Relatório da 3° Avaliação de Inteligência Artificial

Luciana P. Nedel[1], Rafael H. Bordini[2], Flávio Rech Wagner[1], Jomi F. Hübner[3]

[1]Instituto de Computação – Universidade Federal do Amazonas (UFAM)
Av. General Rodrigo Octavio, 6200, Coroado I, CEP: 69080-900

[2]Departamento Faculdade de Tecnologia – Universidade Federal do Amazonas (UFAM)
Av. General Rodrigo Octavio, 6200, Coroado I, CEP: 69080-900

{dpb,lucas.oliveira,joao.reis,daniel.nunes,gabriel.assis}@icomp.ufam.edu.br

1. Introdução

A inteligência artificial (IA) tem sido como utilizada por empresas como ferramenta para automatização e aprimoramentos de eficiência nos seus serviços. Além disso, Com o advento de ferramentas como chatGPT® da OpenAI a IA começou a permear atividades humanas além do âmbito empresarial e acadêmico. Há uma tendência de que pessoas que saibam utilizar melhor essas ferramentas serão as que serão melhor sucedidas que as que não souberem utilizá-las [Lakhani 2023].

Tendo em vista que a IA está se popularizando cada vez mais, questões éticas relativas à confiabilidade e segurança da utilização da IA. Uma dessas questões se deve ao fato de modelos de aprendizagem de máquina e apredizagem profunda serem do tipo *black box*, i.e. podemos ver os resultados desses modelos mas não como esses resultados foram obtidos.

Uma alternativa para contornar esse problema e usar abordagens de IA conexionista com IA simbólica de forma a ter um sistema de aprendizado que possa ser avaliado. O foco desse trabalho é mostrar um exemplo de aplicação da combinação dessas técnicas para o problema do trem de Michalski.

2. Referencial Teórico

Apredizagem de Máquina e Aprendizagem profunda usam métodos estátisticos e, como dito na seção anterior, são do tipo *black blox* e não representam as inferências do raciocínio lógico simbólico das abordagens baseadas em lógica. Existe uma abordagem que combina as duas conhecida como IA neuro-simbólica. [Garcez and Lamb 2023]

O objetivo dessa abordagem é descrever o problema usando uma linguagem lógica simbólica e deixar o processamento de aprendizado para os algoritmos de aprendizagem conexionistas. Essa combinação é buscada devido aos seguintes motivos:

- os sistemas simbólicos possuem algumas limitações no que diz respeito a manipulação de informações quantitativas e dados imprecisos e/ou inexatos, e mesmo quando não tratam com esse tipo de informação o processamento tende a ser oneroso [Bratko 2001];
- os sistemas simbólicos permitem uma representação de conhecimento que pode ser facilmente interpretada por um especialista da área;
- os sistemas conexionistas tratam muito bem as informações quantitativas e permitem que se trabalhe com relativa facilidade com dados aproximados e/ou incorretos devido aos métodos estatísticos da rede;

- nos sistemas a representação do conhecimento é inerente à rede e não interpretáveis por métodos manuais convencionais

A abordagem de união entre desses métodos consiste de um processamento em conjunto em que cada abordagem, conexionista e simbólica processa em paralelo e realimentam a rede passando as informações via métodos de inserção de conhecimentos em redes neurais.

3. Descrição do Problema

No problema conhecido como "O trem de Michalski", a meta é classificar quais os trens que vão para leste e os que vão para oeste. A figura 1 ilustra todos os 10 exemplos utilizados no trabalho original dos trens que vão para o leste e os que vão para o oeste.

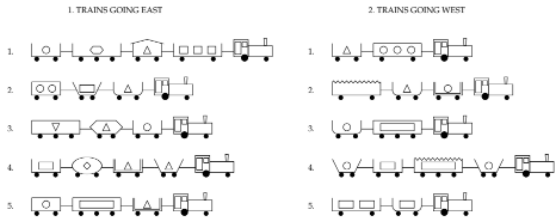

Figure 1. Original dataset para o problema do trem de Michalski

O *dataset* dos dez trens originais é descrito da seguinte forma:

- Quantidade de vagões (valor entre 3 e 5)
- Quantidade de cargas diferentes que pode levar (valor entre 1 a 4);

Cada vagão de trem é descrito por:

- Quantidade de eixo com rodas (valor entre 2 e 3);
- O comprimento (valor curto ou longo); (valor entre 1 a 4);
- Formato da carroceria do vagão:
  - retangulo-fechado
  - retangulo-aberto
  - duplo retangulo-aberto
  - elipse
  - locomotiva
  - hexagono
  - topo dentado
  - trapezio aberto
  - topo triangular-fechado;
- Quantidade de cargas no vagao (0 a 3);
- O formato da carga (círculo, hexágono, retângulo ou triângulo);

