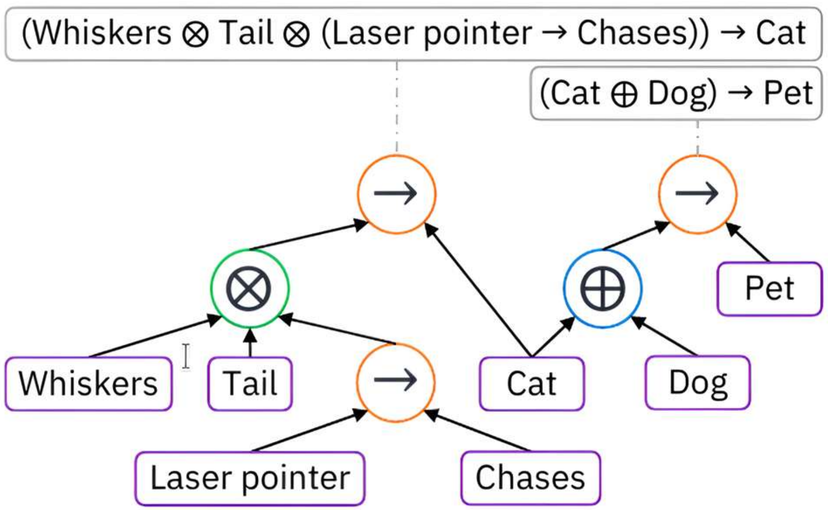
Logical Neural Networks (LNNs)

Martin Dallinger, Lisa Krimbacher

JOHANNES KEPLER UNIVERSITY LINZ

Brief Introduction to LNNs

- Neural networks combined with fuzzy logic (Riegel et al., 2020)
- Intervals for degree of truth (Shakarian et al., 2023)
- A-priori defined knowledge base in propositional logic or first order logic (Riegel et al., 2020)
- Explainable outputs (Shakarian et al., 2023)



Taken from Shakarian et al., 2023

Theoretical Foundations

Weighted Real-Valued Logic

(Riegel et al., 2020)

Boolean Logic

false **true**

Weighted Real-Valued Logic

(Riegel et al., 2020)

$0.5 < \alpha \leq 1$

Weighted Real-Valued Logic

(Riegel et al., 2020)

Each formula is assigned two truth bounds **L**, **U** ∈ [0, 1].

Every neuron is in one of four primary states:

	true			
Lower (**L**)	[α, 1]			
Upper (**U**)	[α, 1]			

Weighted Real-Valued Logic

(Riegel et al., 2020)

Each formula is assigned two truth bounds **L**, **U** ∈ [0, 1].

Every neuron is in one of four primary states:

	true	false		
Lower (**L**)	[α, 1]	[0, 1 – α]		
Upper (**U**)	[α, 1]	[0, 1 – α]		

Weighted Real-Valued Logic

(Riegel et al., 2020)

Each formula is assigned two truth bounds **L**, **U** ∈ [0, 1].

Every neuron is in one of four primary states:

	true	false	uncertain	
Lower (**L**)	[α, 1]	[0, 1 – α]	[0, 1 – α]	
Upper (**U**)	[α, 1]	[0, 1 – α]	[α, 1]	

Weighted Real-Valued Logic

(Riegel et al., 2020)

Each formula is assigned two truth bounds **L**, **U** $\in [0, 1]$.

Every neuron is in one of four primary states:

	true	**false**	**uncertain**	**contradiction**
Lower (**L**)	$[\alpha, 1]$	$[0, 1 - \alpha]$	$[0, 1 - \alpha]$	$L > U$
Upper (**U**)	$[\alpha, 1]$	$[0, 1 - \alpha]$	$[\alpha, 1]$	

Logical Operators

(Shakarian et al., 2023)

Weighted generalizations of **Łukasiewicz operators**

- **„AND“**

$$\beta\left(\bigotimes_{i\in I} x_i^{\otimes w_i}\right) = f\left(\beta - \sum_{i\in I} w_i(1 - x_i)\right)$$

$I\ldots$ input set

$f: \mathbb{R} \to [0,1], f(1-x) = 1 - f(x)$

$\beta \geq 0, w_i \geq 0, x_i \in [0,1]$

Logical Operators

(Shakarian et al., 2023)

Weighted generalizations of **Łukasiewicz operators**

- **„AND“**
- **„OR“**

$$\beta\left(\bigoplus_{i \in I} x_i^{\oplus w_i}\right) = f\left(1 - \beta + \sum_{i \in I} w_i x_i\right)$$

I… input set

$$f : \mathbb{R} \to [0,1], f(1-x) = 1 - f(x)$$

$$\beta \geq 0, w_i \geq 0, x_i \in [0,1]$$

Logical Operators

(Shakarian et al., 2023)

Weighted generalizations of **Łukasiewicz operators**

- **„AND“**
- **„OR“**
- **„IMPLIES“**

$$^{\beta}\left(x^{\otimes w_x} \rightarrow y^{\oplus w_y}\right) = f\left(1 - \beta + w_x(1 - x) + w_y y\right)$$

I… input set

$$f: \mathbb{R} \rightarrow [0,1], f(1-x) = 1 - f(x)$$

$$\beta \geq 0, w_i \geq 0, x_i \in [0,1]$$

JOHANNES KEPLER UNIVERSITY LINZ

Logical Operators

(Riegel et al., 2020)

Weighted generalizations of **Łukasiewicz operators**

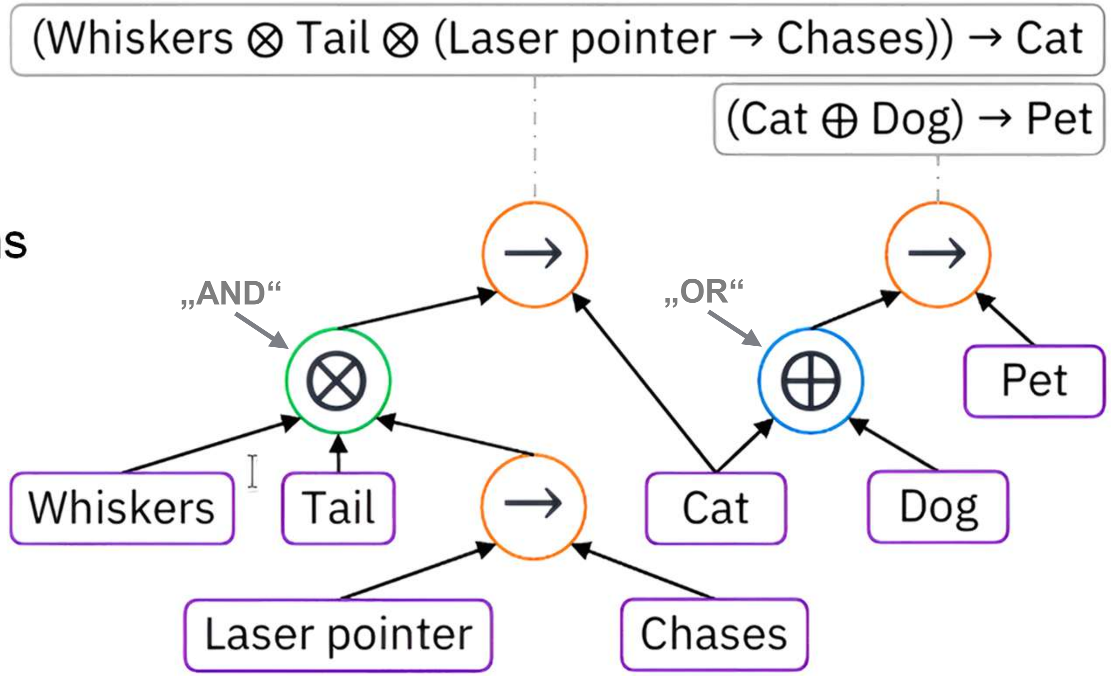
- **„AND“**
- **„OR“**
- **„IMPLIES“**
- **„NOT“**

$$(\neg x) = 1 - x$$
$$x \in [0,1]$$

Syntax Tree

(Riegel et al., 2020)

