

Relatório da 3ª Avaliação de Inteligência Artificial

Erlon Bié, Antônio Vinicius, Thayná Rosa, Glenn Fonseca

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

{erlon.bie, vinicius.cabral, thayna.rosa, glenn.fonseca}@icomput.ufam.edu.br

1 Introdução

As redes neurais artificiais ganharam notoriedade nas últimas décadas, através de técnicas como *Deep Learning*, que podem conseguir bastante precisão para executar determinadas tarefas. Porém, para conseguir essa precisão, modelos como *Deep Learning* precisam de muitos dados de entrada, tornando-se ineficientes para situações onde a quantidade de dados é escassa. Além disso, outro problema relacionado à modelos que utilizam grande quantidade de dados é a capacidade de representar o aprendizado do próprio modelo de uma forma que seja explicável e compreensível para os humanos.

Por conta desses problemas citados, o campo de conhecimento *Neuro-Symbolic AI (NSAI)* tem sido uma alternativa para evitar alguns desses problemas, combinando o poder de extração de atributos que o *Deep Learning* provou eficiente em realizar, com o capacidade expressiva da lógica para formalizar o conhecimento, também tendo como vantagem a fácil generalização de alguns problemas necessitando de poucos dados de entrada [1].

Áreas como carros autônomos, medicina e outras envolvendo segurança necessitam de modelos que possam explicar de forma clara e auditável em todas as etapas da geração do conhecimento, por isso sistemas *Neuro-simbólicos* podem se tornar essenciais num futuro próximo.

2 Referencial Teórico

A inteligência artificial é a área da ciência da computação que busca soluções para automação do comportamento inteligente [2], em qualquer domínio. Nesta seção vamos apresentar alguns métodos e conceitos da área que serão utilizados neste trabalho.

2.1 Sistemas Neuro-simbólicos

Conforme [3], os sistemas neuro-simbólicos tem o objetivo de integrar o raciocínio de alto nível com a percepção de baixo nível. A computação neural trabalha a percepção de baixo nível enquanto o raciocínio de alto nível é definido por proposições lógicas ou probabilísticas. A parte simbólica utiliza técnicas para resolver tarefas em grandes horizontes, mas em espaços discretos, enquanto a parte neural tem resultados em espaço contínuo, mas com um horizonte limitado [4].

A combinação de ambos os métodos tenta aproveitar as suas vantagens, onde os sistemas simbólicos pode resolver problemas de inferência com provas e os sistemas neurais usam do paralelismo e da generalização para a geração de conhecimento.

2.2 Proposicionalização

Proposicionalização é a conversão de um banco de dados relacional para uma tabela chave-valor, utilizável em aprendizado proposicional. Os algoritmos utilizam conhecimento do domínio em questão e os exemplos para encontrar features distintas e que possam diferenciar subconjuntos de exemplos. Dessa forma, eles devem ser distintos e com forte significado[5].

Conforme [5] os tipos de proposicionalização são: orientado a lógica e orientado ao dados. O orientado a lógica busca construir um conjunto de features de primeira ordem que são distintas e relevantes. Já o tipo orientado aos dados busca as relações e funções que geram as features de proposicionalização.

2.3 Redes Neurais Artificiais

As redes neurais artificiais são um tipo de modelo de machine learning e tem ficado popular para uso em classificação, clustering, reconhecimento de padrões e predição em muitas outras áreas do conhecimento. O potencial dessas redes estão na sua capacidade de paralelismo, velocidade de processamento e muitas outras propriedades de auto-aprendizagem e adaptatividade ao domínio alvo do aprendizado [6].

As redes neurais propagam as informações pelos seus neurônios, que devem interpretar os dados, rotular e agrupar de acordo com seu algoritmo. A classificação das redes neurais artificiais podem ser feitas de acordo com essa propagação: feed forward e feed backward neural network [6]. O single layer e o multi layer perceptron são exemplos de feed forward e Bayesian Regularized Neural Network e Competitive Networks são exemplos de feed backward neural networks. O deep learning é o termo que se refere à subárea de machine learning que trata de redes neurais artificiais com multi camadas complexas e tem formas mais complexas de conectar camadas.

