MAC0425 Inteligência Artificial

Terceira Avaliação (P3)

NOME: GABARITO

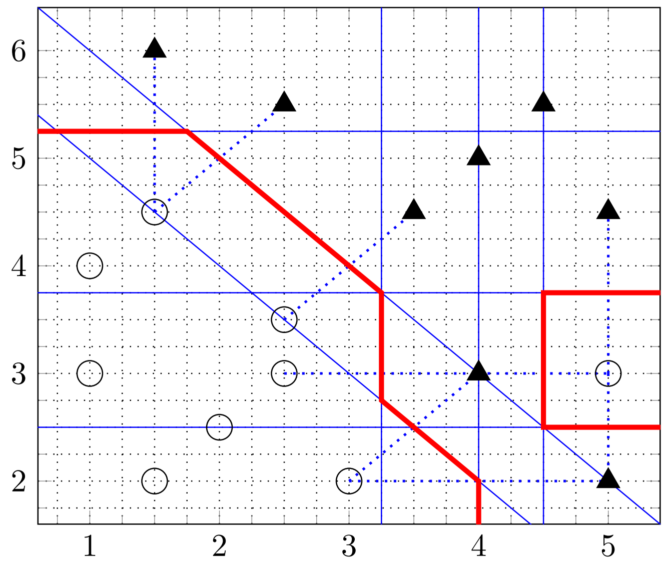
NUSP: **04252019**

ASSINATURA: ....

INSTRUÇÕES:

- Leia atentamente os enunciados.
- Este exame é composto por questões dissertativas; todas as respostas devem ser justificadas.
- Escreva suas resposta nos locais indicados, utilizando caneta esferográfica azul ou preta.
- Use os versos das folhas como rascunho; escreva sua reespota apenas quando tiver certeza.
- Você pode consultar apenas uma folha A4 (frente e verso) individual.
- O uso de equipamentos eletrônicos (calculadoras, celulares, computadores) não é permitido.
- **Duração:** 120 minutos.

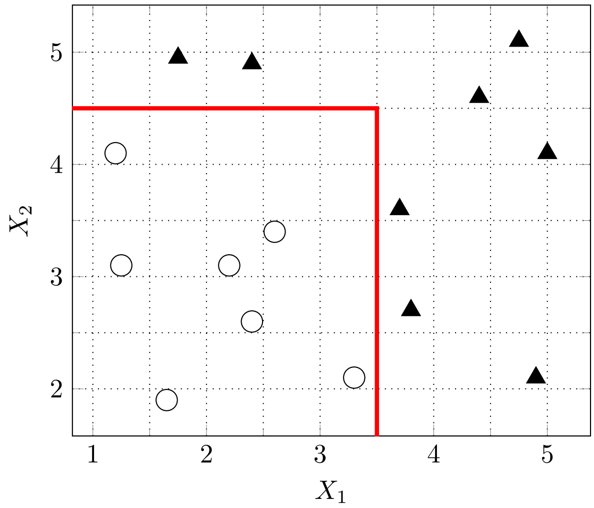
Questão 1. Projete um classificador por vizinho mais próximo (*1-nearest neighbor*) que classifica corretamente o conjunto de dados abaixo. Desenhe a superfície de decisão do classificador na figura e descreva os parâmetros necessários. Assuma distância euclideana e justifique a superfície de decisão.



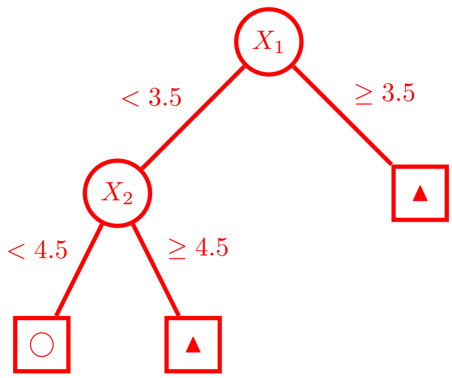
O classificador por vizinho mais próximo consiste em classificar cada ponto de acordo com o rótulo do ponto mais próximo em conjunto de treinamento. Dessa forma, o classificador sempre classifica corretamente os próprios pontos do conjunto de treinamento. O classificador é determinado portanto pelo conjunto de pontos usados como treinamento. Por conveniência podemos considerar que todos os pontos na figura constituem tal conjunto (essa solução não é a única).

