



Tarefa 5 – Redes Neurais

Introdução ao Aprendizado de Máquina 2023.02

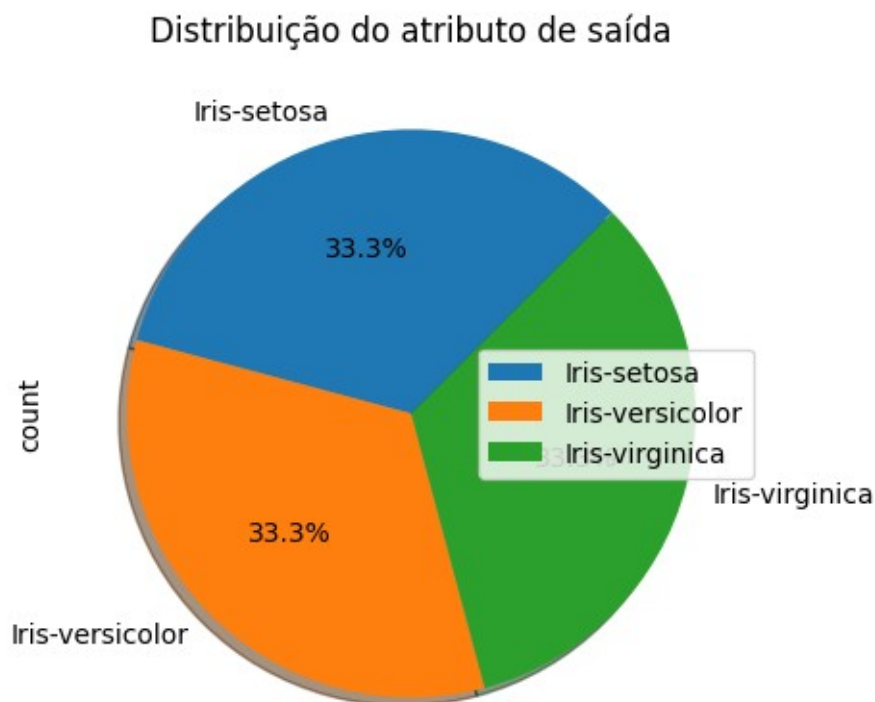
Aluno: Diego Vasconcelos Schardosim de Matos
DRE: 120098723

1. Introdução

Este trabalho foi feito para disciplina Introdução de Aprendizado de Máquina 2023.02 onde devemos praticar os conceitos visto até então em aula sobre Redes Neurais. Usando o popular dataset Iris será treinado uma Rede Neural para classificar uma flor iris entre 'setosa', 'versicolor' e 'virginica' (nomes em inglês, desconheço os termos em português) com 4 parâmetros de entrada comprimento da sépala (em cm), largura da sépala (em cm), comprimento da pétala (em cm) e largura da pétala (em cm), o modelo final tem uma taxa de acerto de 96,67%.

2. Treinamento

Para o treinamento as 150 instâncias foram divididas entre 80% (120 instâncias) para o treinamento e 20% (30 instâncias) para o teste, como as instâncias de cada classe estão bem distribuídas no dataset, o treinamento e teste também ficaram.



Em seguida foi aplicado o método de reamostragem Kfold Cross Validation que consiste em agrupar as instâncias em k grupos e treinar o modelo iterativamente com cada grupo diferente. Para este trabalho foi usado um valor de $k = 5$.

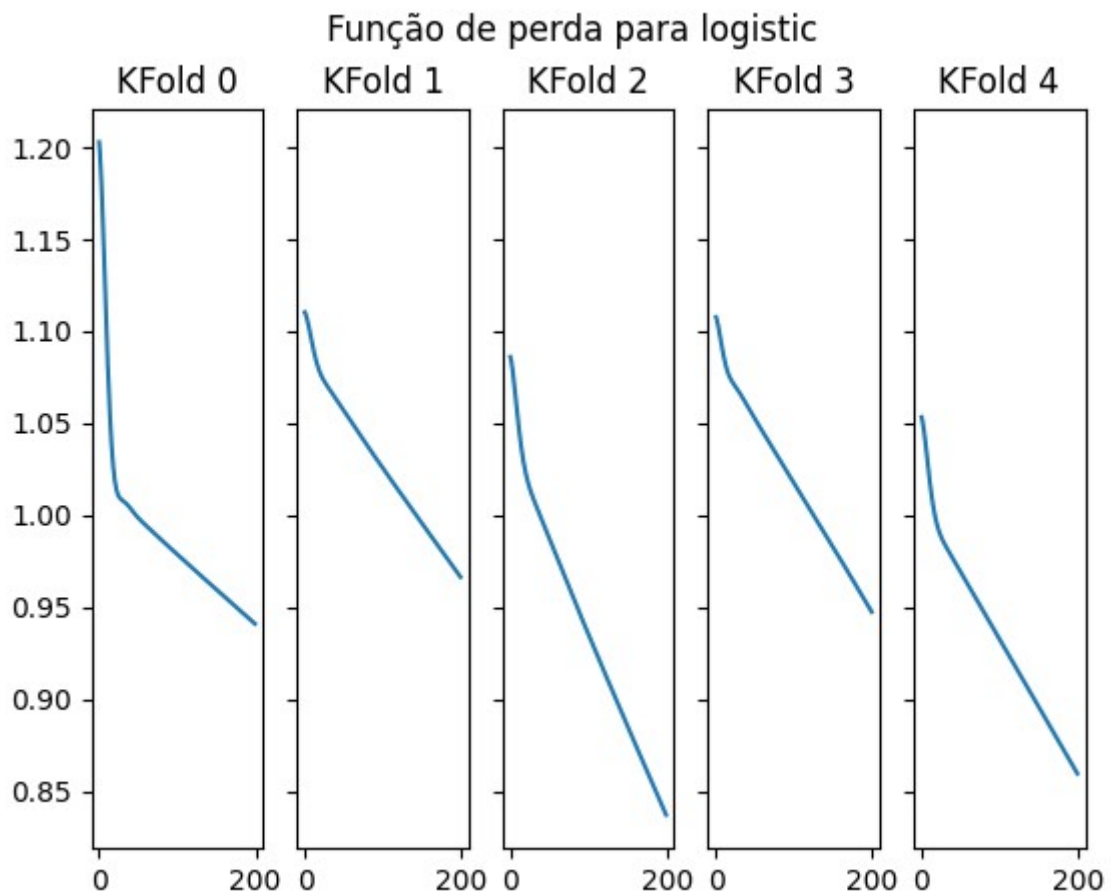
Por fim, foram treinados 6 modelos diferentes de Rede Neural, isto foi feito para descobrirmos de maneira exploratória quais melhores parâmetros para nossa Rede Neural.

