



**MINISTÉRIO DA EDUCAÇÃO**  
**SECRETARIA DE EDUCAÇÃO PROFISSIONAL E**  
**TECNOLÓGICA**  
**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA**  
**E TECNOLOGIA GOIANO - CAMPUS URUTAÍ**  
**CURSO DE SISTEMAS DE INFORMAÇÃO**



**FARLEY RAMOS DE SENA**  
**HIULESSON RENAN ASSUNÇÃO BALBINO**

**RELATÓRIO PERCEPTRON MULTICAMADA**  
**AMOSTRA DE VINHOS**

Urutaí – GO  
Julho – 2022

## Classificar os vinhos usando a rede neural Perceptron Multicamadas

Utilizando Python e suas bibliotecas:

- Carregue e explore os dados da tabela **wine\_quality** que pode ser obtida na biblioteca *Scikit Learn* do Python. Nesse conjunto de dados, temos os resultados de uma análise química de vinhos cultivados na região da Itália, a tarefa é classificar o vinho em três tipos distintos (0,1 ou 2).

([https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load\\_wine.html#sklearn.datasets.load\\_wine](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html#sklearn.datasets.load_wine)).

- Separe os dados de treino e teste.
- Instancie o classificador MLP com as diferentes configurações de arquitetura/topologia e taxa de aprendizagem.
- Treine o algoritmo Perceptron Multicamadas (MLP)
- Opere a rede treinada para prever os valores para os dados de teste.
- Analise e valide os resultados obtidos mostrando: quantidade de épocas, o gráfico da curva do erro e a acurácia obtida para os dados de treinamento e dados de teste.

1) Utilize uma rede Perceptron Multicamadas com uma camada oculta com 2 e 10 neurônios, depois com duas camadas ocultas com (3, 3), (10, 3) e (10, 5) neurônios.

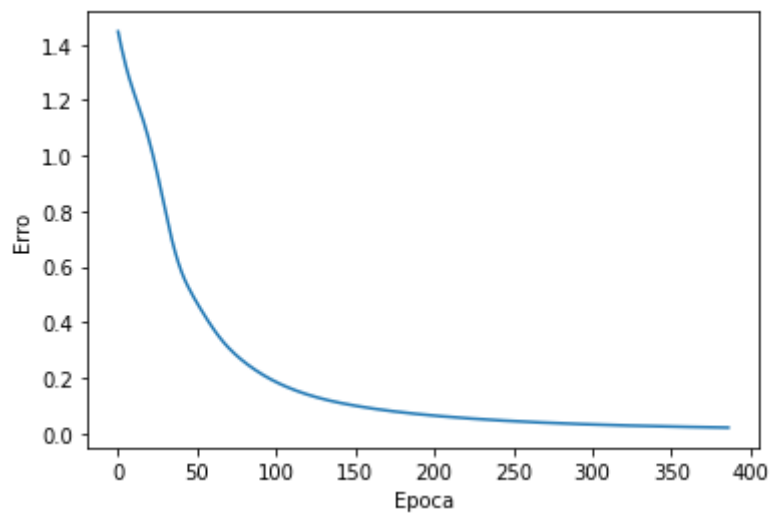
2) Para cada uma das arquiteturas/topologias de rede execute o treinamento para taxa de aprendizagem de 0,1 e 0,01.

3) Para os treinamentos realizados, trace os respectivos gráficos dos valores de erro em função de cada época de treinamento, apresente a quantidade de épocas e a acurácia obtida para os dados de treinamento e dados de teste.

4) Discorra sobre os resultados obtidos e a influência das diferentes configurações na acurácia e convergência da rede.

