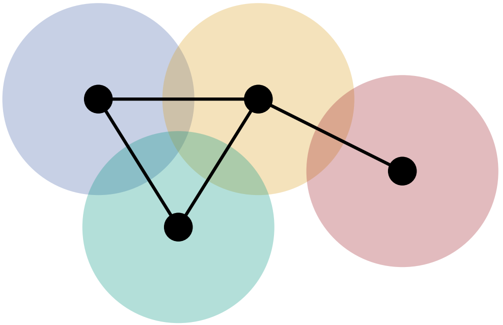
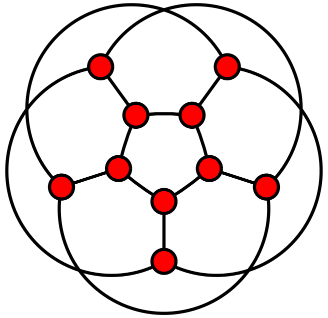
Capturing the Computational Complexity of Geometric Problems by the First-Order Theory of the Reals

Karlsruher Institut für Technologie

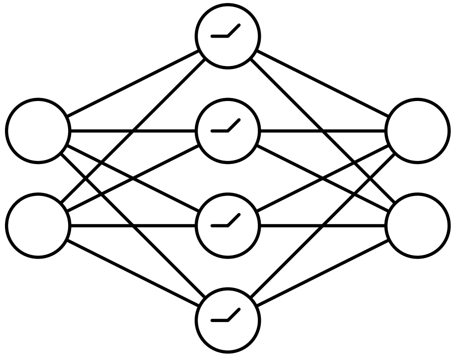
Paul Jungeblut
Betreut durch: PD Dr. Torsten Ueckerdt

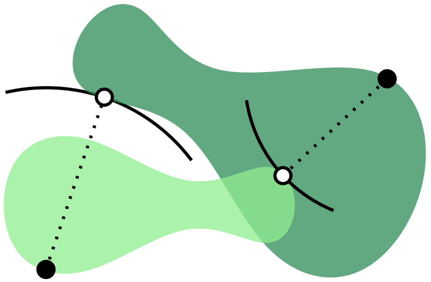
Mündliche Promotionsprüfung · 26. April 2024





$\exists\mathbb{R}$





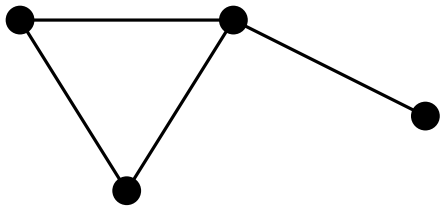
Problem 1: Erkennung von Unit-Disk-Graphen

Problem: Unit-Disk-Graphen erkennen

Eingabe: Graph G

Finde: Schnittrepräsentation von G bestehend aus Einheitskreisen.

Beispiel: Graph G



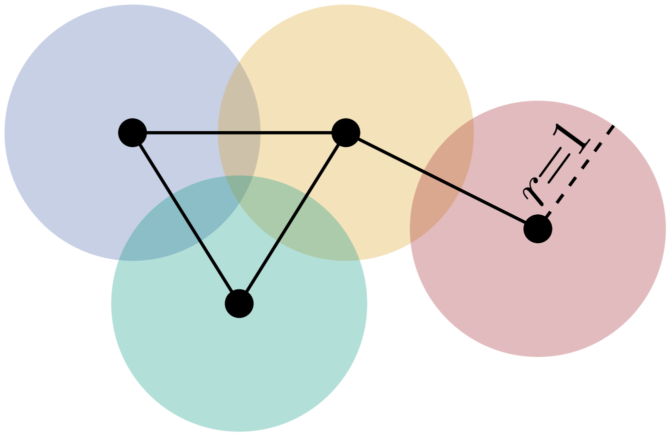
Problem 1: Erkennung von Unit-Disk-Graphen

Problem: Unit-Disk-Graphen erkennen

Eingabe: Graph G

Finde: Schnittrepräsentation von G bestehend aus Einheitskreisen.

Beispiel: Graph G



Problem 1: Erkennung von Unit-Disk-Graphen

Problem: Unit-Disk-Graphen erkennen

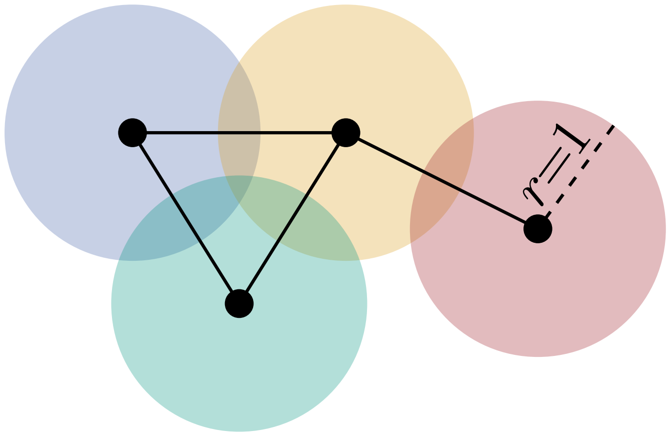
Eingabe: Graph G

Finde: Schnittrepräsentation von G bestehend aus Einheitskreisen.

Lösungen:

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

Beispiel: Graph G



Problem 1: Erkennung von Unit-Disk-Graphen

Problem: Unit-Disk-Graphen erkennen

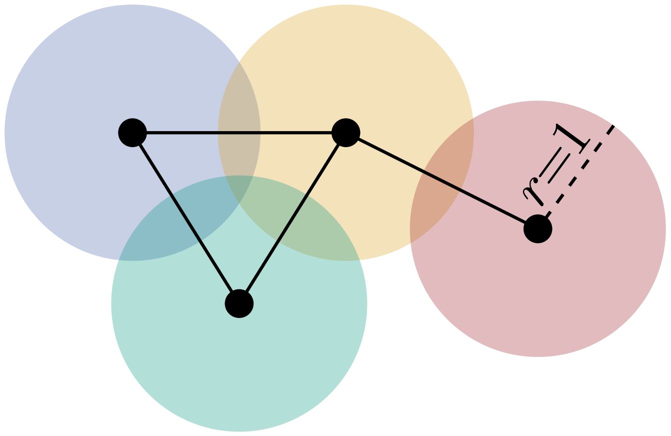
Eingabe: Graph G

Finde: Schnittrepräsentation von G bestehend aus Einheitskreisen.

Lösungen:

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

Beispiel: Graph G



Komplexität:

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

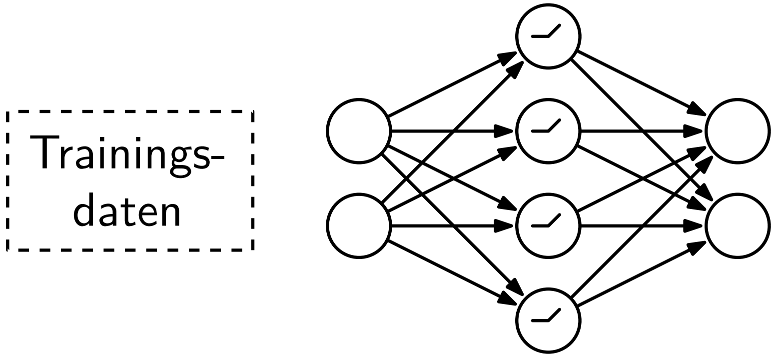
Problem 2: Training neuronaler Netze

Problem: TRAINNN

Eingabe: Netzwerk + Trainingsdaten

Finde: Parameter, sodass Trainingsdaten möglichst gut abgebildet werden.

Beispiel:



Problem 2: Training neuronaler Netze

Problem: TrainNN

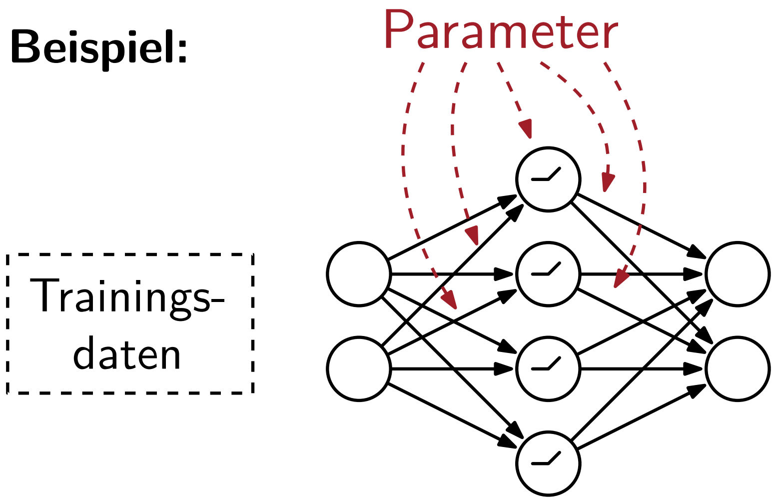
Eingabe: Netzwerk + Trainingsdaten

Finde: Parameter, sodass Trainingsdaten möglichst gut abgebildet werden.

Lösungen:

- Parameter aus $\mathbb{R}$ (Gewichte + Bias)
- Trainingsdaten „gelernt"

Beispiel:



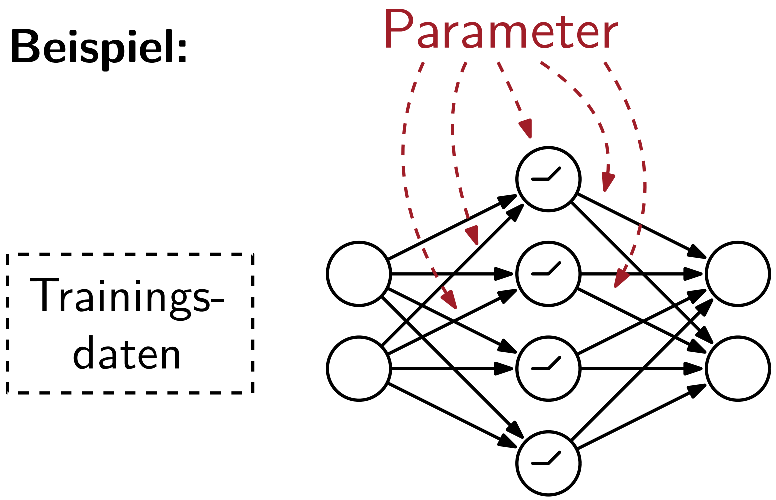
Problem 2: Training neuronaler Netze

Problem: TRAINNN

Eingabe: Netzwerk + Trainingsdaten

Finde: Parameter, sodass Trainingsdaten möglichst gut abgebildet werden.

Beispiel:



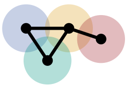
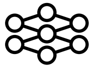
Lösungen:

- Parameter aus $\mathbb{R}$ (Gewichte + Bias)
- Trainingsdaten „gelernt“

Komplexität:

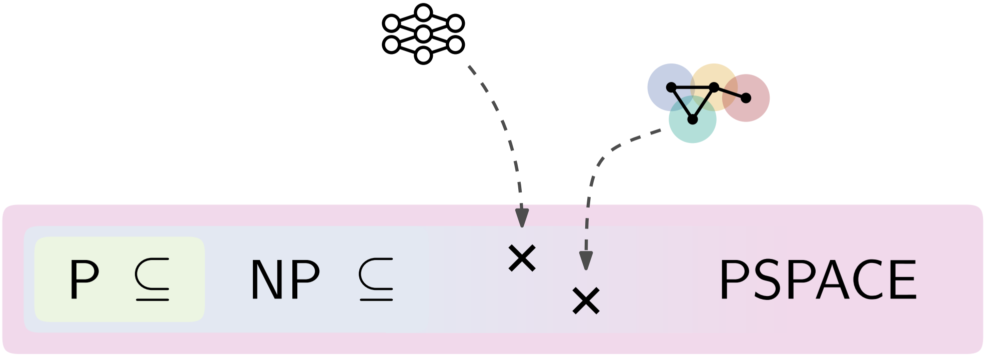
- NP-schwer [Blum, Rivest 1992]
- Überhaupt berechenbar?
  $\rightsquigarrow$ ja: polynomielles Ungleichungssystem

Komplexitätsklasse $\exists\mathbb{R}$

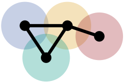
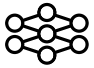
Wie schwierig sind  und  wirklich?

- beide sind NP-schwer
- beide sind in PSPACE

gleiche Wissenslücke bei vielen (geometrischen) Problemen



Komplexitätsklasse $\exists\mathbb{R}$

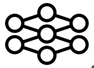
Wie schwierig sind  **und**  **wirklich?**

- beide sind NP-schwer
- beide sind in PSPACE

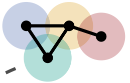
gleiche Wissenslücke bei vielen (geometrischen) Problemen

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

$\rightsquigarrow$ „gleich schwierig“



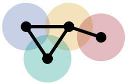
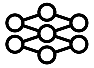
[Abrahamsen, Kleist, Miltzow 2021]
[Bertschinger, Hertrich, **Jungeblut**, Miltzow, Weber 2023]



[McDiarmid, Müller 2013]

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

Komplexitätsklasse $\exists\mathbb{R}$

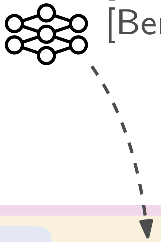
Wie schwierig sind  und  wirklich?

- beide sind NP-schwer
- beide sind in PSPACE

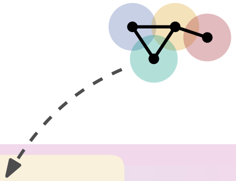
gleiche Wissenslücke bei vielen (geometrischen) Problemen

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

$\rightsquigarrow$ „gleich schwierig“



[Abrahamsen, Kleist, Miltzow 2021]
[Bertschinger, Hertrich, **Jungeblut**, Miltzow, Weber 2023]



[McDiarmid, Müller 2013]

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

In meiner Arbeit:

- exakte Komplexität bekannter offener Probleme bestimmt
  - zugrundeliegende Geometrie ausgenutzt
  - Wissen über $\exists\mathbb{R}$ bot richtige Perspektive
- neue Beweistechniken entwickelt und bekannte verfeinert
  - bereits in nachfolgender Literatur aufgegriffen worden
- 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:

$$\exists x, y \in \mathbb{R} \colon \; x^2 + y^2 \leq 1 \;\wedge\; y \geq 2x^2 - 1$$

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?

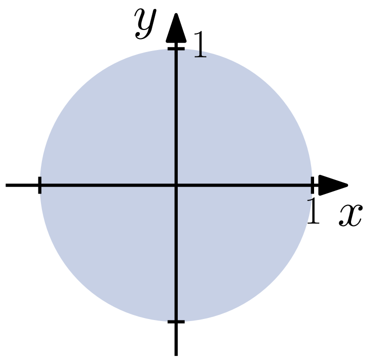
Beispiel 1:

$\exists x \in \mathbb{R} \colon x^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2:

$$\exists x, y \in \mathbb{R} \colon \; x^2 + y^2 \leq 1 \\ \wedge \; y \geq 2x^2 - 1$$



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?

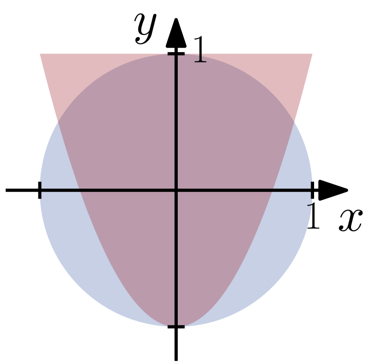
Beispiel 1:

$\exists x \in \mathbb{R} \colon x^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2:

$$\exists x, y \in \mathbb{R} \colon \; x^2 + y^2 \leq 1 \;\wedge\; y \geq 2x^2 - 1$$



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?

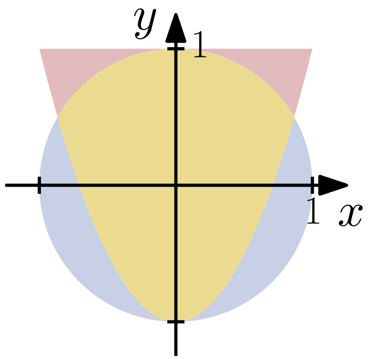
Beispiel 1:

$\exists x \in \mathbb{R} \colon x^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2:

$$\exists x, y \in \mathbb{R} \colon \ x^2 + y^2 \leq 1$$
$$\boxed{\wedge} \ y \geq 2x^2 - 1$$ ✓



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?

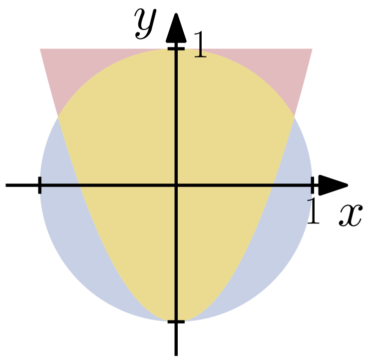
Beispiel 1:

$\exists x \in \mathbb{R}\colon x^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2:

$$\exists x, y \in \mathbb{R}\colon \; x^2 + y^2 \leq 1 \;\wedge\; y \geq 2x^2 - 1$$ ✓



Wie schwierig ist ETR?

- überabzählbar viele Lösungskandidaten
- entscheidbar [Tarski 1951]

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?

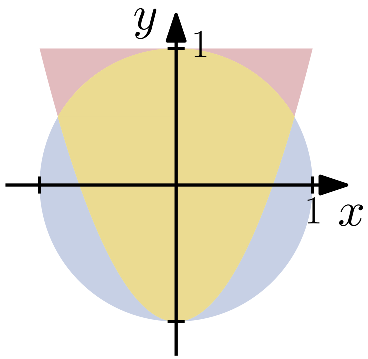
Beispiel 1:

$\exists x \in \mathbb{R} \colon x^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2:

$$\exists x, y \in \mathbb{R} \colon \; x^2 + y^2 \leq 1 \; \wedge \; y \geq 2x^2 - 1$$ ✓



Wie schwierig ist ETR?

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

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?

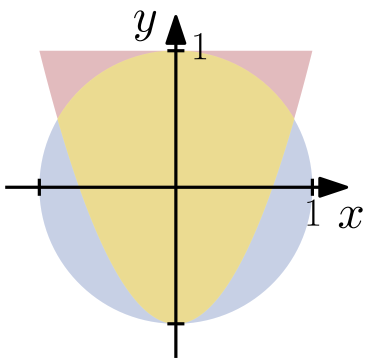
Beispiel 1:

$\exists x \in \mathbb{R}\colon x^2 < 0$ ✗

Quadrate sind immer nichtnegativ

Beispiel 2:

$$\exists x, y \in \mathbb{R}\colon \; x^2 + y^2 \leq 1 \;\wedge\; y \geq 2x^2 - 1$$ ✓



Wie schwierig ist ETR?

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

Definition: [Schaefer 2009]
$\exists\mathbb{R}$ enthält alle Probleme, die in polynomieller Zeit auf ETR reduziert werden können.