- Set of logical sentences (knowledge base) specified a-priori (propositional logic or first order logic)
- Syntax tree of formulas
- Nodes are neurons
- Truth bounds [L, U] for all neurons



Adapted from Shakarian et al., 2023

JOHANNES KEPLER UNIVERSITY LINZ

Activation Functions

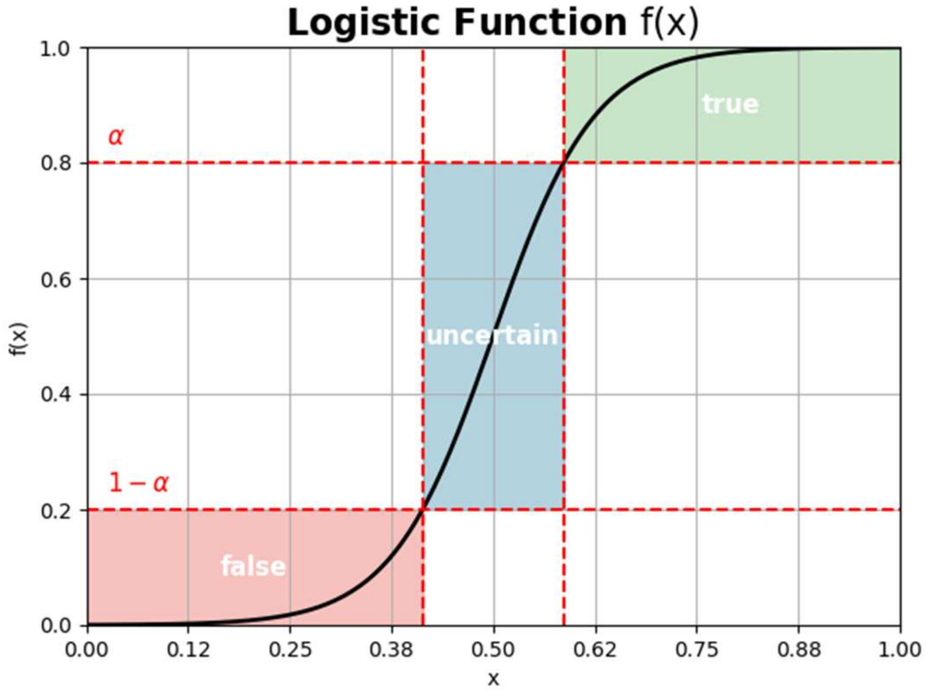
(Riegel et al., 2020)

Neurons corresponding to logical operators

- Monotonicity required
- (Can be) related via De Morgan Laws
- (Can be) commutative, associative

Logistic Function

$$f(x) = \frac{1}{1 + e^{-5(2x-1)}}$$



Adapted from Shakarian et al., 2023, and Riegel et al., 2020

Activation Functions

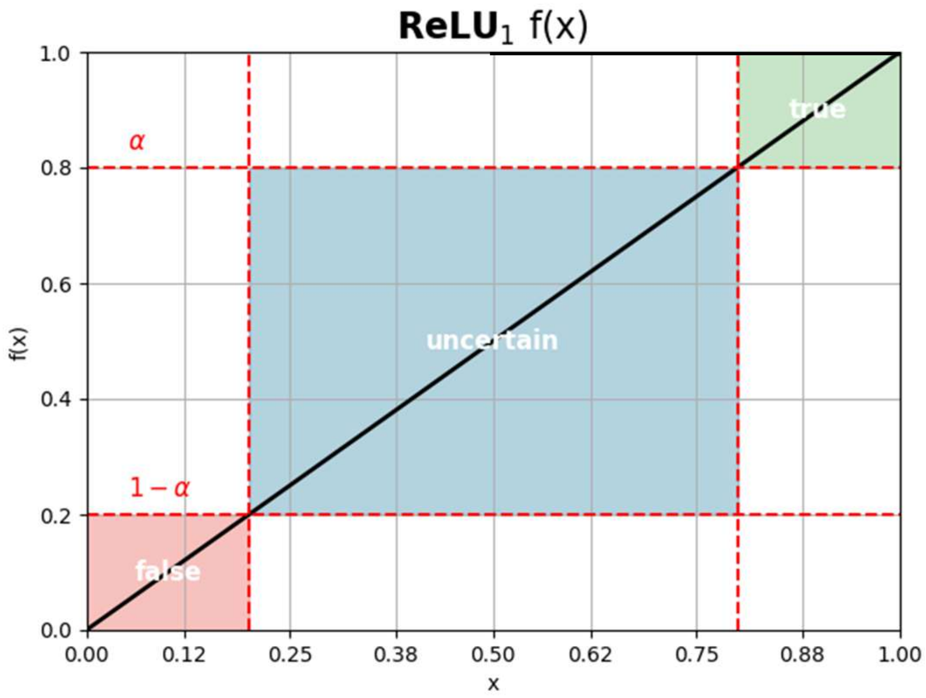
(Riegel et al., 2020)

Neurons corresponding to logical operators

- Monotonicity required
- (Can be) related via De Morgan Laws
- (Can be) commutative, associative

