

Perceptron para Classificação Binária

Lucas Dantas Igarashi
Centro Tecnológico de Joinville (CTJ)
Universidade Federal de Santa Catarina
Joinville, Brasil
lucas20020712@gmail.com

Resumo—Este documento apresenta a implementação de um Perceptron de camada única para classificação binária, visando avaliar seu desempenho em diferentes bases de dados sintéticas. O modelo foi treinado e avaliado utilizando conjuntos de dados sintéticos com diferentes características (linearmente separáveis e não linearmente separáveis, balanceados e desbalanceados). Foi utilizada a técnica de validação cruzada para estimar a capacidade de generalização do modelo. Os resultados demonstram a influência da separabilidade linear e do balanceamento dos dados no desempenho do Perceptron. A análise da perda e da acurácia ao longo das épocas permitiu observar a convergência do modelo. O Perceptron implementado obteve bons resultados em conjuntos de dados linearmente separáveis, demonstrando a eficácia do algoritmo para problemas de classificação. (Resumo)

Keywords—perceptron, classificação,

I. INTRODUÇÃO

Antes de mais nada, os problemas de classificação envolvem a tarefa de separar dados em diferentes classes conforme os seus atributos, utilizando um hiperplano para distinguir as variáveis conhecidas, mas cuja classe não é previamente determinada. Dessa forma, foi criado o Perceptron, a forma mais simples de uma rede neural artificial, sendo introduzida por Frank Rosenblatt em 1958 para solucionar este problema, conforme descrito em seu trabalho seminal[1].

Este estudo visa avaliar a implementação do Perceptron de camada única para solucionar problemas de classificações em duas dimensões, utilizando bases de dados linearmente separáveis e não linearmente separáveis. Para isso, foram empregadas validações cruzadas para estimar o poder preditivo e métricas de precisão e revogação para avaliar o desempenho do modelo.

II. IMPLEMENTAÇÃO

A. Soma Ponderada

O Perceptron de Camada Única consiste em um único neurônio com pesos sinápticos e um viés (peso adicional w_0) ajustáveis. Esses componentes são usados para calcular a soma ponderada das entradas, resultando no conjunto de saídas que contém os resultados da classificação.

A equação da soma ponderada é dada por (1):

$$y = w \cdot x + b \quad (1)$$

Sendo y a saída linear do Perceptron, w o vetor de pesos, x o vetor de entradas e b o viés. Essa soma ponderada é então passada por uma função de ativação, como a função Sigmoid, para produzir a saída final do Perceptron.

B. Função de Ativação

Para a função de ativação, utilizamos a função Sigmoid. A função Sigmoid é preferida devido à sua capacidade de mapear valores reais para um intervalo entre 0 e 1, facilitando a interpretação das saídas como probabilidades. Além disso, a

suavidade e diferenciabilidade da função Sigmoid permitem a aplicação de métodos baseados em gradiente para ajustar os pesos de forma mais precisa.

A função Sigmoid é definida pela seguinte equação (2):

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (2)$$

Sendo, $\sigma(x)$ é a saída da função Sigmoid e x é a soma ponderada das entradas.

C. Validação Cruzada

A validação cruzada foi implementada para avaliar o desempenho do modelo de forma robusta. Utilizamos a técnica de K-Fold, onde o conjunto de dados é dividido em K subgrupos (folds). Em cada iteração, K-1 folds são usados para treinamento e o fold restante é usado para teste. Esse processo é repetido K vezes, garantindo que todas as amostras sejam usadas tanto para treinamento quanto para teste. A validação cruzada ajuda a minimizar o risco de overfitting e fornece uma estimativa mais confiável do poder preditivo do modelo.

D. Treinamento do Modelo

O treinamento do modelo é realizado iterando sobre as épocas e as amostras. Para cada amostra, calculamos a soma ponderada dos pesos e aplicamos a função Sigmoid para obter a probabilidade prevista. Em seguida, calculamos a perda usando a entropia cruzada, que mede a diferença entre a probabilidade prevista e o valor real. Essa perda é então usada para atualizar os pesos e o bias do modelo, ajustando-os para minimizar a perda. A Figura 1 plota a fronteira de decisão com as bases de dados utilizadas para testar para o primeiro fold.

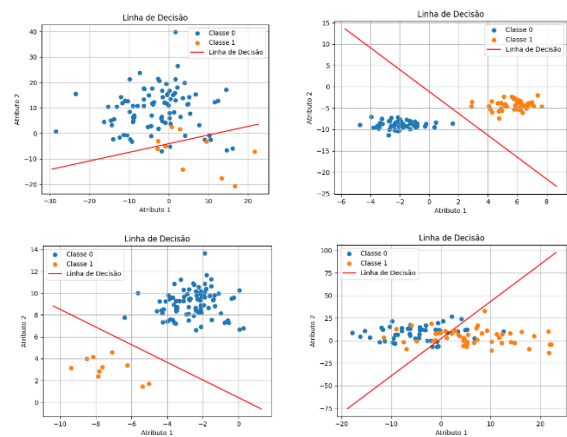


Fig. 1. Fronteiras de decisão para diferentes bases de treinamento

O processo é repetido até que o modelo seja treinado por todas as épocas especificadas. Durante o treinamento, as métricas de avaliação, como acurácia, precisão, revogação, F1-Score e AUC-ROC, são calculadas e armazenadas para

monitorar o desempenho do modelo ao longo do tempo. A Figura 2 mostra perda por época para a base de dados de duas dimensões linearmente separável e balanceada.