P $\subseteq$ NP $\subseteq$ $\exists\mathbb{R}$ $\subseteq$ 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
  $\leadsto$ alle bekannten Algorithmen algebraisch
- 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

Hohe Genauigkeit

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

Perspektivwechsel

- im Kern: reelle algebraische Geometrie
  $\rightsquigarrow$ alle bekannten Algorithmen algebraisch
- 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

Hohe Genauigkeit

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

Perspektivwechsel

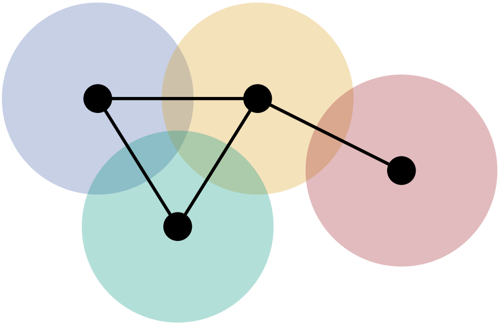
- im Kern: reelle algebraische Geometrie
  $\leadsto$ alle bekannten Algorithmen algebraisch
- algorithmische Fortschritte am ehesten durch
  mathematische Durchbrüche zu erwarten

Lösbarkeit in der Praxis

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

Beitrag und Ergebnisse

Hyperbolische Geometrie

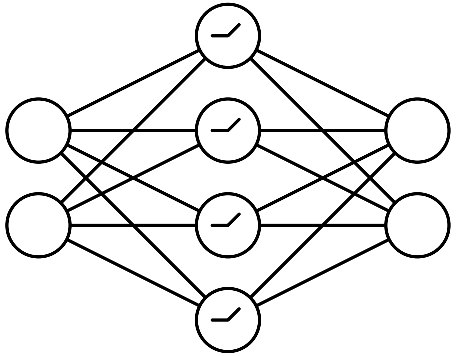


$\exists\mathbb{R}$-Schwere in der hyperbolischen Ebene

[EuroCG 2023]

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

Training neuronaler Netze

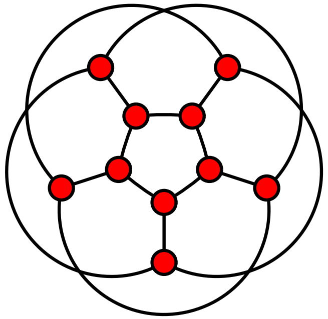


$\exists\mathbb{R}$-vollständig unter realistischen Annahmen

[NeurIPS 2023]

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

Lombardi-Zeichnungen

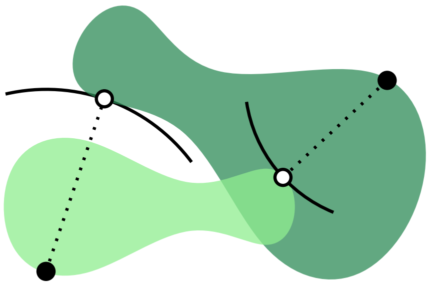


Graphvisualisierung unter geometrischen Nebenbedingungen

[GD 2023]

[**Jungeblut**]

Hausdorff-Distanz



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

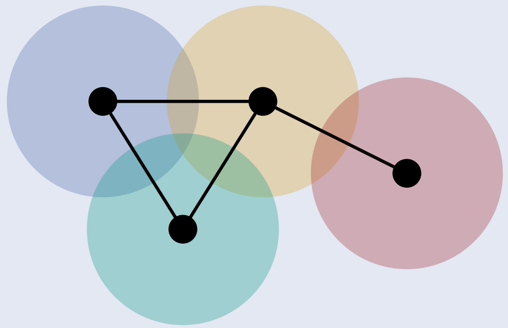
[SoCG 2022]

[Discrete Comput. Geom. 2024]

[**Jungeblut**, Kleist, Miltzow]

Beitrag und Ergebnisse

Hyperbolische Geometrie

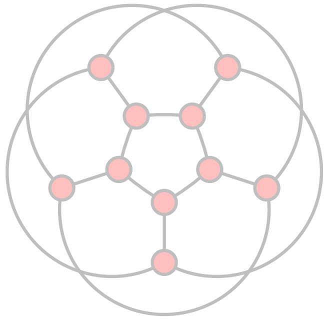


$\exists\mathbb{R}$-Schwere in der hyperbolischen Ebene

[EuroCG 2023]

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

Lombardi-Zeichnungen



Graphvisualisierung unter geometrischen Nebenbedingungen

[GD 2023]

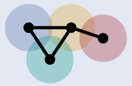
[**Jungeblut**]

STRETCHABILITY-Problem:

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

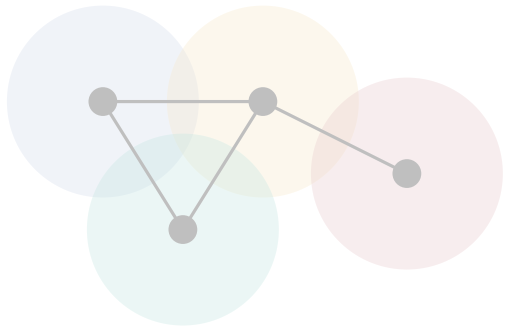
Theorem:
STRETCHABILITY ist $\exists\mathbb{R}$-vollständig in der hyperbolischen Ebene $\mathbb{H}^2$.

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

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

Beitrag und Ergebnisse

Hyperbolische Geometrie

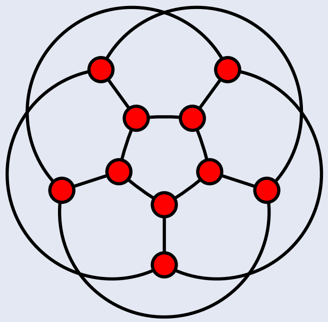


$\exists\mathbb{R}$-Schwere in der hyperbolischen Ebene

[EuroCG 2023]

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

Lombardi-Zeichnungen



Graphvisualisierung unter geometrischen Nebenbedingungen

[GD 2023]

[**Jungeblut**]

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

Kanten: Kreisbögen

gleiche Winkel um jeden Knoten

$\rightsquigarrow$ Symmetrien des Graphen gut erkennbar

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

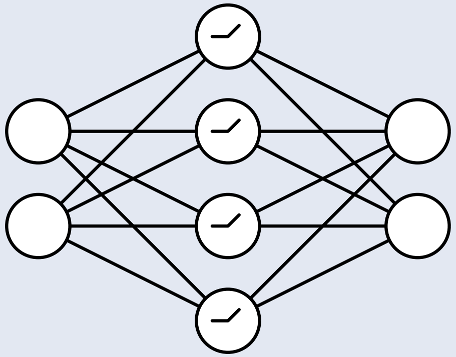
- erstes Komplexitätsresultat
- eleganter Umweg über hyperbolisches $\textsc{Stretchability}$

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

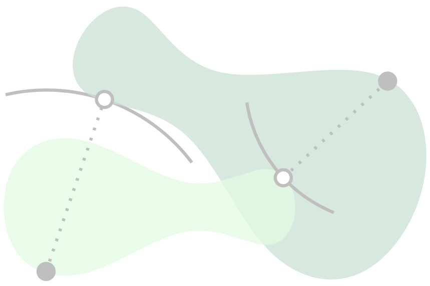


$\exists\mathbb{R}$-vollständig unter realistischen Annahmen

[NeurIPS 2023]

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

Hausdorff-Distanz



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

[SoCG 2022]

[Discrete Comput. Geom. 2024]

[**Jungeblut**, Kleist, Miltzow]

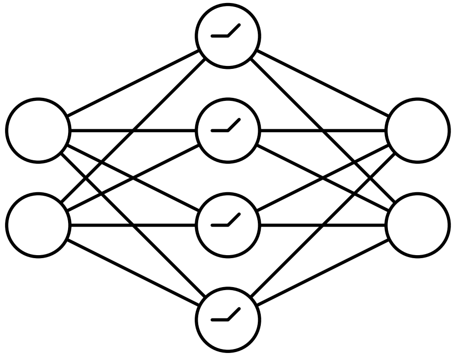
Beitrag und Ergebnisse

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 $\forall\exists\mathbb{R}$-Vollständigkeitsbeweis für ein natürliches Problem

Training neuronaler Netze

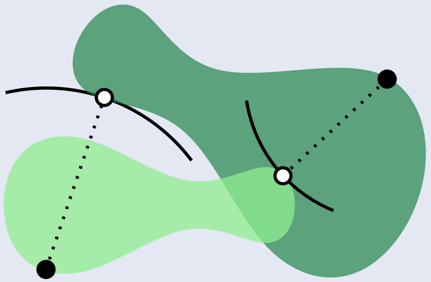


$\exists\mathbb{R}$-vollständig unter realistischen Annahmen

[NeurIPS 2023]

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

Hausdorff-Distanz



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

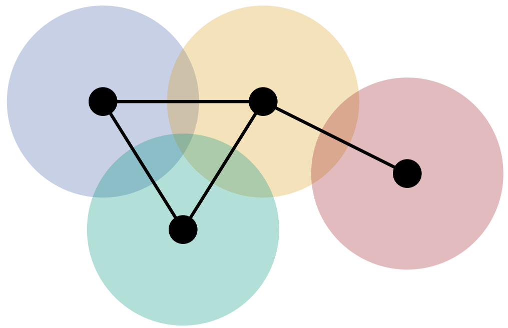
[SoCG 2022]

[Discrete Comput. Geom. 2024]

[**Jungeblut**, Kleist, Miltzow]

Beitrag und Ergebnisse

Hyperbolische Geometrie

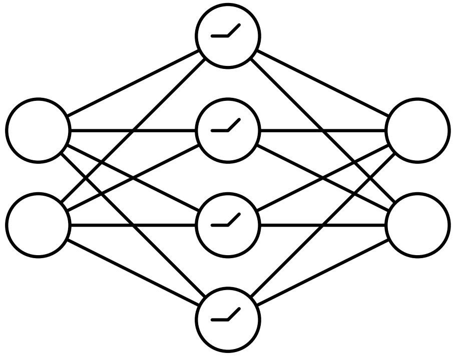


$\exists\mathbb{R}$-Schwere in der hyperbolischen Ebene

[EuroCG 2023]

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

Training neuronaler Netze

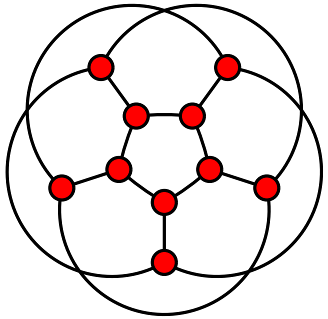


$\exists\mathbb{R}$-vollständig unter realistischen Annahmen

[NeurIPS 2023]

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

Lombardi-Zeichnungen

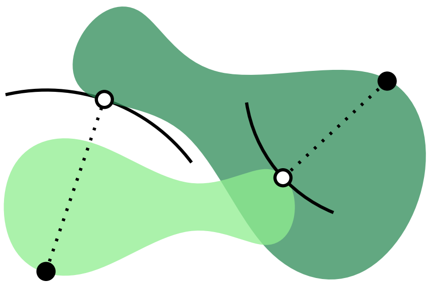


Graphvisualisierung unter geometrischen Nebenbedingungen

[GD 2023]

[**Jungeblut**]

Hausdorff-Distanz



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

[SoCG 2022]

[Discrete Comput. Geom. 2024]

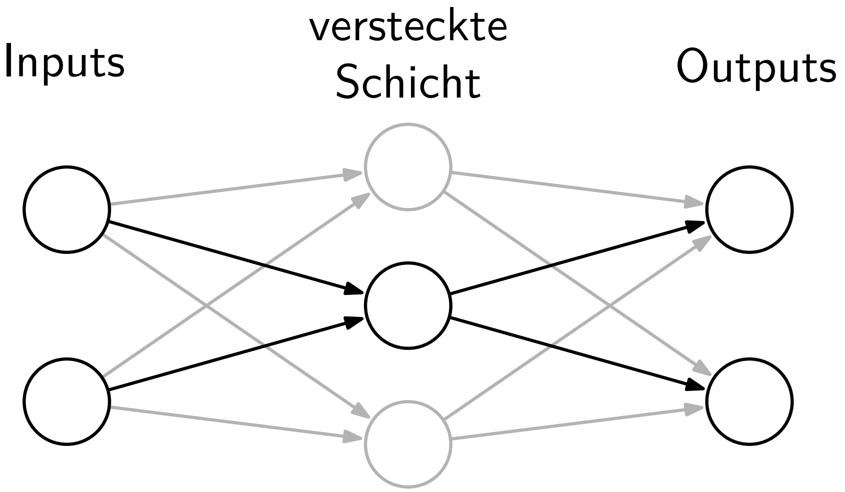
[**Jungeblut**, Kleist, Miltzow]

Neuronale Netze

Neuronales Netz $\longleftrightarrow$ Funktion

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

Parameter

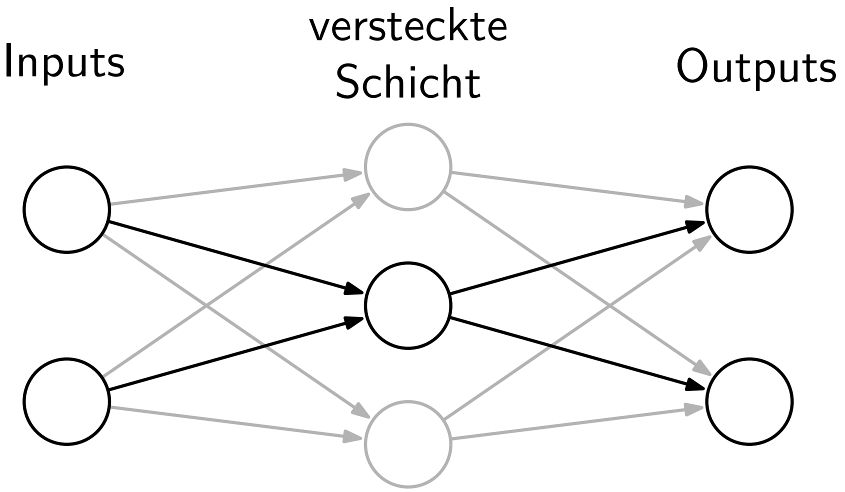


Neuronale Netze

Neuronales Netz $\longleftrightarrow$ Funktion

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

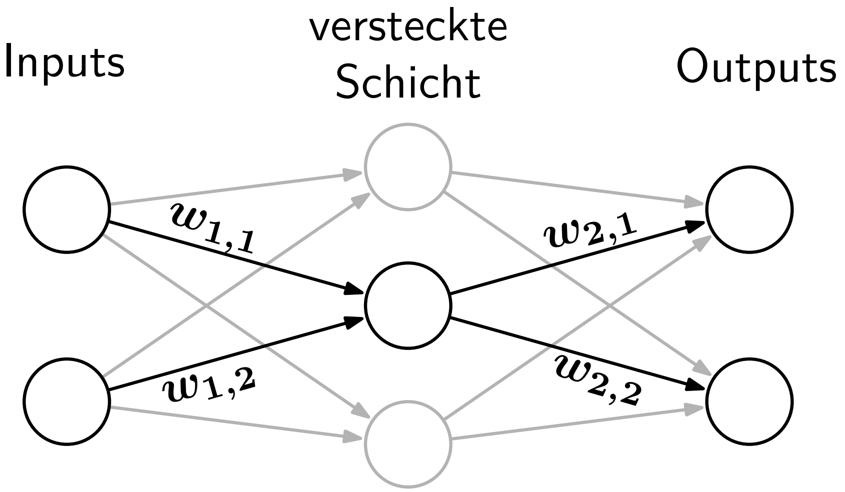
Parameter



Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Neuronale Netze

Neuronales Netz $\longleftrightarrow$ Funktion
Hier: $f_\Theta : \mathbb{R}^2 \to \mathbb{R}^2$
Parameter



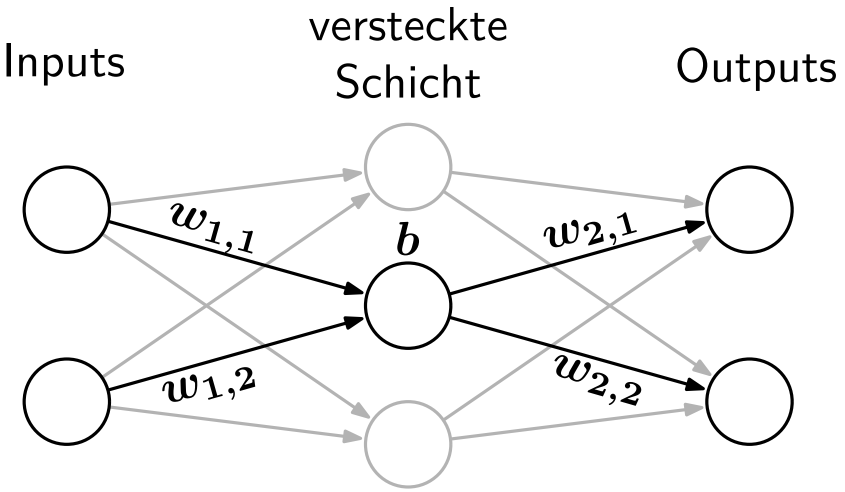
Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Parameter Θ:

Gewichte auf den Kanten

Neuronale Netze

Neuronales Netz $\longleftrightarrow$ **Funktion**
Hier: $f_\Theta : \mathbb{R}^2 \to \mathbb{R}^2$
Parameter



Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Parameter Θ:

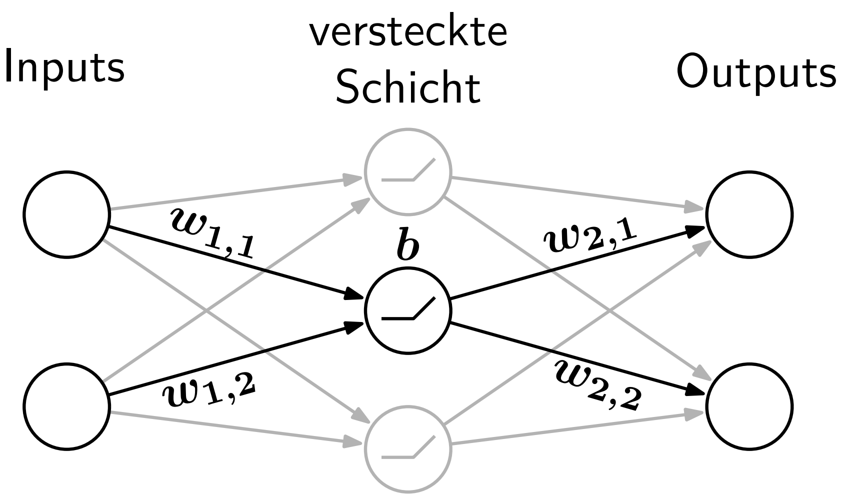
Gewichte auf den Kanten
Bias pro verstecktem Neuron

Neuronale Netze

Neuronales Netz $\longleftrightarrow$ Funktion

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

Parameter

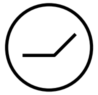


Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

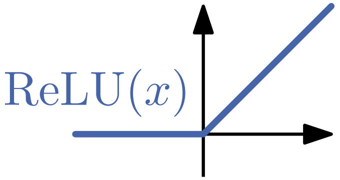
Parameter Θ:

Gewichte auf den Kanten

Bias pro verstecktem Neuron

Aktivierungsfunktion: $\mathrm{ReLU} =$ 

$$\begin{aligned}\mathrm{ReLU} \colon \mathbb{R} &\to \mathbb{R}\\ x &\mapsto \max\{0, x\}\end{aligned}$$

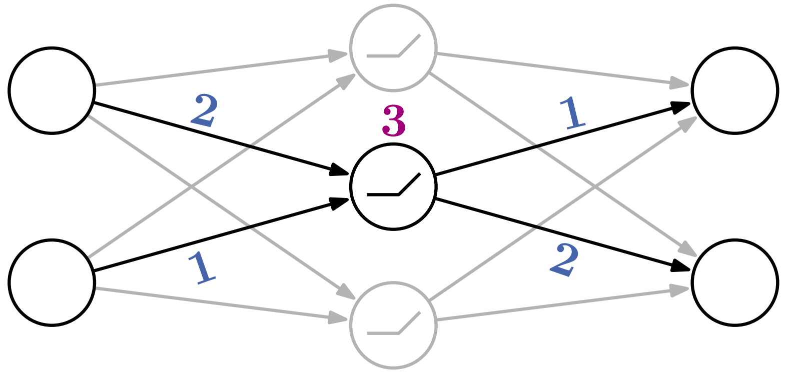


Neuronale Netze

Neuronales Netz $\longleftrightarrow$ Funktion

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

Parameter



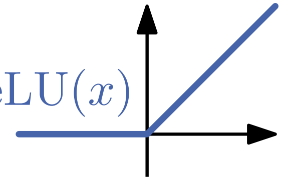
Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Parameter Θ:

Gewichte auf den Kanten
Bias pro verstecktem Neuron

Aktivierungsfunktion: $\mathrm{ReLU} =$

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

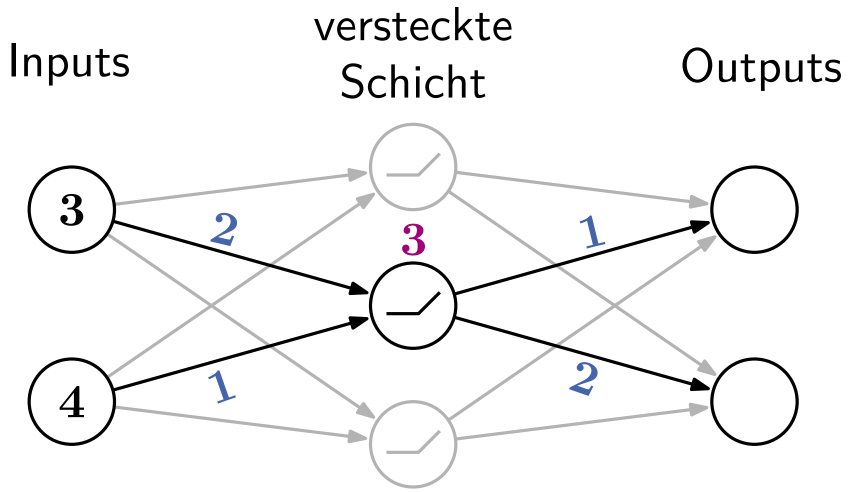


Neuronale Netze

Neuronales Netz $\longleftrightarrow$ Funktion

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

Parameter



Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

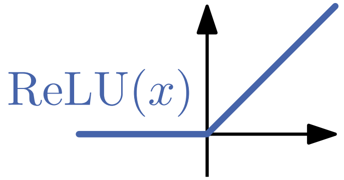
Parameter Θ:

Gewichte auf den Kanten

Bias pro verstecktem Neuron

Aktivierungsfunktion: $\mathrm{ReLU} =$

$$\begin{aligned}\mathrm{ReLU} \colon \mathbb{R} &\to \mathbb{R}\\ x &\mapsto \max\{0, x\}\end{aligned}$$

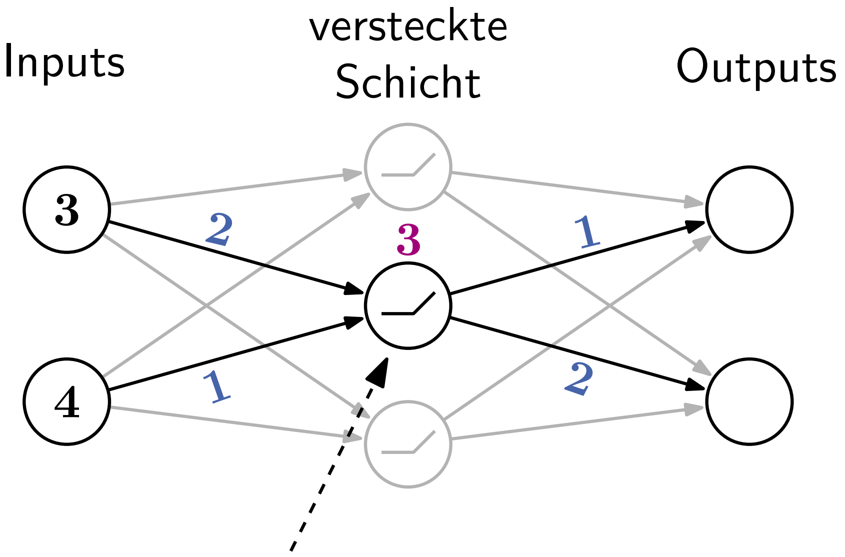


Neuronale Netze

Neuronales Netz $\longleftrightarrow$ Funktion

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

Parameter



$$\mathbf{ReLU}(3 \cdot 2 + 4 \cdot 1 + 3) = 13$$

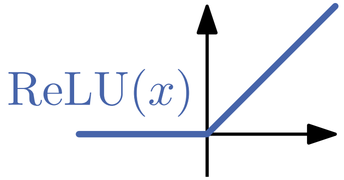
Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Parameter Θ:

Gewichte auf den Kanten
Bias pro verstecktem Neuron

Aktivierungsfunktion: $\mathrm{ReLU} =$

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

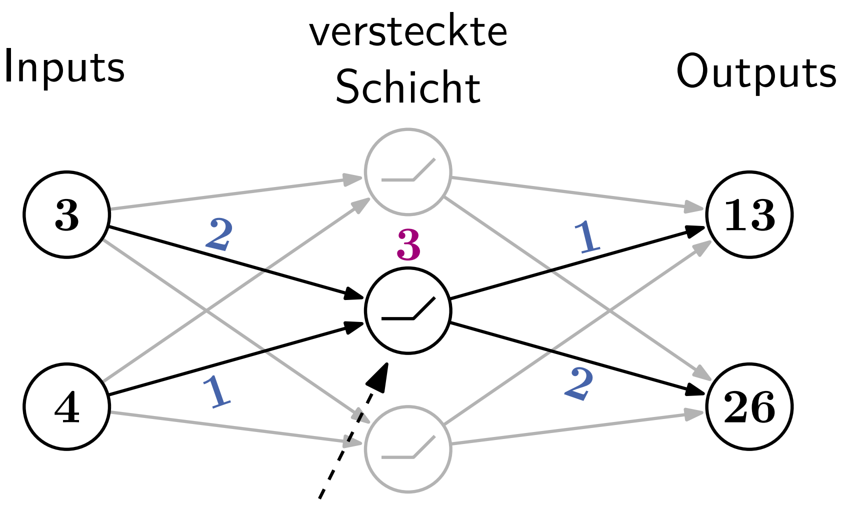


Neuronale Netze

Neuronales Netz $\longleftrightarrow$ Funktion

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

Parameter



$$\mathbf{ReLU}(3 \cdot 2 + 4 \cdot 1 + 3) = 13$$

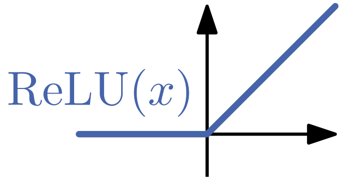
Architektur: gerichteter, azyklischer Graph
(Knoten = Neuronen)

Parameter Θ:

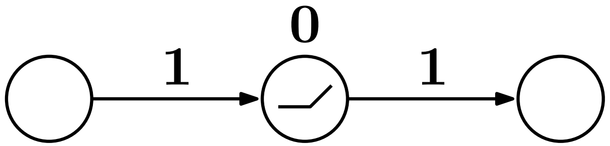
Gewichte auf den Kanten
Bias pro verstecktem Neuron

