Tarih:
11.06.2023

Yapay Sinir Ağı ile Çiçek Türü Sınıflandırması

Yapay Sinir Ağları (YSA), makine öğrenmesi alanında yaygın olarak kullanılan güçlü bir algoritmadır. Bu makalede, YSA kullanarak bir çiçek türü sınıflandırma problemi çözeceğiz. İlk adımda, veri kümesini inceleyeceğiz ve ardından veri önişleme adımlarını uygulayacağız. Daha sonra veriyi eğitim ve test kümelerine böleceğiz ve son olarak YSA modelini oluşturup eğiteceğiz.

1. Veri Kümesinin Adı ve Özellikleri

Veri kümesinin adı ve özellikleri: Veri kümesi 'iris.csv' olarak adlandırılır ve 150 satır ve 5 sütun içerir. Sütunlar sepal uzunluğu, sepal genişliği, petal uzunluğu, petal genişliği ve türdür. Türler setosa, versicolor ve virginica'dır. Veri kümesi, özelliklerine göre iris çiçeklerini sınıflandırmak için kullanılır.

2. Veri Önişleme

Veri önişleme adımları, veri kümesini modelimize uygun hale getirmek için kullanılır. İlk olarak, gerekli kütüphaneleri ve modülleri içe aktarıyoruz: pandas, numpy, matplotlib, seaborn, sklearn.preprocessing, sklearn.model_selection, keras.models ve keras.layers.

Veri kümesini "iris.csv" dosyasından okuyoruz ve bir DataFrame olan "df" değişkenine atıyoruz. Daha sonra, veri kümesinin şeklini ve bilgilerini yazdırarak veri hakkında önemli detaylara göz atıyoruz. Ayrıca, veri kümesinin betimsel istatistiklerini de görüntülüyoruz.

3. Veri Kümesinin Eğitim ve Test Olarak Bölünmesi

Veri kümesini eğitim ve test kümelerine bölmek için train_test_split fonksiyonunu kullanıyoruz. Veriyi %70 eğitim ve %30 test olacak şekilde ayırıyoruz. Bölme işleminden sonra, giriş ve çıkış değişkenlerini ayrı değişkenlere atıyoruz.

Tarih:
11.06.2023

4. Yapay Sinir Ağı Yapısı

Yapay sinir ağının yapısı: Yapay sinir ağı, keras modülünden sequential sınıfı kullanılarak oluşturulur. İki yoğun katmanı vardır: ilki 8 birim ve ReLU aktivasyon fonksiyonuna sahip olan, ikincisi ise 3 birim ve softmax aktivasyon fonksiyonuna sahip olan. İlk katmanın giriş boyutu giriş özelliklerinin sayısına (4) dayanır ve ikinci katmanın çıkış boyutu çıkış sınıflarının sayısına (3) dayanır.

5. Yapay Sinir Ağında Kullanılan Fonksiyonlar

a. Toplama Fonksiyonu:

Yapay sinir ağında, yoğun katmanlarda kullanılan toplama fonksiyonu, ağırlıklar ve girişler arasındaki çarpımların toplamını hesaplar.

$$z_j = \sum_{i=1}^{n} w_{ij} x_i + b_j$$

Burada zj birim j için toplama fonksiyonunun çıktısıdır, wij önceki katmandaki birim i ile mevcut katmandaki birim j arasındaki ağırlıktır, xi önceki katmandaki birim i den giriştir, bj mevcut katmandaki birim j için yanlılıktır ve n önceki katmandaki birimlerin sayısıdır.

b. Gizli Katman Aktivasyon Fonksiyonu:

Bu fonksiyon, bir katmandaki her birimin toplama fonksiyonunun çıktısına doğrusal olmayan bir dönüşüm uygular. Bir birimin ne kadar aktive edildiğini veya ateşlendiğini belirler. Bu koddaki gizli katman için ReLU (düzeltilmiş doğrusal ünite) aktivasyon fonksiyonu kullanılır. Şöyle verilir:

$$a_j = \max(0, z_j)$$

Burada aj birim j için aktivasyon fonksiyonunun çıktısıdır ve zj birim j için toplama fonksiyonunun çıktısıdır.

