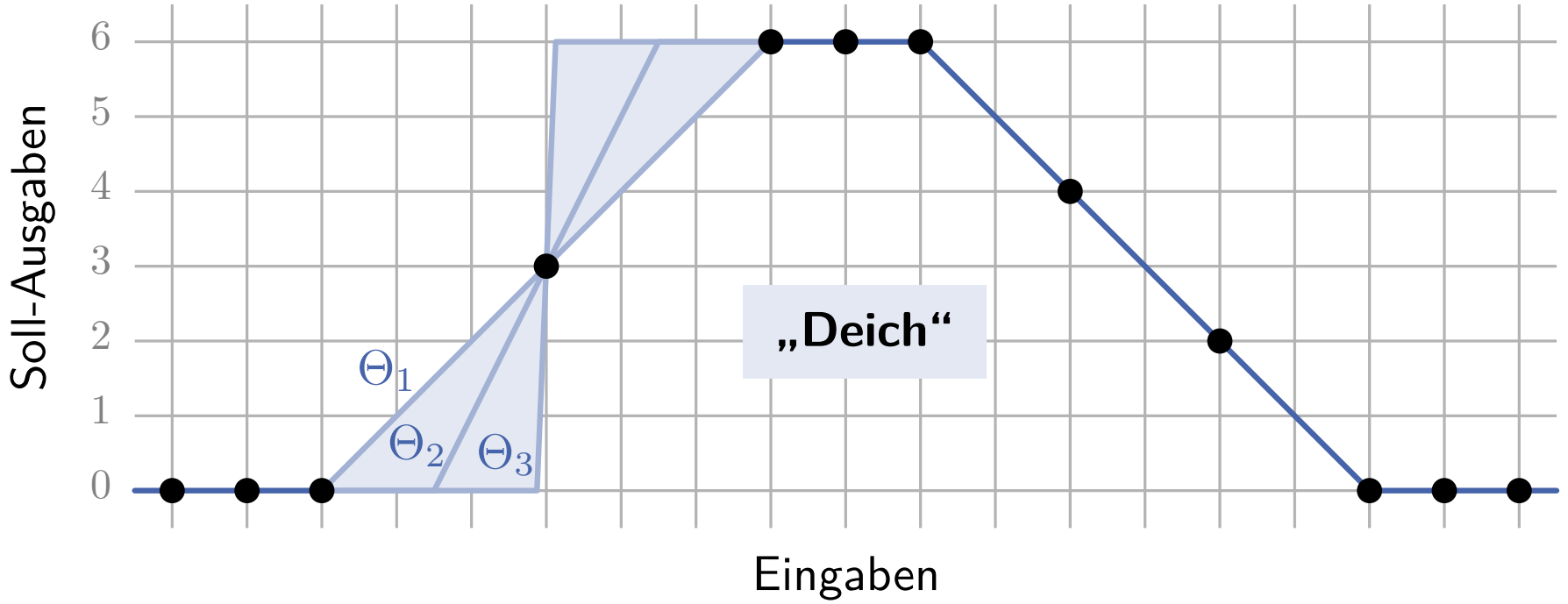
Beobachtungen:

- unendlich viele gültige Parameter Θ
- unterscheiden sich nur in der Steigung
  $\rightsquigarrow$ Steigung $\sim$ Wert

Konstruktion reellwertiger Variablen

Ziel: ReLU-Netzwerk, dass sich wie eine reellwertige Variable verhält

- wir konstruieren 12 Datenpunkte und wählen Architektiur mit 4 ReLU-Neuronen
- Training: Finden von Parametern Θ, sodass f_Θ alle Datenpunkte trifft



Beobachtungen:

- unendlich viele gültige Parameter Θ
- unterscheiden sich nur in der Steigung
  $\rightsquigarrow$ Steigung $\sim$ Wert

