



Trabalho 10

Nome da Atividade: MLP para reconhecimento de flores

Nome e Matrícula: Lucas Lima do Nascimento - 12111ECP024

Introdução e Desenvolvimento:

Nesse trabalho, tentei explorar as diferenças entre um MLP com uma fração da base de dados sendo usada para validar e sem essa fração.

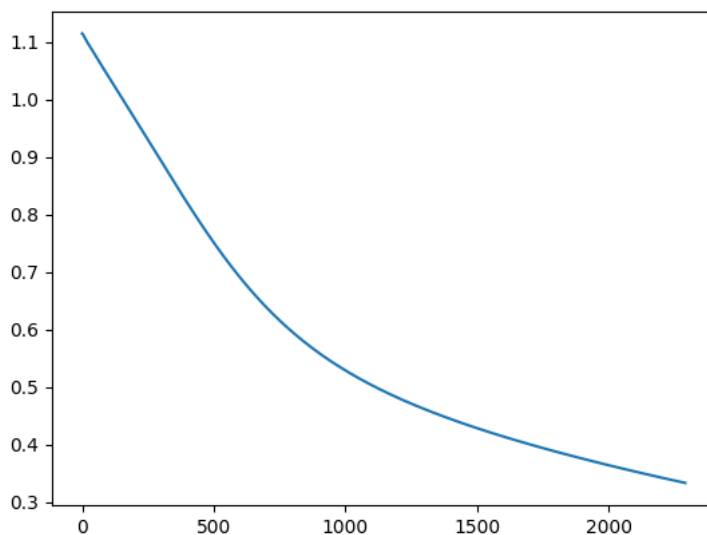
Para isso, utilizei o SKLearn para implementar o MLP com todos os ferramentais necessários para poder aprender.

Obtive resultados interessantes. Como a base de dados era muito pequena (apenas 150 informações), a rede neural que fracionava sua base de dados acaba perdendo muito poder de reconhecimento dos dados.

```
gitpod /workspace/ML/iris (master) $ python3 iris.py
Acerto do MLP sem separação da base de dados dos testes: 0.9798657718120806
Acerto do MLP com separação da base de dados dos testes: 0.33557046979865773
Digite o valor do comprimento da setala 5.0
Digite o valor da largura da setala 3.6
Digite o valor do comprimento da petala 1.4
Digite o valor da largura da petala 0.2
Previsão do MLP sem separação dos dados: Iris-setosa
Previsão do MLP com separação dos dados: Iris-virginica
```

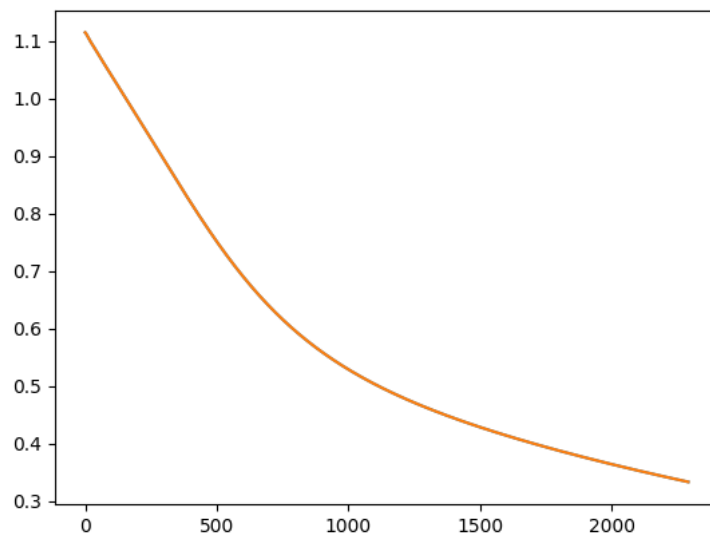
Também plotei as curvas relativas às perdas de cada um dos MLPs. Esses foram os gráficos obtidos:

Sem separação:



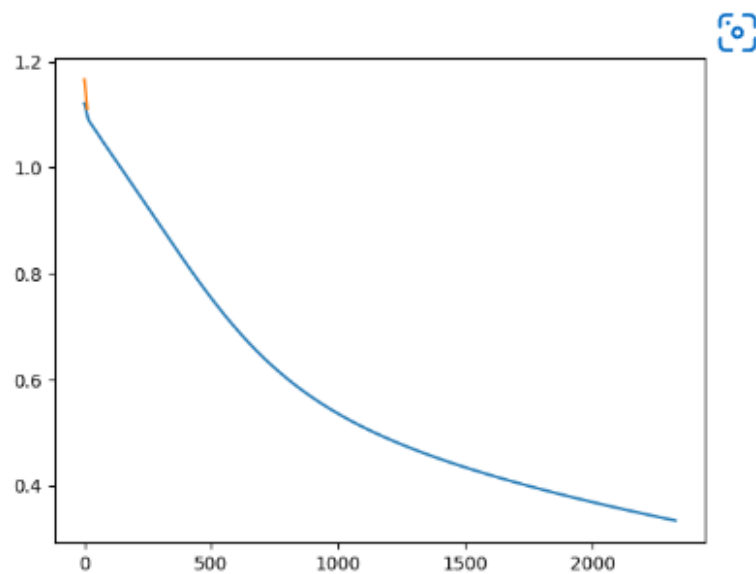
No eixo Y - Valor da perda, no eixo X - número de iterações

Com separação:



No eixo Y - Valor da perda, no eixo X - número de iterações

Sobrepostas:



No eixo Y - Valor da perda, no eixo X - número de iterações

Código:

```
from sklearn.neural_network import MLPClassifier
from pandas import read_excel as rx
from matplotlib import pyplot
import numpy as np

#Loading the data
dataset_init=rx('iris_data.xlsx',names=['Setal_length','Setal_width','Petal_length','Petal_width','Class'])
dataset=dataset_init.values
values=dataset[:,0:4] #Values
classes=dataset[:,4] #Classes
mlp=MLPClassifier(hidden_layer_sizes=(100,), activation='logistic', solver='sgd', alpha=0.00001, max_iter=3000)
```

```

mlp_with_separation_data=MLPClassifier(hidden_layer_sizes=(100,), early_stopping=True, validation_fraction=0.1, activation='logistic',

#Training
mlp.fit(values,classes)
mlp_with_separation_data.fit(values,classes)
print("Acerto do MLP sem separação da base de dados dos testes: ",mlp.score(values,classes))
print("Acerto do MLP com separação da base de dados dos testes: ",mlp_with_separation_data.score(values,classes))

#Plotting
pyplot.plot(mlp.loss_curve_)
pyplot.savefig('mlp_loss.png')

pyplot.plot(mlp_with_separation_data.loss_curve_)
pyplot.savefig('mlp_separation_loss.png')

#Guessing
setal_length = float(input("Digite o valor do comprimento da setala "))
setal_width = float(input("Digite o valor da largura da setala "))
petal_length = float(input("Digite o valor do comprimento da petala "))
petal_width = float(input("Digite o valor da largura da petala "))
new_flower = np.array([setal_length,setal_width,petal_length,petal_width]).reshape(1,-1)

pred_mlp = mlp.predict(new_flower)
pred_mlp_with_separation_data = mlp_with_separation_data.predict(new_flower)

print("Previsão do MLP sem separação dos dados: ",pred_mlp[0])
print("Previsão do MLP com separação dos dados: ",pred_mlp_with_separation_data[0])

```

Repositório com o código e imagens geradas:

ML/iris at master · LLxD/ML

Contribute to LLxD/ML development by creating an account on GitHub.



<https://github.com/LLxD/ML/tree/master/iris>

1
Contributor

0
Issues

0
Stars

0
Forks