Tarih:
11.06.2023

c. Çıkış Katmanı Aktivasyon Fonksiyonu:

Çıkış katmanı için kullanılan aktivasyon fonksiyonu: Bu fonksiyon, bir katmandaki her birimin toplama fonksiyonunun çıktısına doğrusal olmayan bir dönüşüm uygular. Bir birimin nihai tahmine ne kadar katkıda bulunduğunu belirler. Bu koddaki çıkış katmanı için softmax aktivasyon fonksiyonu kullanılır. Şöyle verilir:

$$a_j = \frac{e^{z_j}}{\sum_{k=1}^{m} e^{z_k}}$$

Burada aj birim j için aktivasyon fonksiyonunun çıktısıdır, zj birim j için toplama fonksiyonunun çıktısıdır ve m çıkış katmanındaki birimlerin sayısıdır.

6. Öğrenme Algoritması

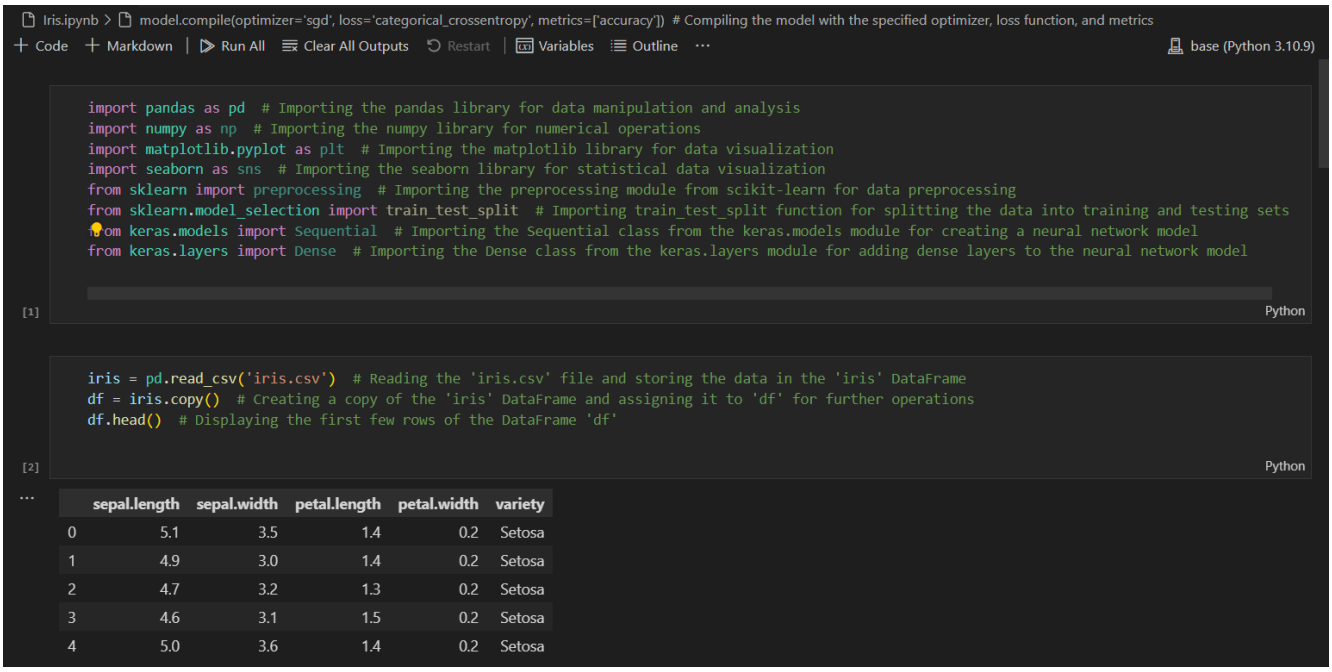
Öğrenme algoritması (pseudo code): Öğrenme algoritması, ağın eğitim verilerinde ne kadar iyi performans gösterdiğine bağlı olarak ağın ağırlıklarını ve yanlılıklarını günceller. Bu koddaki stokastik gradyan inişi (SGD) optimizasyonu öğrenme algoritması olarak kullanılır. Şöyle verilir:

- Ağırlıkları ve yanlılıkları rastgele başlat
- Her bir dönem için:
  - Eğitim verilerini karıştır
  - Her bir küme için:
    - Girişleri ağdan ileri yayınla ve çıktıları hesapla
    - Çıktılara ve gerçek etiketlere dayalı kayıp fonksiyonunu (kategorik çapraz entropi) hesapla
    - Hataları ağdan geri yayınla ve gradyanları hesapla
    - Ağırlıkları ve yanlılıkları gradyanlar ve öğrenme oranı kullanarak güncelle
- Son ağırlıkları ve yanlılıkları döndür

Tarih:
11.06.2023

7. Seçilen Veri Kümesi için Yapay Sinir Ağı Kodu

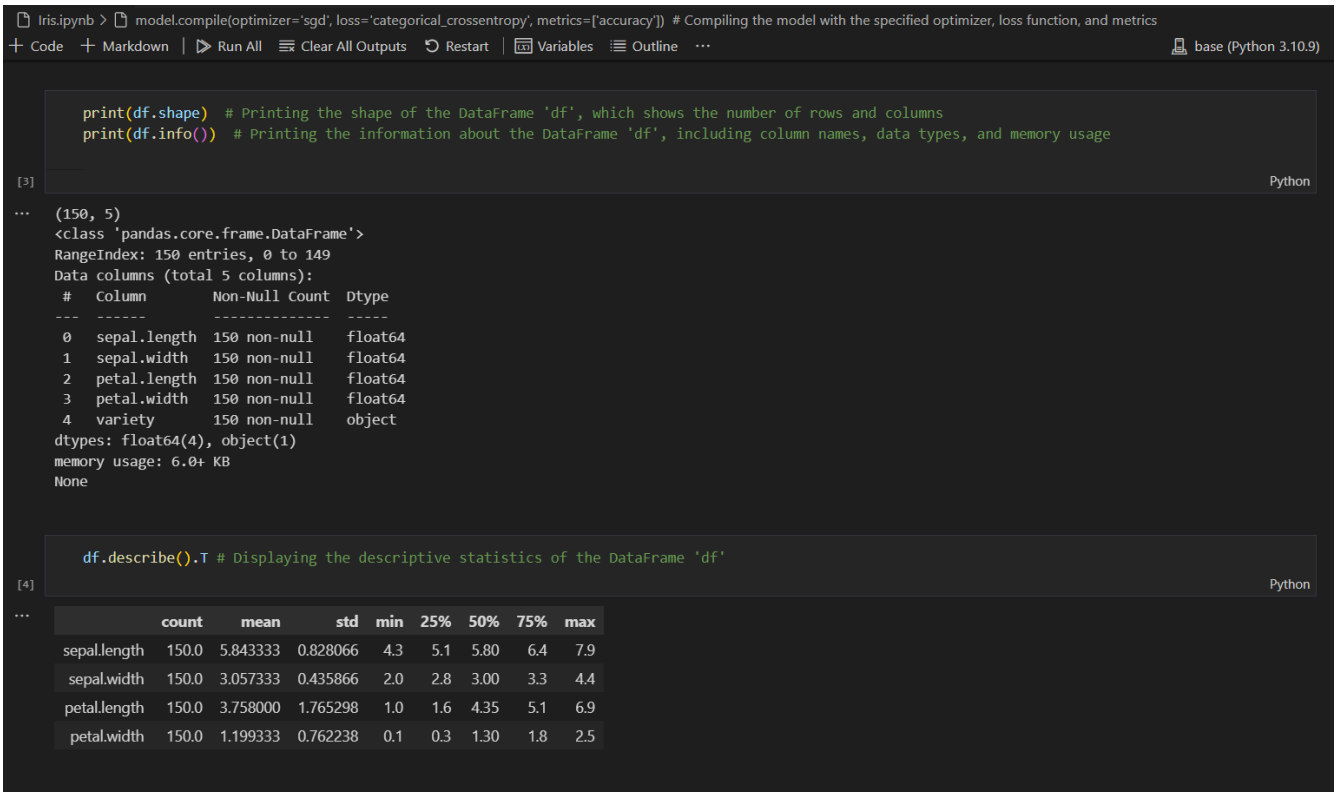
Yukarıdaki adımları takip ederek oluşturduğumuz yapay sinir ağı modelinin Python kodu aşağıdaki gibidir:



```
import pandas as pd  # Importing the pandas library for data manipulation and analysis
import numpy as np  # Importing the numpy library for numerical operations
import matplotlib.pyplot as plt  # Importing the matplotlib library for data visualization
import seaborn as sns  # Importing the seaborn library for statistical data visualization
from sklearn import preprocessing  # Importing the preprocessing module from scikit-learn for data preprocessing
from sklearn.model_selection import train_test_split  # Importing train_test_split function for splitting the data into training and testing sets
from keras.models import Sequential  # Importing the Sequential class from the keras.models module for creating a neural network model
from keras.layers import Dense  # Importing the Dense class from the keras.layers module for adding dense layers to the neural network model
```

```
iris = pd.read_csv('iris.csv')  # Reading the 'iris.csv' file and storing the data in the 'iris' DataFrame
df = iris.copy()  # Creating a copy of the 'iris' DataFrame and assigning it to 'df' for further operations
df.head()  # Displaying the first few rows of the DataFrame 'df'
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa



```
print(df.shape)  # Printing the shape of the DataFrame 'df', which shows the number of rows and columns
print(df.info())  # Printing the information about the DataFrame 'df', including column names, data types, and memory usage
```

```
(150, 5)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal.length  150 non-null    float64
 1   sepal.width   150 non-null    float64
 2   petal.length  150 non-null    float64
 3   petal.width   150 non-null    float64
 4   variety       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
```

```
df.describe().T # Displaying the descriptive statistics of the DataFrame 'df'
```

	count	mean	std	min	25%	50%	75%	max
sepal.length	150.0	5.843333	0.828066	4.3	5.1	5.80	6.4	7.9
sepal.width	150.0	3.057333	0.435866	2.0	2.8	3.00	3.3	4.4
petal.length	150.0	3.758000	1.765298	1.0	1.6	4.35	5.1	6.9
petal.width	150.0	1.199333	0.762238	0.1	0.3	1.30	1.8	2.5

Tarih:
11.06.2023

Iris.ipynb > model.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy']) # Compiling the model with the specified optimizer, loss function, and metrics

+ Code + Markdown | Run All | Clear All Outputs | Restart | Variables | Outline ... base (Python 3.10.9)

```
# Splitting the input and output variables
x = df.iloc[:, 0:4]  # Selecting all rows and the first 4 columns as the input variables and assigning it to 'x'
y = df.iloc[:, 4:5]  # Selecting all rows and the 5th column as the output variable and assigning it to 'y'
```

[5] Python

```
x.head() # Displaying the first few rows of the input variables
```

[6] Python

	sepal.length	sepal.width	petal.length	petal.width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

```
y.head() # Displaying the first few rows of the output variable
```

[7] Python

	variety
0	Setosa
1	Setosa
2	Setosa
3	Setosa
4	Setosa

Iris.ipynb > model.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy']) # Compiling the model with the specified optimizer, loss function, and metrics

+ Code + Markdown | Run All | Clear All Outputs | Restart | Variables | Outline ... base (Python 3.10.9)

```
encoded_y = y.apply(preprocessing.LabelEncoder().fit_transform)  # Applying label encoding to the output variable 'y' using the LabelEncoder from th
ohenc = preprocessing.OneHotEncoder()  # Creating an instance of the OneHotEncoder from the preprocessing module
y = ohenc.fit_transform(encoded_y).toarray()  # Applying one-hot encoding to the encoded output variable 'encoded_y' and converting it to a NumPy ar
```

[8] Python