ReLU_1 Function

$$f(x) = \max(0,\min(1,x))$$



Adapted from Shakarian et al., 2023, and Riegel et al., 2020

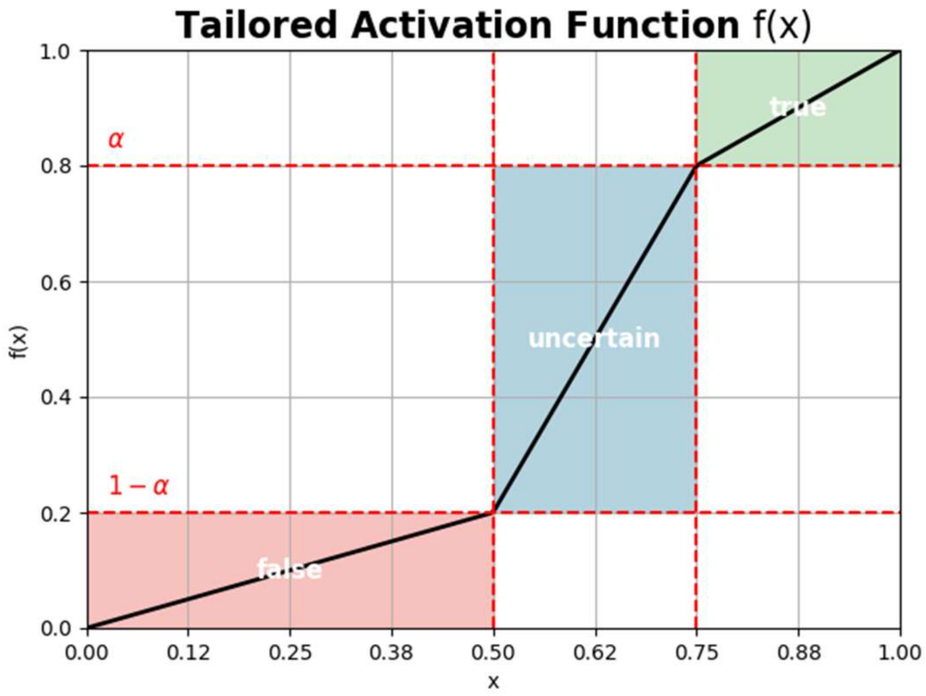
Activation Functions

(Riegel et al., 2020)

Neurons corresponding to logical operators

- Monotonicity required
- (Can be) related via De Morgan Laws
- (Can be) commutative, associative

Tailored Activation Function



Adapted from Shakarian et al., 2023, and Riegel et al., 2020

Activation Functions

(Riegel et al., 2020)

Neurons corresponding to atoms

- Aggregation of truth values

Example for atom **y**:

- $(x_1 \otimes x_2) \rightarrow y$: $[\mathbf{L_1}, \mathbf{U_1}]$
- $(x_2 \otimes x_4) \rightarrow y$: $[\mathbf{L_2}, \mathbf{U_2}]$

Return $[\mathbf{L_{new}}, \mathbf{U_{new}}] = [\max(\mathbf{L_1}, \mathbf{L_2}), \min(\mathbf{U_1}, \mathbf{U_2})]$

Activation Functions

(Riegel et al., 2020)

Neurons corresponding to atoms

- Aggregation of truth values

Return $[L_{new}, U_{new}] = [\max(L_1, L_2), \min(U_1, U_2)]$

Example for atom **y**:

- $(x_1 \otimes x_2) \rightarrow y$: $[L_1, U_1]$
- $(x_2 \otimes x_4) \rightarrow y$: $[L_2, U_2]$

y: $[L_{new}, U_{new}] = [L_2, U_1]$

Activation Functions

(Riegel et al., 2020)

Neurons corresponding to atoms

- Aggregation of truth values

Importance weighting

- Use the learned weights of the operator neurons

Example for atom **y**:

- $x_1^{\otimes 3} \rightarrow y^{\oplus 2}$: [$\mathbf{L_1}$, $\mathbf{U_1}$]
- $x_1^{\otimes 1} \rightarrow y^{\oplus 0.5}$: [$\mathbf{L_2}$, $\mathbf{U_2}$]

$$\mathbf{L_{new}} = \max(0, \min(1, 2\mathbf{L_1} + 0.5\mathbf{L_2}))$$

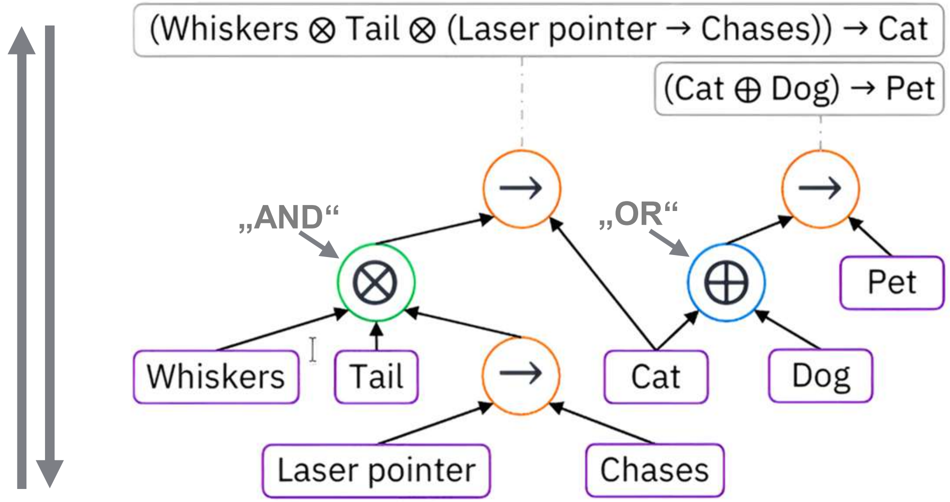
$$\mathbf{U_{new}} = \max(0, \min(1, 1 - 2(1 - \mathbf{U_1}) + 0.5(1 - \mathbf{U_2})))$$

JOHANNES KEPLER UNIVERSITY LINZ

Bidirectional Inference

(Riegel et al., 2020)

- Multiple passes over neurons until convergence (in finite amount of steps)



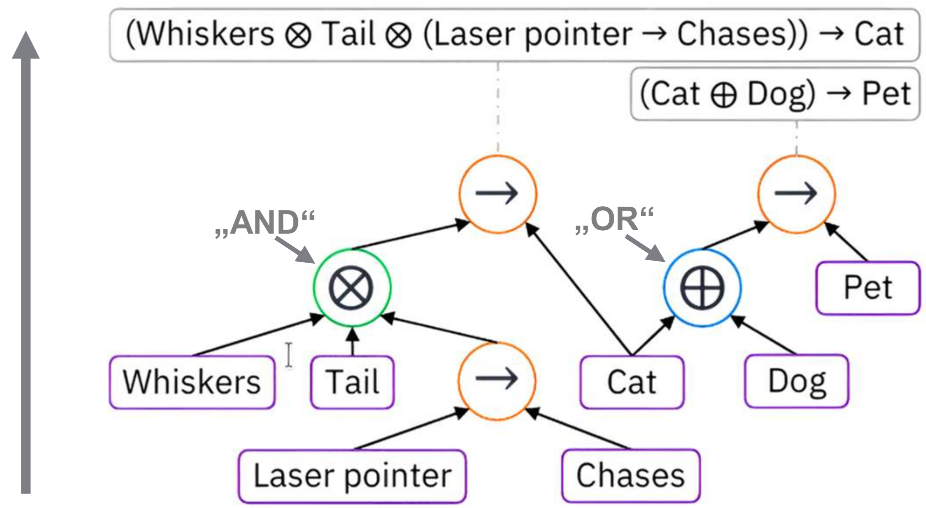
Adapted from Shakarian et al., 2023

Bidirectional Inference

(Riegel et al., 2020)

Upward pass

- Compute the truth bounds of a formula based on its subformulas
- Start from atoms, propagate upwards towards complex formulas