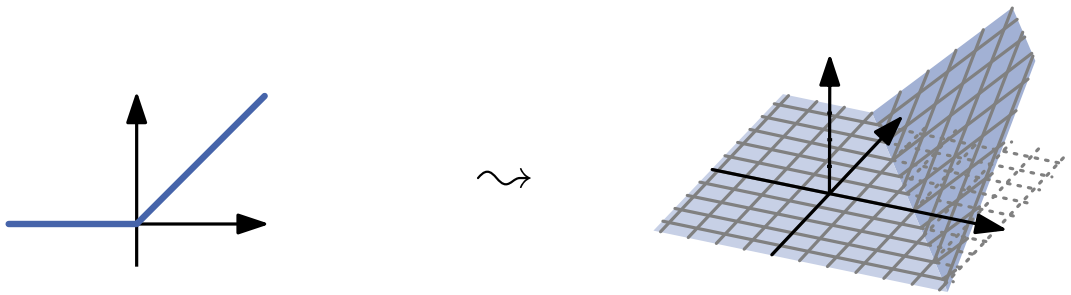
Konstruktion linearer Zusammenhänge

Ziel: linearen Zusammenhang zwischen zwei Variablen X und Y erzwingen

Konstruktion linearer Zusammenhänge

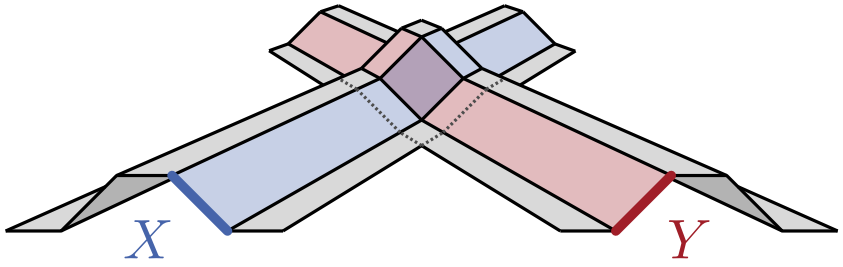
Ziel: linearen Zusammenhang zwischen zwei Variablen X und Y erzwingen

Idee: ausnutzen, dass es zwei Input-Neuronen gibt

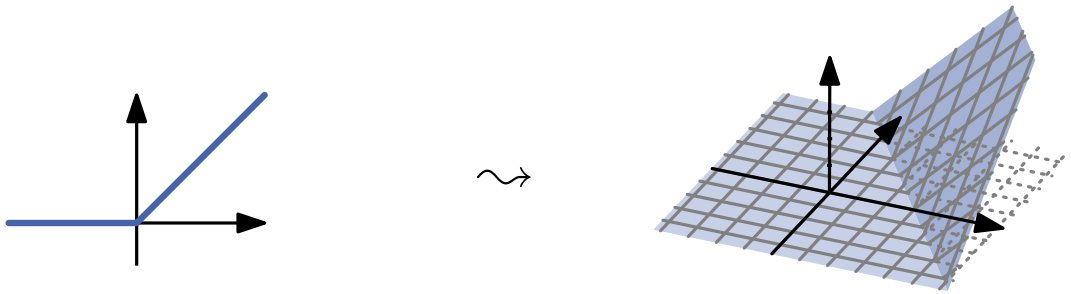


Konstruktion linearer Zusammenhänge

Ziel: linearen Zusammenhang zwischen zwei Variablen X und Y erzwingen



Idee: ausnutzen, dass es zwei Input-Neuronen gibt

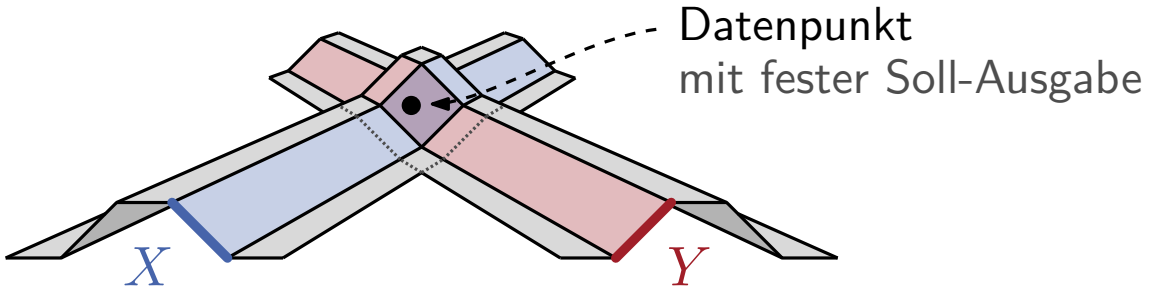


Beobachtungen:

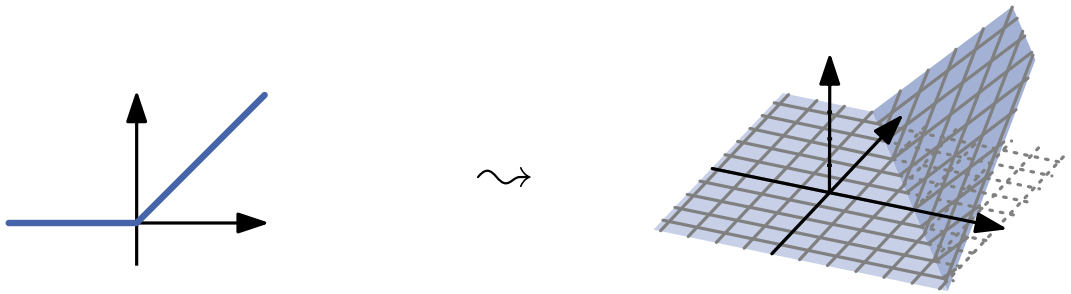
- Variablen-Deiche: 2D $\rightsquigarrow$ 3D
  Wie vorher: Steigung $\sim$ Wert

Konstruktion linearer Zusammenhänge

Ziel: linearen Zusammenhang zwischen zwei Variablen X und Y erzwingen



Idee: ausnutzen, dass es zwei Input-Neuronen gibt

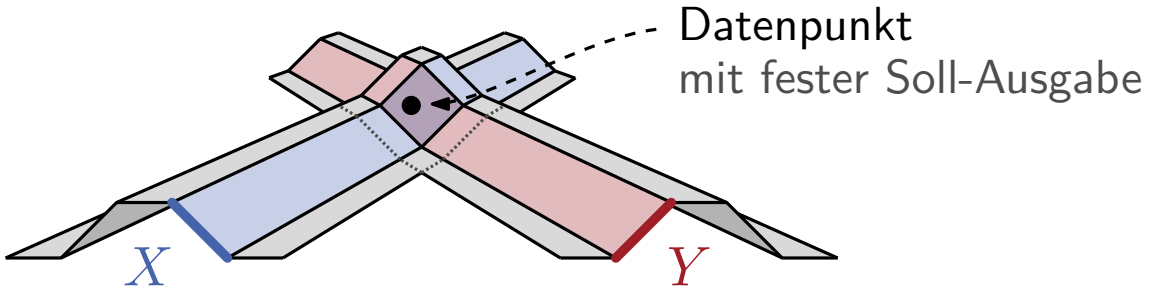


Beobachtungen:

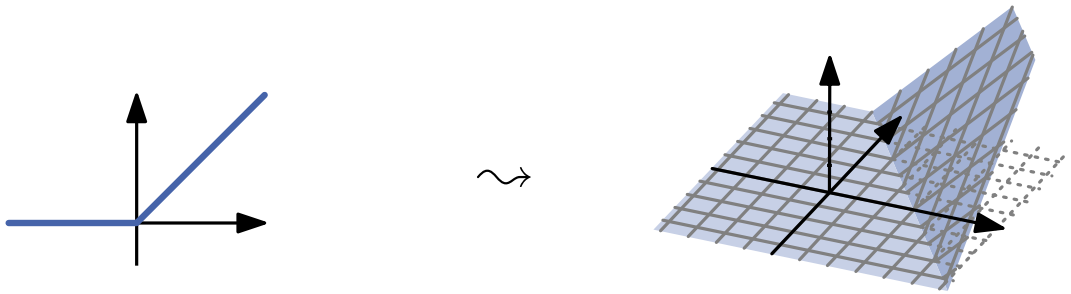
- Variablen-Deiche: 2D $\rightsquigarrow$ 3D
  Wie vorher: Steigung $\sim$ Wert
- Überschneidung
  - blau steiler $\rightsquigarrow$ rot flacher
  - rot steiler $\rightsquigarrow$ blau flacher

