

1. **O que acontece com o algoritmo do Perceptron se o problema for não-linearmente separável?**
 - O algoritmo do Perceptron é baseado no princípio de que os dados são linearmente separáveis. Se o problema for não-linearmente separável, o Perceptron não conseguirá convergir para uma solução. Isso significa que, durante o processo de treinamento, o algoritmo continuará ajustando os pesos indefinidamente sem encontrar uma linha que separe completamente as classes.
2. **Como poderia ser tratado a questão acima no algoritmo do Perceptron?:**
 - Uma possibilidade é transformar o espaço de características de forma que os dados se tornem linearmente separáveis no novo espaço. Isso pode ser feito através de algumas técnicas de transformação de características.
3. **Qual seria os eventuais inconvenientes de se usar valores muito grandes para a taxa de aprendizagem? E para valores muito pequenos?**
 - **Valores Altos:** Uma taxa de aprendizagem muito alta pode levar a atualizações de peso excessivas, fazendo com que o algoritmo "salte" sobre o mínimo ótimo durante a otimização. Isso pode causar instabilidade e impedir a convergência do modelo.
 - **Valores Baixos:** Uma taxa de aprendizagem muito baixa resultará em atualizações de peso muito pequenas, tornando o processo de treinamento extremamente lento. Isso significa que o modelo pode levar muito tempo para convergir, ou pode até ficar preso em mínimos locais.
4. **Dois projetistas estão implementando o mesmo problema de classificação de padrões por meio do Perceptron. Comente se, após o processo de treinamento, o vetor de pesos final terá que ser o mesmo em ambos os projetos.**
 - Se dois projetistas estão implementando o mesmo problema de classificação usando o Perceptron e ambos começam com a mesma inicialização de pesos e utilizam os mesmos dados no mesmo ordenamento para treinamento, eles deveriam chegar ao mesmo vetor de pesos final, assumindo que todos os outros parâmetros do algoritmo são os mesmos. Porém na prática, pequenas diferenças no processo de treinamento (como a ordem em que as amostras são apresentadas ao modelo) pode levar a vetores de pesos finais diferentes. Além disso, se a inicialização dos pesos for aleatória e diferente entre os projetos, é provável que os vetores de pesos finais também sejam diferentes, embora o problema de classificação possa ser resolvido de forma funcional.