Adapted from Shakarian et al., 2023

JOHANNES KEPLER UNIVERSITY LINZ

Bidirectional Inference

Algorithm 1: Upward pass to infer formula truth value bounds from subformulae bounds

function `upwardPass`(formula z):
- **for** operand x_i of z, $i \in I$ **do** `# propagate bounds upward from leaves`
  - `upwardPass`(x_i)
- **if** $z = \neg x$ **then** „NOT“ `# negation`
  - `aggregate`(z, $(1 - U_x,\ 1 - L_x)$)
- **else if** $z = {}^{\beta}(\bigoplus_{i \in I} x_i^{\oplus w_i})$ **then** „OR“ `# multi-input disjunction`
  - `aggregate`(z, $({}^{\beta}(\bigoplus_{i \in I} L_{x_i}^{\oplus w_i}),\ {}^{\beta}(\bigoplus_{i \in I} U_{x_i}^{\oplus w_i}))$)
- `# Other operations are handled as combinations of the above`

function `aggregate`(formula z, (L'_z, U'_z)):
- $(L_z, U_z) := (\max\{L_z, L'_z\},\ \min\{U_z, U'_z\})$ `# tighten existing bounds`

Adapted from Riegel et al., 2020

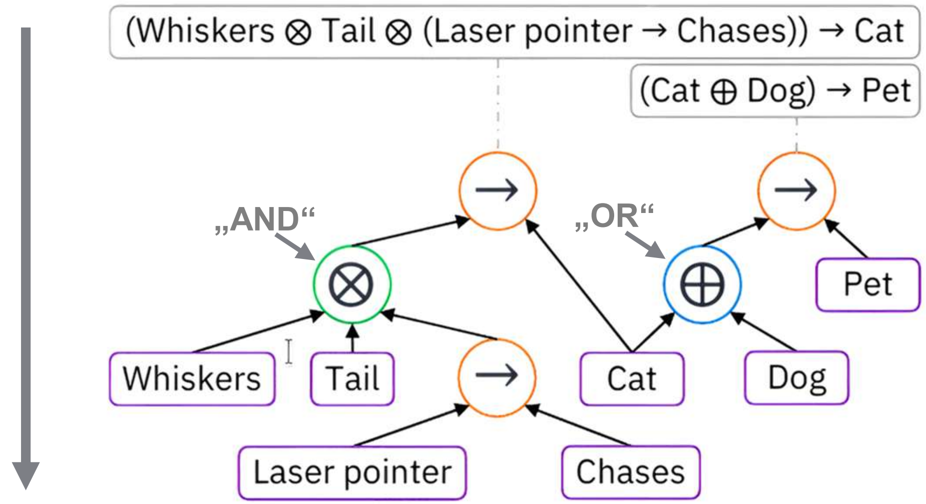
JOHANNES KEPLER UNIVERSITY LINZ

Bidirectional Inference

(Riegel et al., 2020)

Downward pass

- Compute the truth bounds of subcomponents based on the formula
- Logical inverses of upward computations



Adapted from Shakarian et al., 2023

Bidirectional Inference

Algorithm 2: Downward pass to infer subformula truth value bounds from formula bounds

function `downwardPass`(formula z):

- **for** operand x_j of z, $j \in I$ **do**
  - **if** $z = \neg x$ **then** „NOT“ `# negation`
    - `aggregate`$(x, (1 - U_z,\ 1 - L_z))$
  - **else if** $z = {}^{\beta}(\bigoplus_{i \in I} x_i^{\oplus w_i})$ **then** „OR“ `# multi-input disjunction`
    - $L'_{x_j} := {}^{\beta/w_j}((\bigotimes_{i \neq j}(1 - U_{x_i})^{\otimes w_i/w_j}) \otimes L_{\bigoplus_i x_i}^{\otimes 1/w_j})$ if $L_{\bigoplus_i x_i} > 1 - \alpha$, else 0
    - $U'_{x_j} := {}^{\beta/w_j}((\bigotimes_{i \neq j}(1 - L_{x_i})^{\otimes w_i/w_j}) \otimes U_{\bigoplus_i x_i}^{\otimes 1/w_j})$ if $U_{\bigoplus_i x_i} < \alpha$, else 1
    - `aggregate`$(x_j, (L'_{x_j},\ U'_{x_j}))$
  - `downwardPass`(x_j) `# propagate bounds downward to leaves`

Adapted from Riegel et al., 2020

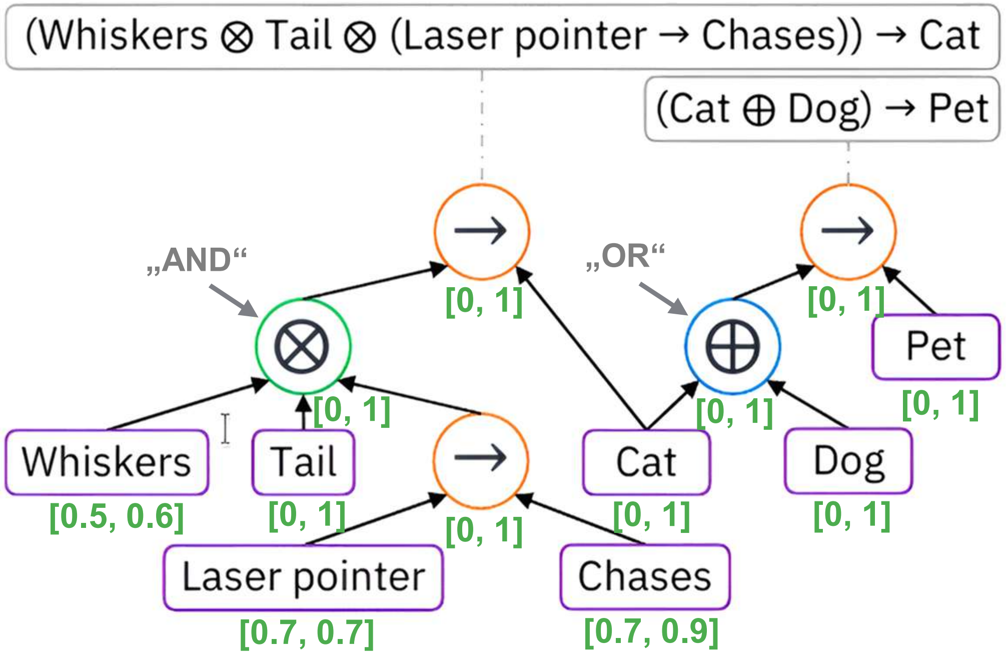
Training

(Riegel et al., 2020)

- Possible via backpropagation

$$\underbrace{\min_{B,W}}_{\text{trainable parameters}} \underbrace{E\,(B,W)}_{\text{e.g. MSE}} + \underbrace{\sum_{k\in N} \max(0, L_{B,W,k} - U_{B,W,k})}_{\text{contradiction loss}}$$

- No consistent knowledge base needed
- Can’t enforce all constraints in the knowledge base



