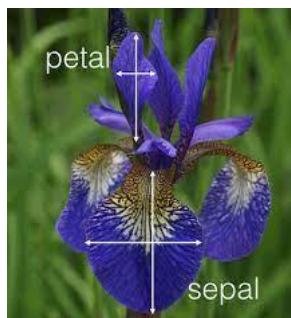
		Instituição:	Universidade positivo		Data:	09/10/2022
Curso:	Ciência da Computação			Disciplina:	Sistemas Inteligentes	
Professora:	Malgarete Rodrigues da Costa			Aluno(s):	Gabriela Cristina Schmitt Felipe dos Santos Fabiensi	

Acesse o código pelo link:

<https://colab.research.google.com/drive/1lGpIHB0BHFBofFKJVJOtcgfJYF2g0qYm?usp=sharing>

1. Introdução


Escolhemos o dataset IRIS comum para aprender sobre Machine Learning. Neste Dataset temos três espécies da Flor Iris, com 50 dados para cada. Dentre os dados apresentados temos os dados em centímetros da Sépala e Pétala de cada uma das espécies.



2. Definição do Tema

O dataset existe dentro da biblioteca sklearn e possui mais especificações através do link: https://en.wikipedia.org/wiki/Iris_flower_data_set

Também é possível obter a descrição do dataset com o seguinte comando:

 UNIVERSIDADE POSITIVO		Instituição:	Universidade positivo	Data:	09/10/2022
Curso:	Ciência da Computação		Disciplina:	Sistemas Inteligentes	
Professora:	Malgarete Rodrigues da Costa		Aluno(s):	Gabriela Cristina Schmitt Felipe dos Santos Fabiensi	

```

from sklearn.datasets import load_iris # Importando o dataset iris da biblioteca sklearn

iris = load_iris()

#DESCR retorna a descrição do dataset.
print(iris['DESCR'][:]) + "\n...")

.. _iris_dataset:

Iris plants dataset
-----

**Data Set Characteristics:**

: Number of Instances: 150 (50 in each of three classes)
: Number of Attributes: 4 numeric, predictive attributes and the class
: Attribute Information:
  - sepal length in cm
  - sepal width in cm
  - petal length in cm
  - petal width in cm
  - class:
    - Iris-Setosa
    - Iris-Versicolour
    - Iris-Virginica

: Summary Statistics:

=====
              Min   Max    Mean   SD   Class Correlation
=====
sepal length:  4.3   7.9   5.84   0.83    0.7826
sepal width:   2.0   4.4   3.05   0.43   -0.4194
petal length:   1.0   6.9   3.76   1.76    0.9490 (high!)
petal width:   0.1   2.5   1.20   0.76    0.9565 (high!)
=====

: Missing Attribute Values: None
: Class Distribution: 33.3% for each of 3 classes.

sepal length:  4.3   7.9   5.84   0.83    0.7826
sepal width:   2.0   4.4   3.05   0.43   -0.4194
petal length:   1.0   6.9   3.76   1.76    0.9490 (high!)
petal width:   0.1   2.5   1.20   0.76    0.9565 (high!)
=====

: Missing Attribute Values: None
: Class Distribution: 33.3% for each of 3 classes.
: Creator: R.A. Fisher
: Donor: Michael Marshall (MARSHALL@PLU@io.arc.nasa.gov)
: Date: July, 1988


The famous Iris database, first used by Sir R.A. Fisher. The dataset is taken
from Fisher's paper. Note that it's the same as in R, but not as in the UCI
Machine Learning Repository, which has two wrong data points.

This is perhaps the best known database to be found in the
pattern recognition literature. Fisher's paper is a classic in the field and
is referenced frequently to this day. (See Duda & Hart, for example.) The
data set contains 3 classes of 50 instances each, where each class refers to a
type of iris plant. One class is linearly separable from the other 2; the
latter are NOT linearly separable from each other.

.. topic:: References

- Fisher, R.A. "The use of multiple measurements in taxonomic problems"
  Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to
  Mathematical Statistics" (John Wiley, NY, 1950).
- Duda, R.O., & Hart, P.E. (1973) Pattern Classification and Scene Analysis.
  (Q327.D83) John Wiley & Sons. ISBN 0-471-22361-1. See page 218.
- Dasarthy, B.V. (1980) "Nosing Around the Neighborhood: A New System
  Structure and Classification Rule for Recognition in Partially Exposed
  Environments". IEEE Transactions on Pattern Analysis and Machine
  Intelligence, Vol. PAMI-2, No. 1, 67-71.
- Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule". IEEE Transactions
  on Information Theory, May 1972, 431-433.
- See also: 1988 MLC Proceedings, 54-64. Cheeseman et al's AUTOCLASS II
  conceptual clustering system finds 3 classes in the data.
- Many, many more ...