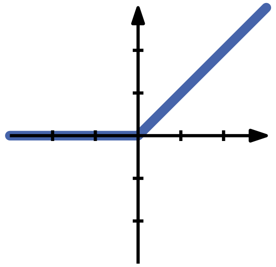
Aktivierungsfunktion: $\mathrm{ReLU} =$

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

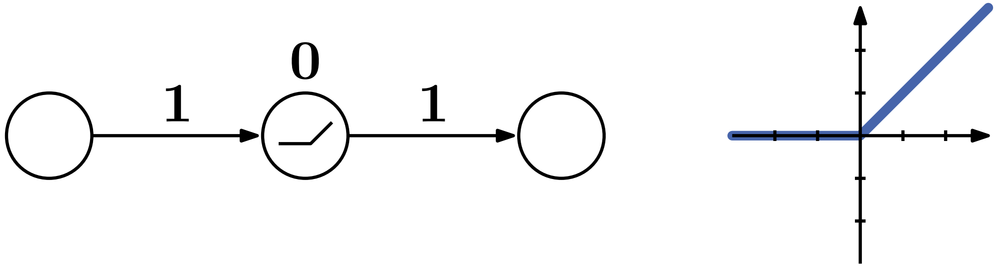


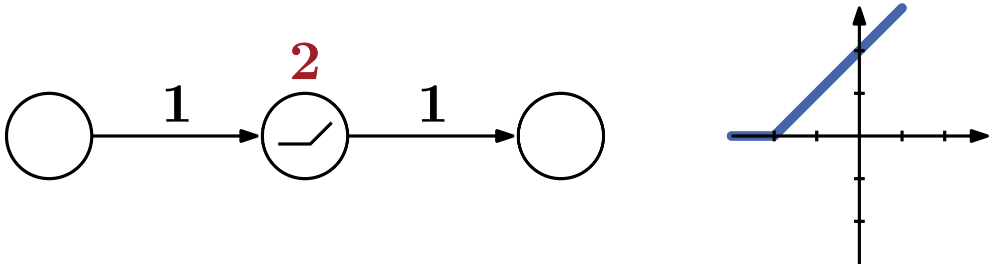
Funktionen von ReLU-Netzwerken



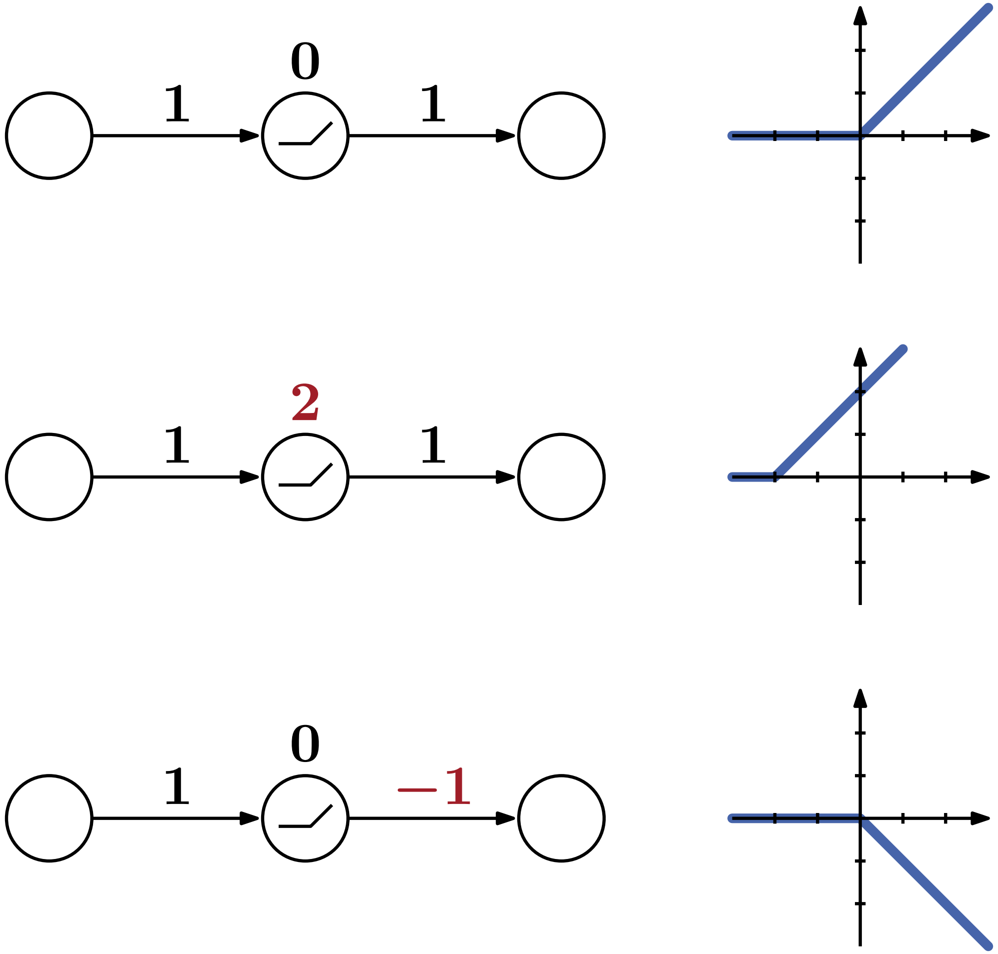


Funktionen von ReLU-Netzwerken

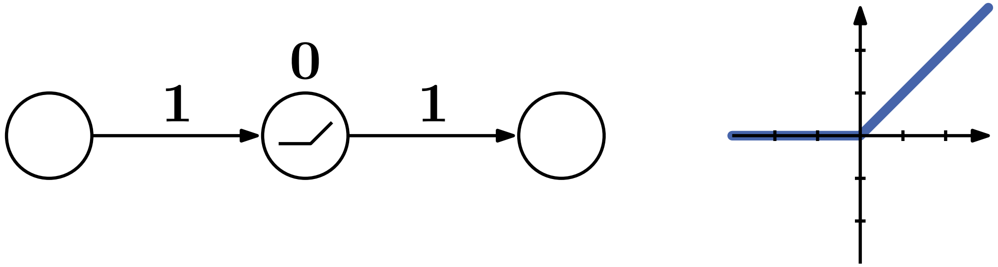


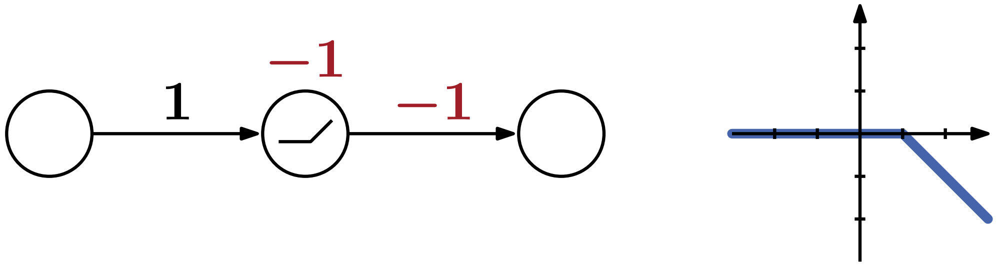


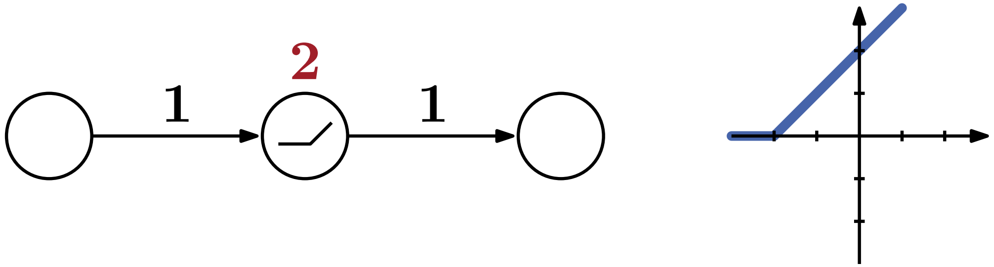
Funktionen von ReLU-Netzwerken

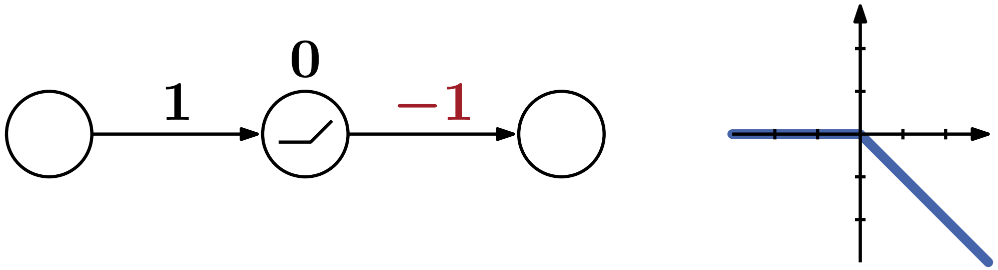


Funktionen von ReLU-Netzwerken

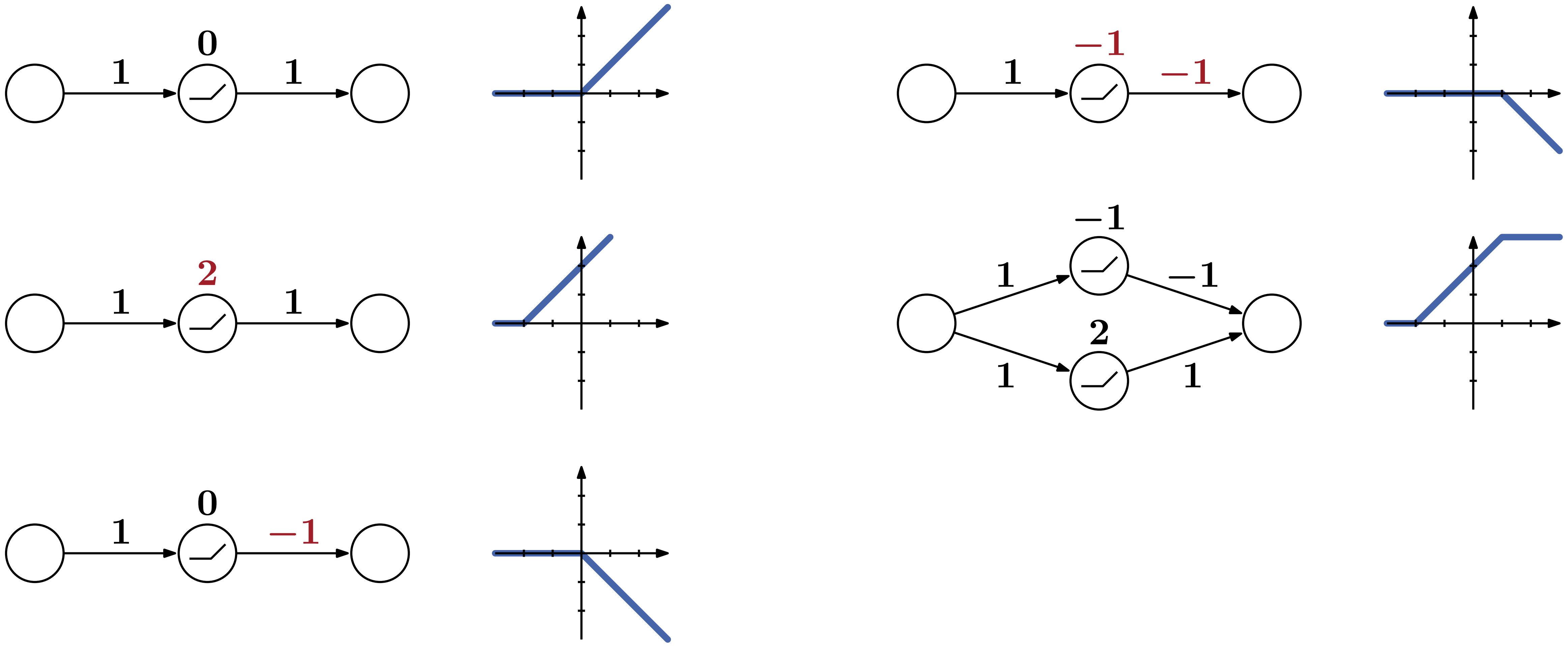




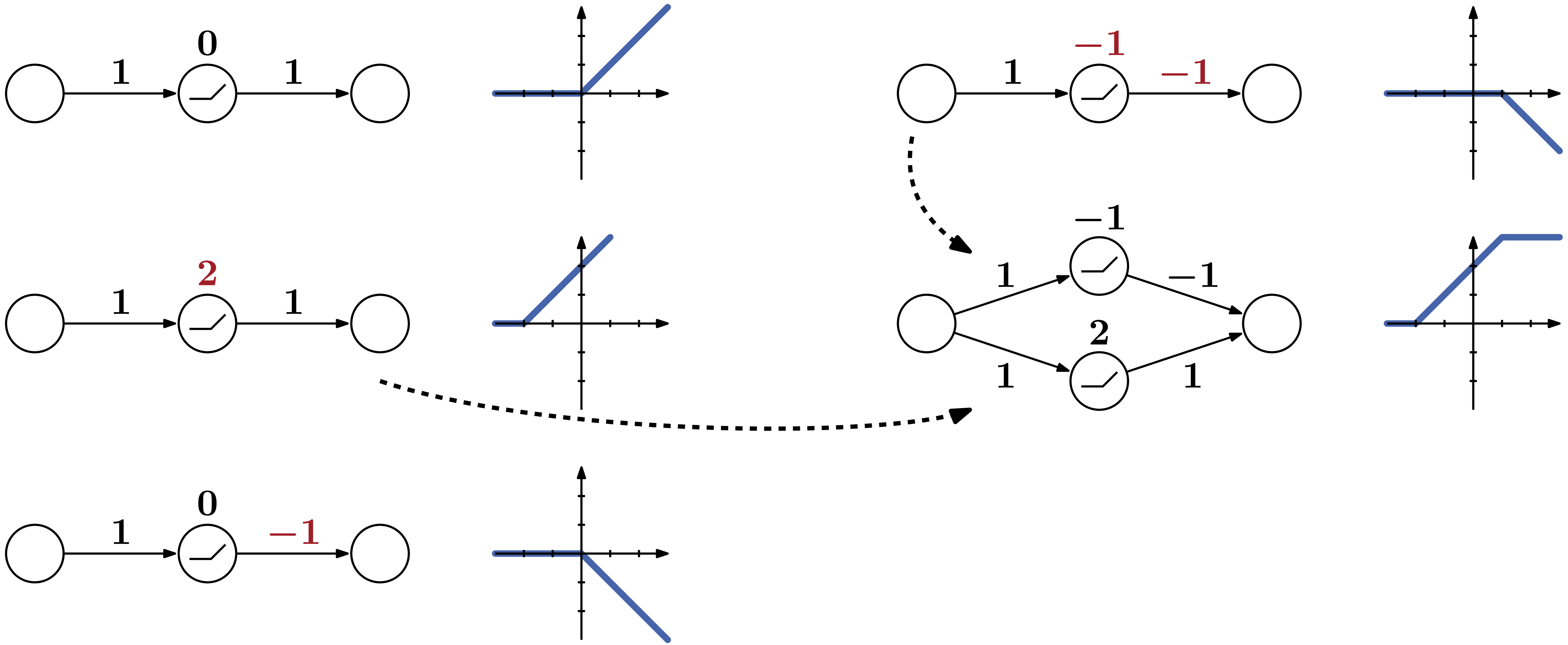




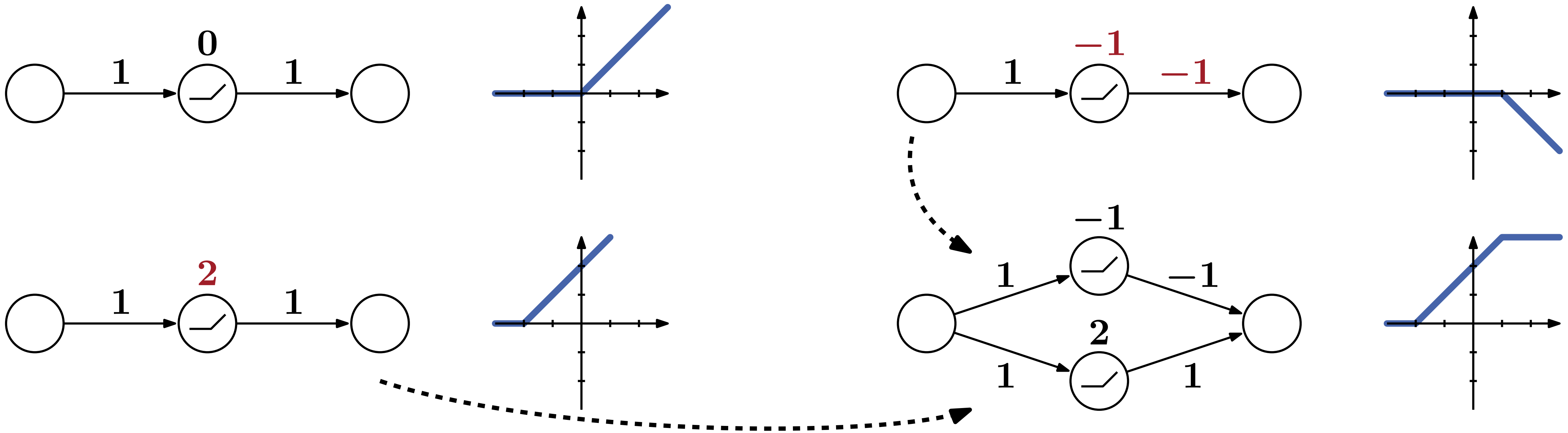
Funktionen von ReLU-Netzwerken

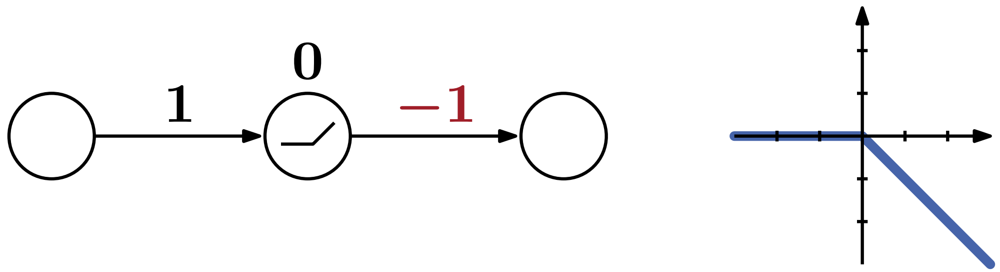


Funktionen von ReLU-Netzwerken



Funktionen von ReLU-Netzwerken





Beobachtungen:

- stetig und stückweise linear
- #Knickstellen = #ReLUs

Das Trainingsproblem

Trainingsproblem: $\textsc{TrainNN}$

Eingabe: Architektur (Graph) $+$
n Datenpunkte: $(x_i, y_i)_{i=1,\ldots,n}$
Eingabe Soll-Ausgabe

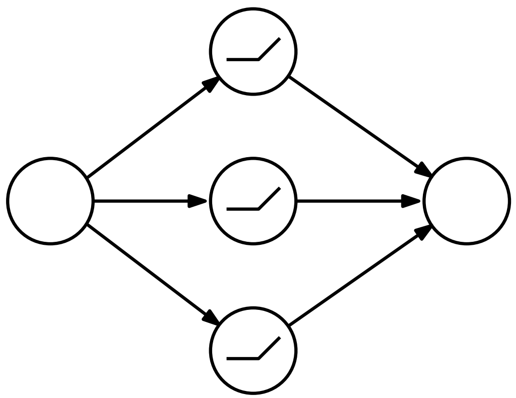
Frage: Existieren Parameter Θ, sodass für alle Datenpunkte gilt: $f_\Theta(x_i) = y_i$

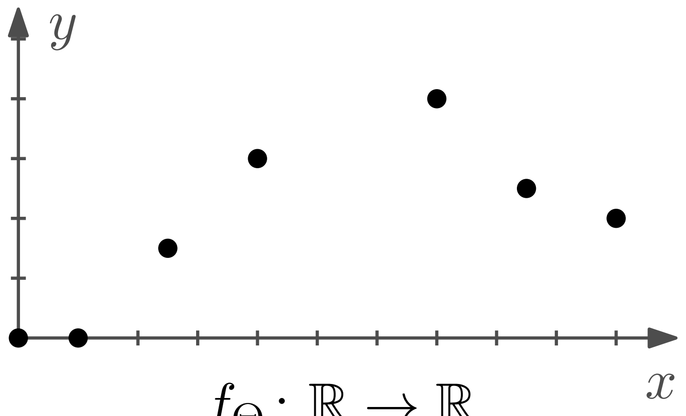
Das Trainingsproblem

Trainingsproblem: $\textsc{TrainNN}$

Eingabe: Architektur (Graph) $+$
n Datenpunkte: $(x_i, y_i)_{i=1,\ldots,n}$
Eingabe ↑ ↑ Soll-Ausgabe

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





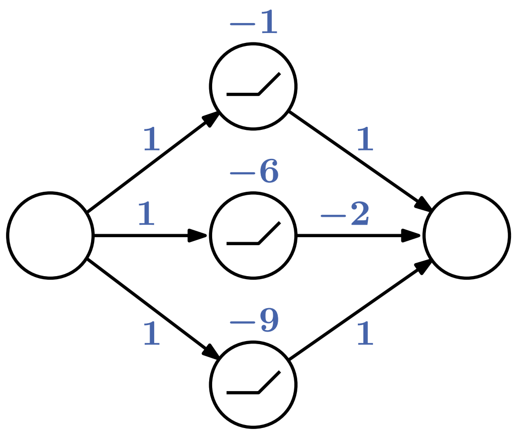
$f_\Theta : \mathbb{R} \to \mathbb{R}$

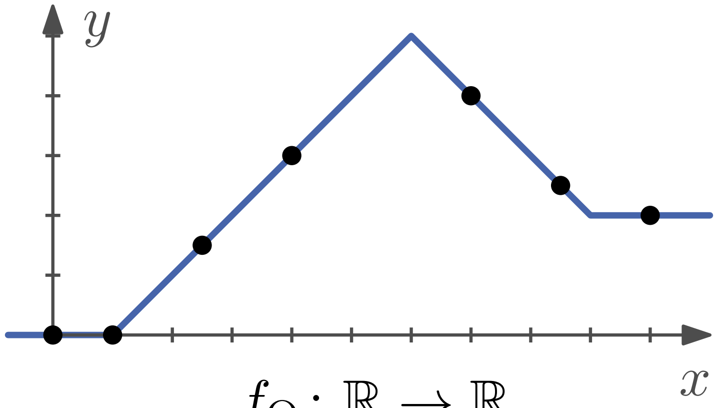
Das Trainingsproblem

Trainingsproblem: $\textsc{TrainNN}$

Eingabe: Architektur (Graph) +
n Datenpunkte: $(x_i, y_i)_{i=1,\ldots,n}$
Eingabe ↑ ↑ Soll-Ausgabe

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





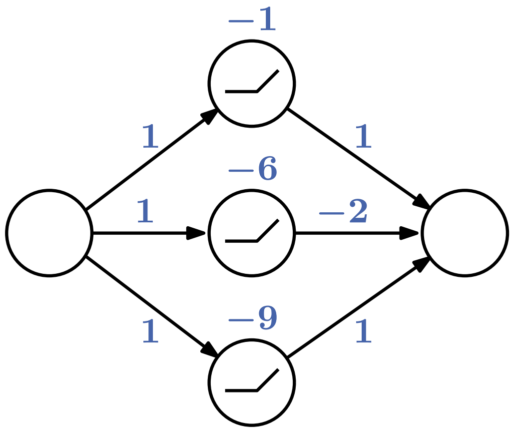
$f_\Theta : \mathbb{R} \to \mathbb{R}$

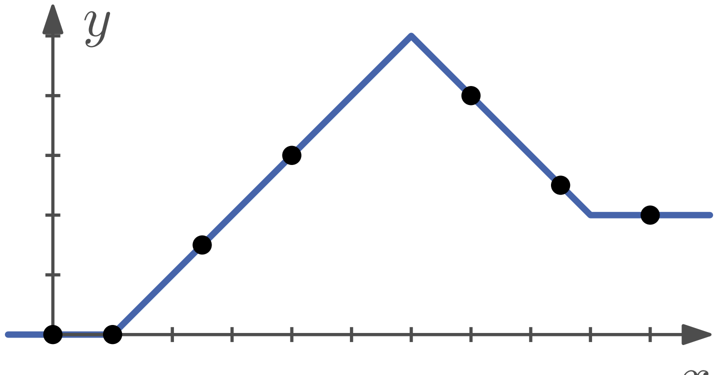
Das Trainingsproblem

Trainingsproblem: $\textsc{TrainNN}$

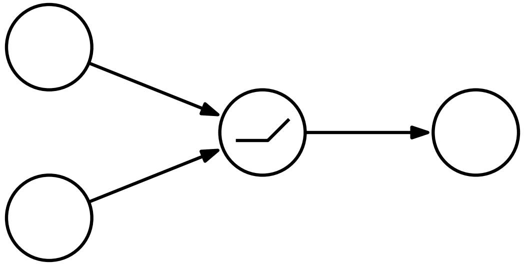
Eingabe: Architektur (Graph) +
n Datenpunkte: $({\color{red}x_i}, {\color{red}y_i})_{i=1,\ldots,n}$
Eingabe Soll-Ausgabe

Frage: Existieren Parameter ${\color{blue}\Theta}$, sodass für alle Datenpunkte gilt: $f_{\color{blue}\Theta}({\color{red}x_i}) = {\color{red}y_i}$

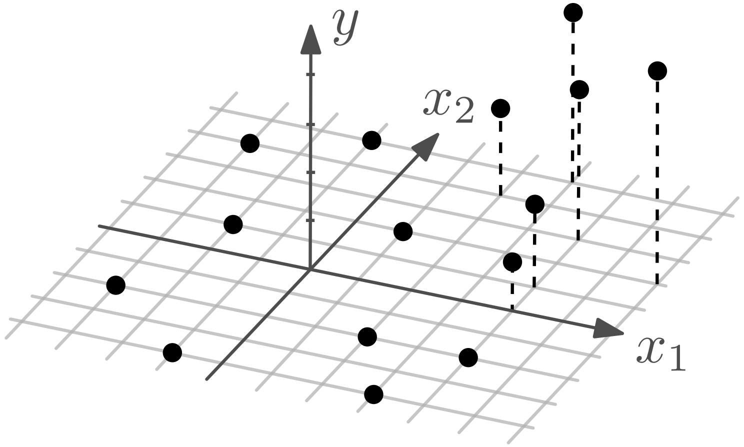




$f_\Theta : \mathbb{R} \to \mathbb{R}$



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

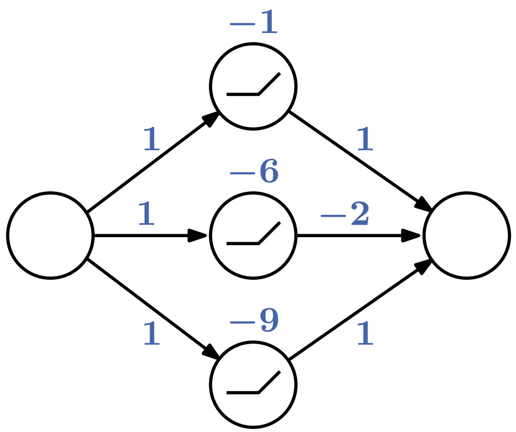


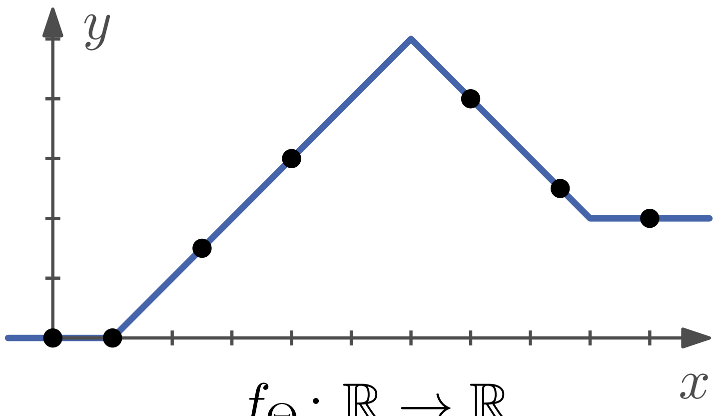
Das Trainingsproblem

Trainingsproblem: $\textsc{TrainNN}$

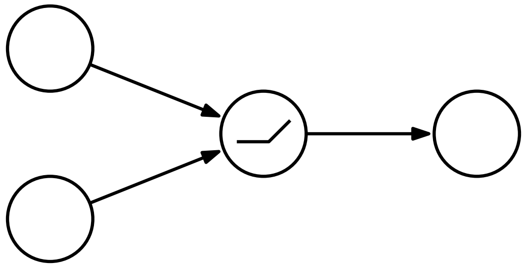
Eingabe: Architektur (Graph) $+$
n Datenpunkte: $(\color{red}{x_i}, \color{red}{y_i})_{i=1,\ldots,n}$
Eingabe ↑ ↑ Soll-Ausgabe

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

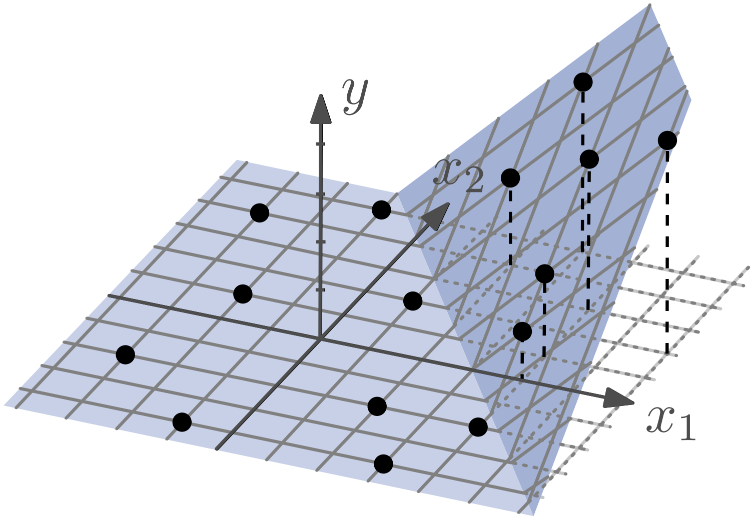




$f_\Theta : \mathbb{R} \to \mathbb{R}$



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



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?

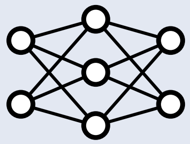
- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

  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]

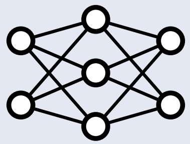
  Frage: Verallgemeinerbar auf kompliziertere Architekturen?

- $\exists\mathbb{R}$-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?

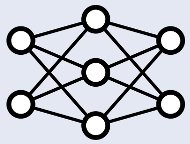
- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

  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):

$\textsc{TrainNN}$ ist ...

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

  Frage: Verallgemeinerbar auf kompliziertere Architekturen?

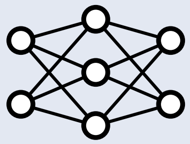
- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

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

Theorem: $\textsc{TrainNN}$ 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?

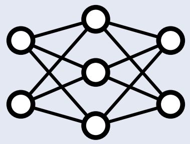
- $\exists\mathbb{R}$-schwer [Zhang 1992] [Abrahamsen, Kleist, Miltzow 2021]

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

Schritt 1: TRAINNN ist in $\exists\mathbb{R}$

„höchstens“ so schwierig wie ETR

Schritt 2: 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: $\textsc{TrainNN} \rightsquigarrow \mathrm{ETR}$

$\textsc{TrainNN}$ als äquivalente ETR-Formel:

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

als Polynomgleichung formulieren

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

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

$\textsc{TrainNN}$ als äquivalente ETR-Formel:

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

als Polynomgleichung formulieren

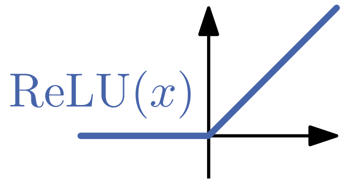
$\rightsquigarrow \textsc{TrainNN} \in \exists\mathbb{R}$ $\square$

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



Reduktion: $\mathrm{ETR} \rightsquigarrow \textsc{Train}\mathrm{NN}$

Gegeben:

Beliebige Instanz von ETR, z.B.:

$$\exists a, b, c \in \mathbb{R} : a + b = c \wedge a \cdot b = 1$$

reelle Variablen

lineare Terme

nichtlineare Terme

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

Gegeben:

Beliebige Instanz von ETR, z.B.:

$$\exists a, b, c \in \mathbb{R} : a + b = c \wedge a \cdot b = 1$$

reelle Variablen

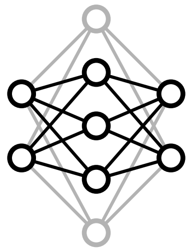
lineare Terme

nichtlineare Terme

Zu konstruieren:

Äquivalente Instanz von $\textsc{TrainNN}$:

- Architektur
  - zwei Inputs und zwei Outputs
  - eine verstecke Schicht
  - feste Anzahl an ReLUs
- Trainingsdaten
  - Eingaben
  - Soll-Ausgaben



Reduktion: ETR $\rightsquigarrow$ TrainNN

Gegeben:

Beliebige Instanz von ETR, z.B.:

$$\exists a, b, c \in \mathbb{R} : a + b = c \land a \cdot b = 1$$

reelle Variablen

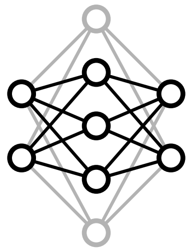
lineare Terme

nichtlineare Terme

Zu konstruieren:

Äquivalente Instanz von TrainNN:

- Architektur
  - zwei Inputs und zwei Outputs
  - eine versteckte Schicht
  - feste Anzahl an ReLUs
- Trainingsdaten
  - Eingaben
  - Soll-Ausgaben



Formel wahr $\iff$ Instanz exakt trainierbar (Funktion geht durch alle Datenpunkte)

Teil 1: Reelle Variablen

ETR: $\exists a \in \mathbb{R}$

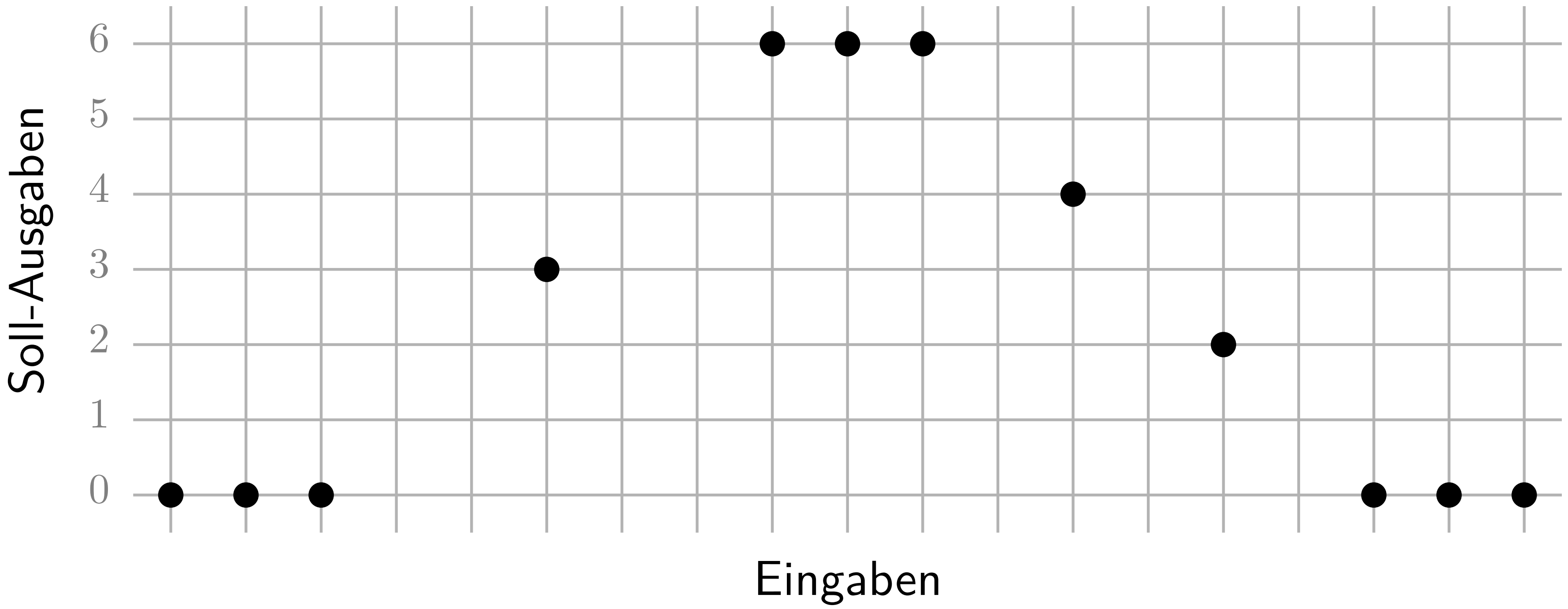
Ziel: Simulation von „$\exists a \in \mathbb{R}$“ als Instanz von $\textsc{Train}\mathrm{NN}$