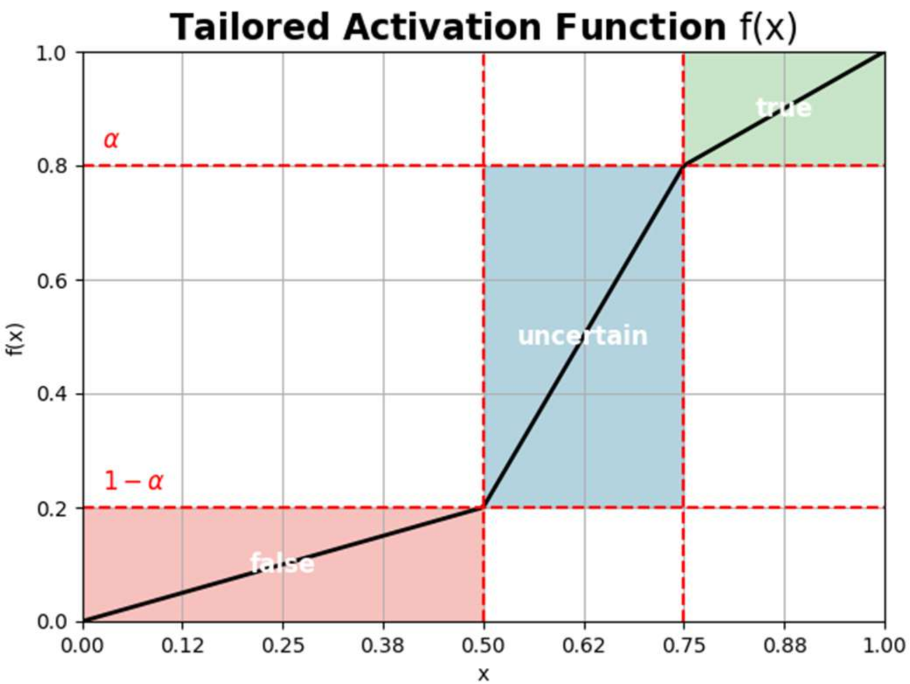
Adapted from Shakarian et al., 2023

Training

(Riegel et al., 2020)

- Possible via backpropagation

$$\underbrace{\min_{B,W}}_{\text{trainable parameters}} \underbrace{E\,(B,W)}_{\text{e.g. MSE}} + \underbrace{\sum_{k\in N} \max(0, L_{B,W,k} - U_{B,W,k})}_{\text{contradiction loss}}$$



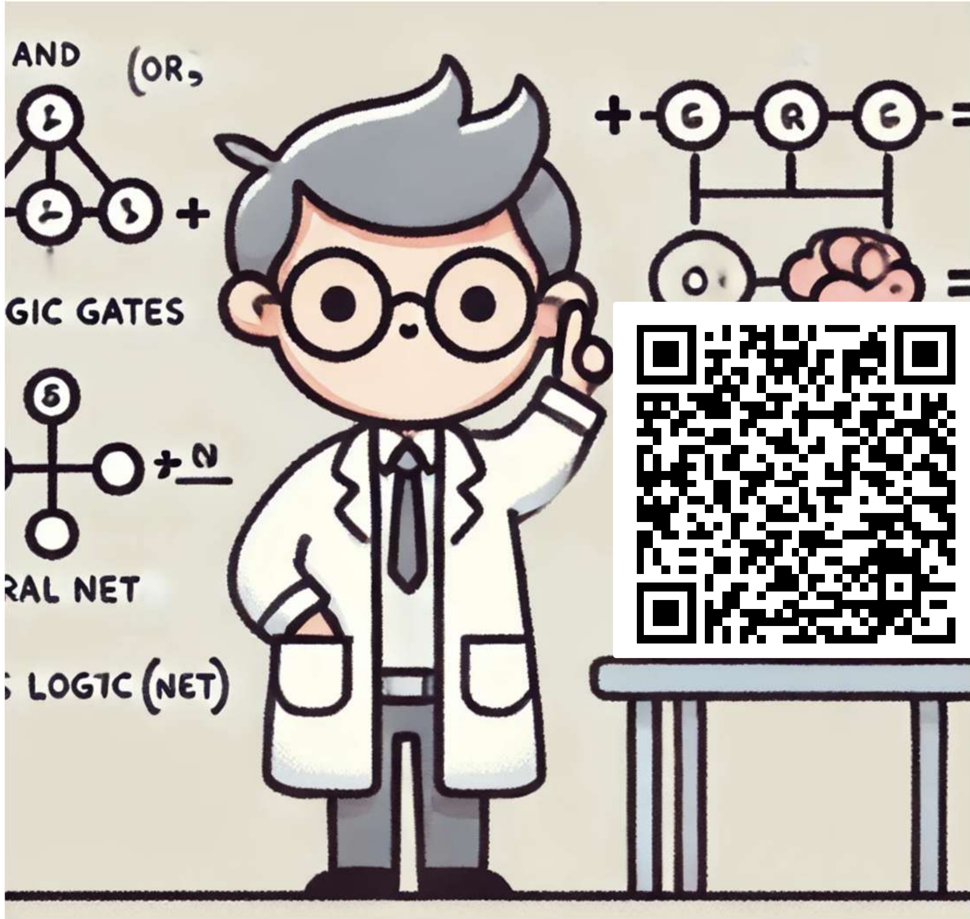
Adapted from Shakarian et al., 2023, and Riegel et al., 2020

Workflow

1. Construct knowledge base
2. Pass samples with truth bounds to the model
3. Perform inference until convergence
4. Compute loss on outputs and backpropagate

Tool Demo

Based on https://github.com/IBM/LNN/tree/master



Generated with DALL-E

Real-World Applications

Taken from ibm.com

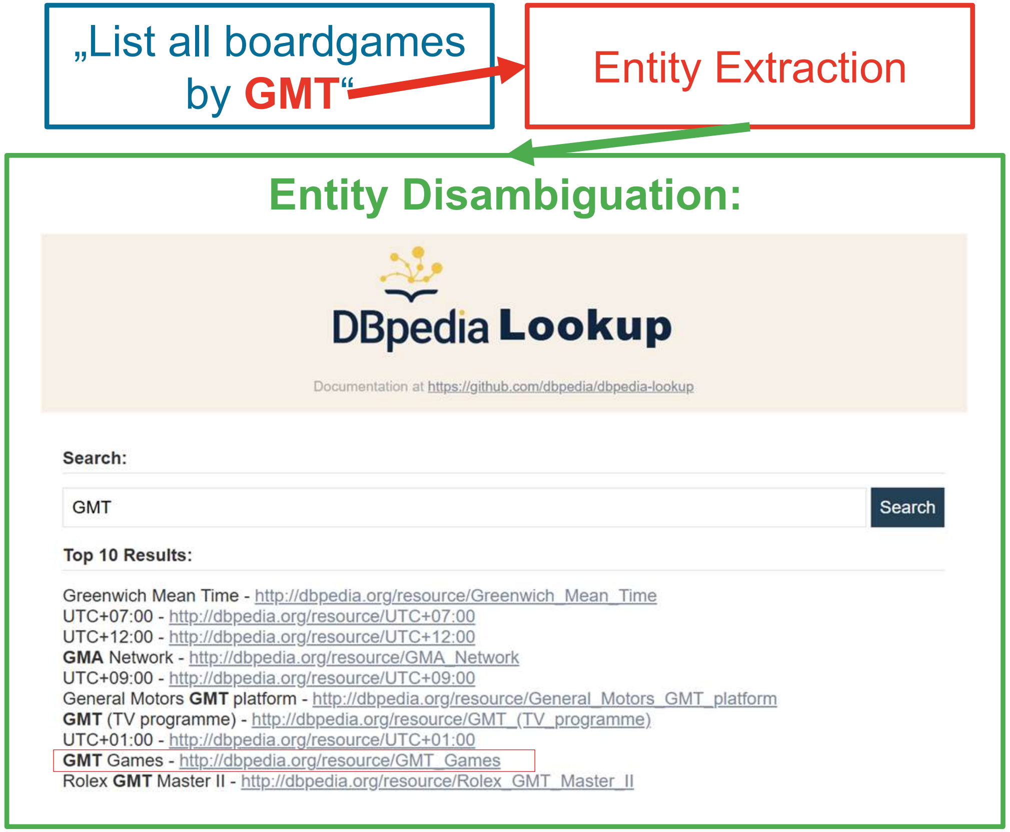
Almost exclusively published by IBM as they do not provide a robust library (perhaps internal) and proper documentation