Konstruktion linearer Zusammenhänge

Ziel: linearen Zusammenhang zwischen zwei Variablen X und Y erzwingen



Idee: ausnutzen, dass es zwei Input-Neuronen gibt



Beobachtungen:

- Variablen-Deiche: 2D $\rightsquigarrow$ 3D
  Wie vorher: Steigung $\sim$ Wert
- Überschneidung
  - blau steiler $\rightsquigarrow$ rot flacher
  - rot steiler $\rightsquigarrow$ blau flacher
- lineare Terme, z.B.
  - Kopieren
  - Addition (3er-Überlagerung)

Reduktion: Zusammensetzen

Reduktion: $\mathrm{ETR} \rightsquigarrow \textsc{TrainNN}$

Gegeben:

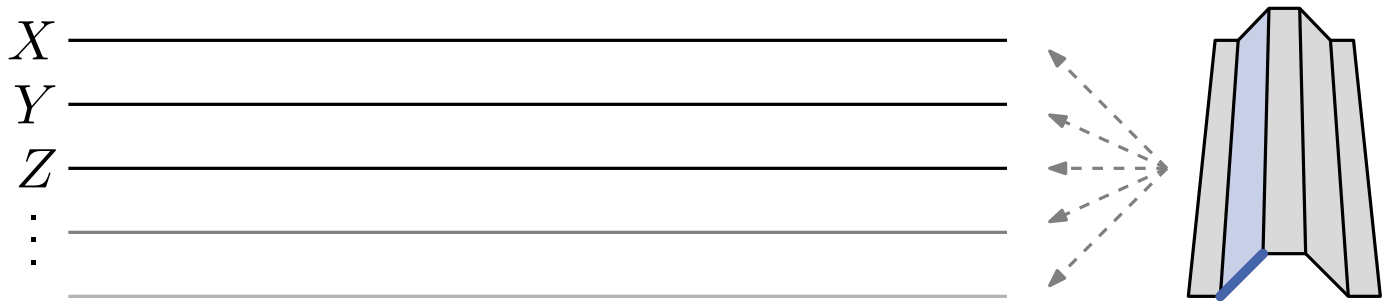
- Variablen: $X, Y, Z, \ldots \in \mathbb{R}$
- lineare Terme: $X + Y = Z$
- nichtlineare Terme: $X \cdot Y = 1$

Reduktion: Zusammensetzen

Reduktion: $\mathrm{ETR} \rightsquigarrow \textsc{TrainNN}$

Gegeben:

- Variablen: $X, Y, Z, \ldots \in \mathbb{R}$
- lineare Terme: $X + Y = Z$
- nichtlineare Terme: $X \cdot Y = 1$

