

Da Lógica ao Aprendizado: Um Framework para Converter Diagramas de Decisão Sentenciais em Redes Neurais Diferenciáveis

Autor: Matheus Sanches Jurgensen

Orientador: Denis Deratani Mauá

Co-Orientador: Jonas Rodrigues Lima Gonçalves

Programação

01

Motivações

02

Definições

03

Metodologia

04

Resultados

05

Conclusão



01

Motivações

Motivações

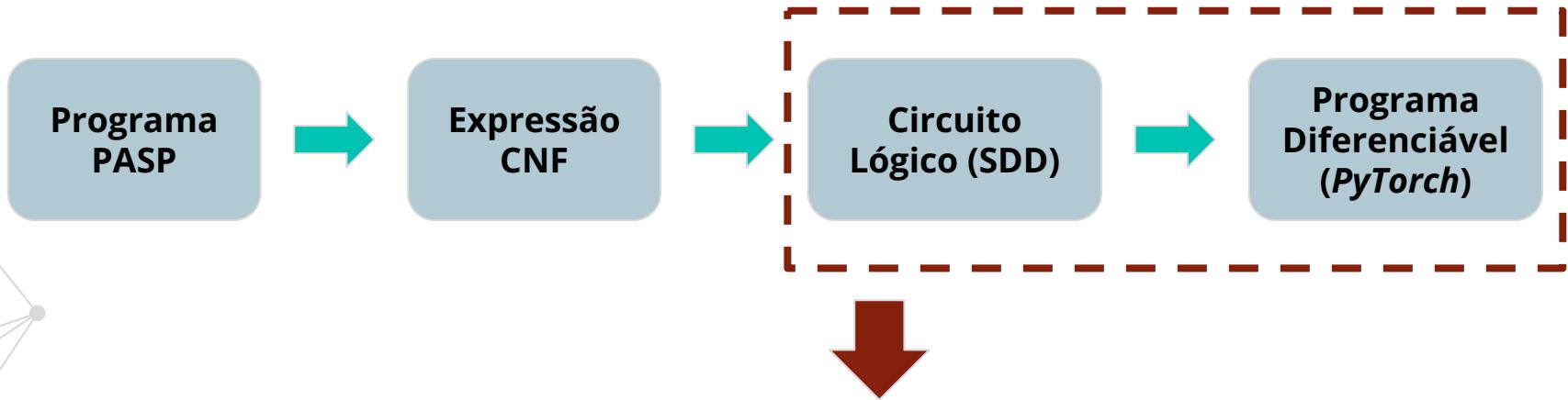
Inteligência Artificial Neuro-Simbólica

- É a área de estudo que busca unir duas abordagens:
 - **Raciocínio lógico/simbólico:** interpretabilidade.
 - **Redes neurais:** generalização e eficiência.
- **Projeto:** dPASP (*Differentiable Probabilistic Answer Set Programming*)
 - **Objetivo central:** criar programas diferenciáveis a partir de problemas definidos em answer set programming probabilístico (PASP).

```
% Daily earthquake probabilities.  
0.05::earthquake(heavy); 0.15::earthquake(mild); 0.8::earthquake(none).  
% Error rates.  
0.90::alarm :- burglary, earthquake(heavy).  
0.85::alarm :- burglary, earthquake(mild).  
0.80::alarm :- burglary, earthquake(none).  
0.05::alarm :- not burglary, earthquake(mild).  
0.10::alarm :- not burglary, earthquake(heavy).  
% Help of neighbors.  
0.8::calls(X) :- alarm, neighbor(X).  
0.1::calls(X) :- not alarm, neighbor(X).  
% Bert and Ernie are Elmo's neighbors.  
neighbor(bert). neighbor(ernie).
```

PASP

Motivações dPASP



- **Objetivos do trabalho:**
 - Conversão de **SDDs** (Diagramas de decisão sentencial) em **redes neurais**.
 - Integrar probabilidades nas estruturas lógicas.
 - Inferência e otimização probabilística de forma otimizada (GPU).
 - Desenvolvimento de um sistema com foco em **qualidade de software**.