3 Descrição do problema

O problema do trem de Michalski foi criado por Ryszard Michalski por meados dos anos 80. O mesmo é definido por apenas uma meta específica: classificar quais os trens que vão para leste e os que vão para oeste. Cada trem possui alguns atributos específicos para a resolução do problema, os quais podem ser observado na Figura 1 e serão descritos a seguir:

1. quantidade de vagões (valor entre 3 a 5)
2. quantidade de cargas diferentes que pode levar (valor entre 1 a 4)

3. para cada vagão de um trem:
 - (a) a quantidade de eixo com rodas (valor entre 2 e 3)
 - (b) o comprimento (valor curto ou longo)
 - (c) o formato da carroceria do vagão, o qual pode ser
 - i. retângulo-fechado
 - ii. retângulo-aberto
 - iii. duplo retângulo-aberto
 - iv. elipse
 - v. locomotiva
 - vi. hexágono
 - vii. topo dentado
 - viii. trapézio aberto
 - ix. topo triangular-fechado
 - (d) quantidade de cargas no vagão (0 a 3)
 - (e) o formato da carga (círculo, hexágono, retângulo ou triângulo)

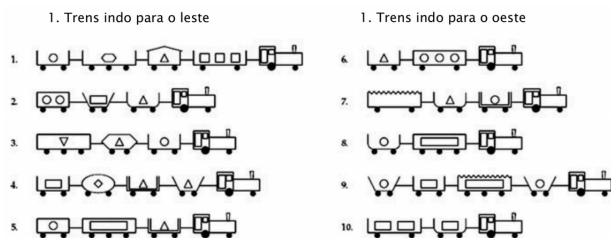


Figure 1: Exemplos dos trens de Michalski

Então, 10 variáveis booleanas (proposicionais) as quais descrevem se qualquer par de tipos de carga estão ou não em vagões adjacentes do trem. Além disso, também temos as seguintes relações a respeito dos vagões de um trem, cujos valores lógicos variam entre -1 (Falso) e 1 (Verdadeiro):

1. Existe um retângulo próximo de um retângulo (V ou F);
2. Existe um retângulo próximo de um triângulo (V ou F);
3. Existe um retângulo próximo de um hexágono (V ou F);
4. Existe um retângulo próximo de um círculo (V ou F);
5. Existe um triângulo próximo de um triângulo (V ou F);
6. Existe um triângulo próximo de um hexágono (V ou F);
7. Existe um triângulo próximo de um círculo (V ou F);
8. Existe um círculo próximo de um círculo (V ou F).

Existe um atributo de classe que define a direção do trem: leste ou oeste. Observe que, para atributos com vários valores, você deve atribuir valores na ordem em que aparecem. Por exemplo, o tipo de carga deve ser 1 para círculo, 2 para hexágono, 3 para retângulo e assim por diante. O neurônio correspondente deve usar uma função de ativação linear, ou seja, $h(x) = x$.

Esse problema tornou-se padrão no teste e demonstração de várias técnicas de aprendizagem de máquina, principalmente implementações de indução lógica.

4 Modelos utilizados

Os modelos utilizados para realizar os experimentos foram implementados com base no livro Neural-symbolic cognitive reasoning [7]. Estes modelos podem ser encontrados no link github.com/erlonbie/Neural-Symbolic-Learner, juntamente com as instruções para execução do experimento.

4.1 Modelo Questão 1

Para responder a primeira questão, foi utilizada uma rede neural seguindo as instruções do enunciado, a qual possui 32 neurônios de entrada e um de saída. O neurônio de saída é o responsável por classificar uma instância como *eastbound* ou *westbound*. A representação gráfica do modelo corresponde a Figura 2.