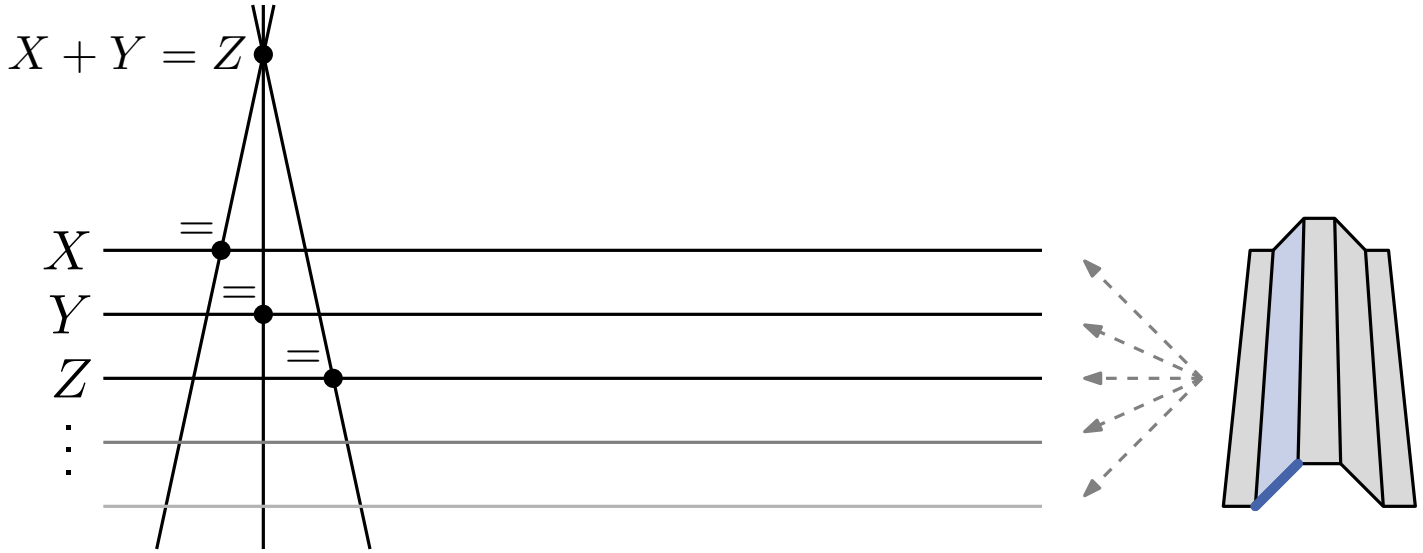


Reduktion: Zusammensetzen

Reduktion: $\mathrm{ETR} \rightsquigarrow \textsc{TrainNN}$

Gegeben:

- Variablen: $X, Y, Z, \ldots \in \mathbb{R}$
- lineare Terme: $X + Y = Z$
- nichtlineare Terme: $X \cdot Y = 1$

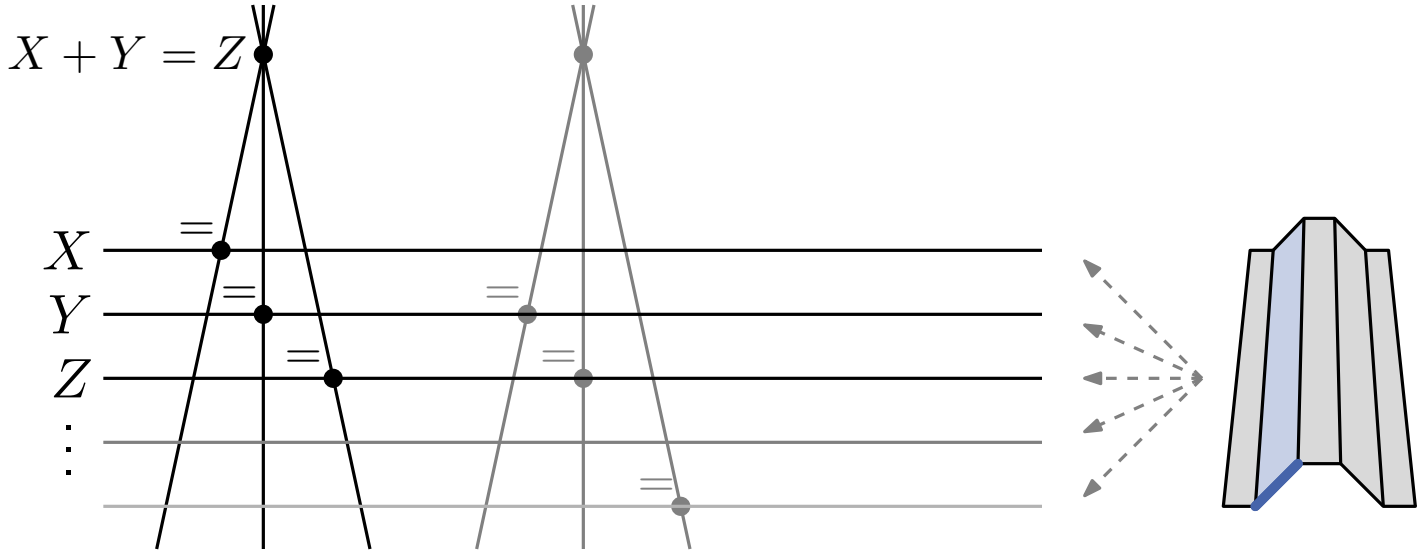


Reduktion: Zusammensetzen

Reduktion: $\mathrm{ETR} \rightsquigarrow \textsc{TrainNN}$

Gegeben:

- Variablen: $X, Y, Z, \ldots \in \mathbb{R}$
- lineare Terme: $X + Y = Z$
- nichtlineare Terme: $X \cdot Y = 1$

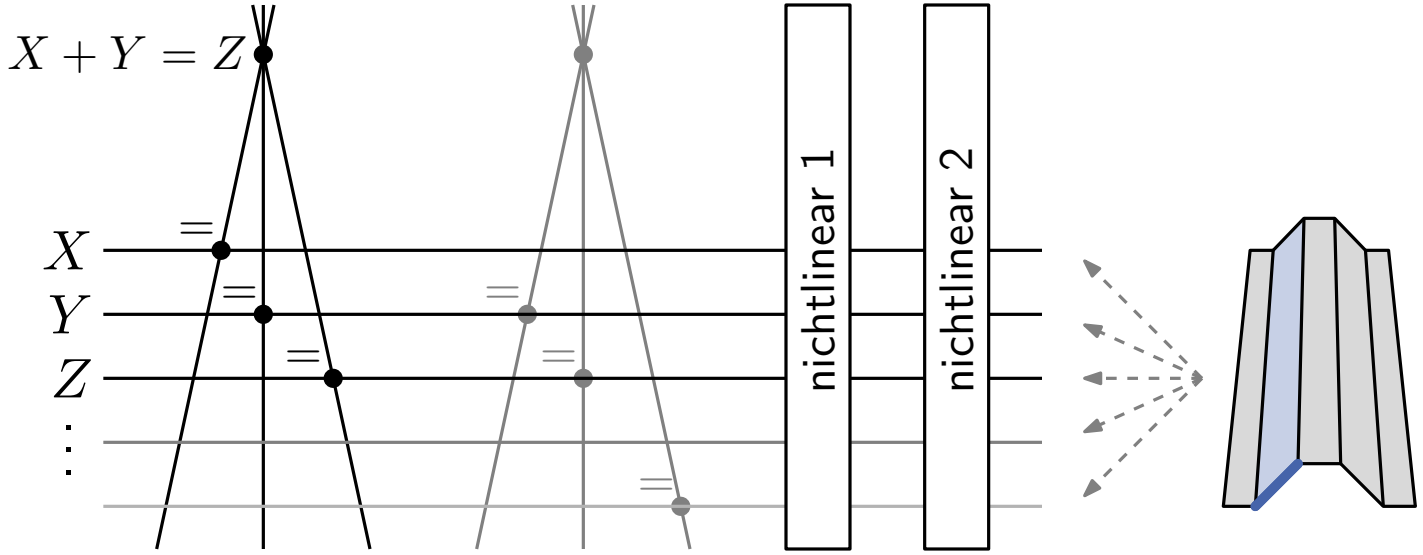


Reduktion: Zusammensetzen

Reduktion: $\mathrm{ETR} \rightsquigarrow \textsc{TrainNN}$

Gegeben:

- Variablen: $X, Y, Z, \ldots \in \mathbb{R}$
- lineare Terme: $X + Y = Z$
- nichtlineare Terme: $X \cdot Y = 1$