Entity Linking with LNN-EL

(Jiang et al., 2020)

- Use rule-based disambiguation models from human experts
- Features:
  - Non-embedding based (Jaccard, etc.)
  - Embedding based (BERT, prior EL methods)

- We have a knowledge graph → LNN 🎉

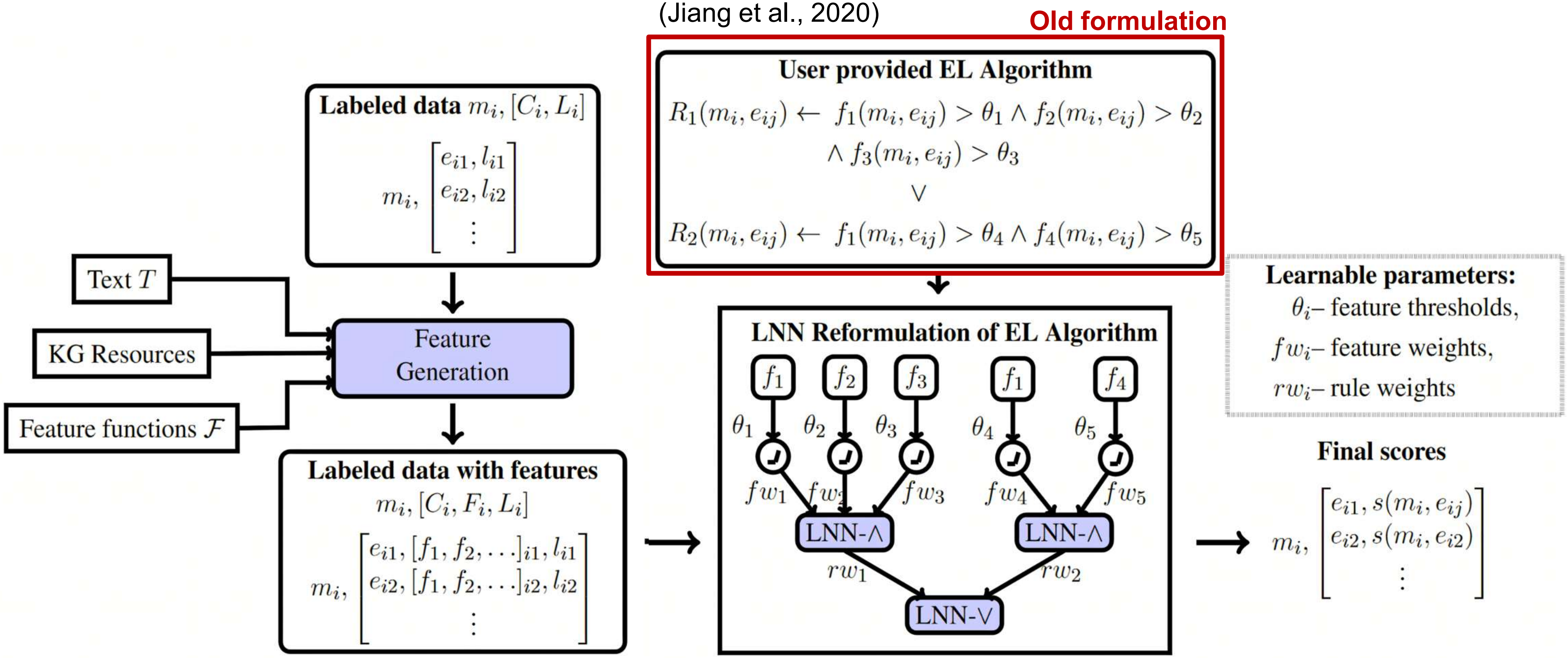


Adapted from Jiang et al., 2020

JOHANNES KEPLER UNIVERSITY LINZ

Entity Linking with LNN-EL

(Jiang et al., 2020)



Adapted from Jiang et al., 2020

Entity Linking with LNN-EL

(Jiang et al., 2020)

- Advantages of LNN-EL: (Jiang et al., 2020)
  - Interpretable
  - Extensible structure
    - Also with embeddings
    - Rule-designers refine the model more informedly

👑 State of the art performance, competitive to black-box models

- Transferable across datasets from the same domain without finetuning

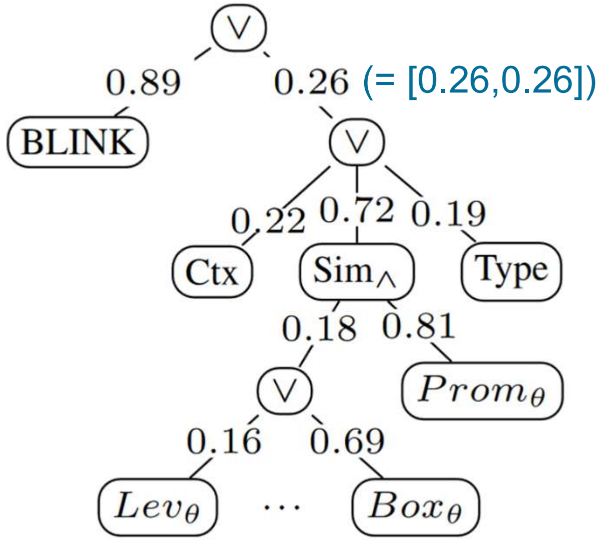
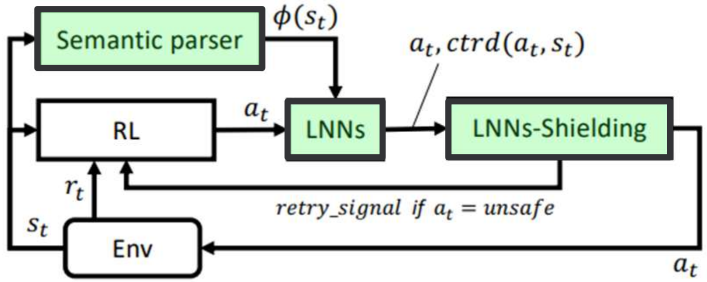