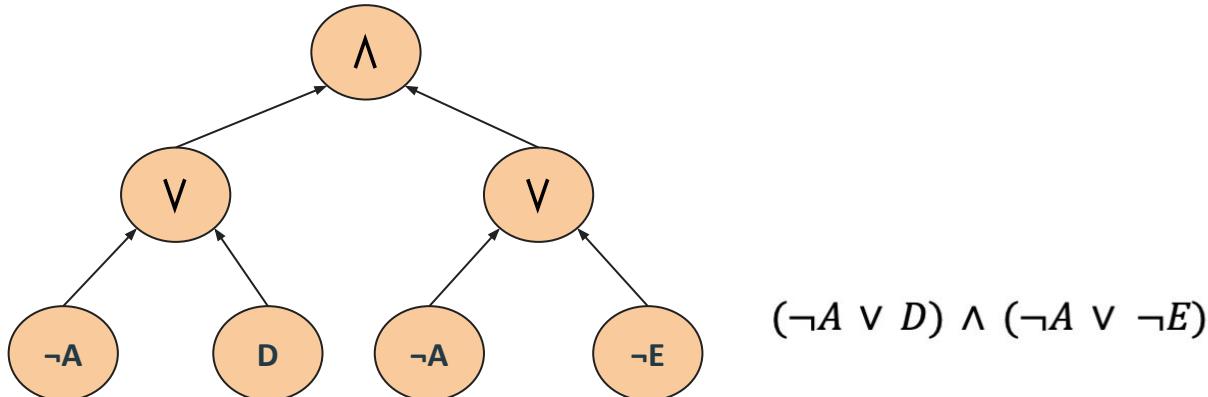
02

Definições

Definições

Círculo Lógico e NNF

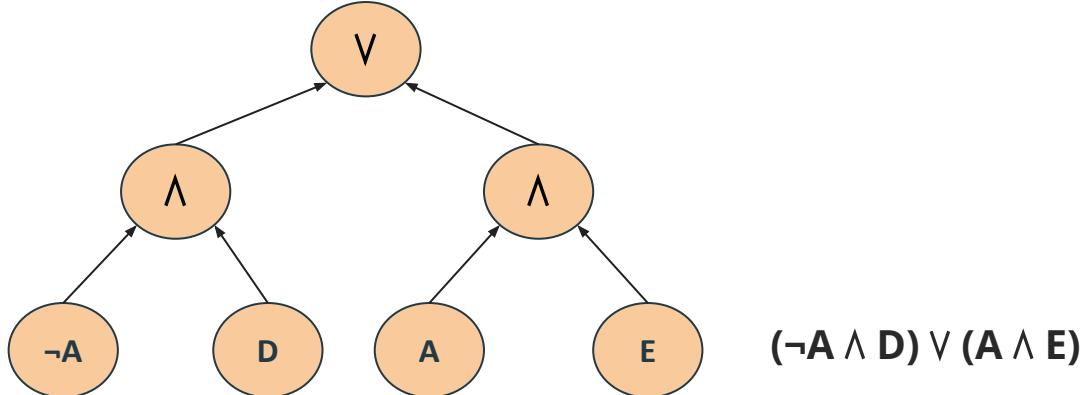
- **NNF (Negation Normal Form):** Grafo acíclico direcionado que representa uma função booleana. É um tipo de **círculo lógico**.
 - **Nós internos:** Operações lógicas (*AND* e *OR*).
 - **Folhas:** *True*, *False*, *X* ou $\neg X$.



Definições

SDD: Diagrama de Decisão Sentencial

- Um **subconjunto dos NNF**, com propriedades adicionais:
 - **Decomponibilidade:** Em toda conjunção (\wedge), os filhos não compartilham variáveis.
 - **Determinismo:** Em toda disjunção (\vee), os filhos são logicamente contraditórios.
 - **Estruturação:** Variáveis apresentam uma organização hierárquica (*vtree*).
- **Vantagens:** tratabilidade de operações, mantendo certa sucintez.



