

Logical Neural Networks (1. Fachsemester M.Sc. Inf.)

Hybrid Intelligence (8/14)

15 + 5 min.

Dr. Ricardo Usbeck
<https://github.com/RicardoUsbeck/HI>

20.07.2022

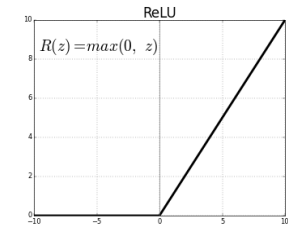
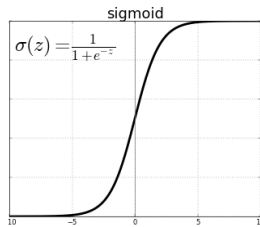
Wdh. Neuronale Netze und Knowledge Graphs

Worüber haben wir die letzten Male gesprochen?

Wdh. Neuronale Netze und Knowledge Graphs

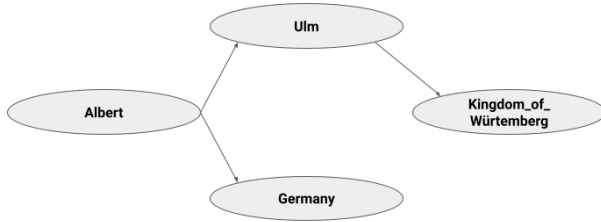
Worüber haben wir die letzten Male gesprochen?

- Was ist ein neuronales Netz?
 - ▶ Wie funktioniert *backpropagation* und *gradient descent*?
 - ▶ Was sind Aktivierungsfunktionen?
- Was sind Grammatiken und Syntaxbäume?
- Was ist FOL? Was ist DL?
- Was ist ein *Knowledge Graph*?



Quelle: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>

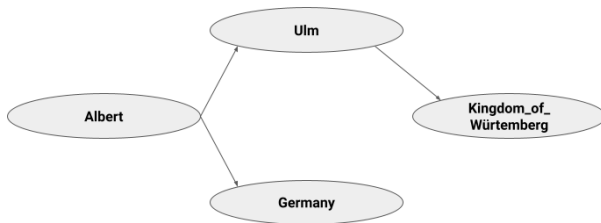
Wdh. Knowledge Graph



bornIn(,)	
Albert	Ulm
Albert	Germany

partOf(,)	
Ulm	Kingdom_of_Württemberg

Wdh. Knowledge Graph

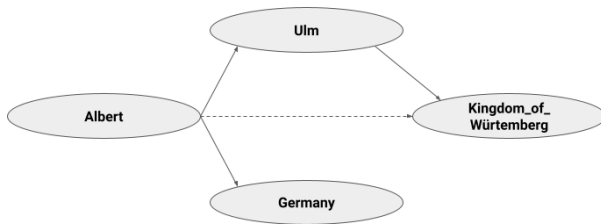


bornIn(,)	
Albert	Ulm
Albert	Germany

partOf(,)	
Ulm	Kingdom_of_Württemberg

Was kann man schlussfolgern?

Wdh. Knowledge Graph



bornIn(,)	
Albert	Ulm
Albert	Germany
Albert	Kingdom_of_Württemberg

partOf(,)	
Ulm	Kingdom_of_Württemberg

Description Logic/First Order Logic Regeln

$$(b(X,A) \wedge b(X,B)) \rightarrow (p(A,B) \vee p(B,A))$$

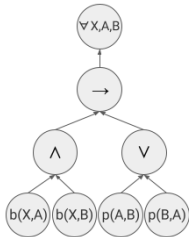
$$(b(X,A) \wedge p(A,B)) \rightarrow b(X,B)$$

bornIn(,)	
Albert	Ulm
Albert	Germany

partOf(,)	
Ulm	Kingdom_of_Würtemberg

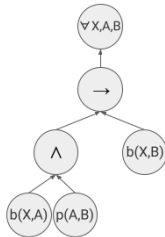
Description Logic/First Order Logic Regeln - Syntaxbäume

$$(b(X,A) \wedge b(X,B)) \rightarrow (p(A,B) \vee p(B,A))$$



bornIn(., .)	
Albert	Ulm
Albert	Germany

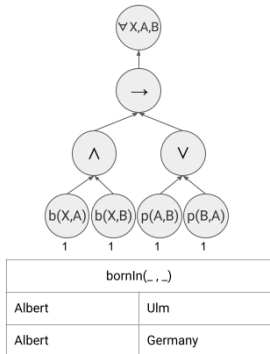
$$(b(X,A) \wedge p(A,B)) \rightarrow b(X,B)$$



