GABARITO da Prova da 2ª Etapa - 4ª Fase da ONIA

Questão 1

Um pesquisador dispõe de 10 000 registros de pacientes sem rótulos sobre a presença de uma doença. Ele quer descobrir se existem subgrupos naturalmente distintos nesses dados. Qual abordagem de IA é a mais apropriada?

A. Aprendizado supervisionado
B. Aprendizado por reforço
C. Aprendizado não supervisionado
D. Aprendizado semi-supervisionado
E. Aprendizado por transferência

Justificativa da escolha: _____

Questão 2

Em um experimento de redes neurais, a precisão no conjunto de treinamento sobe continuamente, mas no conjunto de validação cai após a 5.ª época. Qual medida prática costuma se adotada para mitigar esse problema?

A. Aumentar a taxa de aprendizado
B. Reduzir o número de camadas ocultas
C. Adicionar mais neurônios em cada camada
D. Usar dropout ou regularização L2
E. Trocar a função de ativação para ReLU

Justificativa da escolha: _____

Questão 3

Considere a matriz de confusão para um classificador binário:

Real Positivo (Pred. Pos.): 40 | Real Positivo (Pred. Neg.): 10
Real Negativo (Pred. Pos.): 20 | Real Negativo (Pred. Neg.): 30

Qual é o valor da precisão (precision)?

A. 0,40
B. 0,57
C. 0,67
D. 0,75
E. 0,80

Justificativa da escolha: _____

Questão 4

No algoritmo de gradient descent, qual hiperparâmetro define o tamanho de cada passo dado na direção do gradiente?

A. Momentum
B. Taxa de aprendizado (learning rate)
C. Batch size
D. Número de épocas
E. Função de perda

Justificativa da escolha: _____

Questão 5

O que normalmente acontece ao aumentar o número de filtros (kernels) em uma camada convolucional, mantendo as demais configurações?

A. Diminui a dimensionalidade da saída
B. Reduz o risco de sobreajuste
C. Aumenta a capacidade de extrair diferentes padrões de características
D. Torna a rede invariável a rotações
E. Elimina a necessidade de camadas de pooling

Justificativa da escolha: _____

Questão 6

Em um classificador k-NN, qual efeito esperado ao escolher um valor de k muito grande?

A. Aumento da variância e diminuição do viés
B. Diminuição da variância e aumento do viés
C. Redução do custo computacional
D. Maior sensibilidade a ruídos nos dados
E. Classe minoritária tende a ser privilegiada

Justificativa da escolha: _____

Questão 7

Um sistema de recomendação de vagas de emprego, treinado com dados históricos, passa a exibir uma quantidade menor de oportunidades para mulheres, mesmo quando seus perfis são equivalentes aos dos homens. Qual ação está mais alinhada com as práticas éticas em Inteligência Artificial?

A. Ignorar o problema, pois reflete o histórico real
B. Remover o gênero do conjunto de atributos, sem mais ajustes
C. Monitorar o modelo com métricas de equidade e retreinar com técnicas de mitigação de viés
D. Diminuir a taxa de aprendizado para evitar mudanças bruscas
E. Substituir o algoritmo por uma regra fixa baseada em sorteio aleatório

Justificativa da escolha: _____

Questão 8

No método Q-Learning com política ε-greedy, qual das afirmações define corretamente o papel do ε?

A. É o fator de desconto que pondera recompensas futuras

B. Controla a taxa de atualização dos valores Q

C. Define a probabilidade de escolher uma ação aleatória em vez da melhor ação conhecida

D. Indica a taxa de decaimento da recompensa ao longo do tempo

E. Especifica o número máximo de episódios de treinamento

Justificativa da escolha: _____

Questão 9

As redes neurais convolucionais são bem aplicadas a dados representados em duas ou mais dimensões como as imagens. Uma rede neural convolucional é composta principalmente por 3 tipos de camadas: camada de convolução, camada de pooling e camada fully-connected. A camada de convolução tem como parâmetros N filtros de tamanho m x n que vão ser convoluídos pelo dado de entrada. Por essas características que este modelo é adequado para dados como imagens, as quais são arrays multidimensionais cujos valores representam pixeis.