## TAXA DE APRENDIZAGEM DE 0,01

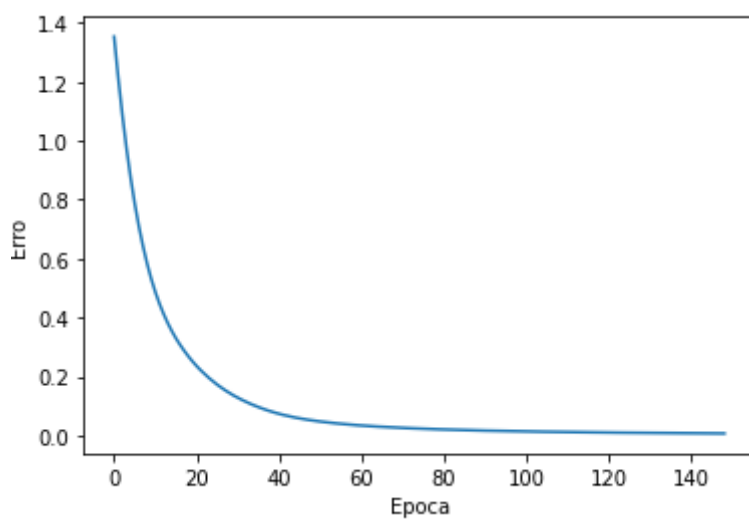
Uma camada oculta com 2 neurônios:



Épocas: 387

Acurácia: 1.0

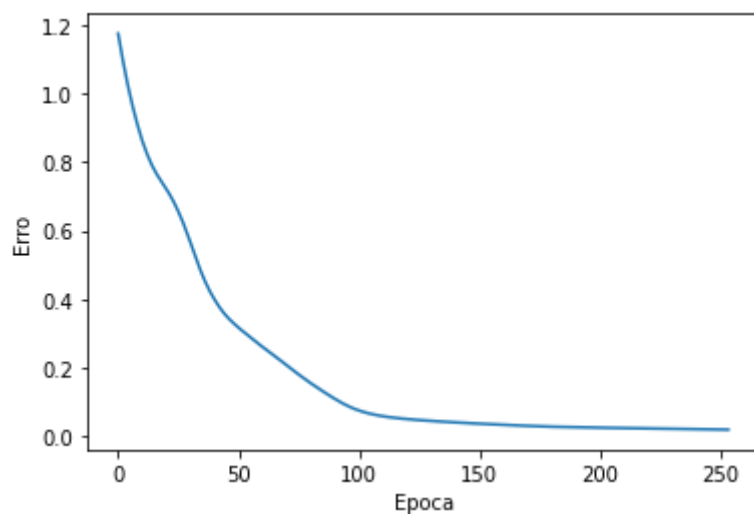
Uma camada oculta com 10 neurônios:



Épocas: 149

Acurácia: 1.0

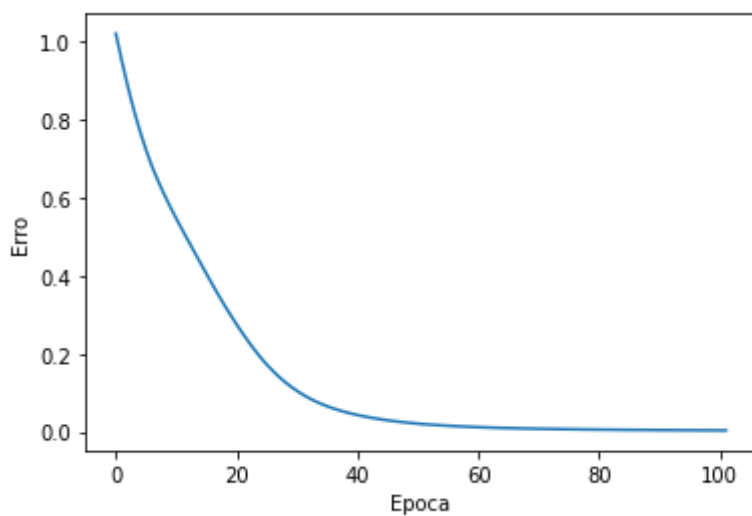
Duas camadas ocultas com (3, 3) neurônios:



Épocas: 254

Acurácia: 1.0

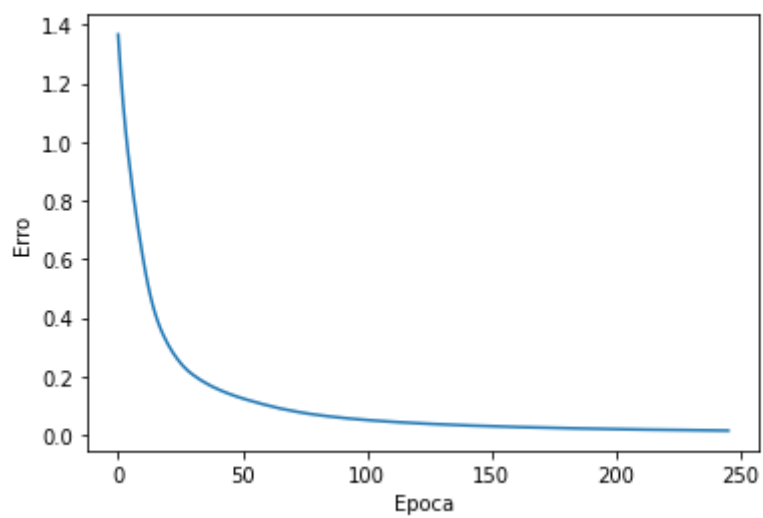
Duas camadas ocultas com (10, 3) neurônios:



Épocas: 102

Acurácia: 1.0

Duas camadas ocultas com (10, 5) neurônios:

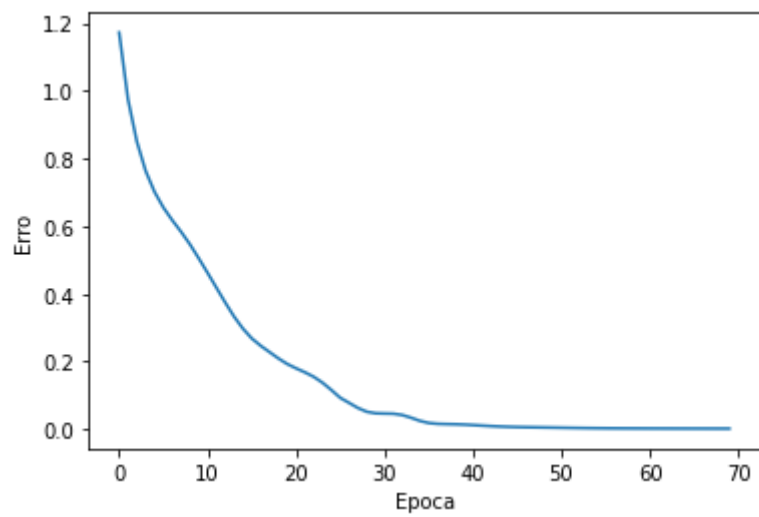


Épocas: 246

Acurácia: 1.0

TAXA DE APRENDIZAGEM DE 0,1

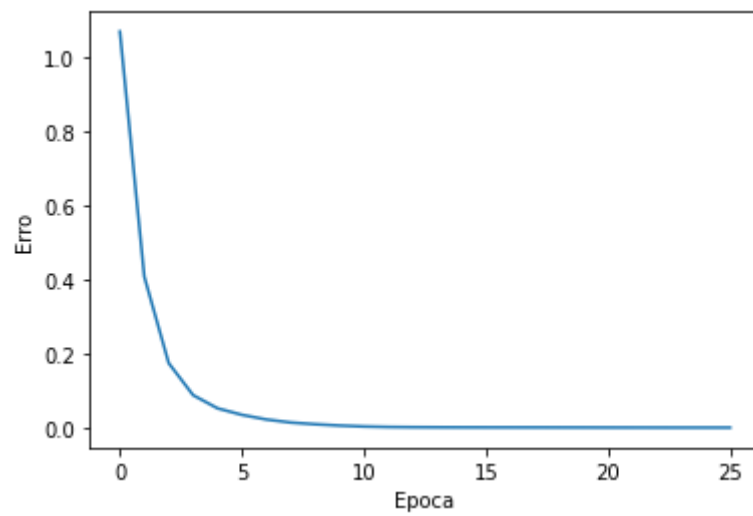
Uma camada oculta com 2 neurônios:



Épocas: 70

Acurácia: 1.0

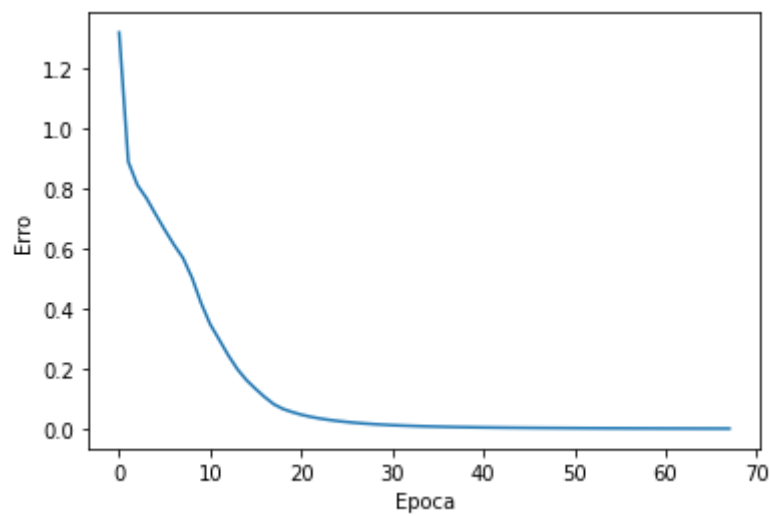
Uma camada oculta com 10 neurônios:



Épocas: 26

Acurácia: 1.0

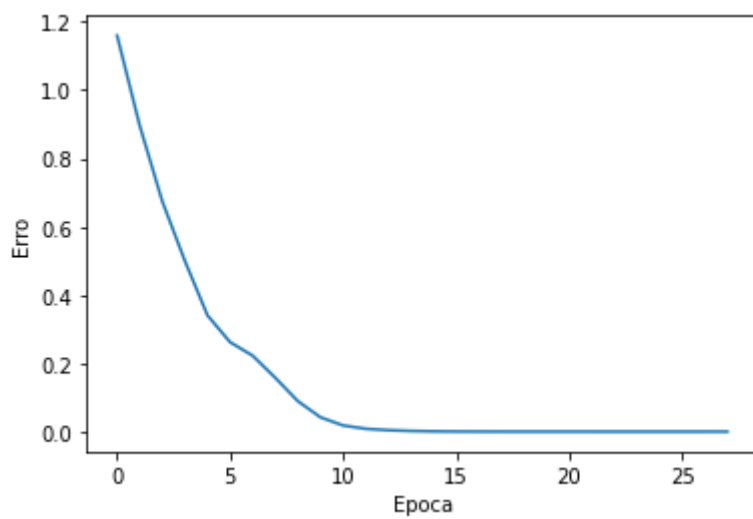
Duas camadas ocultas com (3, 3) neurônios:



Épocas: 68

Acurácia: 1.0

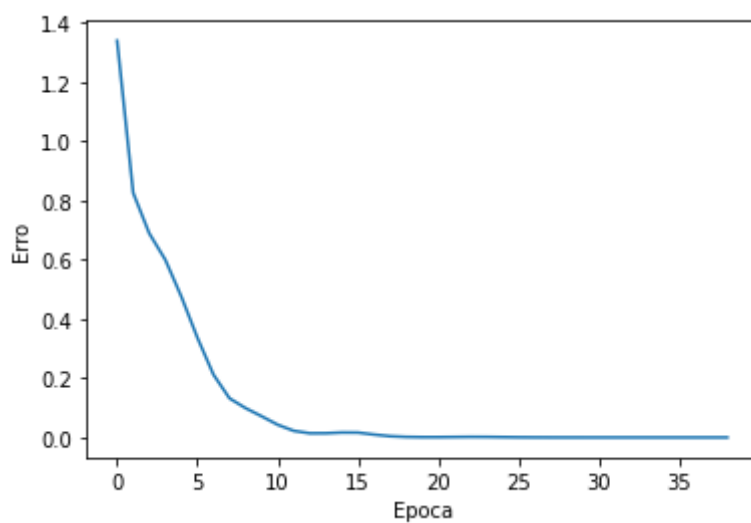
Duas camadas ocultas com (10, 3) neurônios:



Épocas: 28

Acurácia: 1.0

Duas camadas ocultas com (10, 5) neurônios:



Épocas: 39

Acurácia: 1.0

Com base nos testes realizados, podemos observar pelos gráficos a influência das camadas ocultas. Nos testes realizados com uma taxa de aprendizado 0,01 a quantidade de épocas comparada com os testes de taxa de aprendizado 0,1 é excepcionalmente notável.

Outro fator, seria a quantidade de camadas e neurônios que, dispõem da quantidade de épocas. Podemos exemplificar os testes com taxa de aprendizado 0,1, de uma camada de 10 em comparação com o de duas camadas de (3,3) que possui uma diferença de 43 épocas, sendo a de uma camada, mostrando sua superioridade sobre a de duas.