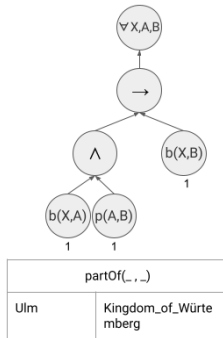
partOf(., .)	
Ulm	Kingdom_of_Würtemberg

Description Logic/First Order Logic Regeln - Wahrheitswerte

$$(b(X,A) \wedge b(X,B)) \rightarrow (p(A,B) \vee p(B,A))$$

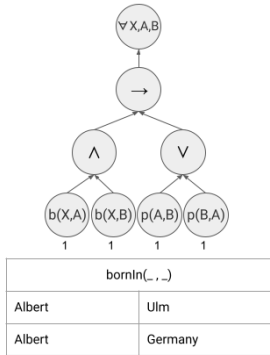


$$(b(X,A) \wedge p(A,B)) \rightarrow b(X,B)$$

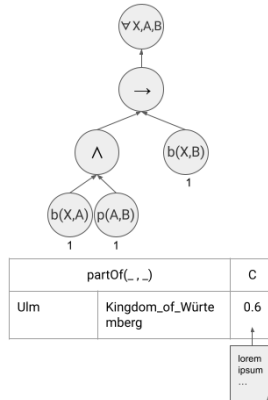


Description Logic/First Order Logic Regeln - realwertige Wahrheitswerte

$$(b(X,A) \wedge b(X,B)) \rightarrow (p(A,B) \vee p(B,A))$$

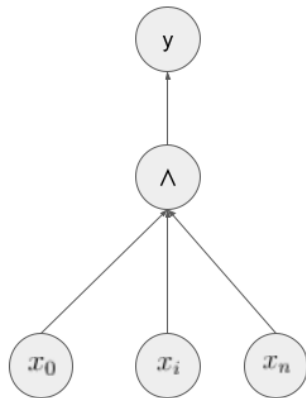


$$(b(X,A) \wedge p(A,B)) \rightarrow b(X,B)$$



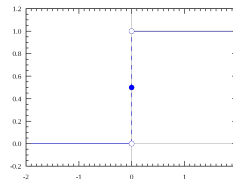
Warum reicht die mathematische Formulierung zur Modellierung von Konfidenzen nicht aus?

Idee: Modelliere Logic mit Neuronalen Netzen (McCulloch und Pitts, 1943)



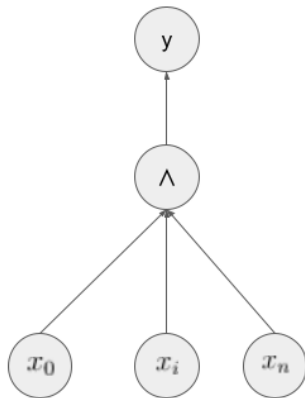
McCulloch-Pitts-Zelle

- Idee: Knoten werden zu Neuronen
- Input: $x_i \in \{0, 1\}$
- Output: $y = \text{step}(\sum x - \theta)$



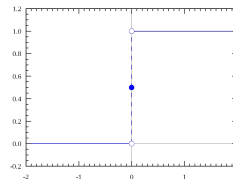
- Mathematische Verhalten:
 - ▶ $p \wedge q = \llbracket p + q > 1.5 \rrbracket$
 - ▶ $p \vee q = \llbracket p + q > 0.5 \rrbracket$
 - ▶ $p \rightarrow q = \llbracket 1 - p + q > 0.5 \rrbracket$

Idee: Modelliere Logic mit Neuronalen Netzen (McCulloch und Pitts, 1943)



McCulloch-Pitts-Zelle

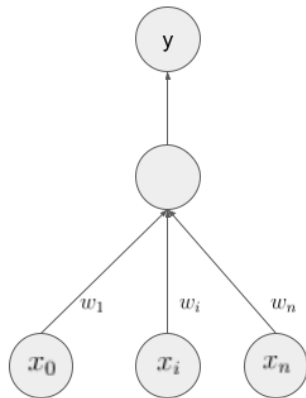
- Idee: Knoten werden zu Neuronen
- Input: $x_i \in \{0, 1\}$
- Output: $y = \text{step}(\sum x - \theta)$



- Mathematische Verhalten:
 - ▶ $p \wedge q = \llbracket p + q > 1.5 \rrbracket$
 - ▶ $p \vee q = \llbracket p + q > 0.5 \rrbracket$
 - ▶ $p \rightarrow q = \llbracket 1 - p + q > 0.5 \rrbracket$

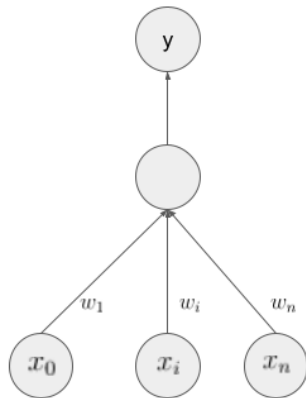
Was fehlt?