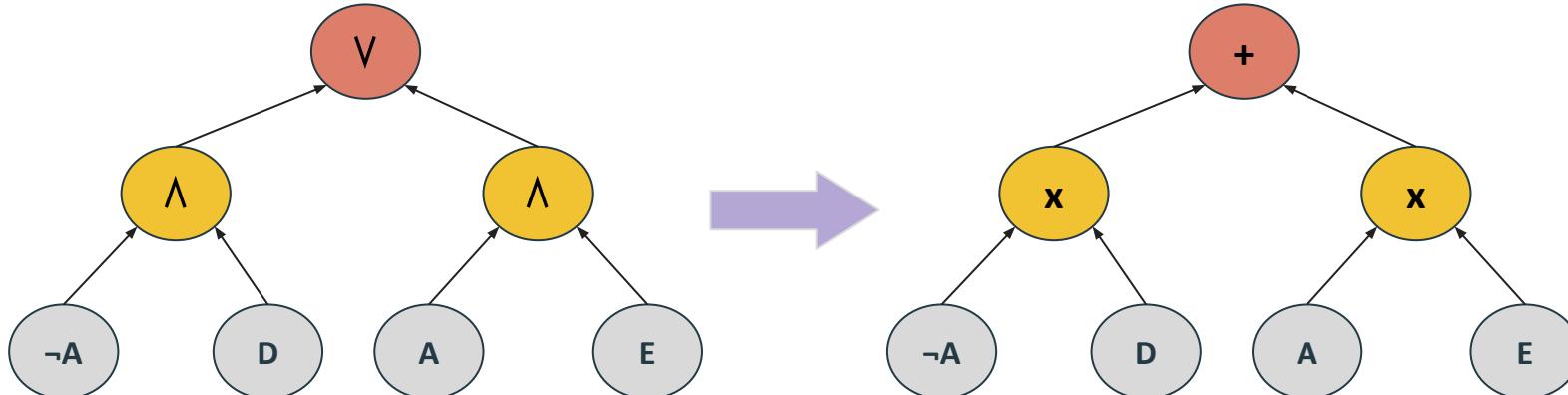
03

Metodologia

Metodologia

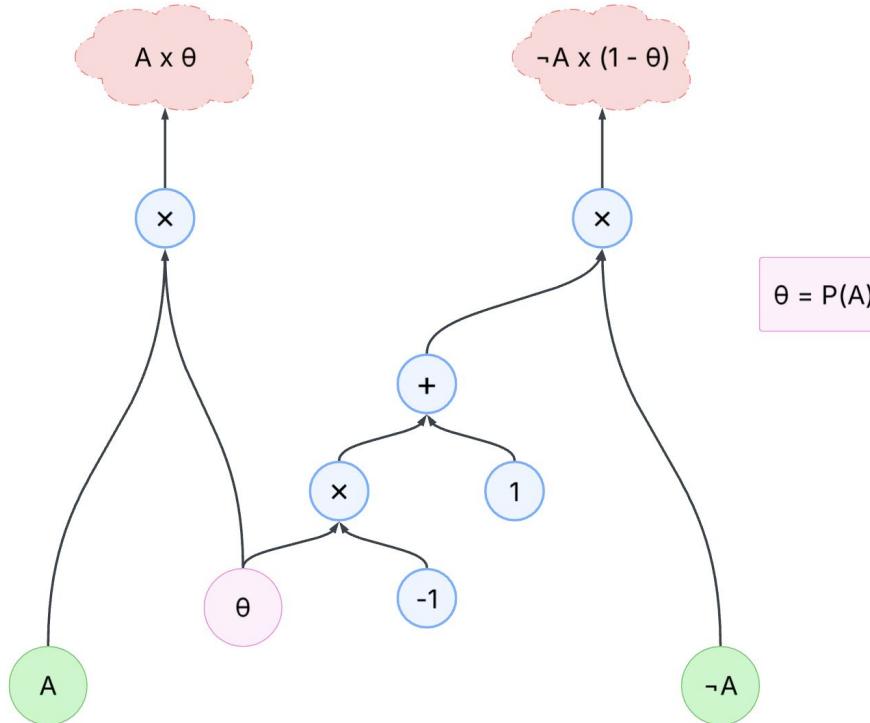
Transformação do Circuito Lógico

- Transformação do SDD em um **círculo aritmético**.
 - Mapeamento das operações lógicas em operações aritméticas diferenciáveis.
 - Nó AND → Nó de **Produto**.
 - Nó OR → Nó de **Soma**.



Metodologia

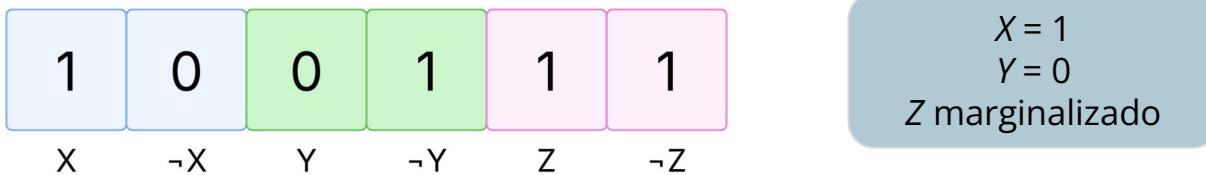
Integração das Probabilidades



Metodologia

Inferência e Treinamento

- **Camada de *input*:**
 - $2N$ nós (N = número de variáveis).
 - X e $\neg X$ em nós de entrada diferentes.



- **Inferência:**
 - **Propagação:** camada de input (folhas) → nó de saída (raiz).
 - Para o vetor de exemplo acima, a propagação computa $P(X=1, Y=0)$
- **Treinamento:**
 - Descobrir probabilidades desconhecidas por meio de um **conjunto de dados**.
 - Elas se tornam parâmetros a serem otimizados por **retropropagação**.
 - Técnica do **gradiente descendente**.

04

Resultados

Resultados

Caso de Estudo: Problema do Alarme

```
% ground probabilistic facts
0.1::burglary.
0.2::earthquake.
0.7::hears_alarm(john).

% ground rules
alarm :- burglary.
alarm :- earthquake.
calls(john) :- alarm, hears_alarm(john).