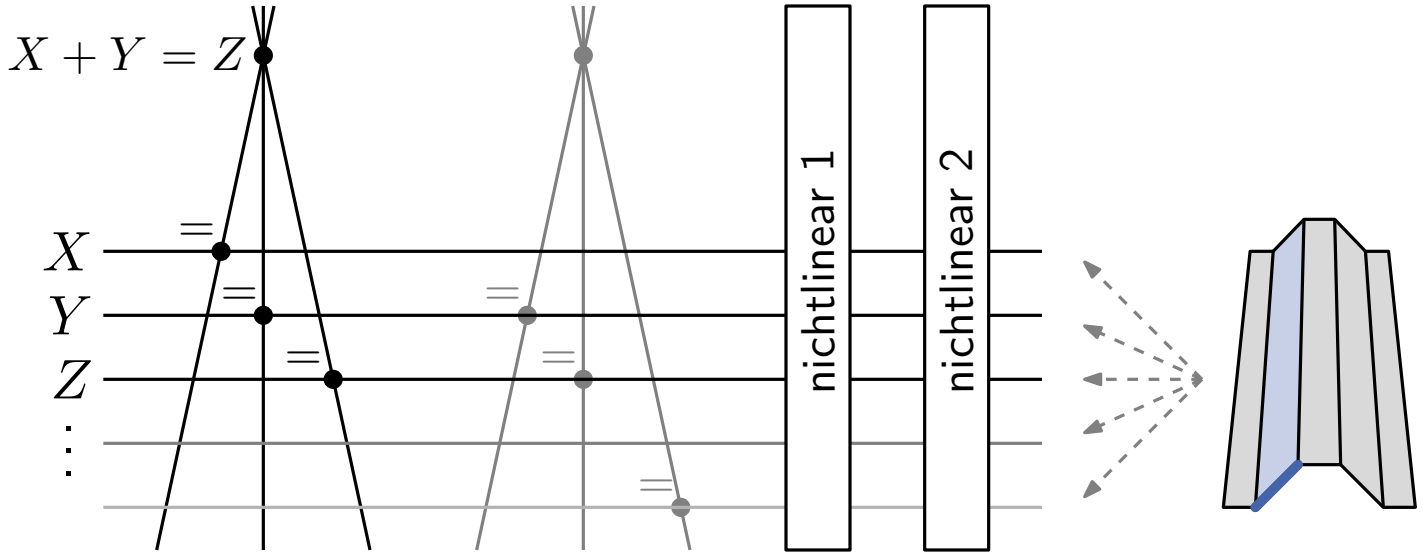


Reduktion: Zusammensetzen

Reduktion: $\mathrm{ETR} \rightsquigarrow \textsc{TrainNN}$

Gegeben:

- Variablen: $X, Y, Z, \ldots \in \mathbb{R}$
- lineare Terme: $X + Y = Z$
- nichtlineare Terme: $X \cdot Y = 1$

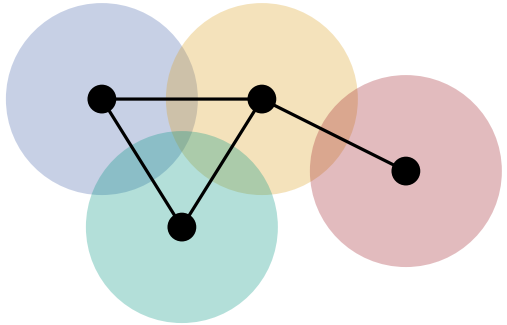


Zusammenfassung:

- $\textsc{TrainNN}$ simuliert ETR
  $\rightsquigarrow$ $\textsc{TrainNN}$ is $\exists\mathbb{R}$-schwer
- Schwere des Spezialfalls
  $\rightsquigarrow$ Schwere im Allgemeinen
- unter realistischen Annahmen
- insgesamt 4-dimensional:
  - 2 für Input $\rightsquigarrow$ linear
  - 2 für Output $\rightsquigarrow$ nichtlinear

Zusammenfassung

Hyperbolische Geometrie

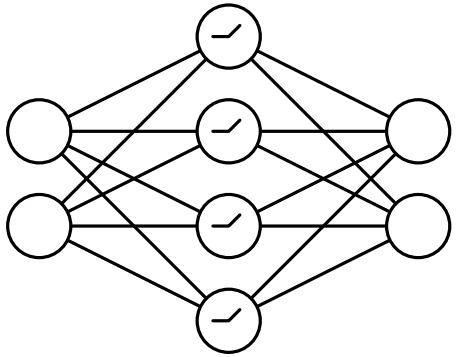


$\exists\mathbb{R}$-Schwere in der hyperbolischen Ebene

[EuroCG 2023]

[Bieker, Bläsius, Dohse, **Jungeblut**]

Training neuronaler Netze

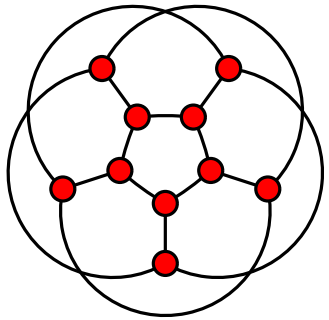


$\exists\mathbb{R}$-vollständig unter realistischen Annahmen

[NeurIPS 2023]