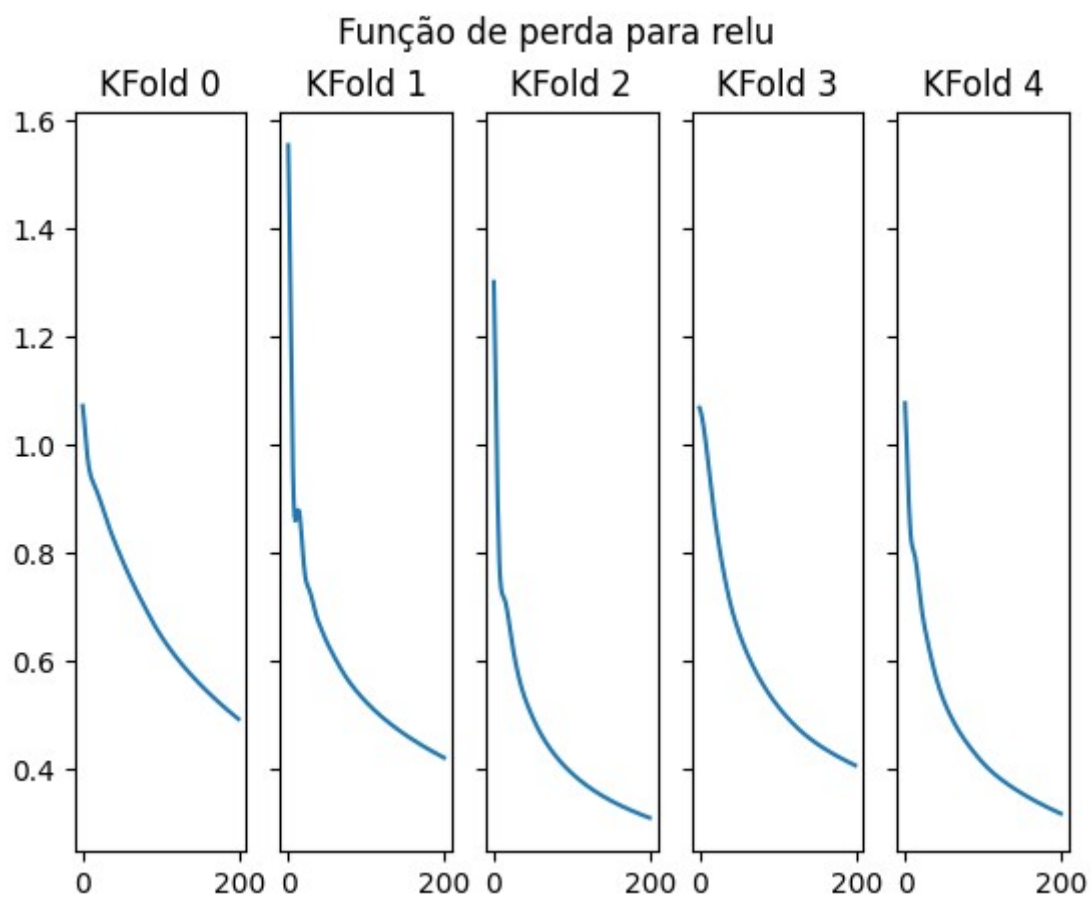
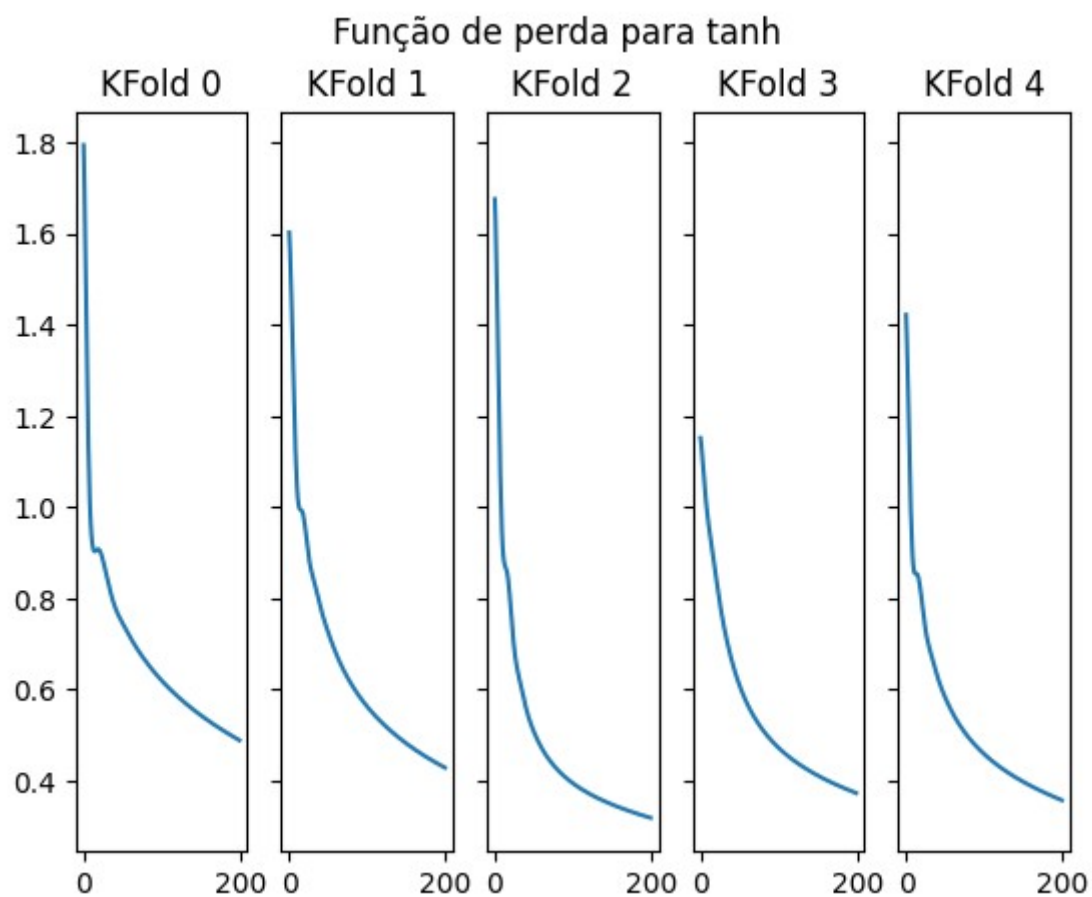
3. Comparação de desempenho entre os modelos