Para utilização durante o trabalho foram propostas 10 variáveis proposicionais que descrevem se qualquer par de tipos de carga estão ou não em vagões adjacentes do trem (já que cada carro carrega um único tipo de carga). Temos as seguintes relações com respeito aos vagões de um trem, cujo valor lógico varia entre -1 (Falso) e 1 (Verdadeiro):

- 1. existe um retângulo próximo de um retângulo

- 2. existe um retângulo próximo de um triângulo
- 3. existe um retângulo próximo de um hexágono
- 4. existe um retangulo próximo de um círculo
- 5. existe um triângulo próximo de um triângulo
- 6. existe um triângulo próximo de um hexágono
- 7. existe um triângulo próximo de um círculo
- 8. existe um hexágono próximo de um hexágono
- 9. existe um hexagono próximo de um círculo
- 10. existe um círculo próximo de um círculo

Há um único atributo de classe que define a direção de um trem: leste ou oeste. Observe que para atributos com múltiplos valores deve-se assinalar valores numéricos na ordem em que surgem, por exemplo, o tipo de carga deve ser 1 para denotar círculo, 2 para hexágono, 3 para retângulo, e assim por diante. Os neurônios correspondentes devem usar função de ativação linear, i.e. h(x) = x.

Note que basta o modelo precisa apenas classificar trens que vão para um dos lados, leste ou oeste, os que não forem classificados vão para o lodo oposto.

4. Descrição do Modelo e Implementação

Os modelos implementados neste trabalho baseiam-se na seção 10.6 do livro *Neural-symbolic cognitive reasoning*. As implementações e resultados foram feitos em Python, utilizando a biblioteca Keras, o repositório com as implementações podem ser vistas em [14]. Esta seção está dividida em duas partes, na qual a primeira comenta sobre a implementação de um modelo neuro simbólica mais profunda com 11 sub-redes que representam regras específicas (predicados). A segunda seção implementa uma solução em LTNTorch para resolver o problema dos trens de Michalsky considerando os 11 predicados iniciais, fornecido no arquivo de especificações.

4.1. Implementação I

Para a primeira implementação foi utilizado um modelo neuro simbólico com 11 sub-redes que representam predicados. A camada escondida utiliza a função de ativação ReLu, enquanto a camada de saída utiliza a função de ativação Sigmoid. O otimizador utilizado na rede é o RAdam (Rectified Adam). Portanto, a saída da rede está no intervalo [0, 1], no qual 1 representa leste e 0 oeste.

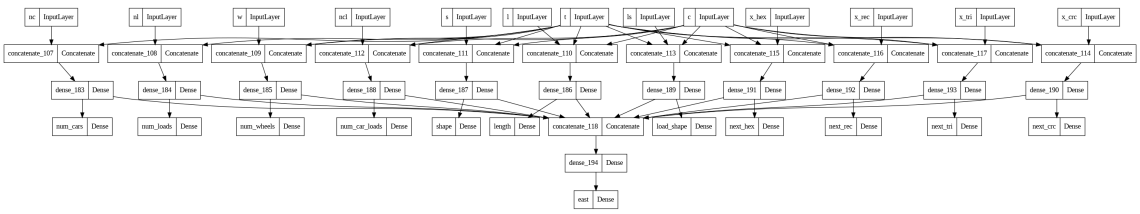

Figure 2. Modelo da rede neuro simbólica

Para o treinamento do modelo foram utilizadas as estratégias de divisão de modelo (70% para testes e 30% para validação) e *cross-validation*. Para o primeiro treinamento, 70 trens foram utilizados para treino e 30 para testes. Foi utilizado o otimizador RAdam, com learning rate 0.001 e a função de perda utilizada no treinamento foi a *Binary Cross*

Entropy. No segundo treinamento foi adotada a estratégia de *cross-validation*, que consiste em dividir o *dataset* em 5 partes diferentes, sendo uma delas utilizada para testes e as outras para treino. Assim como na primeira estratégia, foi utilizado o otimizador RAdam e a função de perda *Binary Cross Entropy*. Em ambas as estratégias foram executadas 100 épocas e os resultados serão disponibilizados na seção de Resultados.

5. Resultados

A partir da análise dos dados fornecidos pelos modelos de abordagem Neuro-Simbólicas, foi possível constatar resultados consistentes para os problemas elucidados, sem perda de desempenho em comparação com as abordagens somente conexionistas. Sendo assim, as técnicas abordadas provaram-se suficientemente satisfatórias para classificação binária a partir de predicados desejados.

References

Bratko, I. (2001). *Prolog programming for artificial intelligence*. Pearson education.

Garcez, A. d. and Lamb, L. C. (2023). Neurosymbolic ai: The 3 rd wave. *Artificial Intelligence Review*, 56(11):12387–12406.

Lakhani, K. (2023). Ai won't replace humans — but humans with ai will replace humans without ai.