[Bertschinger, Hertrich, **Jungeblut**, Miltzow, Weber]

Lombardi-Zeichnungen

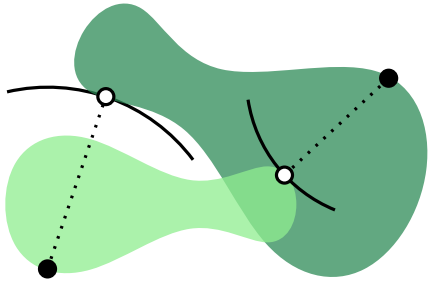


Graphvisualisierung unter geometrischen Nebenbedingungen

[GD 2023]

[**Jungeblut**]

Hausdorff-Distanz



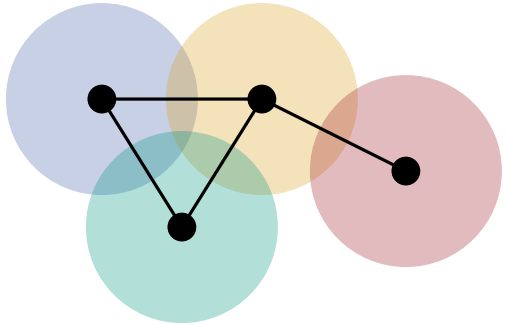
Distanzmaß für Mengen
$\rightsquigarrow \forall\exists\mathbb{R}$-vollständig

[SoCG 2022] [DCG 2023]

[**Jungeblut**, Kleist, Miltzow]

Probleme in $\exists\mathbb{R}$

Recog(UDG):



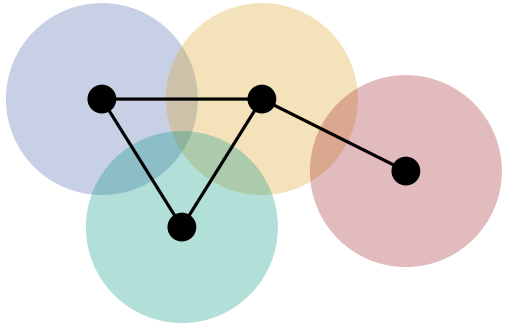
$\Longleftrightarrow$

$$\exists p_1, \ldots, p_n \in \mathbb{R}^2 : \bigwedge_{uv \in E} \|p_u - p_v\| \leq 2 \quad \text{<----- Kanten}$$

$$\wedge \bigwedge_{uv \notin E} \|p_u - p_v\| > 2 \quad \text{<----- Nicht-Kanten}$$

Probleme in $\exists\mathbb{R}$

Recog(UDG):

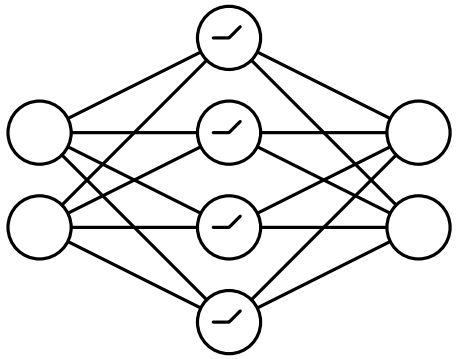


$\iff$

$$\exists p_1, \ldots, p_n \in \mathbb{R}^2 : \bigwedge_{uv \in E} \|p_u - p_v\| \leq 2 \quad \text{<----- Kanten}$$

$$\wedge \bigwedge_{uv \notin E} \|p_u - p_v\| > 2 \quad \text{<----- Nicht-Kanten}$$

TrainNN:



$\iff$

$$\exists \underbrace{w_1, \ldots}_{\text{Parameter}} \in \mathbb{R} : \bigwedge_{\text{Datenpunkt } d} [[d \text{ wird exakt berechnet}]]$$

18 Capturing the Computational Complexity of Geometric Problems by the First-Order Theory of the Reals
Paul Jungeblut

Nichtlineare Zusammenhänge: Inversion