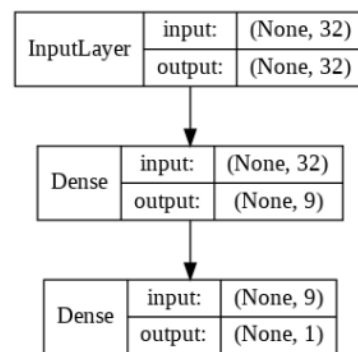


Figure 2: Flat Network

4.2 Modelo Questão 2

Para a segunda questão do trabalho, utilizou-se uma meta-rede com 11 regras propostas na atividade, que equivalem a 11 redes neurais. Uma meta-rede é composta por várias redes neurais, que representam proposições, com sua última camada realizando a classificação. As regras propostas foram:

- $\text{num_cars}(t, nc)$ onde o número de vagões é $nc \in [3..5]$ em um trem $t \in [1..10]$
- $\text{num_loads}(t, nl)$ onde a quantidade de carga é $nl \in [1..4]$ em um trem $t \in [1..10]$
- $\text{num_wheels}(t, c, w)$ onde o número de rodas é $w \in \{2, 3\}$ em um vagão $c \in [1..4]$ de um trem $t \in [1..10]$
- $\text{length}(t, c, l)$ onde o comprimento do vagão é $l \in \{-1, 1\}$ ⁶ em um vagão $c \in [1..4]$ de um trem $t \in [1..10]$
- $\text{shape}(t, c, s)$ onde o formato do vagão é $s \in [1..10]$ ⁷ em um vagão $c \in [1..4]$ de um trem $t \in [1..10]$
- $\text{num_car_loads}(t, c, ncl)$ onde o número de cargas do vagão é $ncl \in [0..3]$ em um vagão $c \in [1..4]$ de um trem $t \in [1..10]$

- `load_shape(t,c,ls)` onde o formato da carga é `ls` $\in [1..4]$ em um vagão `c` $\in [1..4]$ de um trem `t` $\in [1..10]$
- `next_crc(t,c,x)` o vagão `c` $\in [1..4]$ de um trem `t` $\in [1..10]$ está adjacente a um vagão com carga circular `x` $\in \{0, 1\}^8$
- `next_hex(t,c,x)` o vagão `c` $\in [1..4]$ de um trem `t` $\in [1..10]$ está adjacente a um vagão com carga hexagonal `x` $\in \{0, 1\}^8$
- `next_rec(t,c,x)` o vagão `c` $\in [1..4]$ de um trem `t` $\in [1..10]$ está adjacente a um vagão com carga retangular `x` $\in \{0, 1\}$
- `next_tri(t,c,x)` o vagão `c` $\in [1..4]$ de um trem `t` $\in [1..10]$ está adjacente a um vagão com carga triangular `x` $\in \{0, 1\}^8$

Para o treinamento da meta-rede foi utilizado *leaving-one-out* e *cross-validation* com relação a cada trem, aumentando a quantidade de instâncias através de uma *data augmentation*, de 10 para 40, fazendo com que cada trem possa ter até 4 vagões. Sendo assim, a meta-rede foi treinada com 36 exemplos e validada com 4.

5 Resultados e Análises

Conforme citado anteriormente sobre os experimentos, no qual o primeiro (i) uma rede plana/simples, e o segundo (ii) uma meta-rede, para cada um dos experimentos, foram avaliados *MSE* (Mean Squared Error) e Acurácia de cada modelo (rede), e por fim gerada uma tabela de resultados para cada uma das experimentos mencionados.

5.1 Questão 1

No primeiro experimento, utilizando *leave-one-out*, podemos observar que na 3 que o modelo não foi capaz de generalizar corretamente o conceito, não convergindo os valores de treino e validação.

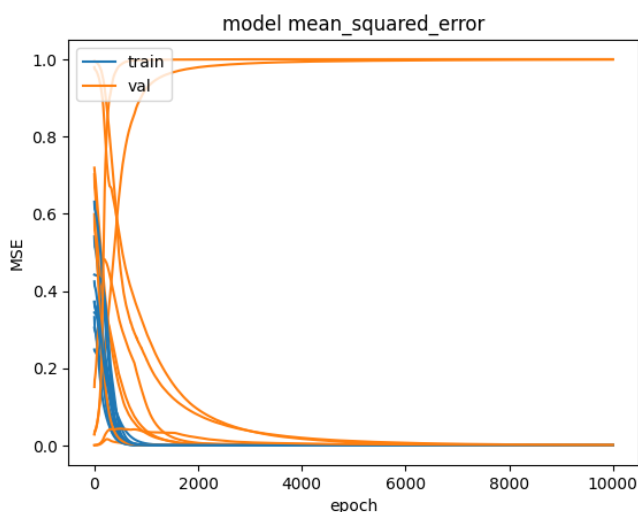


Figure 3: MSE Flat NetWork - Questão 1

O mesmo comportamento pode ser notado na 4, onde os extremos verticais do gráfico não convergem para um mesmo valor.