Figure adapted from Jiang et al., 2020

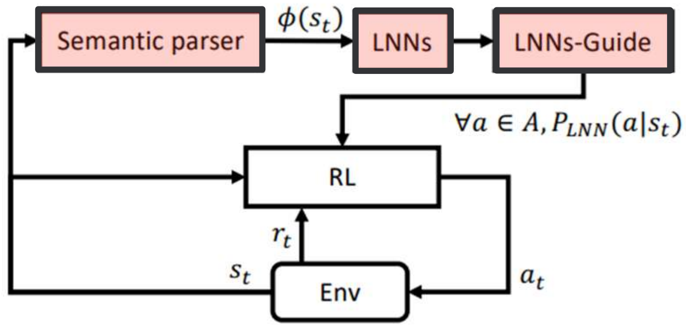
LNNs in Reinforcement Learning

- Augmenting Reinforcement Learning agents with rules about the world ([A] Kimura et al., 2021)
- LNN Shielding and/or Guiding ([B] Kimura et al., 2021)
    - Contradiction Loss
- Speed up the agent‘s learning ([B] Kimura et al., 2021)

- Hand-crafted rules about “TextWorld Commonsense“ (TWC) (Murugesan et al., 2021)
    - Text-based adventures
    - E.g., “if the agent is carrying an item, it cannot also be in that item”



i. Reinforcement learning with LNNs-Shielding



ii. Reinforcement learning with LNNs-Guide

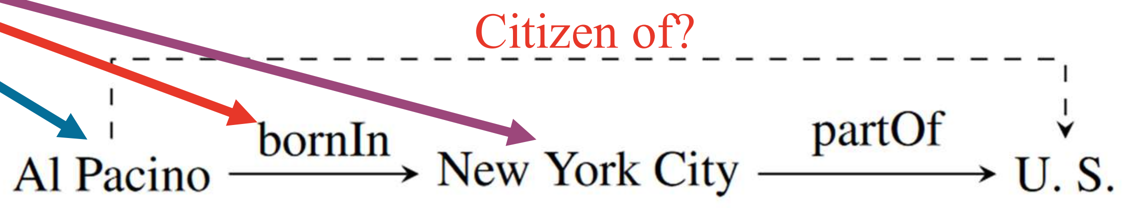
Adapted from [B] Kimura et al., 2021

Knowledge Base Completion

([A] Sen et al., 2020)

- First order logic is useful for inferring missing facts
- Instead: Mine rule triplets <h,r,t>
  - Contrastive Learning

$$\forall P, N \exists L : \mathrm{bornIn}(P, C) \land \mathrm{partOf}(C, N) \rightarrow \mathrm{citizenOf}(P, N)$$



Adapted from [A] Sen et al., 2022

- Rule-based methods:
  Interpretable
- Embedding-based:
  High accuracy
  e.g., ConvE (Dettmers et al., 2017)
- Real-world can better be mapped with degrees of truth
- Basically: Enumerate all paths (up to constraints) and weight them;
  Time-Constraints… → Details after the presentation

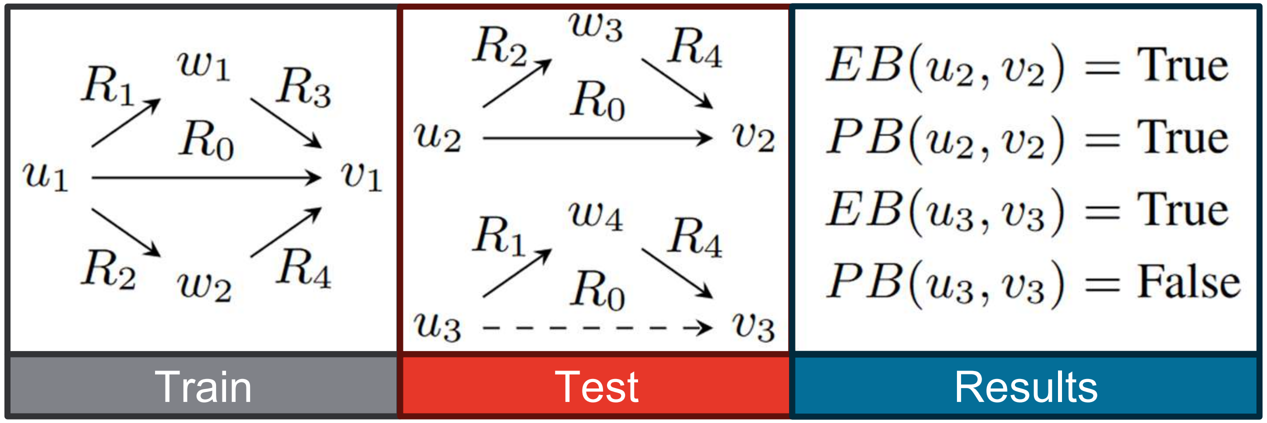
Knowledge Base Completion

([A] Sen et al., 2020)

- Allows for learning from real-world, noisy data

Datasets:

- Major Rule-based approaches:
  - Edge-based (EB):
    - Individual relations
    - Generalizes better
  - Path-based (PB):
    - Entire paths
    - Preserve order and context
- Hence, EB- and PB-LNN were introduced



Adapted from [A] Sen et al., 2022

Knowledge Base Completion

([A] Sen et al., 2020)

- **Path Based** LNN even beats embedding based methods (metric: Mean Reciprocal Rank and Hits)
  - With and without inducing graph embeddings
  - 👑 Best performance on all tested datasets, state of the art (even similar “speed”)
  - Training with rank loss (positive and negative examples, no consistency enforcement)

- Inner-Workings:
  - Each relation path <h,**r**,t> is a conjunction of relations
  - Disjunction over these conjunctions (only one path must hold)
  - Count how many paths can be established (brute-force up to length, $P_r(h,t)$) from h to t and how the model weights them $\phi_w(e_r)$

$$h \xrightarrow{r_1} x_1 \xrightarrow{r_2} \cdots \xrightarrow{r_m} t$$

$$\text{PB-LNN}(h,r,t) = \sum_{r=r_1,\ldots,r_m} |P_r(h,t)|\ \phi_w(e_r)$$

Further Applications

- Decision support and planning (Hoang, 2022)
  - Could sadly only access the abstract
  - Builds on a standard RL or MDP solver (learn a policy)
  - Train LNN based on the obtained policy
  - Especially: Supply-Chain optimization

- Inductive Logic Programming (ILP) for noisy data ([B] Sen et al., 2022)
  - Task: Learn new first order rules to explain the data
  - Still “primitive“ (yet state of the art performance on the given datasets)
  - Applicability of many rule templates (User-Guided)
  - Neural selection of predicates

Strengths and Weaknesses

Taken from https://de.m.wikipedia.org/wiki/Datei:Kahoot_Logo.svg

Strengths of LNNs

(Shakarian et al., 2023)

- Integration of symbolic and subsymbolic learning

- No need for (perfect) consistency in knowledge base ✓ **LNNs** ✕ **Logic Tensor Networks**
- Truth intervals approximate confidence intervals ✓ **LNNs** ✕ **Logic Tensor Networks**
- Explainable outputs ✓ **LNNs** ✕ **Ontological Networks**
- Open world reasoning: Nodes are initialized as uncertain (truth bounds: [0, 1]) rather than false ✓ **LNNs** ✕ **Logic Tensor Networks**

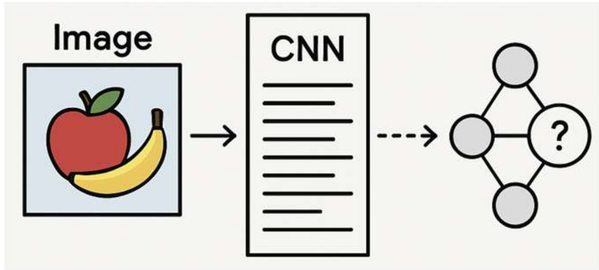
Weaknesses of LNNs

- A-priori definition of the knowledge base (knowledge bottleneck) (Shakarian et al., 2023)

- Not applicable if constraints need to be satified at all times (Shakarian et al., 2023)
- No interpretable combination with perceptual networks possible (yet) (Shakarian et al., 2023)
- Multiple complex mechanisms needed (bidirectional inference, backpropagation)

Future Work

Future Work



Generated with GPT-4o image generation

- Integration with perception: symbol grounding (Shakarian et al., 2023)
  - How the symbols get their meaning from raw data
  - E.g., what is this CNN output neuron?

