Título do Trabalho

Yuri Carneiro Parente[1]

[1]Universidade Federal do Tocantins - UFT

yuri.carneiro@mail.uft.edu.br

Resumo. *Este artigo apresenta uma aplicação prática de uma das técnicas de aprendizado de máquina aprendida na matéria de machine learning, nesse caso, utilizando redes neurais para a classificação de espécies botânicas no conjunto de dados Iris. Através da utilização de uma abordagem baseada em redes neurais, exploramos o potencial dessa técnica na resolução de problemas de classificação. O artigo descreve a metodologia adotada, incluindo o pré-processamento dos dados, a criação da arquitetura da rede neural e o treinamento do modelo. Os resultados obtidos são analisados e discutidos, destacando a eficácia da abordagem e suas implicações na classificação de espécies botânicas.*

1. Introdução

A classificação de espécies botânicas é uma tarefa desafiadora e de grande importância em diversas áreas, como biologia, ecologia e agronomia. A técnica de aprendizado de máquina utilizando redes neurais tem se mostrado promissora para a resolução desse tipo de problema. Neste artigo, utilizamos o conjunto de dados Iris, amplamente conhecido na comunidade de aprendizado de máquina, para explorar a eficácia das redes neurais na classificação de três espécies de plantas: Iris setosa, Iris versicolor e Iris virginica. O objetivo é demonstrar como essa técnica pode ser aplicada com sucesso nesse contexto.

2. Metodologia

2.1. Conjunto de Dados Iris

O conjunto de dados Iris contém informações sobre três espécies de flores Iris: Iris setosa, Iris versicolor e Iris virginica. Cada uma das espécies é representada por 50 amostras, totalizando 150 amostras. Cada amostra possui quatro características, sendo elas: comprimento da sépala, largura da sépala, comprimento da pétala e largura da pétala. O conjunto de dados Iris é amplamente utilizado na área de aprendizado de máquina para problemas de classificação e foi obtido do seguinte link: https://www.kaggle.com/datasets/uciml/iris

2.2. Pré-processamento dos Dados

Inicialmente, foi realizada uma breve análise exploratória dos dados, exibindo informações estatísticas e visualizando as primeiras e últimas entradas do conjunto de dados. Em seguida, atribuímos valores numéricos às espécies para facilitar a classificação. Os dados foram normalizados utilizando a técnica de Normalização, a fim de padronizar as escalas das variáveis de entrada, para assim obter um melhor rendimento com a rede neural, uma vez que é recomendado que os dados sejam bem tratados antes de serem treinados na rede.

2.3. Divisão do Conjunto de Dados

Para avaliar o desempenho do modelo de aprendizado de máquina, o conjunto de dados Iris foi dividido em conjuntos de treinamento e teste. A divisão foi realizada usando a função *train_test_split* da biblioteca *sklearn.model_selection*. O tamanho do conjunto de teste foi definido como 20% do conjunto de dados total.

A divisão do conjunto de dados em treinamento e teste é uma prática comum para avaliar a capacidade de generalização do modelo. O conjunto de treinamento é utilizado para treinar o modelo, ajustando os pesos e parâmetros da rede neural. Já o conjunto de teste é usado para avaliar o desempenho do modelo em dados não vistos anteriormente.

Ao dividir o conjunto de dados Iris, garantimos que o modelo seja avaliado em dados independentes do treinamento, evitando assim problemas de superestimação do desempenho. Essa divisão é essencial para verificar se o modelo é capaz de generalizar bem para novos dados e não apenas decorar os exemplos do conjunto de treinamento.

2.4. Divisão do Conjunto de Dados

A arquitetura da rede neural utilizada consiste em uma sequência de camadas densas, intercaladas com camadas de dropout. A primeira camada possui 50 neurônios, pois esse é o número de entrada definido para este problema, e nela é utilizada a função de ativação ReLU. Já a segunda camada possui 30 neurônios e também utiliza a função ReLU. Além disso, Foi aplicado dropout com taxa de 0.3 para tentar evitar o overfitting. A terceira camada possui 15 neurônios e função de ativação ReLU. Por fim, a camada de saída possui um número de neurônios igual ao número de classes (3) e utiliza a função de ativação softmax. Lembrando que apenas a camada de entrada e saída são previsíveis de acordo com o problema, porém as outras camadas são definidas de forma abitrária e, geralmente, a melhor configuração tem que ser encontrada através da tentativa e erro.

2.5. Treinamento do Modelo

O modelo foi compilado com a função de perda *"categorical_crossentropy"* e o otimizador "adam". Além disso, foram definidas 50 épocas de treinamento e utilizadas as técnicas de *Early Stopping* e *Model Checkpoint* para evitar o overfitting e salvar o melhor modelo obtido durante o treinamento. Essas estratégias contribuem para o aprimoramento do modelo, permitindo que ele aprenda padrões relevantes nos dados de treinamento sem ocorrer sobreajuste (overfitting).