Teil 1: Reelle Variablen

ETR: $\exists a \in \mathbb{R}$

Ziel: Simulation von „$\exists a \in \mathbb{R}$“ als Instanz von $\textsc{TrainNN}$

12 Datenpunkte und 4 ReLUs:

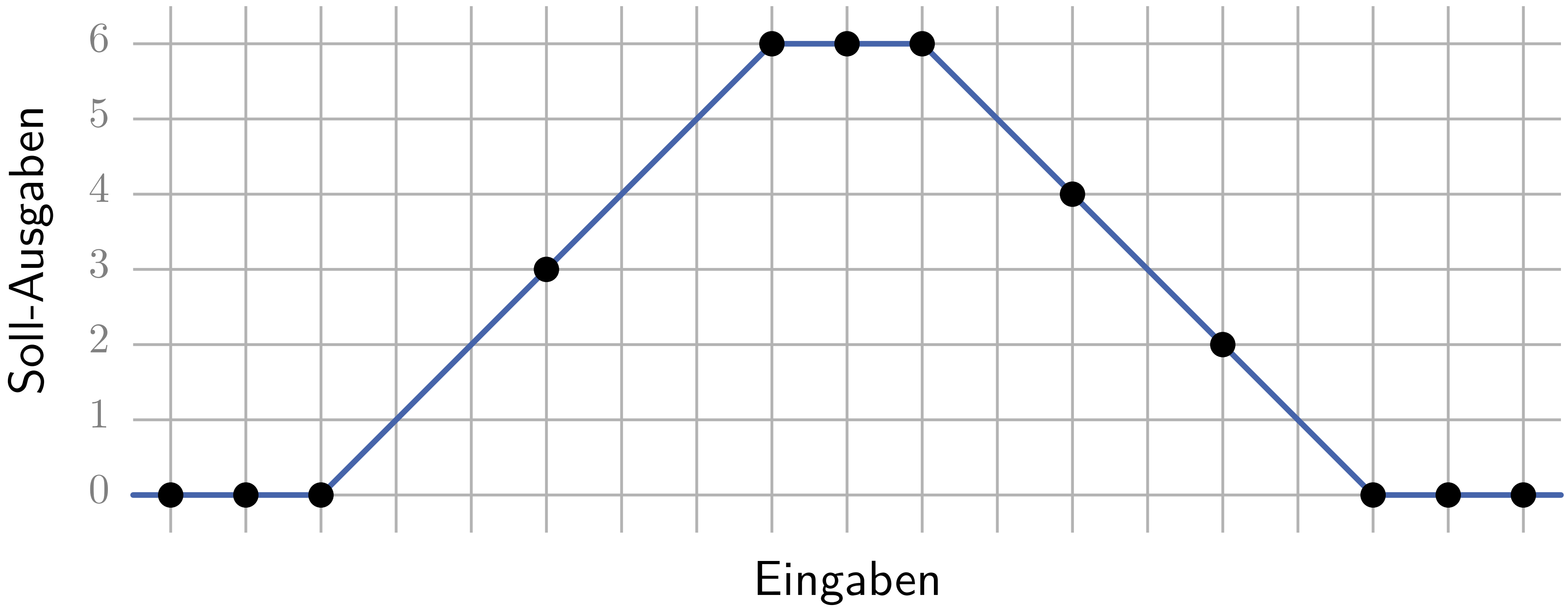


Teil 1: Reelle Variablen

ETR: $\exists a \in \mathbb{R}$

Ziel: Simulation von „$\exists a \in \mathbb{R}$“ als Instanz von $\textsc{TrainNN}$

12 Datenpunkte und 4 ReLUs:

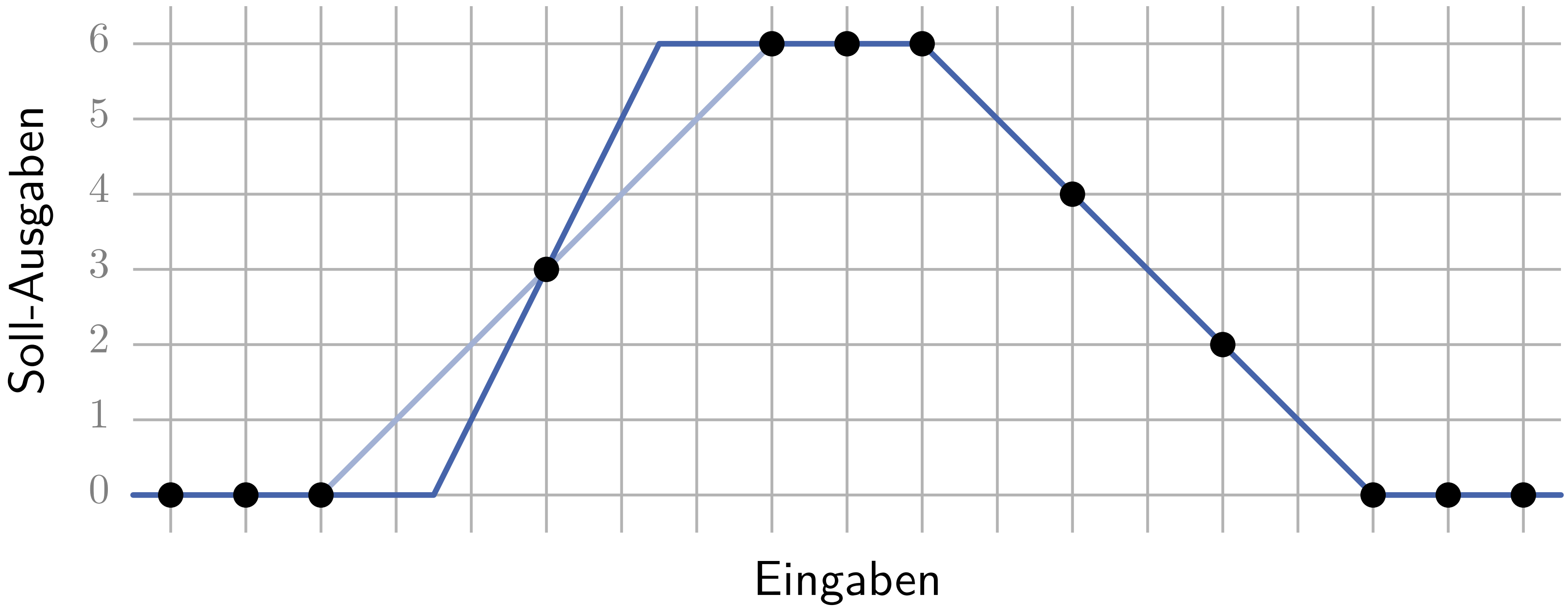


Teil 1: Reelle Variablen

ETR: $\exists a \in \mathbb{R}$

Ziel: Simulation von „$\exists a \in \mathbb{R}$“ als Instanz von $\textsc{TrainNN}$

12 Datenpunkte und 4 ReLUs:

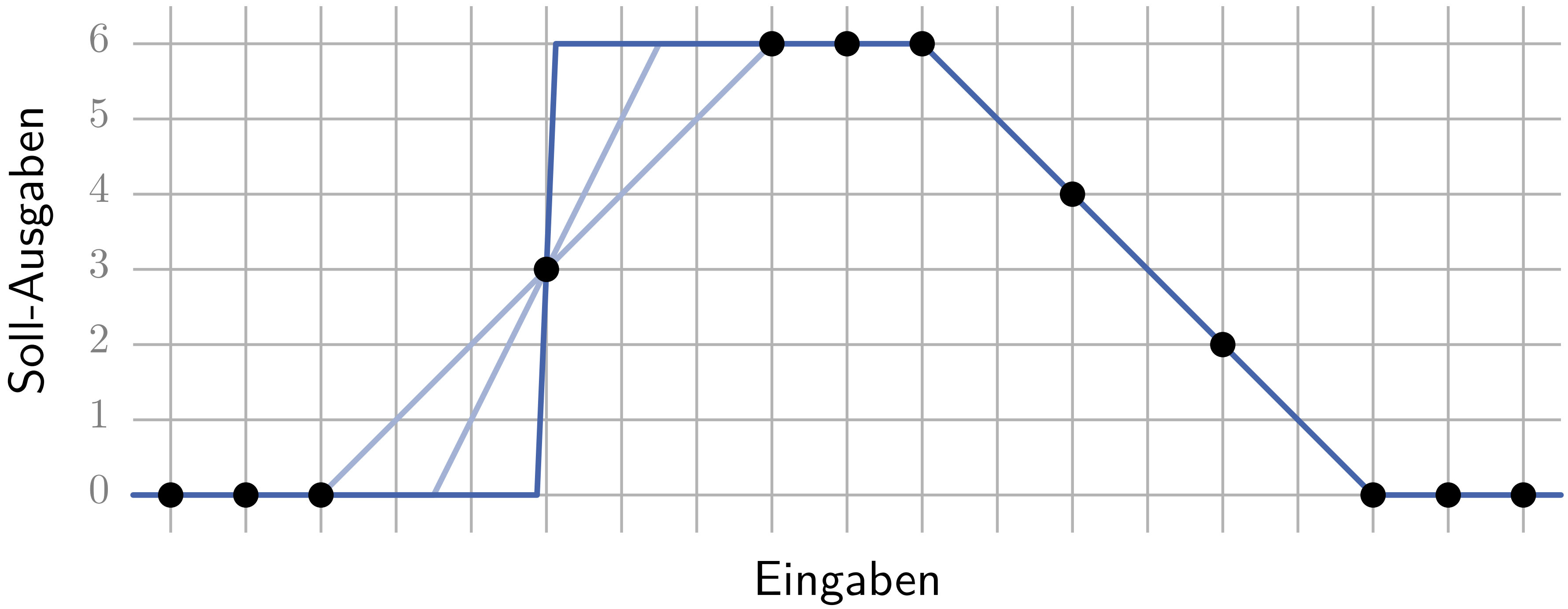


Teil 1: Reelle Variablen

ETR: $\exists a \in \mathbb{R}$

Ziel: Simulation von „$\exists a \in \mathbb{R}$“ als Instanz von $\textsc{TrainNN}$

12 Datenpunkte und 4 ReLUs:

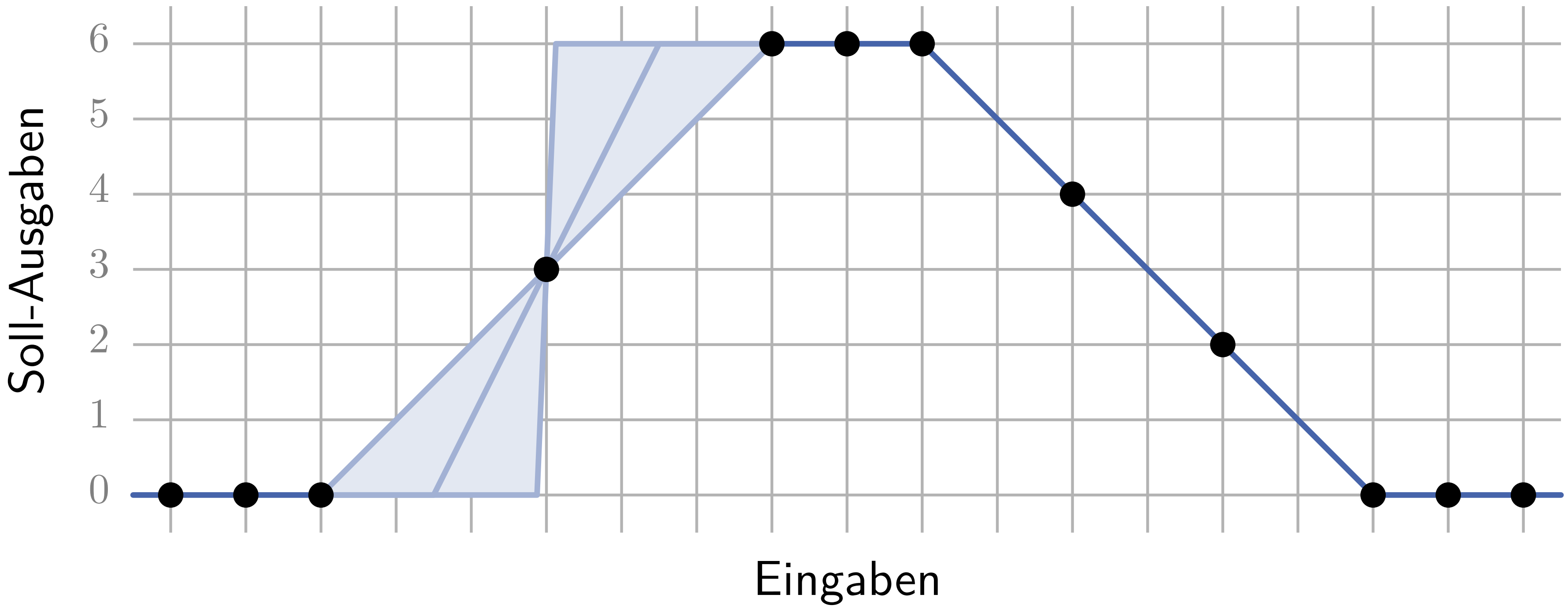


Teil 1: Reelle Variablen

ETR: $\exists a \in \mathbb{R}$

Ziel: Simulation von „$\exists a \in \mathbb{R}$“ als Instanz von $\textsc{TrainNN}$

12 Datenpunkte und 4 ReLUs:

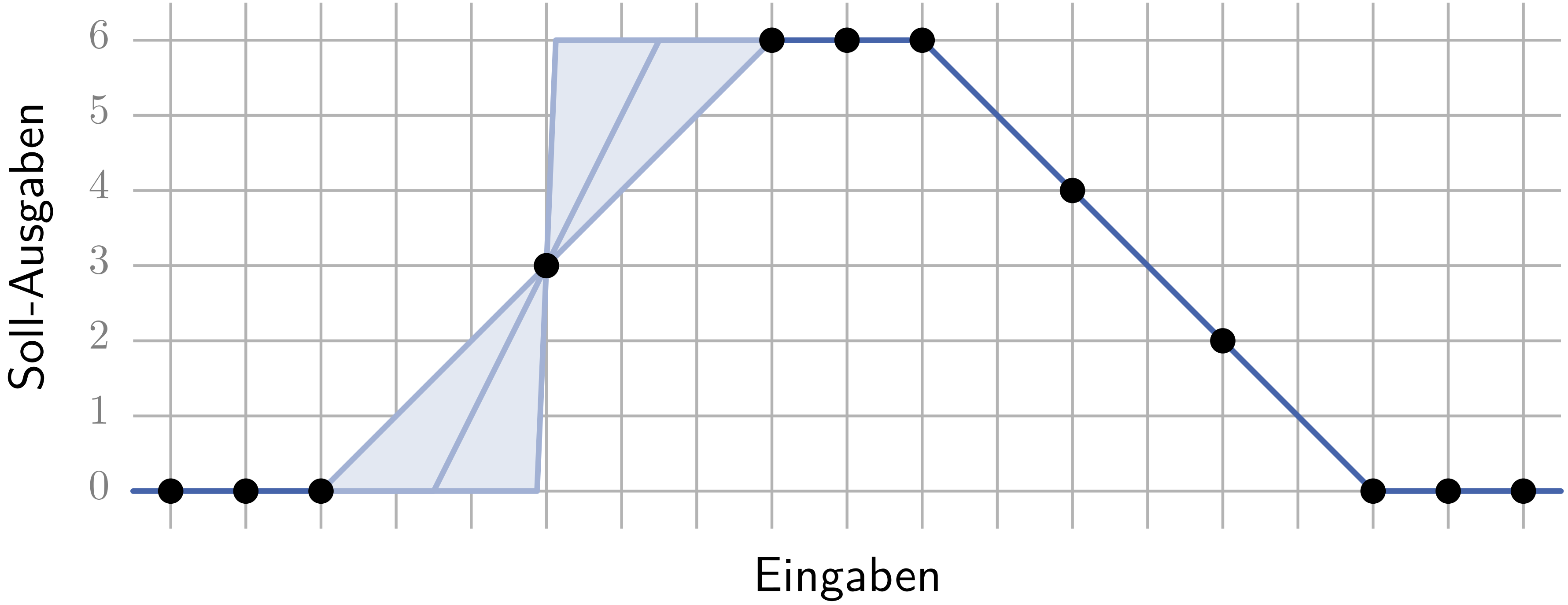


Teil 1: Reelle Variablen

ETR: $\exists a \in \mathbb{R}$

Ziel: Simulation von „$\exists a \in \mathbb{R}$“ als Instanz von $\textsc{TrainNN}$

12 Datenpunkte und 4 ReLUs:



Beobachtung:

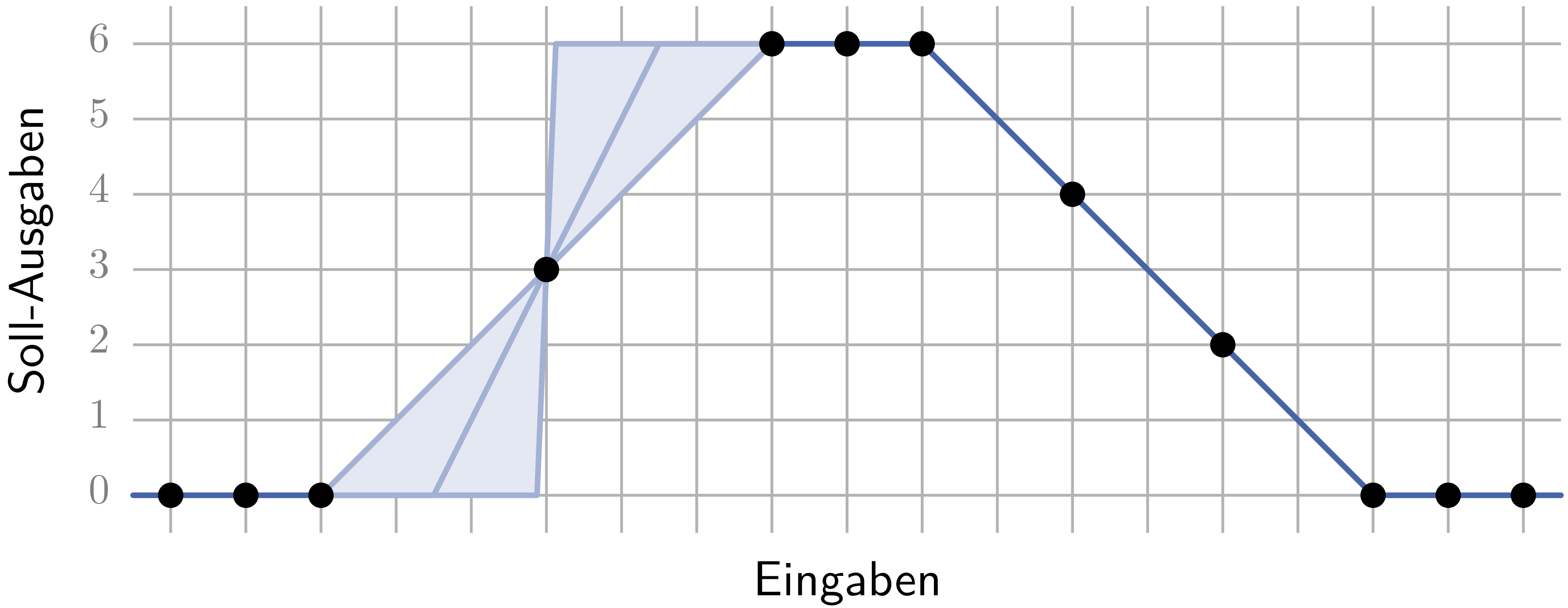
Funktionen unterscheiden sich nur in der Steigung.