```

alarm.pasp

- **3 variáveis probabilísticas**
 - *burglary, earthquake e hears_alarm(john)*
- **2 variáveis lógicas**
 - *alarm e calls(john)*
- **Conversão em SDD**
 - Circuito com 19 nós

Resultados

Caso de Estudo: Problema do Alarme

Inferência Probabilística

- Execução de múltiplas **consultas de probabilidade condicional**.
- **Resultados:**
 - Corretude do circuito para inferências.
 - Eficiência e rapidez.

Query	Probability	Time (ms)
$P(\text{burglary} \mid \text{calls(john)})$	0.357	0.399
$P(\text{earthquake} \mid \text{calls(john)})$	0.714	0.386

Probabilidade das causas (não observáveis), dadas as consequências (observáveis)

Resultados

Caso de Estudo: Problema do Alarme

Aprendizado Probabilístico

- **Objetivo:** Aprendizado das probabilidades das variáveis probabilísticas.
- **Conjunto de dados:**
 - Apenas as variáveis observáveis (*calls(john)* e *alarm*).
 - Geração de um conjunto sintético de dados (utilizando as probabilidades já conhecidas mas omitindo as variáveis probabilísticas).
- **Metodologia:**
 - Hiperparâmetros: número de épocas e tamanho do conjunto de dados.
 - Cada combinação de hiperparâmetros executada 10 vezes.

