

## **Universidade Positivo**

### **Relatório Rede Neural de qualidade de Vinhos**

**Alunos:** Nathan Henrique Lucindo dos Santos e Paulo Alexandre Haliski

## **INTRODUÇÃO**

### **DESCRIÇÃO DO DATASET**

O dataset utilizado neste projeto é o *RedWineQuality*, que contém dados relacionados à composição química de diferentes amostras de vinho tinto e sua respectiva qualidade. Os dados foram coletados considerando variáveis como acidez fixa, acidez volátil, citratos, teor de álcool, entre outros. Para a análise, os dados foram transformados em uma classificação binária: amostras com nota maior ou igual a 6 foram consideradas de "boa qualidade" (classe 1), enquanto as demais foram classificadas como "baixa qualidade" (classe -1).

### **SELEÇÃO DOS ATRIBUTOS**

O dataset conta com 12 colunas, sendo 1 delas a de qualidade, usada como saída da rede neural, e as outras 11 são atributos do vinho, sendo elas: Acidez Fixa, Acidez Volátil, Ácido Cítrico, Açúcar Residual, Cloreto de Sódio, Dióxido de Enxofre Livre, Dióxido de Enxofre Total, Densidade, pH, Sulfatos e Teor Alcoólico. Inicialmente, foram escolhidos todos os 11 atributos para fazer parte do treinamento da rede, porém, com a obtenção de uma taxa de erro acima do esperado, a seleção de atributos foi diminuída para apenas 6, buscando uma melhora no treinamento do modelo. Essa alteração não resultou em mudanças significativas, então foi mantido o número anterior de 11 colunas.

### **ESTRUTURA DA REDE NEURAL**

Foi utilizada uma rede neural com múltiplas camadas, projetada para realizar uma tarefa de classificação binária, a estrutura segue o modelo de aprendizado supervisionado, onde os pesos são ajustados com *backpropagation*. Possui 11 neurônios na camada de entrada, sendo 1 para cada atributo, 3 camadas ocultas com 6 neurônios cada uma, 1 neurônio na camada de saída, e 4 vetores de pesos para armazenar os pesos entre camadas.

### **PROCESSO DE TREINAMENTO**

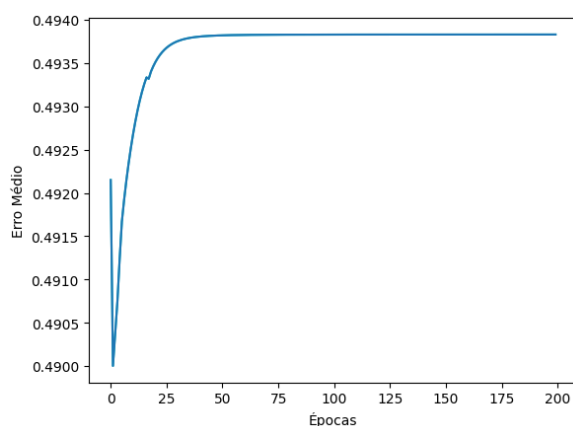
Primeiro, os vetores de pesos são inicializados com valores aleatórios entre -0.5 e 0.5, para evitar valores muito desbalanceados, e a taxa de aprendizado e número de épocas é definida na implementação do código.

Para cada época, a rede processa todas as amostras do conjunto de dados, a entrada dos dados com o bias passam pelas 3 camadas ocultas, e por fim para a saída. Também é calculado o erro através da diferença entre a saída desejada e a saída real, representadas por  $d[j]$  e  $Y$ , respectivamente. Na propagação de erro (*backpropagation*), é ajustado o vetor de pesos de cada camada com base no gradiente de erro. Ao final de cada época, é mostrado no

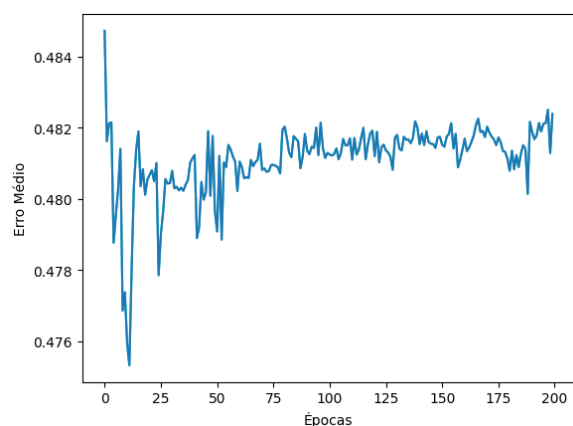
console o Erro Total Médio (Etm), que recebe a média do vetor de erros através da função 'mean()' da biblioteca numpy.

## RESULTADOS E ANÁLISE

Após o a implementação da rede e seu treinamento, foram obtidos os seguintes resultados: O gráfico gerado após a execução resultou em um bom treinamento, porém com a taxa de erro elevada, após consulta e orientações a Professora, foram realizadas mudanças em quantidade épocas e taxa de aprendizado, além de mudarmos qual função utilizar. Com isso teve uma tendência decrescente no erro, alterando o desenho do gráfico, mostrando que o modelo aprendeu os padrões a partir dos dados de uma forma com menos erros.



Função Sigmoid



Função Tangente hiperbólica

A redução também se dá para na mudança de parâmetros iniciais como citado acima onde foi aumentado a taxa de aprendizado, e diminuído o número de épocas para melhor ajuste. Como falado a função final escolhida foi a Tangente Hiperbólica, onde se mostrou mais eficaz para a classificação, diferentemente da Relu e sigmoid, que apresentaram desempenho inferior para esses dados.

Com isso vemos que o modelo teve um bom desempenho para a classificação dos vinhos baseado em atributos químicos. As escolhas de quantidades de camadas, taxa de aprendizado, épocas, e a função, foram suficientes para captar o resultado esperado.