

[4 valores = 2+2]

1. Conhecimento contextual.

1.1

Kai-Fu Lee, ex-presidente Google China, hoje CEO de SinoVation Ventures, completou o seu PhD em Carnegie Mellon, em 1988. Quando começou o curso, o seu orientador sugeriu-lhe que trabalhasse em sistemas periciais por inteligência artificial "clássica", mas Lee argumentou que essa abordagem era "não escalável" e acabou por tornar-se um pioneiro em Machine-Learning (ML).

→ O que poderá significar esta perspetiva de um sistema pericial clássico não ser "escalável"?, e em que medida ML resolve esse problema?

Num sistema pericial (Expert System) clássico, perito(s) fornecem explicitamente afirmações e regras entre elas, que codificam o conhecimento que têm sobre algum assunto. Tipicamente, o conhecimento fica codificado numa notação que permite a algoritmos de inferência lógica, aferir a validade de outras proposições.

Esta abordagem pode ser entendida como "não escalável", pelos fatores que limitam o crescimento do conhecimento codificado e/ou extraível pelos algoritmos lógicos que trabalham a partir da base, como a disponibilidade de peritos e limitações dos próprios sistemas lógicos binários (True/False).

Por seu lado, Machine-Learning (ML) é uma abordagem estatística em que são calculadas probabilidades (valores de 0 a 1, com infinitas possibilidades entre os extremos) para as situações em aplicação, ou valores reais em geral. Basta haver dados que possam constituir "exemplo" para um sistema de ML poder ser alimentado.

O sistema pericial só se alimenta de "verdades" que um perito anote. O próprio valor lógico anotado terá exigido muitas observações. Em ML, qualquer ocorrência pode constituir anotação e a descoberta de novo conhecimento não depende necessariamente da sua observação explícita por um perito; nesta perspetiva, consegue-se melhor escalabilidade.

1.2

Alguns autores escrevem que a Inteligência Artificial vive um "verão", mas que não se deve esquecer o "inverno" passado.

→ O que é isto do "inverno da IA"? Caracterize esse período por pelo menos dois atributos.

"Invernos" de IA são períodos em que o interesse por, e a disponibilidade de financiamento para projetos de IA, diminuem significativamente, por considerar-se que as soluções da área não satisfazem as expectativas. Note-se que frustrar expectativas é tanto mais fácil, quando mais elevadas essas.

O final da década de 1970 corresponde a um destes períodos. Nos EUA, a DARPA limitou severamente o financiamento de projetos de IA, depois de problemas no desempenho de sistemas ambiciosos, como a leitura por voz, em tempo real, dos

valores em painéis de instrumentos de aviões militares; ou a tradução entre algumas línguas naturais.

Estes períodos são assim sempre caracterizados por uma percepção de "under-deliverance", ou funcionalidade abaixo das expectativas, e a consequente retração de interesse na área e da disponibilidade em financiar-se atividades relacionadas.

[4 valores = $8 \cdot 0.5$]

2. As questões deste grupo são no contexto de lógica proposicional, com notação adaptada.

→ 2.1 O que entende por "satisfação" de uma expressão booleana, por um modelo?

Uma expressão booleana diz-se "satisfazer" um modelo, quando as atribuições que constam do modelo a tornam verdadeira.

→ 2.2 O que entende por "consequência lógica" entre expressões?

Uma expressão e_2 diz-se "consequência lógica" de uma expressão e_1 , quando todos os modelos que satisfazem e_1 também satisfazem e_2 .

Responda V (Verdade) ou F (Falso), consoante, respetivamente, o modelo M seguinte satisfaça a expressão booleana que aparece à direita do operador de satisfação \models .

$M = \{a/\text{True}, b/\text{True}, c/\text{False}\}$

→ 2.3.1

$M \models (a \text{ and } b \text{ and } c)$ **F**

→ 2.3.2

$M \models a \text{ and } b \text{ or } c$ **V**

→ 2.3.3

$M \models a \Rightarrow c \text{ and } b$ **F**

→ 2.3.4

$M \models a \Rightarrow c \text{ or } b$ **V**

2.4

Sejam a , b , c quaisquer proposições booleanas.

Indique V (Verdade) ou F (Falso), consoante verificar-se implicação lógica.

Nos casos em que a resposta for F, indique TODOS os modelos possíveis que invalidam a implicação lógica.

→ 2.4.1

$a, b \text{ or } c \models c$

F

0 modelo $\{a/T, b/T, c/F\}$ invalida a implicação lógica.

→ 2.4.2

$a, b, c \models c$

V

[2 valores = $1 + 2 \cdot 0.5$]

3. Recorde os algoritmos de procura que estudou.

→ 3.1 Qual a relevância da disciplina de acesso (LIFO ou FIFO) em algoritmos de procura não informada, para a estrutura de dados "fronteira"/"frontier"?

Com LIFO acontece DFS.

Com FIFO acontece BFS.

→ 3.2 Verdadeiro (V) ou Falso (F)?

0 algoritmo BFS (Breadth-First Search) vai, sempre e garantidamente, encontrar o caminho/solução óptima. **V**

Nota: a utilização da palavra "caminho" pretende indicar um "problema de labirinto", de custo unitário. Nesse contexto, a função de custo é nunca decrescente, e o BFS encontra sempre, garantidamente, o caminho ótimo.

→ 3.3 Verdadeiro (V) ou Falso (F)?

0 algoritmo BFS (Breadth-First Search) pode ser o mais lento a encontrar a solução. **V**

[2.5 valores]

4. Em problemas de classificação multi-classe, é muito comum encontrar-se código Python como o seguinte:

```
from sklearn import model_selection
X_train, X_test, y_train, y_test = \
model_selection.train_test_split (amostras, classesPorAmostra)
```

→ O que representam os valores X_train, X_test, y_train, y_test? Porque é que é importante separá-los?

X_train são as amostras a usar para treino ou "fit" do modelo;

X_test são as amostras reservadas para aferir o modelo, logo NÃO serão usadas para treino;

y_train são as classificações conhecidas para as amostras em X_train;

y_test são as classificações conhecidas para as amostras em X_test.

A separação entre amostras de treino e teste é fundamental para aferir a precisão do modelo, depois de treinado.

[2.5 valores]

5. Interpretação de soluções.

Um modelo programático/rede neuronal baseado na framework Keras, para a classificação de imagens, foi criado da seguinte forma:

```
from keras.models import Sequential
from keras.layers import Dense

iUnits = 32
model = Sequential([
    Dense(iUnits, activation="relu", input_shape=(28*28,)),
    Dense(iUnits, activation="relu"),
    Dense(8, activation="softmax")])
```

→ Qual o número de parâmetros da primeira camada da rede?

Basta apresentar a expressão matemática, não é preciso calcular o valor.

$(N^{\circ} \text{ inputs por nó}) * n^{\circ} \text{ de nós} + (1 \text{ param de bias} * n^{\circ} \text{ de nós})$
 $(28*28) * 32 + (1*32)$

[5 valores]

6. Autoria de soluções.

→ Escreva, integralmente, em Python, uma função de nome "KNN1", capaz de receber:

- pD = uma lista de amostras (o dataset de treino), cada amostra uma lista de inteiros;
- pC = uma lista de classificações paralela a pD;
- pF = uma função, a utilizar para medir a distância entre quaisquer duas amostras;

A sua solução deverá corresponder ao algoritmo KNN na sua máxima complexidade (K=1), retornando a classificação adequada para pF.

TODO