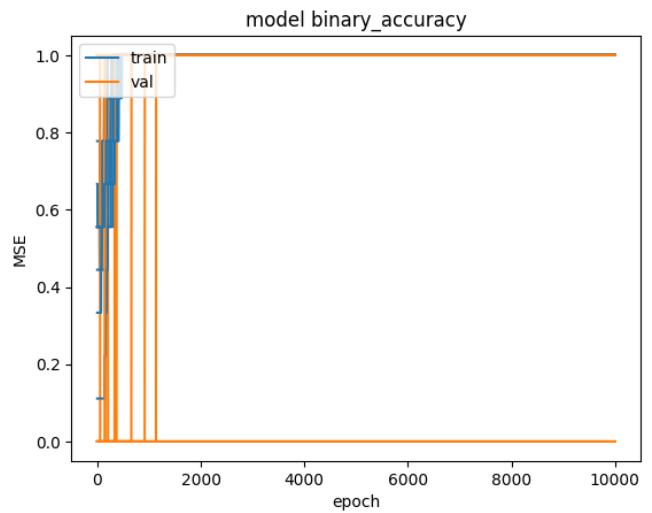


Figure 4: Acurácia Flat Network - Questão 1

Utilizando 10 exemplos para esse modelo, podemos notar na tabela 1 que a rede não foi capaz de classificar corretamente todos os exemplos, resultando em uma acurácia de 80%. Não é o melhor dos resultados, mas pode-se dizer que é satisfatório pela simplicidade da rede.

Table 1: Resultados obtidos - Questão 1

Trem	Saída da rede	Saída desejada	Classe
0	0.00	1	leste
1	0.98	1	leste
2	1.00	1	leste
3	0.99	1	leste
4	0.99	1	leste
5	0.00	0	oeste
6	0.03	0	oeste
7	0.00	0	oeste
8	1.00	0	oeste
9	0.03	0	oeste

5.2 Questão 2

O segundo experimento é mais eficaz, porém, mais complexo de implementar, menos escalável em relação a execução. Na tabela 2, o resultado a meta-rede aponta que todas as saídas tendem ao resultado desejado, sendo iguais em dois casos, mas nenhum dos resultados da rede preveem a direção errada para o trem. Desse modo, podemos dizer que o experimento da questão 2 foi mais bem sucedido que o da questão 1, que apontou a direção errada para 1 dos 10 trens.

As figuras mostram o resultado das estatísticas para 10 grupos do *leave-one-out*. Este experimento obtém convergência em todos os 10 grupos. Ao observar a figura 6, temos que a acurácia converge a totalidade em todos os casos a partir da epoch 400 e, da mesma forma, o erro quadrado médio (MSE) fica linear na mesma sobreposição, conforme a 5.