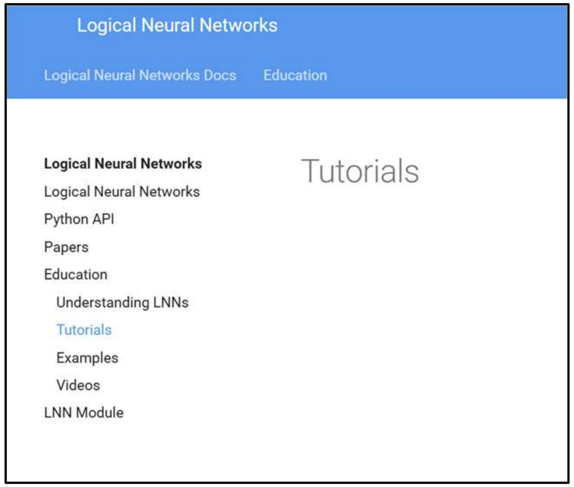
- Enforcing hard constraints for consistency (Shakarian et al., 2023)
  - Requirement for e.g., robotics, aerospace
  - Only hypothesized in the original paper, no implementations

Generated with DALL-E

- Automated knowledge learning
  - Learn knowledge base automatically – and make it more dynamic
  - Starts in Neuro-Symbolic ILP ([B] Sen et al., 2022)

Future Work

- Documentation (Our opinion)
    - Especially Riegel et al., 2020
    - Had to reverse-engineer the GitHub repository https://github.com/IBM/LNN
        - Also half-finished (just like docs Web-pages)
        - Developments stopped 2 years ago



https://ibm.github.io/LNN/education/tutorials/tutorials.html, retrieved on Mach 20th, 2025

- More benchmarks for neurosymbolic architectures (Ott et al., 2023)
    - Example: Interactive fiction
        - “TextWorld Commonsense“ Environment (Murugesan et al., 2021)

References

- Dettmers, T., Minervini, P., Stenetorp, P., & Riedel, S. (2017). Convolutional 2D Knowledge Graph Embeddings. AAAI Conference on Artificial Intelligence.
- Hoang, N. L., & Zadorojniy, A. (2022, February). Logical Neural Networks to Serve Decision Making with Meaning. In AAAI Conference on Artificial Intelligence. https://research.ibm.com/publications/logical-neural-networks-to-serve-decision-making-with-meaning
- Jiang, H., Gurajada, S., Lu, Q., Neelam, S., Popa, L., Sen, P., Li, Y., & Gray, A. (2021). LNN-EL: A neuro-symbolic approach to short-text entity linking. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 775–787). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.64

References cont.

- [A] Kimura, D., Chaudhury, S., Ono, M., Tatsubori, M., Agravante, D. J., Munawar, A., Wachi, A., Kohita, R., & Gray, A. (2021). LOA: Logical optimal actions for text-based interaction games. In H. Ji, J. C. Park, & R. Xia (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations* (pp. 227–231). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-demo.27
- [B] Kimura, D., Chaudhury, S., Wachi, A., Kohita, R., Munawar, A., Tatsubori, M., & Gray, A.G. (2021). Reinforcement Learning with External Knowledge by using Logical Neural Networks. ArXiv, abs/2103.02363. https://doi.org/10.48550/arXiv.2103.02363

References cont.

- Murugesan, K., Atzeni, M., Kapanipathi, P., Shukla, P., Kumaravel, S., Tesauro, G., Talamadupula, K., Sachan, M., & Campbell, M. (2021). Text-based RL agents with commonsense knowledge: New challenges, environments and baselines. *Thirty-Fifth AAAI Conference on Artificial Intelligence*. http://dx.doi.org/10.1609/aaai.v35i10.17090
- Ott, J., Ledaguenel, A., Hudelot, C., & Hartwig, M. (n.d.). How to Think About Benchmarking Neurosymbolic AI? Retrieved March 20, 2025, from https://ceur-ws.org/Vol-3432/paper22.pdf

References cont.

- Riegel, R., Gray, A., Luus, F., Khan, N., Makondo, N., Akhalwaya, I. Y., ... & Srivastava, S. (2020). Logical neural networks. arXiv preprint arXiv:2006.13155. https://doi.org/10.48550/arXiv.2006.13155
- [A] Sen, P., Carvalho, B. W., Abdelaziz, I., Kapanipathi, P., Roukos, S., & Gray, A. (2022, December). Logical neural networks for knowledge base completion with embeddings & rules. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (pp. 3863-3875). https://doi.org/10.18653/v1/2022.emnlp-main.255

References cont.

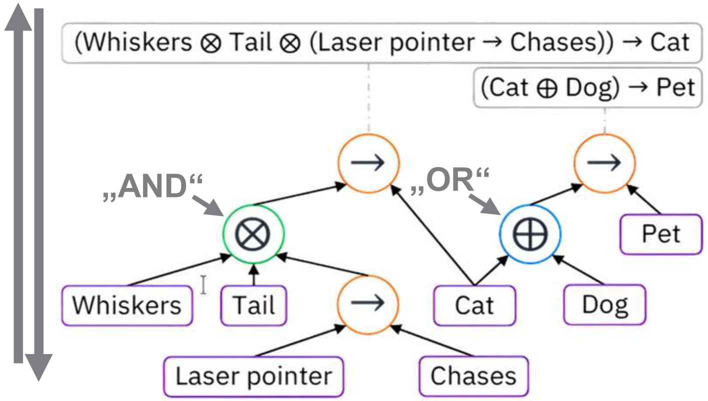
- [B] Sen, P., de Carvalho, B. W., Riegel, R., & Gray, A. (2022, June). Neuro-symbolic inductive logic programming with logical neural networks. In Proceedings of the AAAI conference on artificial intelligence (Vol. 36, No. 8, pp. 8212-8219). https://doi.org/10.1609/aaai.v36i8.20795
- Shakarian, P., Baral, C., Simari, G. I., Xi, B., & Pokala, L. (2023). *Neuro Symbolic Reasoning and Learning*. Springer. https://doi.org/10.1007/978-3-031-39179-8

Questions?

Martin Dallinger,
Lisa Krimbacher

JOHANNES KEPLER
UNIVERSITY LINZ

Recap



Adapted from Shakarian et al., 2023

Slides