```

 UNIVERSIDADE POSITIVO		Instituição:	Universidade positivo	Data:	09/10/2022
Curso:	Ciência da Computação		Disciplina:	Sistemas Inteligentes	
Professora:	Malgarete Rodrigues da Costa		Aluno(s):	Gabriela Cristina Schmitt Felipe dos Santos Fabiensi	

3. Parametrização da Rede Neural

Para fazer uso do código utilizado em aula, diminuámos o dataset mantendo apenas as primeiras espécies. Desta forma nosso dataset será treinado com 100 dados, sendo 50 de cada flor.

Também não utilizamos as 4 características que o dataset oferece. Utilizamos apenas as características da Sépala.

```
iris = load_iris()

numEpocas = 700
numAmostras = 100

# atributos (sepal_lenght, sepal_width, petal_lenght, petal width) vamos usar os dois primeiros
features = iris.data.T

sepal_lenght = features[0, 0:100]
sepal_width = features[1, 0:100]

# saída esperada (class)
expected_output = iris['target'][0:100] # dividindo o dataset para trabalhar com apenas duas classificações, 0 = setosa and 1 = versicolor

#bias
bias = 1

# Perceptron

X = np.vstack((sepal_lenght, sepal_width)) #matriz de duas colunas
Y = np.vstack(expected_output) # saída ==ou seja== método supervisionado

# Taxa de aprendizado
eta = 0.1

# Definindo vetor de pesos inicializando com ZERO
W = np.zeros([1, 3]) # (uma linha e 3 colunas =) duas entradas + bias // uma camada com 3 nós


# Array de erros
e = np.zeros(100)

def funcaoAtivacao(valor):
    return 1 if valor > 0. else 0
```

4. Analise de treinamento

Ao utilizarmos as 100 entradas diferentes com 2 recursos por cada entrada (sepal_lenght e sepal_width) e 500 épocas o algoritmo ainda apresentava números diferentes de 0 no vetor de erros. Desta forma, fomos aumentando o número de épocas e com os novos testes podemos perceber que o modelo de aprendizagem tem um taxa de erros nula a partir de 700 épocas.

Na função de ativação utilizamos a variação 0 e 1.

 UNIVERSIDADE POSITIVO	Instituição:	Universidade positivo	Data:	09/10/2022
Curso:	Ciência da Computação	Disciplina:	Sistemas Inteligentes	
Professora:	Malgarete Rodrigues da Costa	Aluno(s):	Gabriela Cristina Schmitt Felipe dos Santos Fabiensi	

```
def funcaoAtivacao(valor):
    return 1 if valor > 0. else 0
```

Nosso algoritmo de treinamento:

```
#-----Treinamento-----
#-----

for j in range(numEpocas):
    for k in range(numAmostras):

        # insere bias no vetor
        Xb = np.hstack((bias, X[:,k])) #hstack empilha o bias em todas as linhas

        # Calcula o vetor de campo induzido (multiplicação vetorial // a matriz de entrada )
        V = np.dot(W, Xb)

        # Calcula a saída do Perceptron
        Yr = funcaoAtivacao(V)
        # Yr = np. tanh(V) # deve ser usado em camadas do meio e nunca de saída pois mascara a saída
        # Yr = np.sign(V)

        # Calcula o erro
        e[k] = Y[k] - Yr

        # Treinamento da rede neural
        W = W + eta*e[k]*Xb # peso + taxa de aprendizado * erro * entrada ajustada com o bias
        #print("Vetor de erros (e) = " + str(e))

# colocar dentro do for se quiser olhar cada época detalhadamente
print("Vetor de erros (e) = " + str(e))

# Plot sepal length against sepal width:
plt.scatter(sepal_length, sepal_width, c=iris.target[0:100])
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])

plt.show()

plt.plot(sepal_length, sepal_width, color = 'r')
plt.show()]
```

Plotamos dois gráficos diferentes, no primeiro há a separação das duas espécies por cor, dentro de uma escala de centímetros de Sépala. Já no segundo gráfico, esta sendo analisado o caminho percorrido pela inteligência artificial.