Teil 1: Reelle Variablen

ETR: $\exists a \in \mathbb{R}$

Ziel: Simulation von „$\exists a \in \mathbb{R}$“ als Instanz von $\textsc{TrainNN}$

12 Datenpunkte und 4 ReLUs:



Beobachtung:

Funktionen unterscheiden sich nur in der Steigung.

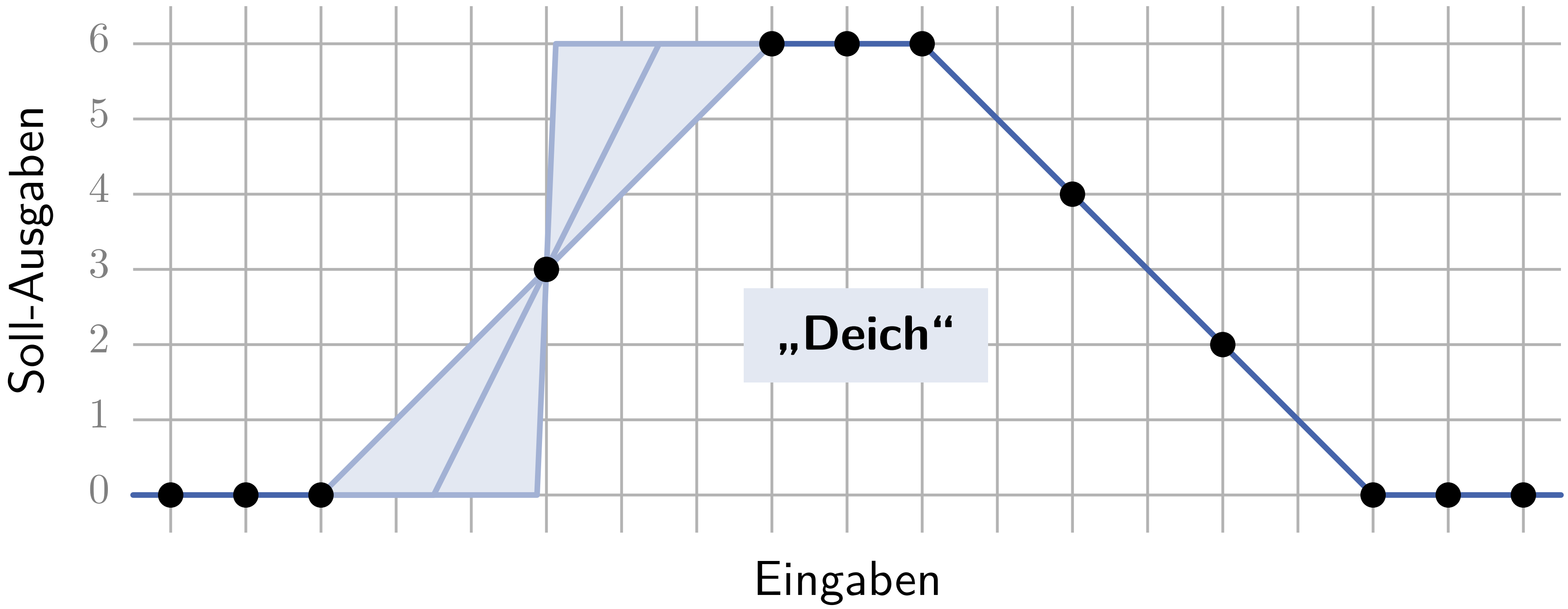
$\rightsquigarrow$ Steigung $\hat{=}$ Wert

Teil 1: Reelle Variablen

ETR: $\exists a \in \mathbb{R}$

Ziel: Simulation von „$\exists a \in \mathbb{R}$“ als Instanz von $\textsc{TrainNN}$

12 Datenpunkte und 4 ReLUs:



Beobachtung:

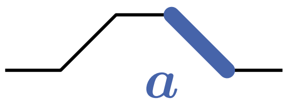
Funktionen unterscheiden sich nur in der Steigung.

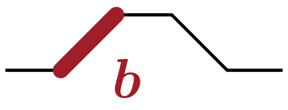
$\rightsquigarrow$ Steigung $\hat{=}$ Wert

Teil 2: Lineare Zusammenhänge

ETR: lineare Zusammenhänge, z.B. $a + b = 5$

Ziel: Simulation von „$a + b = 5$“ als Instanz von $\textsc{TrainNN}$



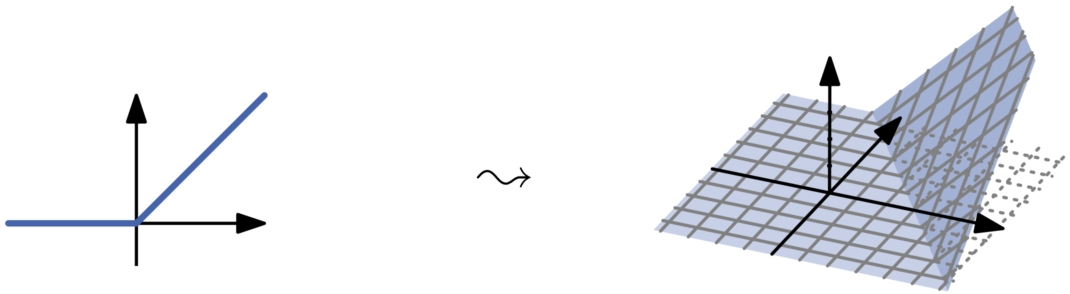


Teil 2: Lineare Zusammenhänge

ETR: lineare Zusammenhänge, z.B. $a + b = 5$

Ziel: Simulation von „$a + b = 5$“ als Instanz von $\textsc{TrainNN}$

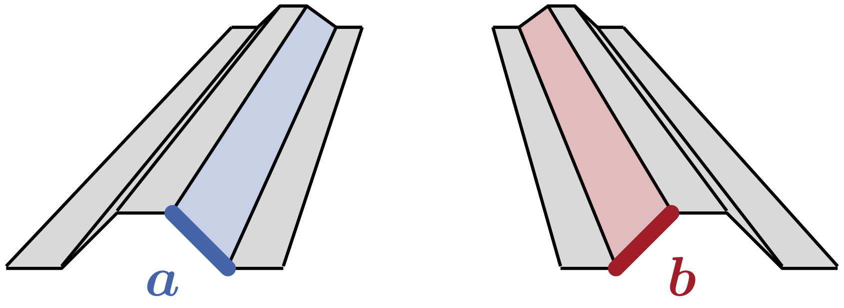
Idee: ausnutzen, dass es zwei Input-Neuronen gibt



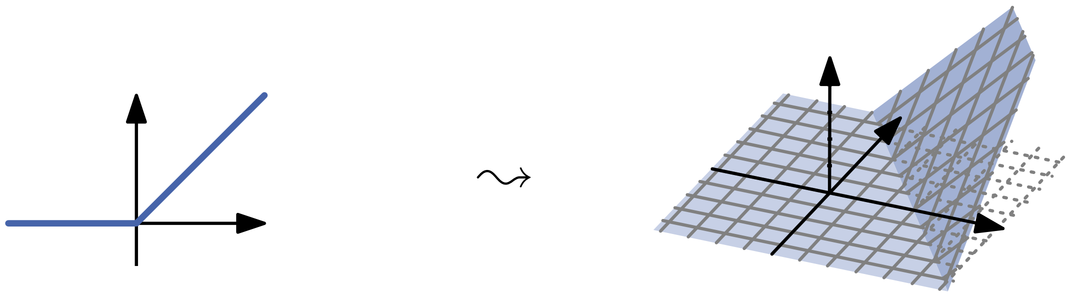
Teil 2: Lineare Zusammenhänge

ETR: lineare Zusammenhänge, z.B. $a + b = 5$

Ziel: Simulation von „$a + b = 5$“ als Instanz von $\textsc{TrainNN}$



Idee: ausnutzen, dass es zwei Input-Neuronen gibt



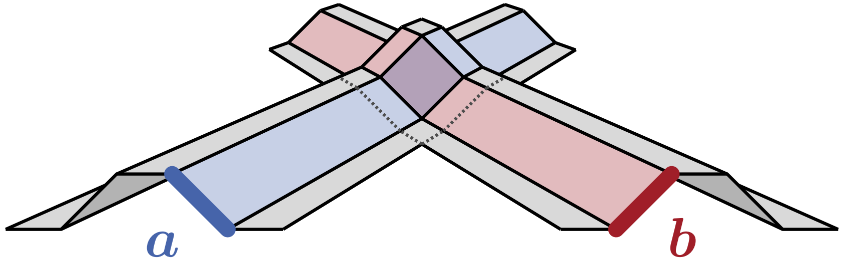
Beobachtungen:

- Variablen-Deiche: 2D $\rightsquigarrow$ 3D
  Wie vorher: Steigung $\hat{=}$ Wert

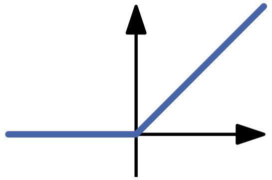
Teil 2: Lineare Zusammenhänge

ETR: lineare Zusammenhänge, z.B. $a + b = 5$

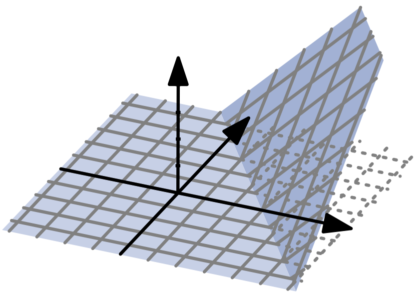
Ziel: Simulation von „$a + b = 5$“ als Instanz von $\textsc{TrainNN}$



Idee: ausnutzen, dass es zwei Input-Neuronen gibt



$\rightsquigarrow$



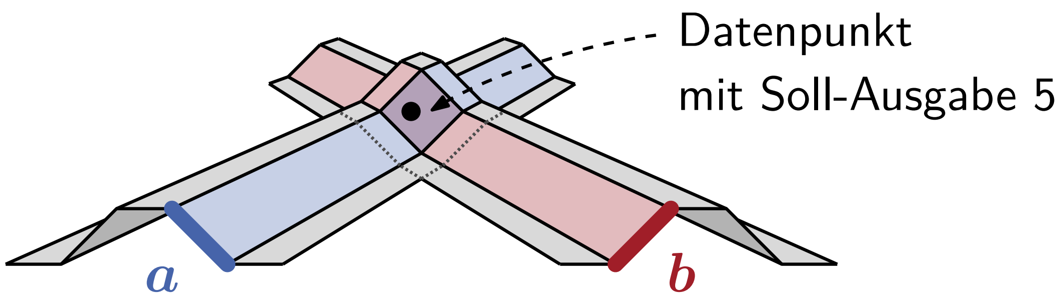
Beobachtungen:

- Variablen-Deiche: 2D $\rightsquigarrow$ 3D
  Wie vorher: Steigung $\hat{=}$ Wert
- „Interferenz“ in Überlagerung

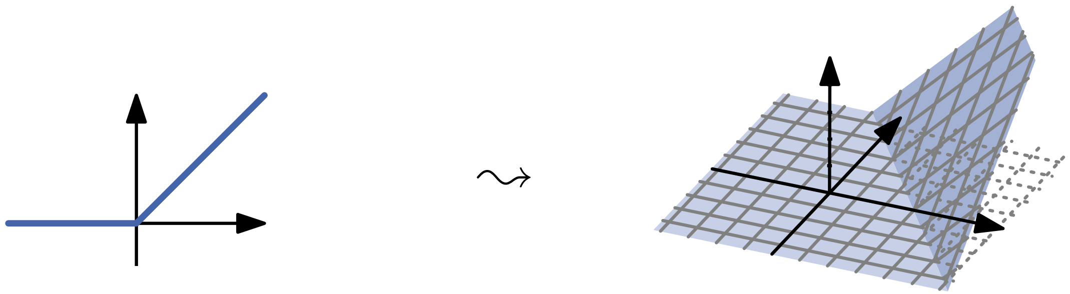
Teil 2: Lineare Zusammenhänge

ETR: lineare Zusammenhänge, z.B. $a + b = 5$

Ziel: Simulation von „$a + b = 5$“ als Instanz von $\textsc{TrainNN}$



Idee: ausnutzen, dass es zwei Input-Neuronen gibt



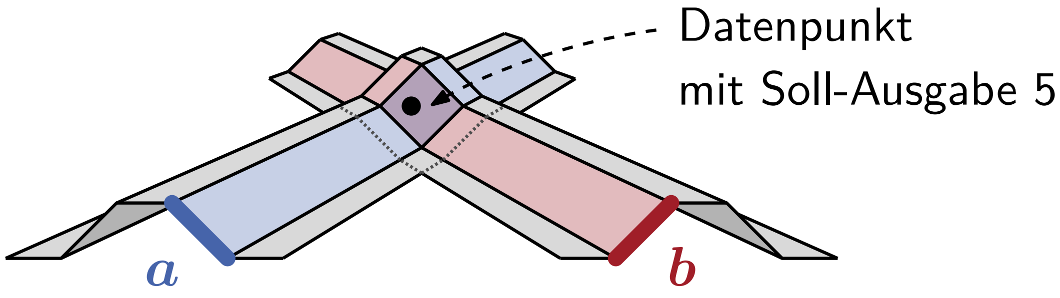
Beobachtungen:

- Variablen-Deiche: 2D $\rightsquigarrow$ 3D
  Wie vorher: Steigung $\hat{=}$ Wert
- „Interferenz“ in Überlagerung
- Datenpunkt in Überlagerung
  - blau steiler $\rightsquigarrow$ rot flacher
  - rot steiler $\rightsquigarrow$ blau flacher

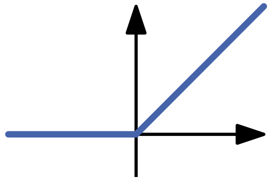
Teil 2: Lineare Zusammenhänge

ETR: lineare Zusammenhänge, z.B. $a + b = 5$

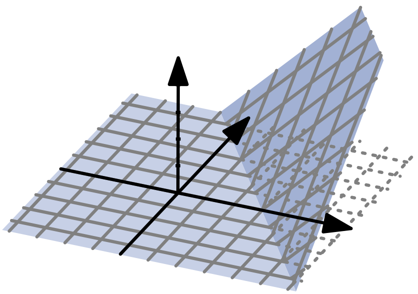
Ziel: Simulation von „$a + b = 5$“ als Instanz von $\textsc{TrainNN}$



Idee: ausnutzen, dass es zwei Input-Neuronen gibt



$\rightsquigarrow$



Beobachtungen:

- Variablen-Deiche: 2D $\rightsquigarrow$ 3D
  Wie vorher: Steigung $\hat{=}$ Wert
- „Interferenz“ in Überlagerung
- Datenpunkt in Überlagerung
  - blau steiler $\rightsquigarrow$ rot flacher
  - rot steiler $\rightsquigarrow$ blau flacher
- weitere lineare Terme, insb.:
  - Gleichheit
  - Addition (3er-Überlagerung)

Teil 3: Zusammensetzen

Reduktion: $\mathrm{ETR} \rightsquigarrow \textsc{Train}\mathrm{NN}$

Gegeben:

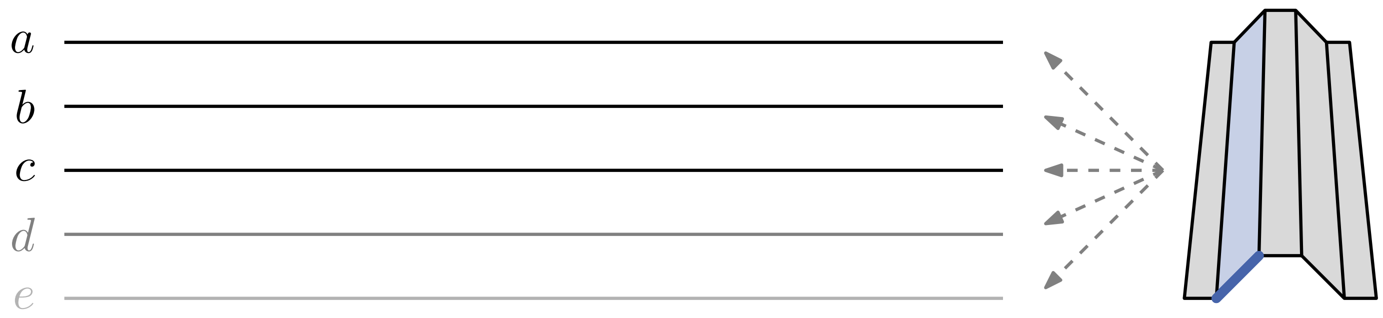
- Variablen: $a, b, c, \ldots \in \mathbb{R}$
- lineare Terme: $a + b = c$
- nichtlineare Terme: $a \cdot b = 1$

Teil 3: Zusammensetzen

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

Gegeben:

- Variablen: $a, b, c, \ldots \in \mathbb{R}$
- lineare Terme: $a + b = c$
- nichtlineare Terme: $a \cdot b = 1$

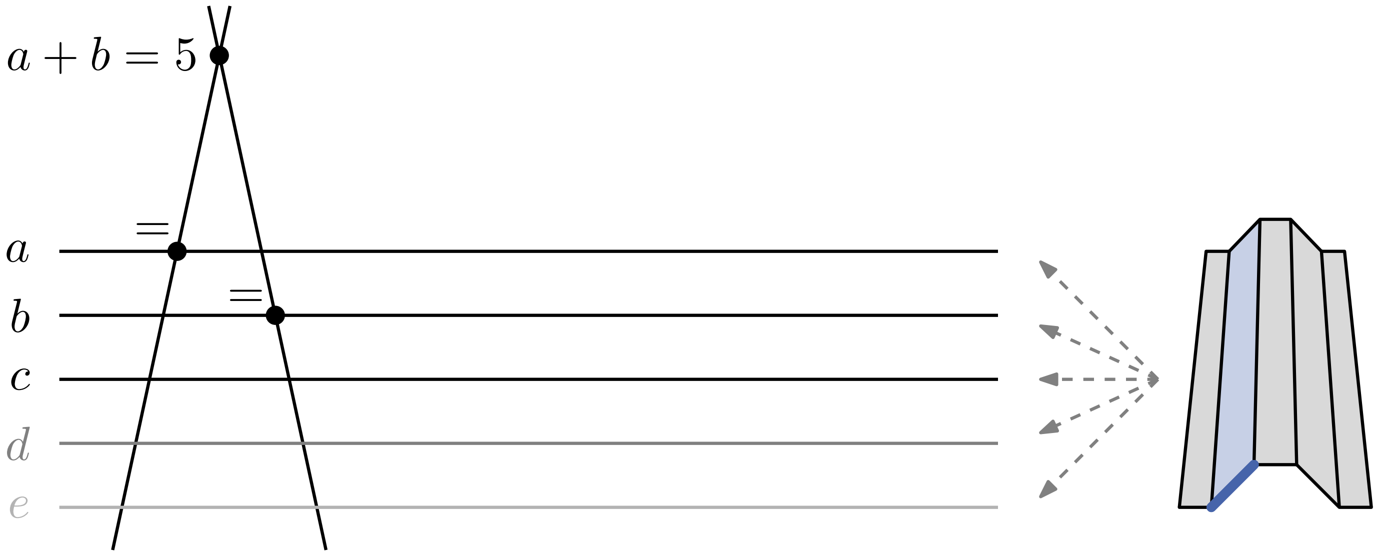


Teil 3: Zusammensetzen

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

Gegeben:

- Variablen: $a, b, c, \ldots \in \mathbb{R}$
- lineare Terme: $a + b = c$
- nichtlineare Terme: $a \cdot b = 1$