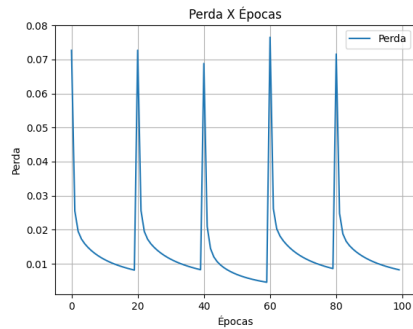


Fig. 2. Perda por época para a base de dados de duas dimensões linearmente separável e balanceada

III. RESULTADOS

A. Base de dados de duas dimensões

Linearmente Separável e Balanceada

O Perceptron conseguiu separar perfeitamente as classes devido à linearidade e ao balanceamento dos dados. Todas as métricas de avaliação, como acurácia, precisão, revocação, F1-Score e AUC-ROC, atingiram o score máximo de 1.

Linearmente Separável e Não Balanceada

Embora todas as métricas tenham atingido o score máximo, a precisão, revocação, F1-Score e AUC-ROC apresentaram mais erros no início do treinamento devido ao desequilíbrio das classes. No início, o modelo teve dificuldade em classificar corretamente a classe minoritária, mas ajustou os pesos ao longo do tempo, melhorando a classificação e convergindo os valores das métricas para um score ótimo.

Não Linearmente Separável e Balanceada

Nenhuma métrica conseguiu atingir o valor máximo devido à não linearidade dos dados, pois o Perceptron não consegue lidar adequadamente com este tipo de base. Os scores das métricas se mantiveram constantes, indicando que o modelo não conseguiu melhorar significativamente a classificação ao longo das épocas. No entanto, a curva ROC apresentou o melhor score, pois considera a taxa de verdadeiros positivos e falsos positivos, proporcionando uma visão mais equilibrada do desempenho do modelo.

Não Linearmente Separável e Não Balanceada

Nenhuma métrica atingiu um score perfeito, mas a precisão e o F1-Score tiveram pior desempenho devido ao desequilíbrio das classes e à não linearidade dos dados. A precisão foi afetada porque o modelo teve dificuldade em prever corretamente a classe minoritária, resultando em um maior número de falsos positivos. O F1-Score, sendo a média harmônica da precisão e da revocação, também foi impactado negativamente por essa dificuldade em classificar corretamente a classe minoritária. A curva ROC apresentou o melhor score, pelo mesmo motivo do caso anterior.

B. Base de dados de cinco dimensões

Linearmente Separável e Balanceada

Apesar do aumento no número de atributos, o Perceptron conseguiu encontrar um score perfeito para todas as métricas devido à linearidade e ao balanceamento dos dados. O modelo pode separar as classes de forma eficaz em um espaço de maior dimensão.

Linearmente Separável e Não Balanceada:

Mesma explicação para o caso em 2D desta base de dados. O desequilíbrio das classes causou mais erros no início do treinamento, mas o modelo eventualmente ajustou os pesos para melhorar a classificação.

Não Linearmente Separável e Balanceada

A precisão e o F1-Score não ficaram abaixo das demais métricas porque o balanceamento dos dados ajudou o modelo a aprender melhor a classificação das duas classes, mesmo que a separação não fosse linear. O modelo ainda enfrentou dificuldades devido à não linearidade, mas o balanceamento dos dados mitigou parte desse problema.

Não Linearmente Separável e Não Balanceada:

A revocação e a AUC-ROC tiveram uma pontuação de quase que máxima porque o modelo conseguiu identificar a maioria dos verdadeiros positivos, apesar do desequilíbrio das classes. A curva ROC é menos afetada pelo desequilíbrio das classes e fornece uma visão mais completa do desempenho do modelo.

IV. CONCLUSÕES

O Perceptron de Camada Única implementado neste trabalho demonstrou ser eficaz na classificação de dados linearmente separáveis, tanto em espaços 2D quanto 5D. A utilização da função de ativação Sigmoid permitiu uma melhor interpretação das saídas como probabilidades, facilitando o ajuste dos pesos durante o treinamento.

A validação cruzada foi essencial para avaliar o desempenho do modelo de forma robusta, minimizando o risco de overfitting e proporcionando uma estimativa confiável do poder preditivo do modelo. Os resultados mostraram que o Perceptron conseguiu atingir score máximo de acurácia em conjuntos de dados linearmente separáveis e balanceados, independentemente do número de dimensões.

No entanto, o desempenho do Perceptron foi comprometido em conjuntos de dados não linearmente separáveis e não balanceados. A precisão e o F1-Score apresentaram valores mais baixos devido à dificuldade do modelo em lidar com a não linearidade e o desequilíbrio das classes. A curva ROC, por outro lado, mostrou-se uma métrica mais robusta, fornecendo uma visão mais equilibrada do desempenho do modelo.

Em resumo, o Perceptron de Camada Única funciona para problemas de classificação linearmente separáveis. No entanto, suas limitações em relação à não linearidade e ao desequilíbrio de classes devem ser consideradas.

REFERÊNCIAS

- [1] Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65(6), 386-40
- [2] Igarashi, Lucas Dantas. "Implementação do Perceptron". Disponível em: [LucasIgarashi/Perceptron](https://lucasigarashi.github.io/Perceptron/).