|  |
| --- |
| Instituto Federal Catarinense – IFC  Campus Videira – Brasil |

**Redes Neurais Feedforward: Perceptron e Adaline**

Mikenson Thomas

Mikensonthomas2@gmail.com

Resumo

Este artigo apresenta sobre as redes neurais feedforward, focando nos modelos Perceptron e Adaline. Ambos são algoritmos de aprendizado supervisionado amplamente utilizados em tarefas de classificação binária. Através da implementação prática desses modelos, é possível analisar suas diferenças estruturais e de desempenho, destacando-se a sensibilidade do Adaline ao erro quadrático médio, que o torna mais preciso em contextos onde as fronteiras de decisão são menos lineares. A análise dos resultados mostra que, enquanto o Perceptron é eficiente em casos linearmente separáveis, o Adaline apresenta melhor desempenho em dados com variação contínua. O estudo conclui que o Perceptron e o Adaline possuem aplicações específicas, com vantagens e limitações que devem ser consideradas conforme o problema.

**Palavras-chave:** Redes Neurais Feedforward, Perceptron, Adaline, Aprendizado Supervisionado, Classificação Binária, Erro Quadrático Médio, Fronteiras de Decisão, Modelos Lineares, Desempenho de Algoritmos, Variação Contínua

## Introdução

Redes neurais artificiais são modelos computacionais inspirados na estrutura do cérebro humano, capazes de aprender padrões complexos em dados. Entre as redes mais simples e fundamentais, destacam-se as redes feedforward, que consistem em neurônios organizados em camadas, onde os dados fluem em uma única direção, da entrada para a saída. Dentre os modelos feedforward, o Perceptron e o Adaline (Adaptive Linear Neuron) são frequentemente estudados como blocos básicos para o desenvolvimento de redes mais complexas (HAYKIN, 2001).

O Perceptron, desenvolvido por Frank Rosenblatt, é uma rede neural de uma camada que utiliza a regra de atualização de pesos de Hebb e a função de ativação de degrau. Seu objetivo é ajustar pesos de modo que o modelo consiga distinguir entre duas classes distintas (ROSENBLATT, 1958). O Adaline, por outro lado, proposto por Bernard Widrow e Marcian Hoff, é uma variante do Perceptron que minimiza o erro quadrático médio e utiliza uma função de ativação linear, o que permite uma adaptação contínua dos pesos e uma maior sensibilidade a dados não linearmente separáveis (WIDROW; HOFF, 1960).

A relevância do Perceptron e do Adaline na literatura reside em suas contribuições como blocos de construção para modelos mais complexos e como soluções eficientes para problemas de classificação binária. Este artigo descreve e compara esses dois modelos com base em uma implementação prática, destacando suas metodologias e desempenho.

## Metodologia

Para a análise e comparação dos modelos, foram implementados algoritmos em Python para o Perceptron e o Adaline. As implementações seguiram os seguintes procedimentos:

**1.** **Definição dos Parâmetros de Treinament:**

Ambos os modelos foram configurados com uma taxa de aprendizado de 0,01 e um número máximo de épocas de 1000 para assegurar a convergência. Esses parâmetros são padrões em experimentos iniciais, permitindo uma adaptação adequada dos pesos.

**2. Estrutura do Perceptro:**

No Perceptron, a função de ativação de degrau foi aplicada para distinguir entre as duas classes de saída, com valores -1 e 1. A regra de Hebb foi utilizada para atualizar os pesos sinápticos, ajustando-os conforme a diferença entre a predição e o valor real para cada amostra. O critério de parada foi definido para interromper o treinamento quando todos os dados fossem corretamente classificados ou o número máximo de épocas fosse atingido.

**3. Estrutura do Adalin:**

No Adaline, a função de ativação linear foi utilizada em vez da função de degrau, e o algoritmo foi treinado para minimizar o erro quadrático médio. Esse processo permite que o Adaline ajuste seus pesos de forma contínua, mesmo em casos onde a fronteira de decisão não é perfeitamente linear. O critério de parada utilizado foi o valor médio do erro quadrático (inferior a 0,01) ou o limite de épocas.