Idee: Logical Neural Networks (Riegel et al. 2020)



- Ideen:
 - ▶ Knoten werden zu Neuronen
 - ▶ Kanten werden gewichtet
- Input: $x_i \in [0, 1]$
- Output: $y = f(w \cdot x - \theta)$ (*Upward Inference*)
 - ▶ Mit $w_i, i, \theta \geq 0$
- Einführung eines *truth parameters* $0.5 < \alpha \leq 1.0$
 - ▶ $p \geq \alpha$ interpretiert als wahr
 - ▶ $p \leq 1 - \alpha$ interpretiert als falsch
- f ist eine monotone Aktivierungsfunktion
 - ▶ $f(\alpha) = \alpha$
 - ▶ $f(1 - \alpha) = 1 - \alpha$
- Optimierung über *Constrained Optimization* (nächstes Mal)

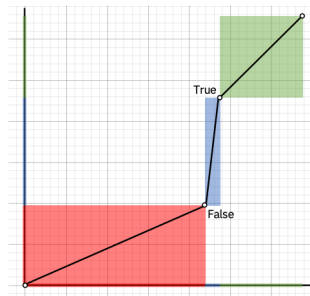
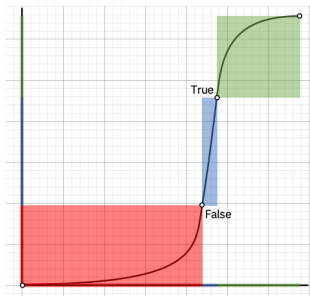
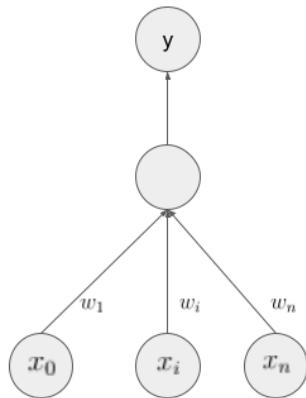
Idee: Logical Neural Networks (Riegel et al. 2020)



- Ideen:
 - ▶ Knoten werden zu Neuronen
 - ▶ Kanten werden gewichtet
- Input: $x_i \in [0, 1]$
- Output: $y = f(w \cdot x - \theta)$ (*Upward Inference*)
 - ▶ Mit $w_i, i, \theta \geq 0$
- Einführung eines *truth parameters* $0.5 < \alpha \leq 1.0$
 - ▶ $p \geq \alpha$ interpretiert als wahr
 - ▶ $p \leq 1 - \alpha$ interpretiert als falsch
- f ist eine monotone Aktivierungsfunktion
 - ▶ $f(\alpha) = \alpha$
 - ▶ $f(1 - \alpha) = 1 - \alpha$
- Optimierung über *Constrained Optimization* (nächstes Mal)

Was bringt uns das bezogen auf die Aussagekraft des Neurons?

Idee: Logical Neural Networks (Riegel et al. 2020) — Interpretierbarkeit



Quelle: <https://arxiv.org/pdf/2006.13155.pdf>

- Klassisches 'Und'-Verhalten

- ▶ $\sum_i w_i \alpha \geq \alpha$ 'wahr'
- ▶ $\forall i \sum_j w_j - w_i \alpha - \theta \leq 1 - \alpha$ 'falsch'

Logical Neural Networks (Riegel et al. 2020) - Überblick

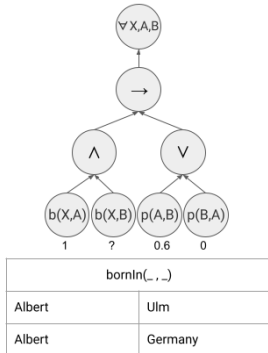
Idee: Gewichte in Neuronalen Netzen können als logische Funktionen interpretiert werden

- **Neuron:** Repräsentiert ein Element in einer Logischen Formel (entweder Variable oder Operator) mit Gewichten an den Verbindungen
- **Training:** Feature-Value-Paare
- **Auswertung:** erfolgt über monotone Verschärfung unterer und oberer Schranken von Subformeln in einem *upward/downward pass* über den Syntaxbaum (nächstes Mal)
- **Eigenschaften:** E2E-differenzierbar, deterministisch, Widersprüche abbildbar, *Open World Assumption*, *disentangle* Repräsentation