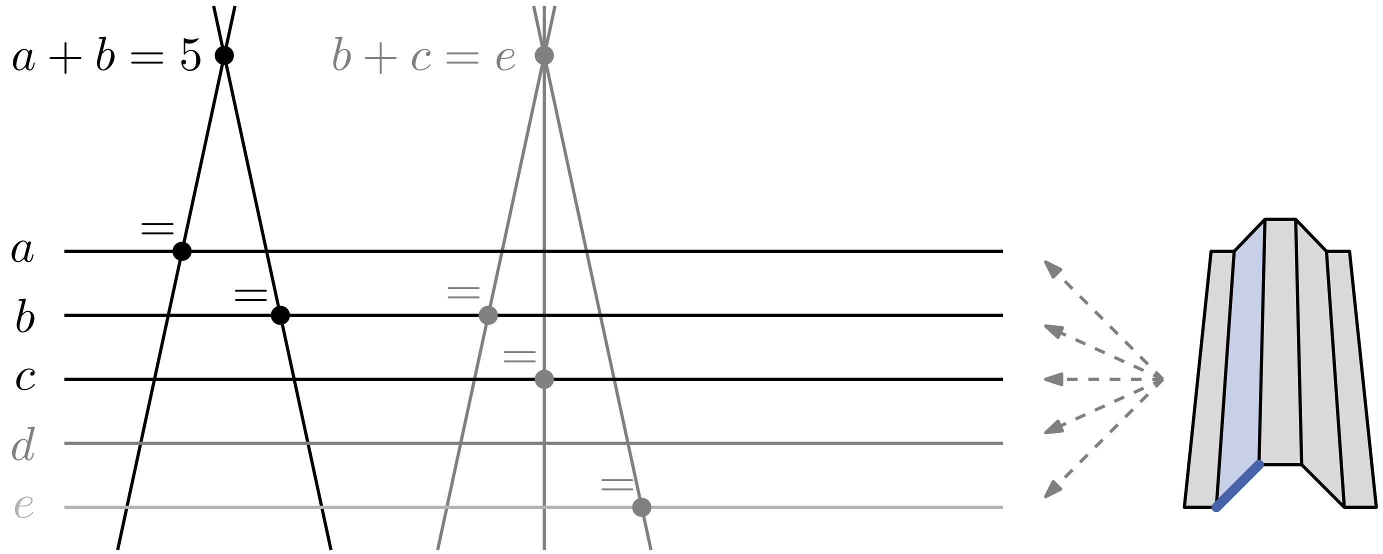


Teil 3: Zusammensetzen

Reduktion: $\text{ETR} \rightsquigarrow \text{TrainNN}$

Gegeben:

- Variablen: $a, b, c, \ldots \in \mathbb{R}$
- lineare Terme: $a + b = c$
- nichtlineare Terme: $a \cdot b = 1$

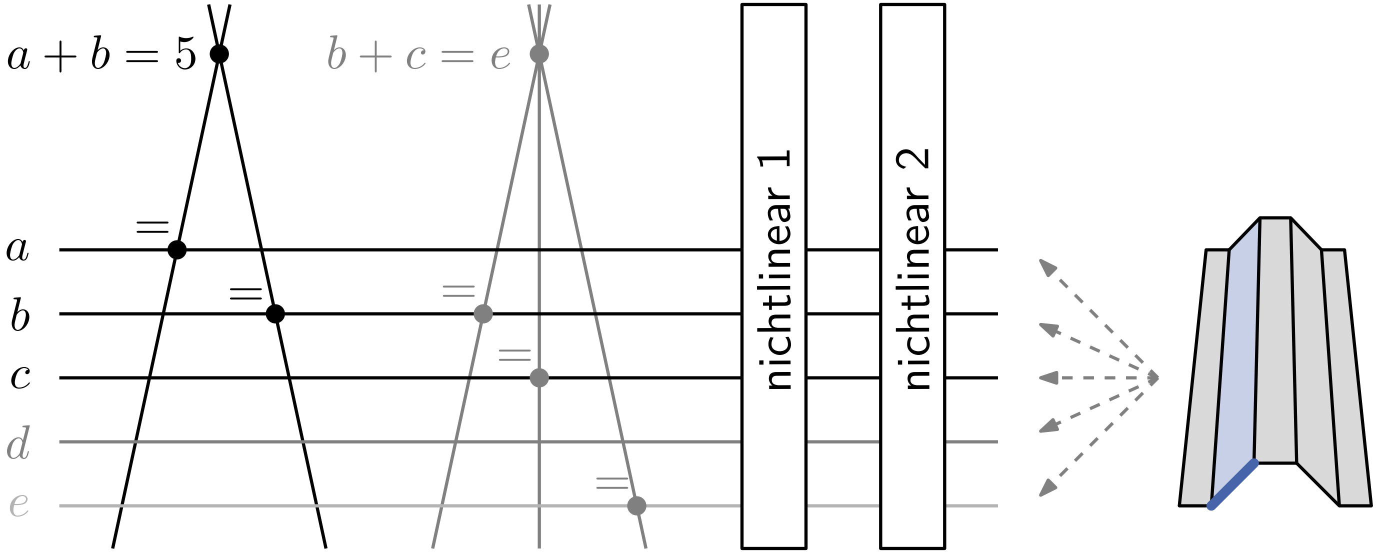


Teil 3: Zusammensetzen

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

Gegeben:

- Variablen: $a, b, c, \ldots \in \mathbb{R}$
- lineare Terme: $a + b = c$
- nichtlineare Terme: $a \cdot b = 1$

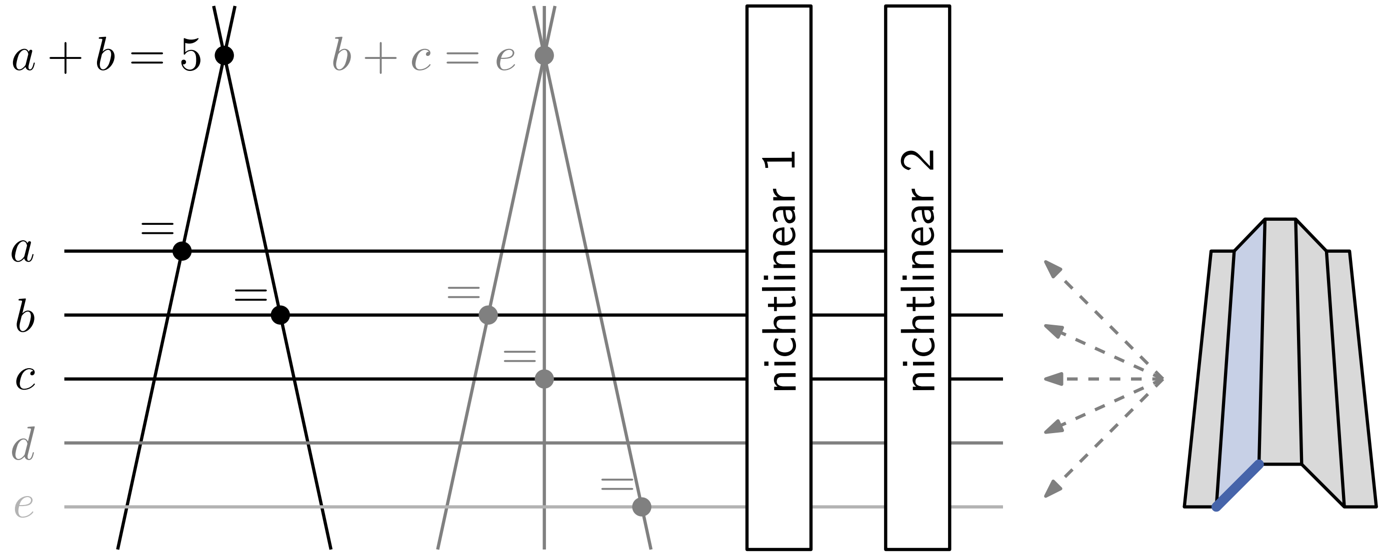


Teil 3: Zusammensetzen

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

Gegeben:

- Variablen: $a, b, c, \ldots \in \mathbb{R}$
- lineare Terme: $a + b = c$
- nichtlineare Terme: $a \cdot b = 1$

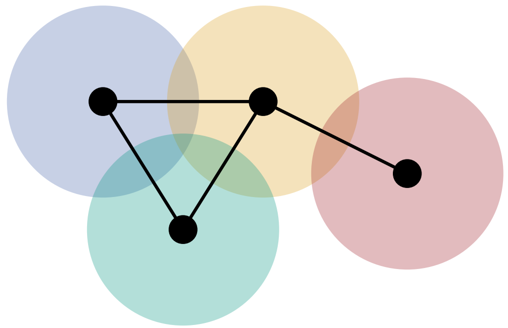


Zusammenfassung:

- $\textsc{TrainNN}$ simuliert ETR
  $\rightsquigarrow$ $\textsc{TrainNN}$ is $\exists\mathbb{R}$-schwer
- Schwere des Spezialfalls
  $\rightsquigarrow$ Schwere im Allgemeinen
- keine unrealistischen Annahmen

Zusammenfassung

Hyperbolische Geometrie

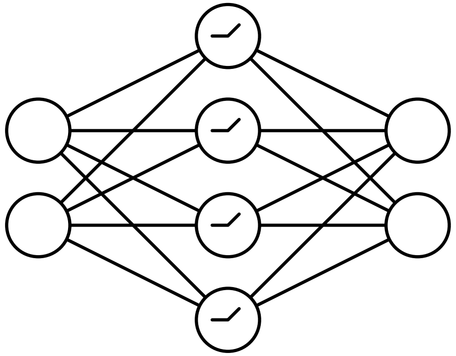


∃ℝ-Schwere in der hyperbolischen Ebene

[EuroCG 2023]

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

Training neuronaler Netze

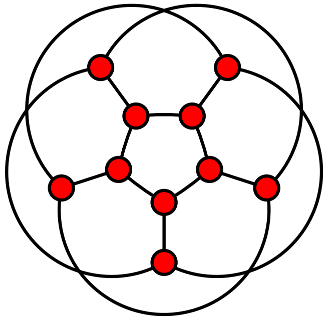


∃ℝ-vollständig unter realistischen Annahmen

[NeurIPS 2023]

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

Lombardi-Zeichnungen

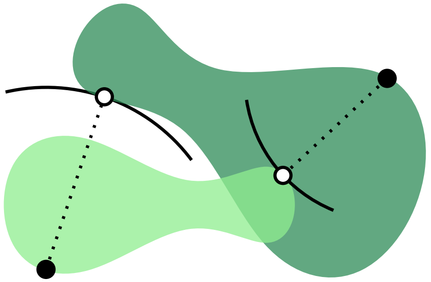


Graphvisualisierung unter geometrischen Nebenbedingungen

[GD 2023]

[**Jungeblut**]

Hausdorff-Distanz



Distanzmaß für Mengen
$\rightsquigarrow$ ∀∃ℝ-vollständig

[SoCG 2022]

[Discrete Comput. Geom. 2024]

[**Jungeblut**, Kleist, Miltzow]

Erkennung von Unit-Disk-Graphen ist in $\exists\mathbb{R}$

Problem: Unit-Disk-Graphen erkennen

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

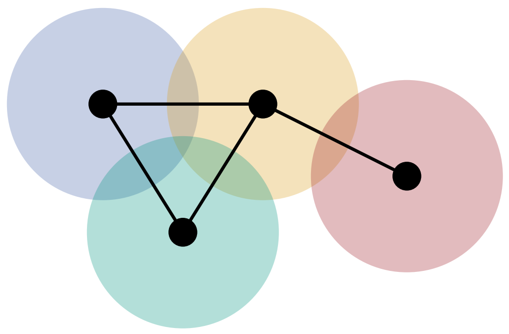
Finde: Schnittrepräsentation von G bestehend aus Einheitskreisen.

$$(x_u - x_v)^2 + (y_u - y_v)^2 \leq 4$$

Quadrieren

$$\sqrt{(x_u - x_v)^2 + (y_u - y_v)^2} \leq 2$$

Satz des Pythagoras



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

Kanten

$$\wedge \bigwedge_{uv \notin E} \|p_u - p_v\| > 2$$

nicht-Kanten

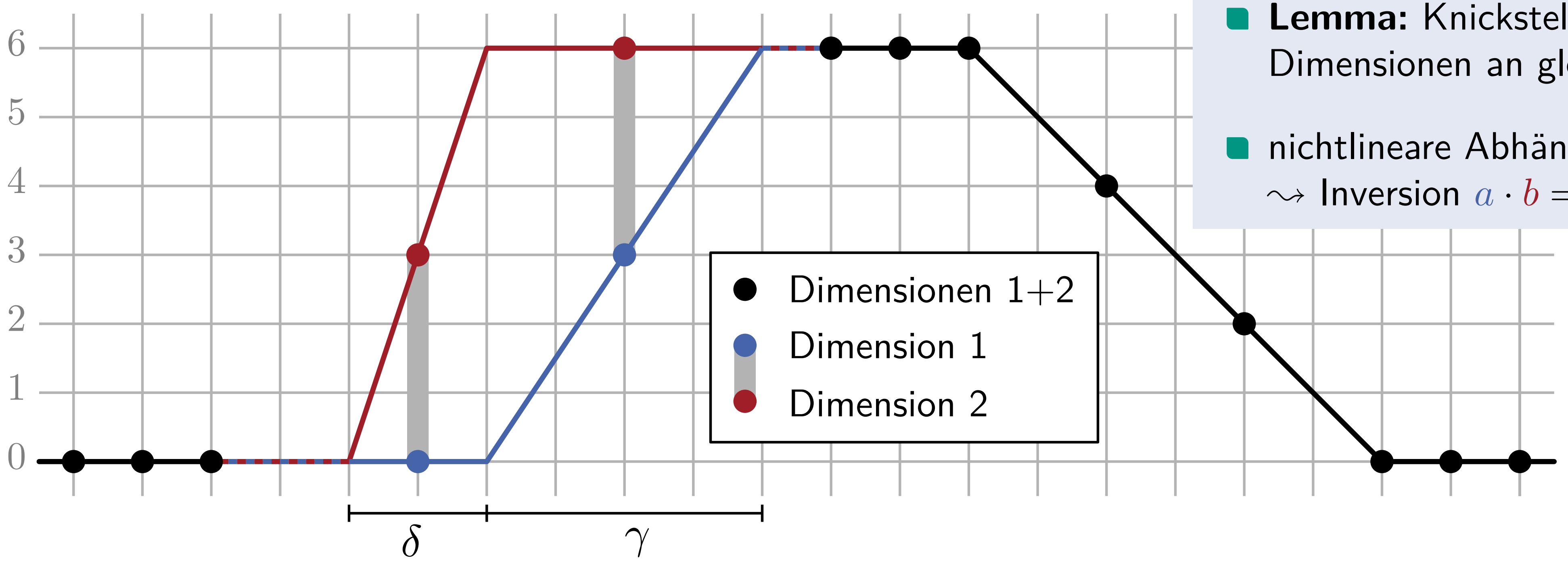
Reduktion: Nichtlineare Terme

ETR: nichtlinearer Zusammenhang, z.B. $a \cdot b = 1$

Ziel: Simulation von „$a \cdot b = 1$“ als Instanz von $\textsc{TrainNN}$

Idee: nutze zweite Output-Dimension

- 5 ReLUs $\rightsquigarrow$ 5 Knickstellen
- **Lemma:** Knickstellen in beiden Output-Dimensionen an gleicher Stelle.
- nichtlineare Abhängigkeit zw. δ und γ $\rightsquigarrow$ Inversion $a \cdot b = 1$

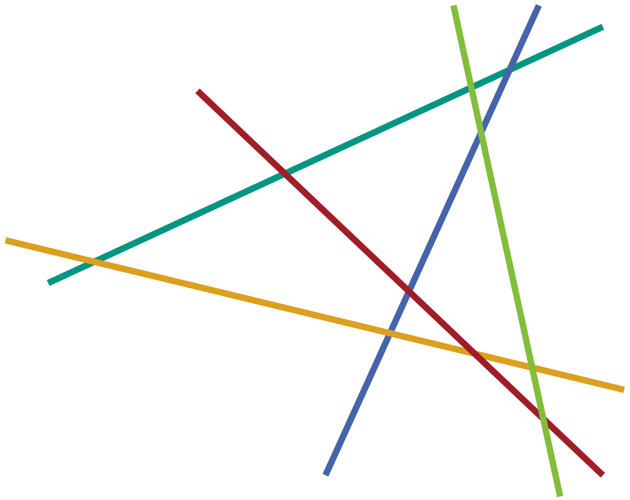


Das STRETCHABILITY-Problem

Eingabe:

Menge von Pseudogeraden

Gesucht:



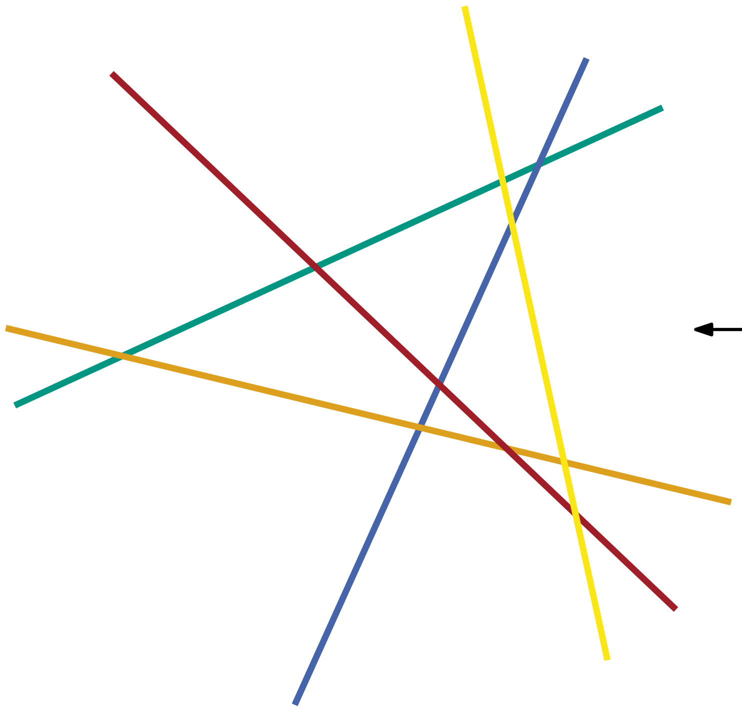
Menge von Geraden
mit gleicher Kombinatorik

Theorem: [Mnëv 1988]
STRETCHABILITY ist $\exists\mathbb{R}$-vollständig.