Sendo assim, a técnica de *Early Stoppin*g interrompe o treinamento se a melhoria na função de perda nos dados de validação não for significativa após um determinado número de épocas, evitando o desperdício de recursos computacionais e prevenindo o overfitting. Pois nesse caso em específico, como a base de dados é pequena, com apenas 150 amostras, se a rede neural for treinada por muitas épocas, existe uma chance muito alta de que ela se ajuste demais ao problema, assim, pode ter um resultado de 100% no treinamento, porém não será boa para generalizar caso algo fora do padrão apareça.

O Model Checkpoint é responsável por salvar o modelo com os melhores resultados durante o treinamento, com base na função de perda nos dados de validação. Dessa forma, podemos recuperar o modelo que obteve o melhor desempenho geral, mesmo que o treinamento tenha sido interrompido precocemente.

Com isso, essas técnicas combinadas garantem um treinamento eficiente e eficaz do modelo, permitindo que ele aprenda padrões relevantes nos dados de treinamento e generalize bem quando novos dados forem apresentados..

3. Resultados

Após o treinamento do modelo, foram obtidos resultados promissores na classificação das diferentes espécies de iris. A acurácia alcançada nos dados de teste foi de aproximadamente 93,33%, indicando uma boa capacidade de generalização do modelo. Além disso, a matriz de confusão dos resultados revelou um desempenho equilibrado na classificação das espécies. Isso demonstra a capacidade do modelo em discriminar corretamente as características distintivas das diferentes espécies de iris.

O relatório de classificação forneceu métricas detalhadas para cada classe, incluindo precisão, recall e F1-score. Os resultados mostram que o modelo obteve altas taxas de precisão para todas as classes, evidenciando sua capacidade de distinguir corretamente entre as espécies de iris.

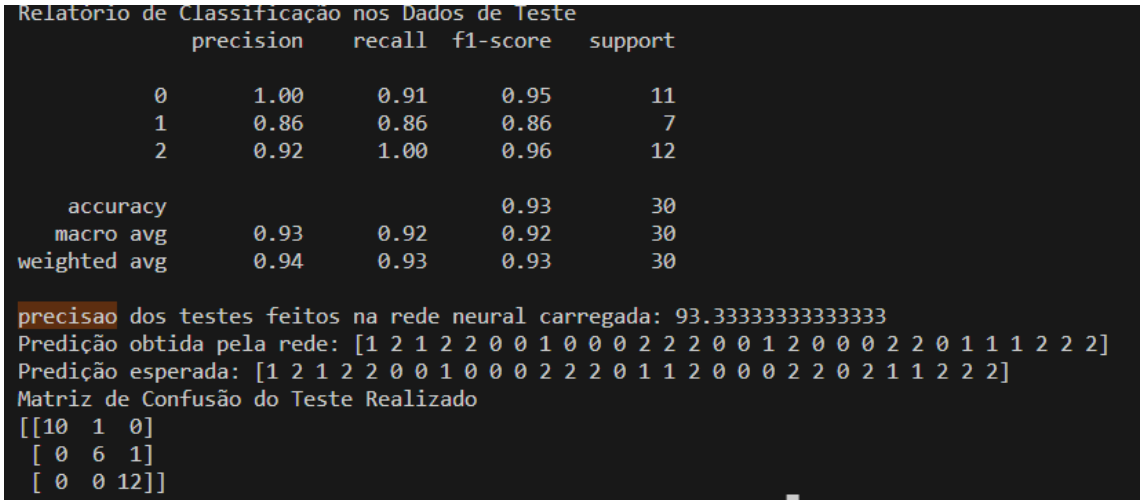

Figura 1. Resultados obtidos

Além disso, a precisão obtida ao carregar o modelo salvo foi consistente com os resultados obtidos durante o treinamento, validando a eficácia da técnica de Model Checkpoint.

4. Conclusão

Este trabalho apresentou uma aplicação prática de redes neurais no conjunto de dados Iris para a classificação de espécies botânicas. Os resultados obtidos confirmaram a eficácia dessa abordagem, evidenciando a capacidade das redes neurais em aprender e identificar padrões nas características das espécies de íris. Assim, reforçando a capacidade do uso de técnicas de aprendizado de máquina em problemas de classificação.

Através da metodologia apresentada, incluindo o pré-processamento dos dados, a criação da arquitetura da rede neural e o treinamento do modelo, foi possível obter resultados satisfatórios. O modelo demonstrou um desempenho equilibrado e alta precisão na classificação das espécies.

Com isso, os resultados alcançados neste trabalho destacam o potencial das redes neurais como ferramentas valiosas para auxiliar em , por exemplo, pesquisas botânicas e áreas afins.

Referências

KAGGLE, Datasets: UCI Machine Learning Repository - Iris. Disponível em: <www.kaggle.com/datasets/uciml/iris>. Acesso em: 24 jun. 2023.