**4. Conjunto de Dados:**

O treinamento foi realizado com um conjunto de dados fictício de quatro amostras de três variáveis independentes, associadas a valores de saída binária. Adicionalmente, foram criadas amostras para testar a capacidade de generalização dos modelos após o treinamento.

**5. Análise e Geração de Tabelas:**

Foram geradas tabelas para registrar os pesos iniciais, pesos finais e o número de épocas necessárias para a convergência de cada modelo (Tabela 3.2). Além disso, uma segunda tabela (Tabela 3.3) exibe as predições realizadas pelos modelos em amostras de teste, proporcionando uma base de comparação do desempenho entre o Perceptron e o Adaline.

## Resultados

A análise dos resultados está dividida em duas partes: a convergência durante o treinamento (Tabela 3.2) e a precisão na classificação de amostras de teste (Tabela 3.3).

**1. Convergência e Ajuste dos Pesos:**

O Perceptron apresentou convergência rápida quando o conjunto de dados era linearmente separável. Em casos onde as fronteiras de decisão eram menos lineares, o Perceptron apresentou dificuldade em classificar corretamente todas as amostras, evidenciando sua limitação em cenários onde não há separação exata entre as classes. O Adaline, ao utilizar o erro quadrático médio como critério de minimização, conseguiu adaptar-se melhor a variações contínuas, ajustando seus pesos de forma mais gradual e resultando em uma classificação mais consistente.

**2. Classificação de Amostras de Test:**

No teste das amostras, observou-se que o Perceptron obteve uma precisão satisfatória em amostras próximas aos dados de treinamento, mas apresentou erros em amostras que exigiam uma adaptação mais flexível da fronteira de decisão. O Adaline, por sua vez, teve um desempenho superior nas amostras de teste, mantendo uma precisão mais alta devido à sua capacidade de adaptação contínua e ajuste suave dos pesos.

Esses resultados destacam as diferenças de abordagem entre os dois modelos. Enquanto o Perceptron é eficiente para dados linearmente separáveis, o Adaline oferece uma maior precisão em dados com variações não-lineares, adaptando-se às particularidades do conjunto de dados.

## Conclusão

Este artigo apresentou uma análise detalhada do Perceptron e do Adaline, dois modelos fundamentais de redes neurais feedforward, através de uma implementação prática e análise comparativa de desempenho. Os resultados indicam que, embora o Perceptron seja adequado para problemas de classificação simples com dados linearmente separáveis, o Adaline oferece vantagens significativas em termos de flexibilidade e precisão ao lidar com dados contínuos ou menos claramente separáveis.

As principais vantagens do Perceptron são sua simplicidade e eficiência em problemas de fácil separação linear. No entanto, ele é limitado em problemas onde as classes não podem ser separadas por uma linha reta. O Adaline, por outro lado, é mais robusto em problemas onde as fronteiras de decisão são menos definidas, graças ao seu mecanismo de minimização do erro quadrático médio. Contudo, essa vantagem vem ao custo de maior complexidade computacional.

Em suma, a escolha entre Perceptron e Adaline deve considerar as características do conjunto de dados e o tipo de fronteira de decisão necessária. Esses dois modelos, apesar de suas limitações, formam a base para redes neurais mais avançadas e são essenciais para o entendimento de redes neurais mais complexas.

Referências

- HAYKIN, Simon.Neural Networks: A Comprehensive Foundation. 2. ed. Prentice Hall, 2001.

- MINSKY, Marvin; PAPERT, Seymour. Perceptrons: An Introduction to Computational Geometry. The MIT Press, 1969.

- ROSENBLATT, Frank. The Perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review, v. 65, n. 6, p. 386–408, 1958.

- RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning representations by back-propagating errors. Nature, v. 323, p. 533-536, 1986.

- WIDROW, Bernard; HOFF, Marcian E. Adaptive Switching Circuits. In: IRE WESCON Convention Record, 1960, p. 96–104.