



Poder Executivo
Ministério da Educação
Universidade Federal do Amazonas
Instituto de Computação - IComp



Disciplina:	Inteligência Artificial	3a Avaliação e Trabalho final	Data: 06/08/24
Professors:	Edjard Mota	Turma: IEC01 e CB500	

Example do trem de Michalski. A meta (goal) é classificar quais os trens que vão para leste e os que vão para oeste (veja figura no material do trabalho do classroom).

Para cada trem temos os seguintes atributos:

1. quantidade de vagões (*car* em inglês) (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, e pode ser
 1. retângulo-fechado,
 2. retângulo-aberto
 3. duplo retângulo-aberto
 4. elipse
 5. locomotiva
 6. hexágono
 7. topo dentado
 8. trapézio aberto
 9. topo triangular-fechado
 - d) quantidade de cargas no vagão (0 a 3)
 - e) o formato da carga (círculo, hexágono, retângulo ou triângulo)

Então, 10 variáveis booleanas (proposicionais) 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).

Informações que podem ser úteis no seu modelo: 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 (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)

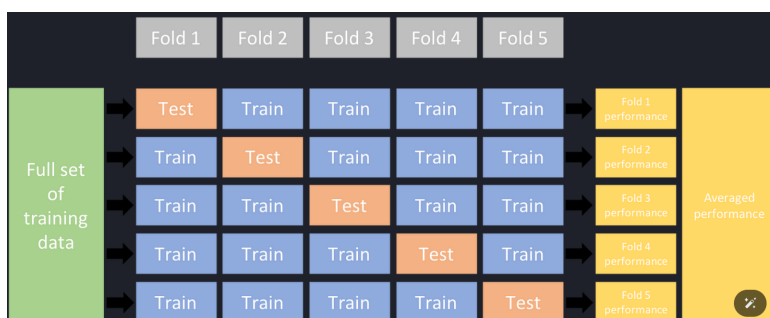
Há um único atributo de classe que define a direção de um trem: *leste* ou *oeste*.

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. $h(x) = x$.

Para as questão 1 que se pede utilize como código base inicial o arquivo fornecido para este trabalho (Neural-Symbolic-Learner-main.zip). Nele contem uma resolucao de turma anterior para este memos problema utilizando uma rede flat e outra neuro-simbolica replicando o teste do livro (Figure 6: Acurácia Meta-Network - Questão 2 daquele trabaho)

Questão 1. Adapte e execute o código apenas com o modelo neuro-simbólico e treine sua adaptação para os 100 trens assumindo que o da coluna esquerda vão para o leste e os da esquerda para o oeste., considerando inicialmente os 11 predicados sugeridos..

1. Analise se são suficientes para o caso dos 100 e se não for, adicione a descrição lógica dos novos predicados.
2. Para treinamento:
 - a. Caso 1 Separe 30% dos casos, treine o modelo com 70%, e teste com os 30% restante.
 - b. Faça um treinamento e teste com validação cruzada conforme figura abaixo, e compare com o resultado de 2.(a).



1. $num_cars(t, nc)$, em que $t \in [1..10]$ e $nc \in [3..5]$.
2. $num_loads(t, nl)$ em que $t \in [1..10]$ e $nl \in [1..4]$.
3. $num_wheels(t, c, w)$ em que $t \in [1..10]$ e $c \in [1..4]$ e $w \in [2..3]$.
4. $length(t, c, l)$ em que $t \in [1..10]$ e $c \in [1..4]$ e $l \in [-1..1]$ (-1 denota curto e 1 longo)
5. $shape(t, c, s)$ em que $t \in [1..10]$ e $c \in [1..4]$ e $s \in [1..10]$ (um número para cada forma).
6. $num_cars_loads(t, c, ncl)$ em que $t \in [1..10]$ e $c \in [1..4]$ e $ncl \in [0..3]$.
7. $load_shape(t, c, ls)$ em que $t \in [1..10]$ e $c \in [1..4]$ e $ls \in [1..4]$.

8. $next_crc(t, c, x)$ em que $t \in [1..10]$ e $c \in [1..4]$ e $x \in [-1..1]$, em que o vagão c do trem t tem um vagão adjacente com cargas em círculo.
9. $next_hex(t, c, x)$ em que $t \in [1..10]$ e $c \in [1..4]$ e $x \in [-1..1]$, em que o vagão c do trem t tem um vagão adjacente com cargas em hexágono.
10. $next_rec(t, c, x)$ em que $t \in [1..10]$ e $c \in [1..4]$ e $x \in [-1..1]$, em que o vagão c do trem t tem um vagão adjacente com cargas em retângulo.
11. $next_tri(t, c, x)$ em que $t \in [1..10]$ e $c \in [1..4]$ e $x \in [-1..1]$, em que o vagão c do trem t tem um vagão adjacente com cargas em triângulo

Questão 2. Implemente uma solução em LTNTorch para o problema dos trens considerando os 11 predicados.

1. Teste com os 10 trens do livro e compare o resultado com o o gerado pelo código fornecido, Table 2: Resultados obtidos - Questão 2
2. Execute o código LTNTorch para os 100 trens usando os mesmos procedimentos de treinamento da questão anterior.

Questão 3. Compare seus resultados do modelo da questão 1 com o desta questão (2 no caso), e

- a. explique o porque das diferenças baseado nas características de arquitetura dos dois modelos e nos resultados numéricos. Cite o referencial teórico, tanto do livro (capítulo 10) quanto dos artigos sobre LTN.
- b. Defina como seria possível extrair, dos dois modelos, a seguinte regra genérica que classifica trens indo para o leste (east), apenas descreva como seria a extração baseado no algoritmo de extração visto em sala (e no material de classe). :

1. $car(T,C) \wedge short(C) \wedge closed_top(C) \rightarrow east(T)$ (no caso dos 10 trens)

2. Qual regra para os 100 trens?

- c. Verifique se sua solução classifica corretamente de acordo com as seguintes teorias

A. Se um trem tem um vagão curto e fechado, então ele vai para o leste, caso contrário, vai para o oeste (note que isto é uma descrição textual da regra lógica da questão 2.b)

B. Se um trem tem dois vagões, ou tem um vagão com teto irregular, então ele vai para o oeste, caso contrário, vai para o leste

C. Se um trem tiver mais de dois tipos diferentes de carga, então ele vai para o leste, caso contrário, vai para o oeste

O que voce deve entregar?

Sua solução disponível no github, código e os dados: **NAO MANDE ARQUIVO COMPACTADO!!!** Deixe tudo no github, e se usar o Colab indique isso no github.

O que voce deve entregar?

1. O relatório em formato tex do overleaf e um arquivo txt com **faça um HOW TO de como executar sua solução para as 3 questões**. Não coloque arquivos que estejam em post local!!!! Deixe tudo em pastas do git e que possam ser lidos do próprio colar ou quando clonados localmente.
2. Relatório no template em tex similar ao trabalho exemplo fornecido, que contenha as seções:
 1. Um Referencial Teórico que brevemente descreve sistemas neuro-simbólicos de aprendizado (use o livro do Garcez, eos artigos sobre LTN como referencia básica, mas pegue outras referencias) ,
 2. Descrição do problema do trem **(escrevam a descrição), inclusive com figuras para cada meta-rede,**