Para obter a superfície de decisão precisamos decidir as regiões que são mais próximas de pontos ◯ e as regiões que são mais próximas de pontos ▲. Uma forma de determinar isso é traçar o hiperplano separador para cada par de pontos de rótulos distintos; tal hiperplano é obtido pela reta perpendicular ao ponto médio da reta que conecta os dois pontos. Na prática, apenas pontos que estão próximos da fronteira precisam ser considerados. As retas separadoras para pares de pontos relevantes são desenhandas em azul na figura acima. As linhas conectando os pontos aparecem em azul pontilhado. A superfície de decisão, desenhanda em vermelho na figura acima, é obtida considerando a vizinhança de cada um desses pontos relevantes e de suas correspondentes retas separadoras.
Uma técnica mais objetiva (porém mais trabalhosa) consiste em obter o diagrama de voronoi desses pontos, e então extrair a superfície como a fronteira de células cujos centros possuem rótulos distintas.

Questão 2. Projete um classificador por árvore de decisão de 2 níveis que classifica corretamente o conjunto de dados abaixo. Desenhe a superfície de decisão do classificador na figura e descreva os parâmetros necessários.



A seguinte árvore de decisão classifica corretamente todos os pontos acima.



A superfície de decisão induzida pela árvore é desenhada em vermelho na figura.

Questão 3. O algoritmo de Q-Learning com aproximação linear considera que a função $Q(s,a)$ é um combinação linear de pesos w_k e funções características $f_k(s,a)$, atualizada através da equação

$$w_k^{(i+1)} = w_k^{(i)} + \alpha[R(s,a) + \gamma \max_{a'} Q^*(s',a') - Q^{(i)}(s,a)]f_k(s,a)\,,$$

onde α é a taxa de aprendizado, γ é o fator de desconto e $Q^*(s',a')$ é uma função oráculo que é considerada fixa (no algoritmo visto em classe usamos $Q^{(i)}(s',a')$ para estimar esse valor, mas para esse exercício considere esse valor como uma constante).

Obtenha uma função objetivo $J(w)$ para a qual a equação acima representa a atualização de um algoritmo de descida do gradiente estocástico.

O algoritmo de descida do gradiente estocástico otimiza os parâmetros w a fim de minimizar uma função $J(w)$ aplicando a regra

$$w_k^{(i+1)} \leftarrow w^{(i)} - \alpha \frac{\partial J_i(w)}{\partial w_k}\,,$$

onde $J(w) = \sum_i J_i(w)$.
Já o algoritmo de Q-Learning com aproximação linear assume que

$$Q(s,a) = \sum_k w_k \cdot f_k(s,a)\,.$$

Portanto procuramos uma função $J(w) = \sum_i J_i(w)$ tal que

$$\frac{\partial J_i(w)}{\partial w_k} = -[R(s,a) + \max_{a'} Q^*(s',a') - Q^{(i)}(s,a)]f_k(s,a)\,.$$

A seguinte função satisfaz essa propriedade:

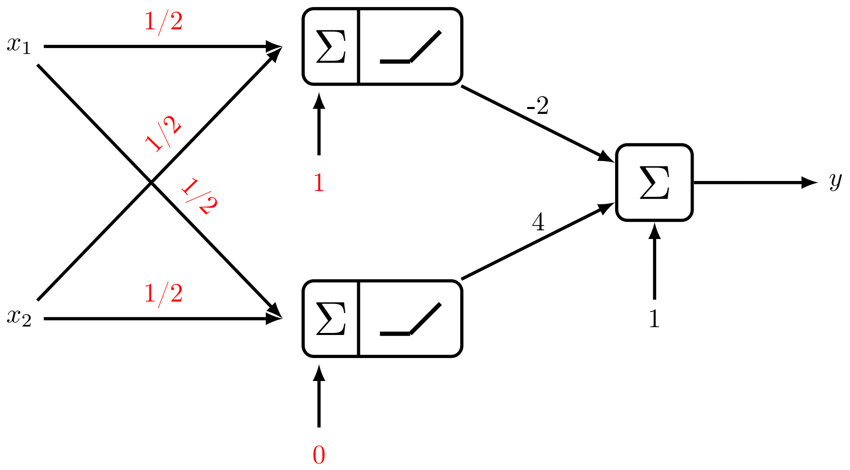
$$J(w) = \frac{1}{2}\sum_s \sum_a \sum_{s'} \big(R(s,a) + \gamma \max_{a'} Q^*(s',a') - Q(s,a)\big)^2\,.$$

A cada iteração o algoritmo "seleciona" uma transição (s,a,s') e aplica a regra de atualização do enunciado.

Questão 4. Complete a especificação da rede neural abaixo para que ela compute a seguinte função:

x_1	x_2	y
-1	-1	1
-1	1	-1
1	-1	-1
1	1	1

Os neurônios da camada oculta possuem função de ativação ReLU ($\text{relu}(z) = \max(0, z)$); o neurônio da camada de saída é linear. Note que os valores dos pesos da camade de saída já estão especificados e não devem ser alterados.



Devido a especificação dos parâmetros da camada de saída, temos que

$$-2h_1 + 4h_2 + 1 = y, \quad h_1, h_2 \geq 0\,,$$

onde h_1 e h_2 são as saídas dos neurônios superior e inferior da camada oculta, respectivamente. Considerando os valores desejados para a função chegamos ao sistema de equações

$$h_1(-1,-1) = 2h_2(-1,-1) \qquad h_1(-1,1) = 2h_2(-1,1) + 1$$
$$h_1(1,-1) = 2h_2(1,-1) + 1 \qquad h_1(1,1) = 2h_2(1,1)$$

O sistema acima possui 8 incógnitas e 4 equações (e as condições de não-negatividade) e é portanto subdeterminado. Isso indica que o problema tem várias (infinitas, de fato) soluções possíveis. Uma forma de resolvê-lo é impor restrições adicionais as variáveis. Por exemplo, dada a característica do problema podemos assumir que h_1 e h_2 são funções simétricas em relação à bissetriz e portanto $h_1(-1,1) = h_1(1,-1)$ e $h_2(-1,1) = h_2(1,-1)$. Podemos também assumir que $h_1(-1,-1) = h_2(-1,-1) = 0$, e que $h_2(1,1) = 1$ (outras suposições levam a respostas distintas). Essas restrições forçam h_1 e h_2 a serem funções crescentes de $x_1 + x_2$. Conseguimos dessa forma preencher, por tentativa e erro, a tabela abaixo com valores de h_1 e h_2.

x_1	x_2	$x_1 + x_2$	z_1	h_1	z_2	h_2	y
-1	-1	-2	≤ 0	0	≤ 0	0	1
-1	1	0	1	1	≤ 0	0	-1
1	-1	0	1	1	≤ 0	0	-1
1	1	2	2	2	1	1	1

O valores z_1 and z_2 na tabela acima correspondem ao resultado da combinação linear dos neurônios da camada oculta (ou seja, antes de ser aplicada a função relu). Novamente por tentativa e erro, conseguimos chegar aos parâmetros da rede acima. Comparando as colunas de soma das entradas e dos das combinações lineares (z) conseguimos selecionar os pesos e limiares dos neurônios.