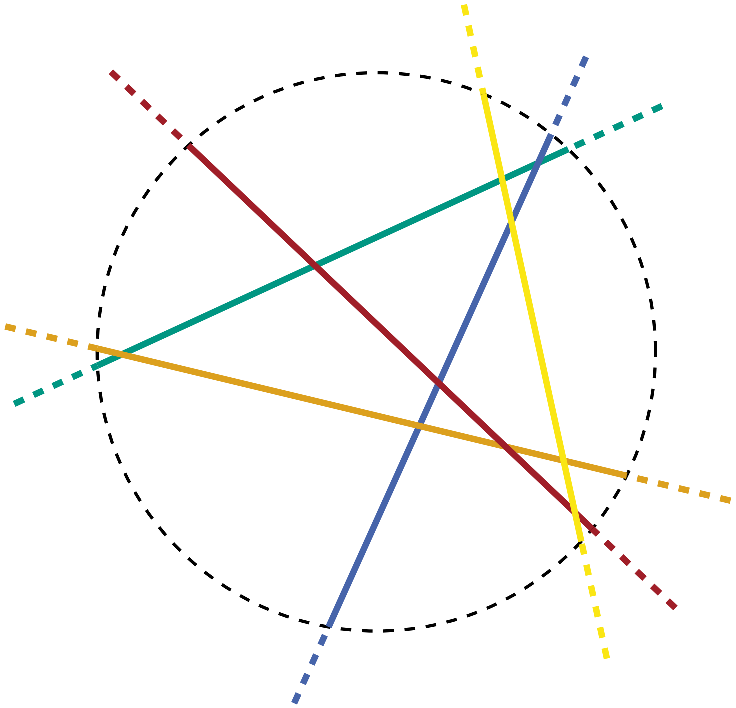
SimpleStretchability

- je zwei Geraden haben genau einen Schnittpunkt
- keine Dreifachschnittpunkte

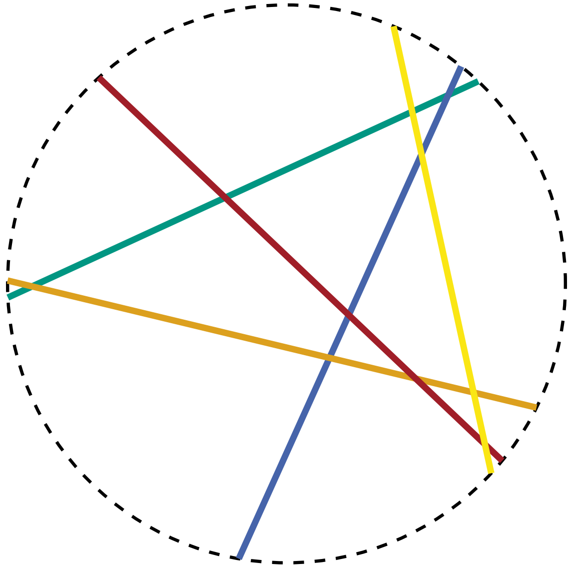
Stretchability in der hyperbolischen Ebene $\mathbb{H}^2$



Geraden in $\mathbb{R}^2$



Kreis um alle Schnittpunkte



Geraden in $\mathbb{H}^2$

(Beltrami-Klein Modell von $\mathbb{H}^2$)

Framework für hyperbolische Unit-Disk-Graphen

Theorem: [McDiarmid, Müller 2013]
STRETCHABILITY kann in polynomieller Zeit auf das Erkennen von Unit-Disk-Graphen reduziert werden. $\rightsquigarrow$ $\exists\mathbb{R}$-schwer

$\mathcal{D}$ - Instanz von SIMPLESTRETCHABILITY

$G_{\mathcal{D}}$ - Graph, konstruiert aus $\mathcal{D}$ nach McDiarmid + Müller (2013)

$\mathcal{D}$ stretchable in $\mathbb{R}^2$ $\overset{(1)}{\Longleftrightarrow}$ $G_{\mathcal{D}}$ ist ein euklidischer Unit-Disk-Graph
[McDiarmid, Müller 2013]

(2) $\Downarrow$ [Bläsius, Friedrich, Katzmann, Stephan 2023]

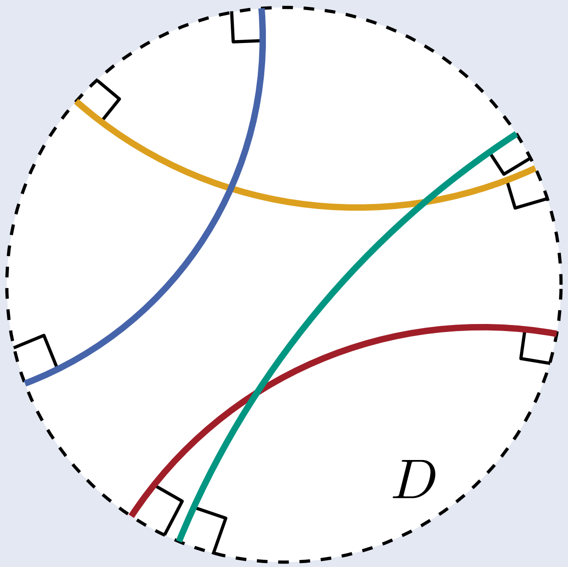
(4) $\Updownarrow$ [Bieker, Bläsius, Dohse, **Jungeblut**, 2023]

$\mathcal{D}$ stretchable in $\mathbb{H}^2$ $\underset{(3)}{\Longleftarrow}$ $G_{\mathcal{D}}$ ist ein hyperbolischer Unit-Disk-Graph

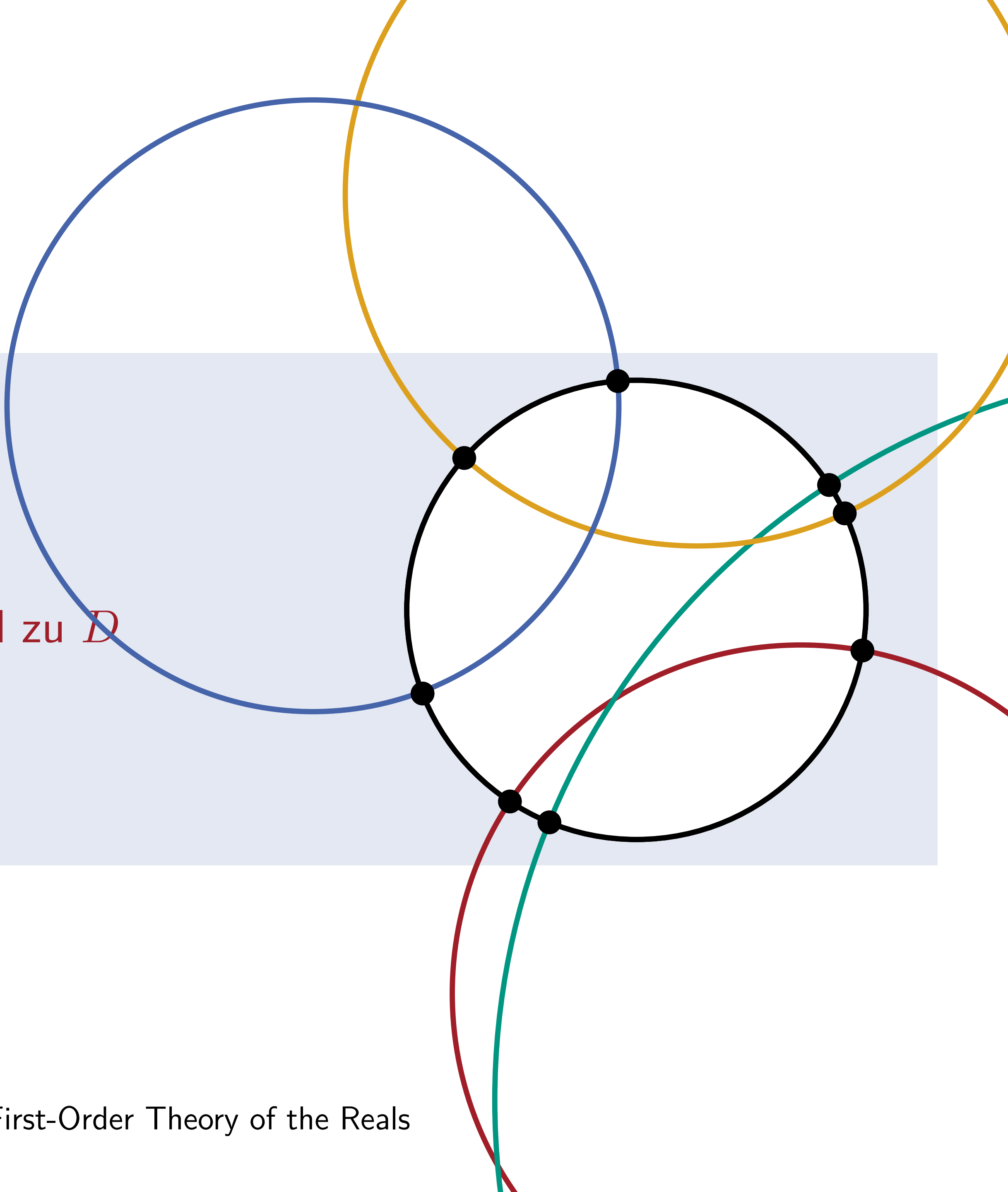
Lombardi und hyperbolische Geometrie

Poincaré Disk Modell:

- bettet $\mathbb{H}^2$ in $\mathbb{R}^2$ ein
- $\mathbb{H}^2$ ist Inneres eines Einheitskreises D
- hyperbolische Geraden $\rightsquigarrow$ Kreisbögen orthogonal zu D
- Konform: winkelerhaltend



Lombardi und hyperbolische Geometrie



Poincaré Disk Modell:

- bettet $\mathbb{H}^2$ in $\mathbb{R}^2$ ein
- $\mathbb{H}^2$ ist Inneres eines Einheitskreises D
- hyperbolische Geraden $\rightsquigarrow$ Kreisbögen orthogonal zu D
- Konform: winkelerhaltend

Hausdorff-Distanz: $\forall\exists\mathbb{R}$-Schwere (Skizze)

Eingabe: $\Phi \;:\equiv\; \forall X \in \mathbb{R}^n \,.\, \exists Y \in \mathbb{R}^m : \varphi(X, Y)$

Reduktion: $A \;:=\; \{x \in \mathbb{R}^n \mid \exists Y \in \mathbb{R}^m : \varphi(x, Y)\} \;\subseteq\; \mathbb{R}^n$

alle x, für die es ein Y gibt

$B \;:=\; \mathbb{R}^n$

Hausdorff-Distanz: $\forall\exists\mathbb{R}$-Schwere (Skizze)

Eingabe: $\Phi \;:\equiv\; \forall X \in \mathbb{R}^n \,.\, \exists Y \in \mathbb{R}^m : \varphi(X, Y)$

Reduktion: $A \;:=\; \{x \in \mathbb{R}^n \mid \exists Y \in \mathbb{R}^m : \varphi(x, Y)\} \;\subseteq\; \mathbb{R}^n$ ← alle x, für die es ein Y gibt

$B \;:=\; \mathbb{R}^n$

Idee: Φ wahr $\rightsquigarrow$ $A = \mathbb{R}^n$ $\rightsquigarrow$ $d_{\mathrm{H}}(A, B) = 0$

Φ falsch $\rightsquigarrow$ $A \subsetneq \mathbb{R}^n$ $\rightsquigarrow$ $d_{\mathrm{H}}(A, B) > 0$

Hausdorff-Distanz: ∀∃ℝ-Schwere (Skizze)

Eingabe: $\Phi \;:\equiv\; \forall X \in \mathbb{R}^n \,.\, \exists Y \in \mathbb{R}^m : \varphi(X, Y)$

Reduktion: $A \;:=\; \{x \in \mathbb{R}^n \mid \exists Y \in \mathbb{R}^m : \varphi(x, Y)\} \;\subseteq\; \mathbb{R}^n$ ← alle x, für die es ein Y gibt

$B \;:=\; \mathbb{R}^n$

Idee:

Φ wahr $\rightsquigarrow$ $A = \mathbb{R}^n$ $\rightsquigarrow$ $d_{\mathrm{H}}(A, B) = 0$

Φ falsch $\rightsquigarrow$ $A \subsetneq \mathbb{R}^n$ $\rightsquigarrow$ $d_{\mathrm{H}}(A, B) > 0$

$d_{\mathrm{H}}(A, B) = 0$
$\Longleftrightarrow$
A und B haben gleichen Abschluss

Problem 1:

nur ein einziges „x ohne passendes Y“
$\not\Longrightarrow d_{\mathrm{H}}(A, B) > 0$

Hausdorff-Distanz: $\forall\exists\mathbb{R}$-Schwere (Skizze)

Eingabe: $\Phi \;:\equiv\; \forall X \in \mathbb{R}^n \,.\, \exists Y \in \mathbb{R}^m : \varphi(X, Y)$

Reduktion: $A \;:=\; \{x \in \mathbb{R}^n \mid \exists Y \in \mathbb{R}^m : \varphi(x, Y)\} \;\subseteq\; \mathbb{R}^n$ ← alle x, für die es ein Y gibt

$B \;:=\; \mathbb{R}^n$

Idee: Φ wahr $\rightsquigarrow$ $A = \mathbb{R}^n$ $\rightsquigarrow$ $d_{\mathrm{H}}(A, B) = 0$

Φ falsch $\rightsquigarrow$ $A \subsetneq \mathbb{R}^n$ $\rightsquigarrow$ $d_{\mathrm{H}}(A, B) > 0$

$d_{\mathrm{H}}(A, B) = 0$
$\Longleftrightarrow$
A und B haben gleichen Abschluss

Problem 1:

nur ein einziges „x ohne passendes Y“
$\not\Longrightarrow d_{\mathrm{H}}(A, B) > 0$

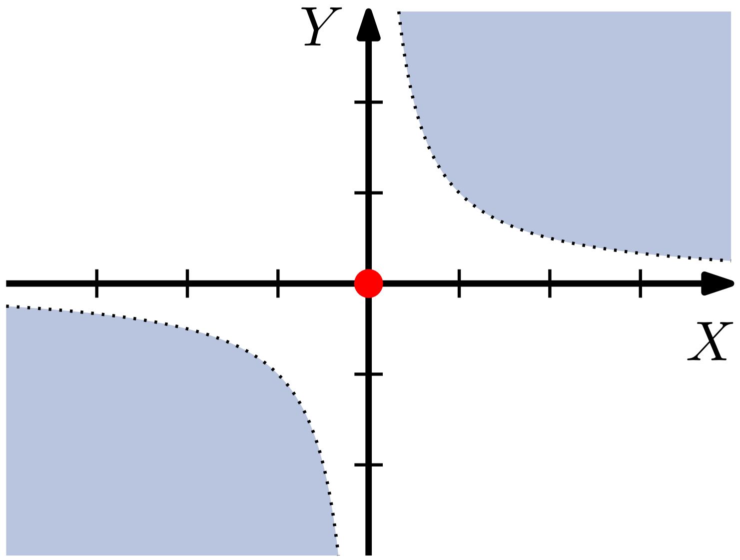
Problem 2:

Definition von A enthält Existenzquantor $\exists$

Lösung 1: falsch und falscher

Idee: Falsche Formeln „noch falscher“ machen.

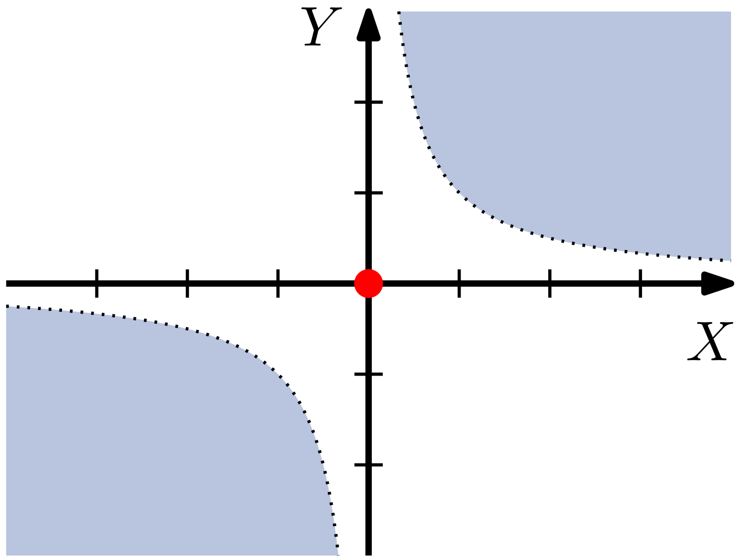
$$\forall X \in \mathbb{R} \,.\, \exists Y \in \mathbb{R} : XY > 1$$



Lösung 1: falsch und falscher

Idee: Falsche Formeln „noch falscher“ machen.

$$\forall X \in \mathbb{R} \, . \, \exists Y \in \mathbb{R} : XY > 1$$



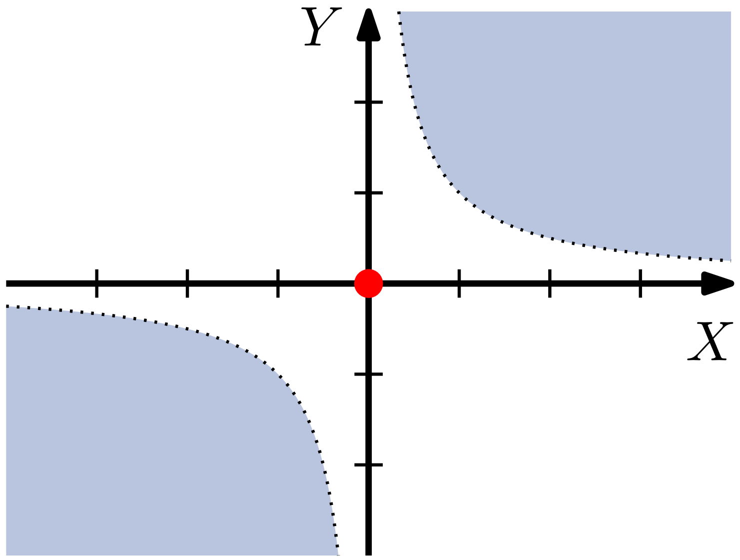
$A = \mathbb{R} \setminus \{0\} \qquad B = \mathbb{R}$

$\rightsquigarrow d_{\mathrm{H}}(A, B) = 0$

Lösung 1: falsch und falscher

Idee: Falsche Formeln „noch falscher“ machen.

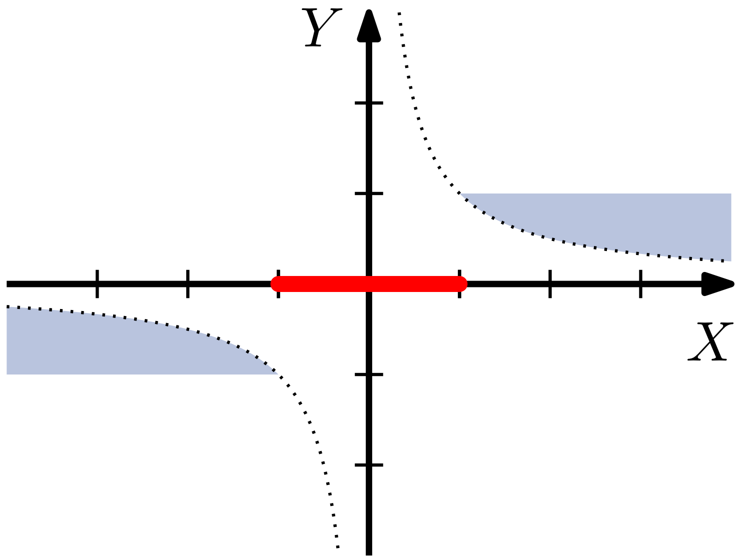
$$\forall X \in \mathbb{R} \,.\, \exists Y \in \mathbb{R} : XY > 1$$



$A = \mathbb{R} \setminus \{0\}$ $\quad B = \mathbb{R}$

$\rightsquigarrow d_{\mathrm{H}}(A, B) = 0$

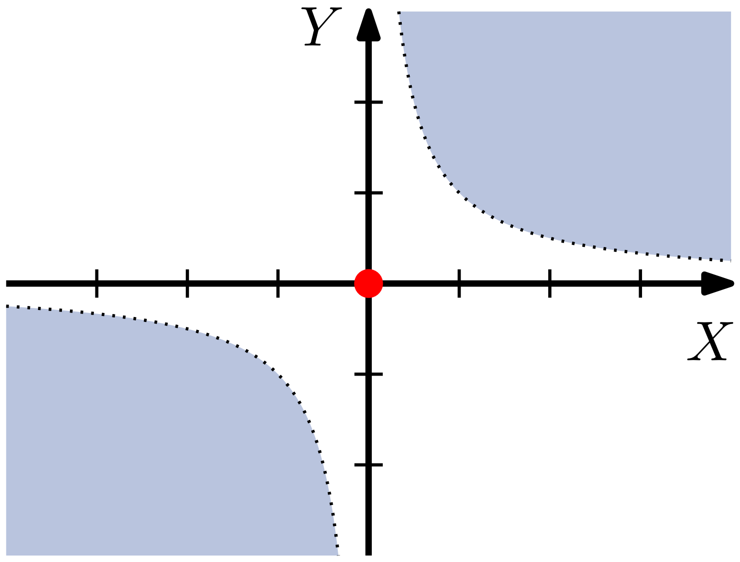
$$\forall X \in \mathbb{R} \,.\, \exists Y \in [-1, 1] : XY > 1$$



Lösung 1: falsch und falscher

Idee: Falsche Formeln „noch falscher" machen.

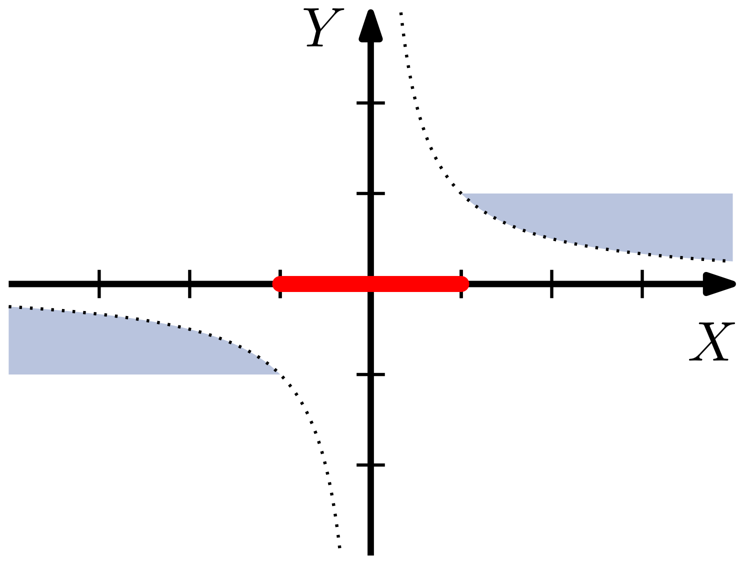
$\forall X \in \mathbb{R} \,.\, \exists Y \in \mathbb{R} : XY > 1$



$A = \mathbb{R} \setminus \{0\} \qquad B = \mathbb{R}$

$\rightsquigarrow d_{\mathrm{H}}(A, B) = 0$

$\forall X \in \mathbb{R} \,.\, \exists Y \in [-1, 1] : XY > 1$



$A = \mathbb{R} \setminus [-1, 1] \qquad B = \mathbb{R}$

$\rightsquigarrow d_{\mathrm{H}}(A, B) = 1$