A matriz abaixo ilustra o input de uma imagem preta e branca de tamanho 12x12. Considere o filtro 3x3 na imagem para simular a operação de convolução deste input, considerando a arquitetura abaixo:

1	1	0	1	1	1	0	0
0	1	1	0	1	1	1	0
1	0	1	0	0	1	1	1
0	0	0	0	0	1	1	0
0	0	1	0	1	1	0	0
1	0	1	1	0	0	0	1
1	1	0	1	0	0	1	0
1	1	1	0	1	0	0	0

```
sobel_horizontal = np.array([
    [-1, -2, -1],
    [ 0,  0,  0],
    [ 1,  2,  1]
], dtype=np.float32)

sobel_kernel = sobel_horizontal.reshape((3, 3, 1, 1))

inputShape = (8, 8, 1)
      model = Sequential()
model.add(Conv2D( filters=1,kernel_size=(3, 3),padding='same',
input_shape=(8, 8, 1),
kernel_initializer=Constant(sobel_kernel),
                            use_bias=False,
                            trainable=False))
      model.add(Activation("relu"))
model.add(Conv2D( filters=1,kernel_size=(3, 3),padding='same',
kernel_initializer=Constant(sobel_kernel),
                            use_bias=False,
                            trainable=False))
      model.add(Activation("relu"))
      model.add(MaxPooling2D((2,2)))
```

Qual das alternativas representa a matriz de saída da camada de pooling.

A)
[[0. 0. 3. 9.]
 [9. 11. 5. 2.]
 [8. 1. 3. 3.]
 [0. 0. 0. 0.]]

B)[[3. 11. 1. 0.]
 [5. 15. 8. 0.]
 [5. 4. 2. 0.]
 [0. 0. 3. 0.]]

C)[[1. 5. 9. 9.]
 [11. 11. 5. 3.]
 [8. 1. 3. 3.]
 [0. 0. 0. 0.]]

D)[[3. 3. 4. 3.]
 [0. 1. 2. 0.]
 [3. 3. 0. 1.]
 [2. 0. 1. 0.]]

E)[[8. 2. 5. 0.]
 [7. 9. 6. 0.]
 [11. 10. 3. 1.]
 [14. 2. 1. 5.]]

Questão 10

Em uma pequena fábrica de montagem de robôs, dois sensores binários (Sensor A e Sensor B) são usados para verificar a presença de peças em duas posições diferentes de uma esteira. Cada sensor pode assumir apenas dois valores: 0 (sem peça) ou 1 (com peça).

Um sistema de controle precisa acionar um braço robótico sempre que pelo menos uma das posições contiver uma peça. Ou seja, o braço só ficará inativo quando ambos os sensores detectarem a ausência de peças. Para isso, os engenheiros estão treinando um modelo de classificação linear simples que compreenda essa lógica. A partir de x e y, calcule como seria o processo de atualização dos pesos e bias para que o modelo faça a predição.

Entrada:
x = [[0, 0], [0, 1], [1, 0], [1, 1]]
y = [0, 1, 1, 1]

Pesos: [0.868, -0.012]

Bias: -0.015

Taxa de Aprendizado: 0.01

Predições feitas pelo modelo:
a) Predição: [0.024, 0.022, -0.800, 0.822]
b) Predição: [0.056, -0.234, 0.899, 0.712]
c) Predição: [-0.015, -0.027, 0.863, 0.861]
d) Predição: [-0.019, -0.156, -0.010, 0.600]
e) Predição: [-0.089, -0.899, 0.7830, 0.345]

GABARITO COMENTADO DA PROVA

Questão 1: C

Explicação: Como não há rótulos, técnicas de agrupamento típicas do aprendizado não supervisionado são as mais indicadas.

Questão 2: D

Explicação: O sintoma é overfitting; dropout ou L2 reduzem a complexidade efetiva do modelo, ajudando a generalizar.

Questão 3: C

Explicação: Precisão = VP / (VP + FP) = 40 / 60 ≈ 0,67.

Questão 4: B

Explicação: A taxa de aprendizado escala o gradiente, definindo o tamanho do passo de atualização.

Questão 5: C

Explicação: Mais filtros produzem mais mapas de ativação, capturando diferentes padrões de características.

Questão 6: B

Explicação: k grande suaviza a fronteira: maior viés, menor variância.

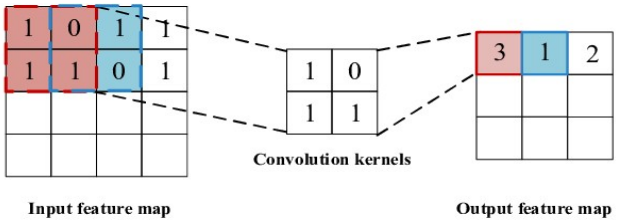
Questão 7: C

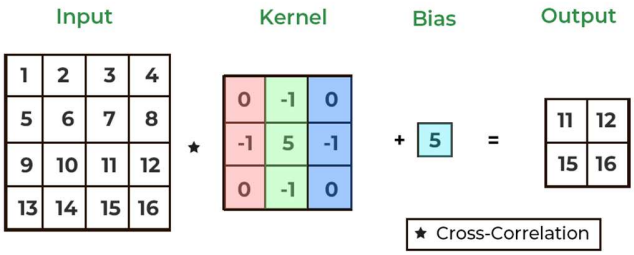
Explicação: É recomendado medir e mitigar viés, re-treinando o modelo com técnicas adequadas.