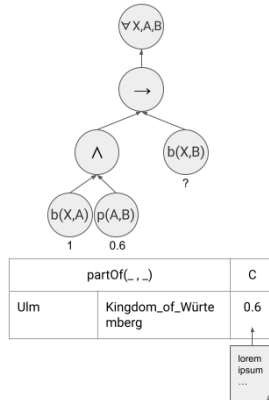
⇒ Modularität führt zu Generalisierbarkeit

Logical Neural Networks (Riegel et al. 2020) - Beispiel

$$(b(X,A) \wedge b(X,B)) \rightarrow (p(A,B) \vee p(B,A))$$

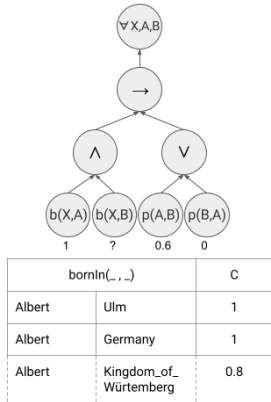


$$(b(X,A) \wedge p(A,B)) \rightarrow b(X,B)$$

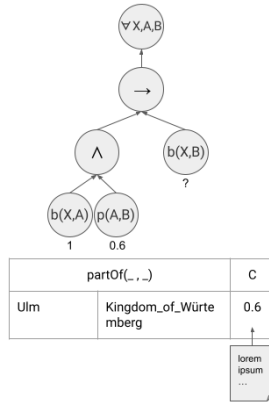


Logical Neural Networks (Riegel et al. 2020) - Beispiel

$$(b(X,A) \wedge b(X,B)) \rightarrow (p(A,B) \vee p(B,A))$$



$$(b(X,A) \wedge p(A,B)) \rightarrow b(X,B)$$



Warum eine Kombination von statistischen Lernverfahren und Logik?

Symbolische Systeme:

Eigenschaften

- - Hoher manueller Aufwand
- + *Compositional Generalizability*

Sub-symbolische Systeme und statistische Lernverfahren (z.B. Deep Learning):

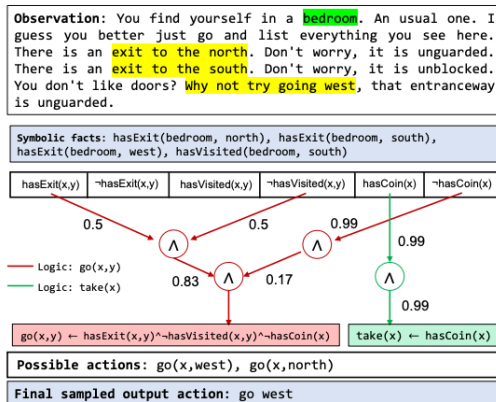
Nachteile

- + Starke Performance, wenn genügend Trainingsdaten verfügbar
- - Keine Interpretierbarkeit

Lösung: Hybride Künstliche Intelligenz (bspw. mit Logical Neural Networks)!

Mehr lesen: Garcez, Artur d'Avila, et al. NNeural-symbolic learning and reasoning: A survey and interpretation. NNeuro-Symbolic Artificial Intelligence: The State of the Art 342 (2022)

Logical Neural Networks (Riegel et al. 2020) - Anwendungen



Quelle: Chaudhury, Subhajit, et al. 'Neuro-Symbolic Approaches for Text-Based Policy Learning.' Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021.

Aber auch bei Entity Linking, Question Answering (NSQA SOTA on QALD-9), ...

Logical Neural Networks (Riegel et al. 2020) - Code

- Inferenz erzeugt:

('Albert', 'Ulm', 'Kingdom_of_Wuerttemberg')
APPROX_TRUE (0.6, 1.0)

```
1 from lnn import Predicate, Variable, Implies, ForAll, Model, World, TRUE, And, Or, Join
2
3
4
5 def example():
6     model = Model()
7
8     born_in = Predicate('bornIn', arity=2)
9     part_of = Predicate('partOf', arity=2)
10
11     x = Variable('x', ctype='person')
12     a = Variable('a', ctype='place')
13     b = Variable('b', ctype='place')
14
15     model["birthplaces_are_part_of"] = ForAll(x,
16                                             a,
17                                             b,
18                                             Implies(
19                                                 And(born_in(x, a), born_in(x, b), join=Join.OUTER),
20                                                 Or(part_of(a, b), part_of(b, a), join=Join.OUTER),
21                                                 join=Join.OUTER,
22                                             ),
23                                             join=Join.OUTER,
24                                             world=World.AXION)
25
26     model["part_of_are_birthplaces"] = ForAll(x,
27                                             a,
28                                             b,
29                                             Implies(
30                                                 And(born_in(x, a), part_of(a, b), join=Join.OUTER),
31                                                 born_in(x, b),
32                                                 join=Join.OUTER),
33                                             join=Join.OUTER,
34                                             world=World.AXION)
35
36     model.add_facts({
37         born_in.name: {
38             ('Albert', 'Ulm'): TRUE,
39             ('Albert', 'Germany'): TRUE
40         },
41         part_of.name: {
42             ('Ulm', 'Kingdom_of_Wuerttemberg'): (0.6, 0.6)
43         }
44     })
45
46     model.print()
47
48     model.infer()
```

Quelle: <https://github.com/RicardoUsbeck/HI>
Kudos to C. Möller

Take-Away und Ausblick

Was haben wir gelernt?

- Grundaufbau Logische Neuronale Netze
- Vorteile Hybrider KI

Wo können wir weiter lernen oder ausprobieren?

- Original Code: <https://github.com/ibm/lmn>
- Beispiel Code: <https://github.com/RicardoUsbeck/HI>

Was passiert beim nächsten Mal?

- Bidirektionale Inferenz: modus ponens, modus tollens
- Optimierung von LNNs

Was sollte man bis zum nächsten Mal gelesen haben?

- Lesen: <https://ibm.github.io/LNN/>

Quellen

- Riegel, Ryan, et al. 'Logical neural networks.' arXiv preprint arXiv:2006.13155 (2020).
- Fagin, Ronald, Ryan Riegel, and Alexander Gray. Foundations of reasoning with uncertainty via real-valued logics. arXiv preprint arXiv:2008.02429 (2020).
- McCulloch, Warren S., and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics 5.4 (1943): 115-133.
- Blog -
<https://skirmilitor.medium.com/logical-neural-networks-31498d1aa9be>
- Youtube - KRHCAI Invited Talk by Alexander Gray: Logical Neural Networks
<https://www.youtube.com/watch?v=m687EBE0Nqw>

Danke für Ihre Aufmerksamkeit!

🎓 Lernmaterial (VL, Selbsttests, Übungen, Links zu Jupyter Notebooks...)

🌐 <http://github.com/RicardoUsbeck/HI>

Welche Fragen haben Sie?

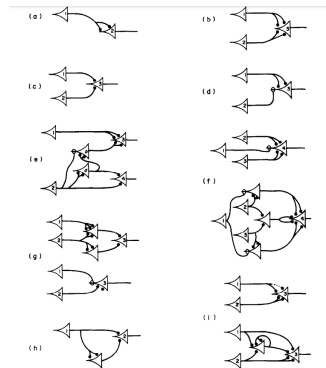


Figure 1. The neuron c_i is always marked with the numeral i upon the body of the cell, and the corresponding action is denoted by " N " with i subscript, as in the text:

Quelle: McCulloch, Warren S., and Walter Pitts. 'A logical calculus of the ideas immanent in nervous activity.' The bulletin of mathematical biophysics 5.4 (1943): 115-133.
<https://www.cs.cmu.edu/~epxing/Class/10715/reading/McCulloch.and.Pitts.pdf>

Bonus: Łukasiewicz Logik als Basis

Definition:

- Grundannahme, dass 'unknown implies unknown' wahr ist is true
- $p \otimes q = \max(0, p + q - 1)$ (and)
- $p \oplus q = \min(1, p + q)$ (or)
- $p \rightarrow q = \min(1, 1 - p + q)$ (or)
- If you plot this, it looks like ReLu!
- But (!) it does not allow for real-valued inputs

Bonus: Logical Neural Networks (Riegel et al. 2020) - Eigenschaften

- LNNs sind *End-to-End* differenzierbar
- Inferenz ist *omnidirektional* und *deterministisch*, d.h. Relationen sind bidirektional inferierbar und Herleitungen wiederholbar
- Inferenz konvergiert beweisbar in endlicher Anzahl Schritte
- *Loss-Function* bildet auch logische Widersprüche ab
- LNNs arbeiten mit der *Open World assumption*
- Formeln sind *modular* und *compositional*, d.h. es kann hierarchische Beziehungen für Formeln geben
- Representation ist *disentangled*, d.h. es gibt nicht einen einzigen Vektor für die Formel