Resultados

Caso de Estudo: Problema do Alarme

- Seleção das piores e melhores combinações:
 - Média das probabilidades calculadas \pm erro padrão
- **Piores casos:** seguiram na **direção correta**, mas com **maior variabilidade** (poucos dados).
- **Melhores casos:** valores bem **precisos** e **menos variáveis** (mais dados e épocas).

Samples / Epochs	Burglary (True: 0.1)	Earthquake (True: 0.2)	Hears Alarm (True: 0.7)	Avg. Time (ms)
Worst Configurations				
1,000 / 1,000	0.114 ± 0.019	0.197 ± 0.019	0.697 ± 0.020	1,010
10,000 / 5,000	0.105 ± 0.005	0.195 ± 0.003	0.697 ± 0.006	19,321
Best Configurations				
50,000 / 10,000	0.099 ± 0.001	0.199 ± 0.001	0.700 ± 0.002	78,622
100,000 / 10,000	0.100 ± 0.001	0.199 ± 0.001	0.699 ± 0.001	107,696

Resultados

Qualidade de Engenharia de Software

- **Princípios de Design (SOLID)**
 - **Princípio da Responsabilidade Única:**
 - Cada classe cumpre uma única função.
 - *SDDParser, NetworkBuilder, QueryExecutor*, etc.
 - **Princípio Aberto-Fechado:**
 - Arquitetura orientada a abstrações.
 - Possibilita extensões sem modificar as classes já existentes.
- **Confiabilidade**
 - 149 testes automatizados (unitários e de integração) em *PyTest*.
- **Manutenibilidade**
 - Código documentado, modular e sem duplicações.

05

Conclusão

Conclusão

Contribuições

- **Framework de Lógica Diferenciável:**
 - Transformação de circuitos lógicos em redes neurais em *PyTorch*.
 - Otimização dos parâmetros probabilísticos via gradiente descendente.
 - Recuperação precisa de probabilidades latentes com dados parcialmente observados.
- **Excelência em Engenharia de Software:**
 - Sistema robusto, modular e com extensa cobertura de testes automatizados.
 - Aderência aos princípios **SOLID** de qualidade de software.

Conclusão

Próximos Passos

- Suporte a **disjunções anotadas**.
 - Uma única cláusula com várias alternativas mutuamente exclusivas possíveis.
 - Exige maiores restrições sobre as probabilidades otimizadas.

% Daily earthquake probabilities.

0.05::earthquake(heavy); **0.15**::earthquake(mild); **0.8**::earthquake(none).

- **Otimizações arquiteturais.**
 - Organização do circuito aritmético em camadas com nós de mesmo tipo (Soma/Produto).
 - Maximização do paralelismo (GPU).

Referências

Darwiche, A. (2002). **A Knowledge Compilation Map**. Journal of Artificial Intelligence Research, 17, 229–264.
<https://doi.org/10.1613/jair.989>

Darwiche, A. (2011). **SDD: A New Canonical Representation of Propositional Knowledge Bases**.
Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI), 819–826.

Garcez, A. S. d., Besold, T. R., & Lamb, L. C. (2019). **Neural-Symbolic Learning and Reasoning: Contributions and Challenges**. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 9489–9496.

Darwiche, A. (1999). **On the Tractable Counting of Theory Models and Its Application to Belief Revision and Truth Maintenance**. Journal of Applied Non-Classical Logics, 9(1), 37–60.

Chavira, M., & Darwiche, A. (2008). **On Probabilistic Inference by Weighted Model Counting**. Artificial Intelligence, 172(6–7), 772–799.

Geh, R. L., Gonçalves, J., Silveira, I. C., Mauá, D. D., & Cozman, F. G. (2023). **dPASP: A Comprehensive Differentiable Probabilistic Answer Set Programming Environment For Neurosymbolic Learning and Reasoning**. arXiv preprint arXiv:2308.02944. <https://arxiv.org/abs/2308.02944>

Obrigado!

Perguntas?

mjurgensen@usp.br

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#) and infographics & images by [Freepik](#)