Questão 8: C

Explicação: ε controla a probabilidade de exploração, equilibrando exploração e exploração.

Questão 9: A





input original 8x8 filtro(3,3) com pading 1

0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	1	0	0	0
0	0	1	1	0	1	1	1	0	0
0	1	0	1	0	0	1	1	1	0
0	0	0	0	0	0	1	1	0	0
0	0	0	1	0	1	1	0	0	0
0	1	0	1	1	0	0	0	1	0
0	1	1	0	1	0	0	1	0	0
0	1	1	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0

-1	-2	-1
0	0	0
1	2	1

1	3	3	2	3	4	3	1
-1	-1	0	-2	-3	0	3	3
-1	-3	-3	-2	-2	-1	0	0
-2	-1	0	1	2	0	-3	-3
2	2	3	3	0	-3	-2	1
3	2	0	0	-2	-2	1	1
1	2	0	-1	1	1	-1	-2
-3	-3	-2	-2	-1	-1	-2	-1

Aplicando relu

0	0	0	0	0	0	0	0	0	0
0	1	3	3	2	3	4	3	1	0
0	0	0	0	0	0	0	3	3	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	2	0	0	0	0
0	2	2	3	3	0	0	0	1	0
0	3	2	0	0	0	0	1	1	0
0	1	2	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

multiplica pelo filtro de novo

0	0	0	0	0	3	9	9
-5	-10	-11	-10	-12	-14	-11	-5
0	0	1	4	5	-1	-9	-9
6	9	11	9	3	0	1	2
8	7	1	-4	-5	-1	3	3
-2	-4	-9	-8	0	3	0	-2
-8	-7	-2	0	0	-1	-3	-3
-4	-5	-2	-1	-3	-3	-1	0

relu de novo

0	0	0	0	0	3	9	9
0	0	0	0	0	0	0	0
0	0	1	4	5	0	0	0
6	9	11	9	3	0	1	2
8	7	1	0	0	0	3	3
0	0	0	0	0	3	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

por fim, aplica-se max_poling(2,2), que resulta em

[[0. 0. 3. 9.]

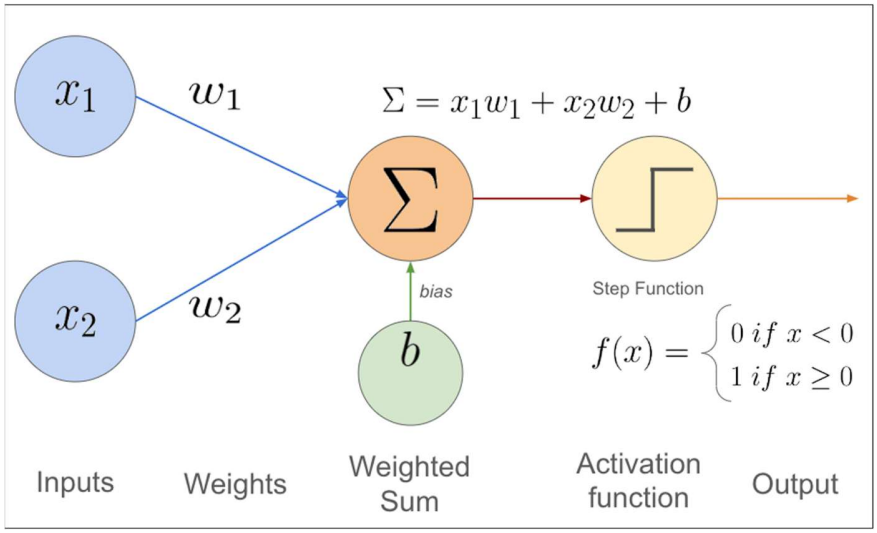
[9. 11. 5. 2.]

[8. 1. 3. 3.]

[0. 0. 0. 0.]]