Para os primeiros 3 modelos foram usados os valores padrões do MLPClassifier da biblioteca `sklearn.neural_network`, são eles: 1 camada com 100 neurônios, taxa de aprendizado

0.001 e 200 iterações máximas até convergir. Usando o Método do Gradiente Descendente Estocástico para otimizar os pesos dos neurônios cada modelo foi treinado com uma função de ativação diferente: logística, tangente hiperbólica e ReLU.

Os resultados desses 3 modelos não foram satisfatórios, sendo que nenhum deles convergiu para uma solução ótima.





Para os últimos 3 modelos eu resolvi alterar alguns parâmetros com o objetivo de simplificar a rede:

- Taxa de aprendizado: O valor da taxa de aprendizado padrão é muito baixo (0.001) fazendo com que demore para convergir, penso em deixa-lo 10x maior.
- Limite de iterações: Aumentar o limite máximo de iterações do modelo para 1000, assim tendo bastante margem para a Rede convergir para um bom resultado.
- Diminuir o número de neurônios da rede para deixá-la mais simples: Como visto em aula, um número maior de neurônios não significa resultados melhores, estarei treinando com apenas 1 camada e 10 neurônios nessa camada.
- De acordo com a documentação da biblioteca, usar o método `solver='lbfgs'` converge melhor a para dataset pequenos. Como nosso caso possui apenas 150 instâncias, estarei testando com ele.

Por fim executei o mesmo treinamento anterior, iterando entre as 3 funções de ativações e todas convergiram, assim consegui selecionar o melhor modelo: usando a função de ativação tangente hiperbólica o modelo tem uma acurácia de 96,67%.

4. Conclusão

Usando todas as features do dataset conseguirmos após experimentar diferentes parâmetros para a Rede Neural uma acurácia muito boa, maior que 96%. Vale ressaltar que dependendo de quais instâncias são divididas entre treinamento e teste junto com o sorteio dos grupos, o resultado varia entre 90% e 99% de acurácia, sendo 96,67% um valor 'moda' após rodar várias vezes o algoritmo.