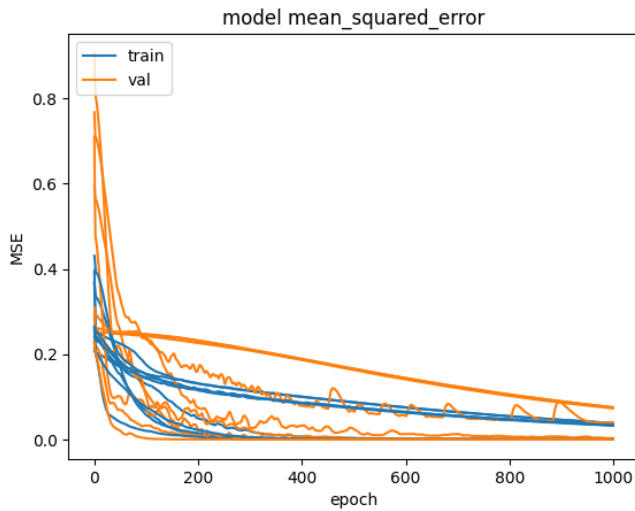


Figure 5: MSE Meta-NetWork - Questão 2

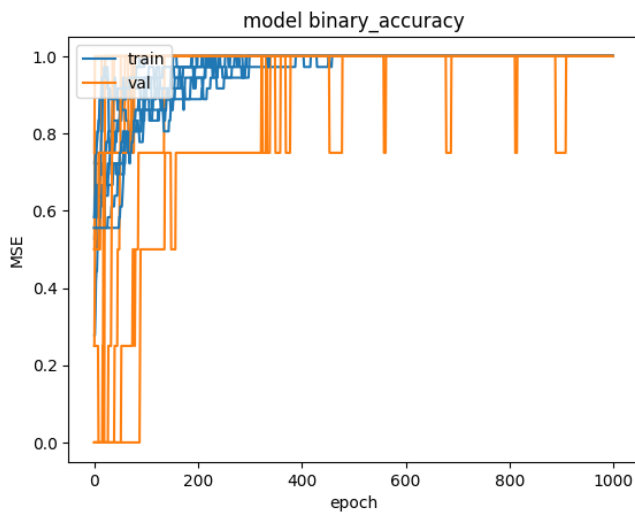


Figure 6: Acurácia Meta-NetWork - Questão 2

Table 2: Resultados obtidos - Questão 2

Trem	Saída da rede	Saída desejada	Classe
0	1.00	1.00	leste
1	0.73	1.00	leste
2	0.97	1.00	leste
3	1.00	1.00	leste
4	0.62	1.00	leste
5	0.02	0.00	oeste
6	0.00	0.00	oeste
7	0.27	0.00	oeste
8	0.27	0.00	oeste
9	0.00	0.00	oeste

6 Conclusão

Com a construção de dois tipos de redes, a partir de algoritmos diferentes, foi possível resolver o problema dos trens de Michalski. O experimento 2 demonstrou ser mais eficiente, mesmo precisando de menos épocas, que o outro experimento, utilizando um método de proposicionalização semelhante ao do algoritmo do

CILP [8], com algumas regras definidas, conseguindo uma acurácia de 100%.

Por fim, foi de grande importância para o aprendizado em na área de Inteligência Artificial, mais especificamente redes neurais que não seguem o modelo conexionista, que é um "padrão" de mercado, o que não significa ser a melhor opção pra todas as situações.

References

- [1] Zachary Susskind, Bryce Arden, Lizy K. John, Patrick Stockton, and Eugene B. John. Neuro-symbolic AI: an emerging class of AI workloads and their characterization. *CoRR*, abs/2109.06133, 2021.
- [2] K. R. Chowdhary. *Fundamentals of Artificial Intelligence*. Springer, 2020.
- [3] Luc De Raedt, Robin Manhaeve, Sebastijan Dumanic, Thomas Demeester, and Angelika Kimmig. Neuro-symbolic= neural+ logical+ probabilistic. In *NeSy'19@ IJCAI, the 14th International Workshop on Neural-Symbolic Learning and Reasoning*, 2019.
- [4] Rohan Chitnis, Tom Silver, Joshua B Tenenbaum, Tomas Lozano-Perez, and Leslie Pack Kaelbling. Learning neuro-symbolic relational transition models for bilevel planning. *arXiv preprint arXiv:2105.14074*, 2021.
- [5] Manoel Vitor Macedo França. Aprendizado relacional através do uso de cláusulas mais específicas no sistema c-il2p. Master's thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2012.
- [6] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat Abdelatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.
- [7] Artur S. D'Avila Garcez, Luís C. Lamb, and Dov Gabbay. *Neural-Symbolic Cognitive Reasoning*. Springer-Verlag Berlin Heidelberg, 2009.
- [8] Manoel Franca, Gerson Zaverucha, and Artur Garcez. Neural relational learning through semi-propositionalization of bottom clauses, 2015.