Questão 10: C

Explicação: Perceptron Básico:



1. Cálculo da saída predita (y_pred):

A equação abaixo calcula a **soma ponderada** das entradas (x) com seus respectivos **pesos (w)**, seguida de um **valor de viés (b)**:

$$y_{pred} = \sum_{i=1}^{n} (x_i \cdot w_i) + b$$

Ou seja, para um vetor de entradas x=[x1, x2, …, xn], e os pesos w=[w1, w2, …, wn], a previsão **Ypred** é dada pela soma ponderada de cada entrada **xi** multiplicada pelo peso **wi**, somada ao **viés b**.

2. Aplicação da função de ativação:

Após calcular **Ypred**, a saída final é definida por uma função de ativação que a transforma em um valor binário (0 ou 1). Essa função pode ser representada como:

$$y_{pred} = \begin{cases} 1 & \text{se } y_{pred} > 0 \\ 0 & \text{se } y_{pred} \leq 0 \end{cases}$$

Ou seja, se a soma ponderada for positiva, o modelo prediz **1** (indicando que o braço robótico deve ser acionado, no contexto da história). Caso contrário, prediz **0** (indicando que o braço não deve ser acionado).

Cálculo:
x = [[0, 0], [0, 1], [1, 0], [1, 1]])
y = [0, 1, 1, 1]

Pesos: [0.868, -0.012]

Bias: -0.015

Taxa de Aprendizado: 0.01

1° STEP
1° Entrada:
x = [0, 0]

y = [0]

Peso = [0.868, -0.012]

Bias = -0.015

Taxa de Aprendizado = 0.01

Predição:

x = (0 * 0.868 + 0 * -0.012) + (-0.015)

x = -0,015

Regras:
x > 0 retorna o valor 1 caso o contrário 0

y_pred = 0

Calculo do Erro:

erro = y - y_pred

erro = 0

Atualização dos pesos:

Peso_n = (Peso_n) + (Taxa de Aprendizado * erro * x_n)

Peso 1 = (0.868) + (0.01*0*0)
Peso 1 = 0.868

Peso 2 = (-0.012) + (0.01*0*0)
Peso 2 = -0.012

Calculo do Bias

Bias = (Bias) + (Taxa de Aprendizado*erro)

Bias = -0.015

2° Entrada

x = [0, 1]

y = [1]

Peso = [0.868, -0.012]

Bias = -0.015

Taxa de Aprendizado = 0.01

Predição:

x = (0 * 0.868 + 1 * -0,012) + (-0.015)

x = -0.027

Regras:
x > 0 retorna o valor 1 caso o contrário 0

y_pred = 0

Calculo do Erro:

erro = y - y_pred

erro = 1

Atualização dos pesos:

Peso_n = (Peso_n) + (Taxa de Aprendizado * erro * x_n)

Peso 1 = (0.868) + (0.01*1*0)
Peso 1 = 0.868

Peso 2 = (-0.012) + (0.01*1*1)
Peso 2 = -0.002

Calculo do Bias

Bias = (Bias) + (Taxa de Aprendizado*erro)

Bias = -0.015 + 0.01*1

Bias = -0.005

3° Entrada

[1, 0]

y = [1]

Peso = [0.868, -0.002]

Bias = -0.005

Taxa de Aprendizado = 0.01

Predição:

x = (1 * 0.868 + 0 * -0.002) + (-0.005)

x = 0.863

Regras:
x > 0 retorna o valor 1 caso o contrário 0

y_pred = 1

Calculo do Erro:

erro = y - y_pred

erro = 0

Atualização dos pesos:

Peso_n = (Peso_n) + (Taxa de Aprendizado * erro * x_n)

Peso 1 = (0.868) + (0.01*0*1)
Peso 1 = 0.868

Peso 2 = (-0.002) + (0.01*0*0)
Peso 2 = -0.002

Calculo do Bias

Bias = (Bias) + (Taxa de Aprendizado*erro)
Bias = -0.005

4° Entrada

[1, 1]

y = [1]

Peso = [0.868, -0.002]

Bias = -0.005

Taxa de Aprendizado = 0.01

Predição:

x = (1 * 0.868 + 1 * -0.002) + (-0.005)

x = 0.861

Regras:
x > 0 retorna o valor 1 caso o contrário 0

y_pred = 1

Calculo do Erro:

erro = y - y_pred

erro = 0

Atualização dos pesos:

Peso_n = (Peso_n) + (Taxa de Aprendizado * erro * x_n)

Peso 1 = (0.868) + (0.01*0*1)
Peso 1 = 0.868

Peso 2 = (-0.002) + (0.01*0*1)
Peso 2 = -0.002

Calculo do Bias

Bias = (Bias) + (Taxa de Aprendizado*erro)

Bias = -0.005