```
# Creating a new DataFrame 'y' with categorical labels
# setosa = [1, 0, 0], versicolor = [0, 1, 0], virginica = [0, 0, 1]
y = pd.DataFrame(y, columns=['setosa', 'versicolor', 'virginica'])
```

[9] Python

```
# Using StandardScaler for feature scaling
scaler = preprocessing.StandardScaler()  # Creating an instance of the StandardScaler from the preprocessing module
x = scaler.fit_transform(x)  # Scaling the input variables 'x' using the fit_transform() method of the scaler
```

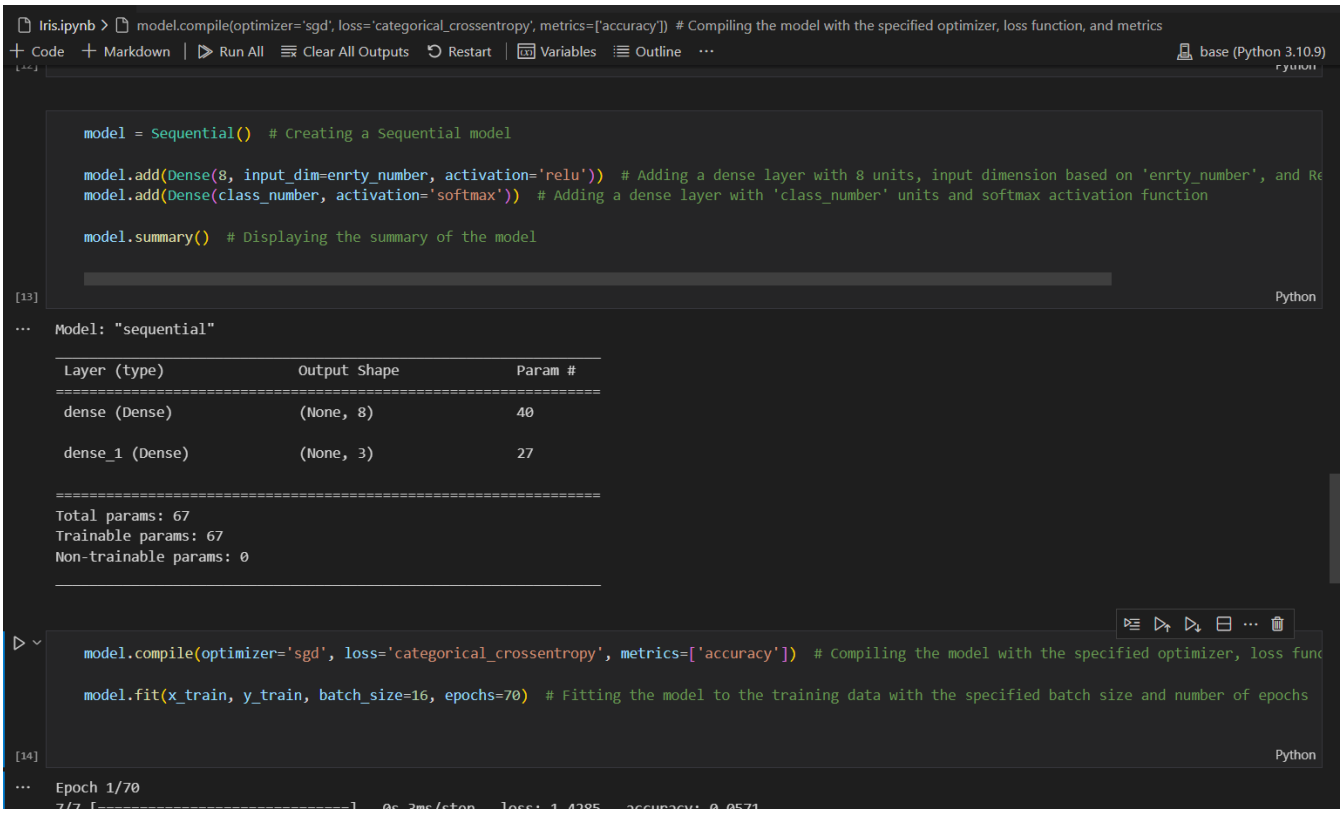
[10] Python

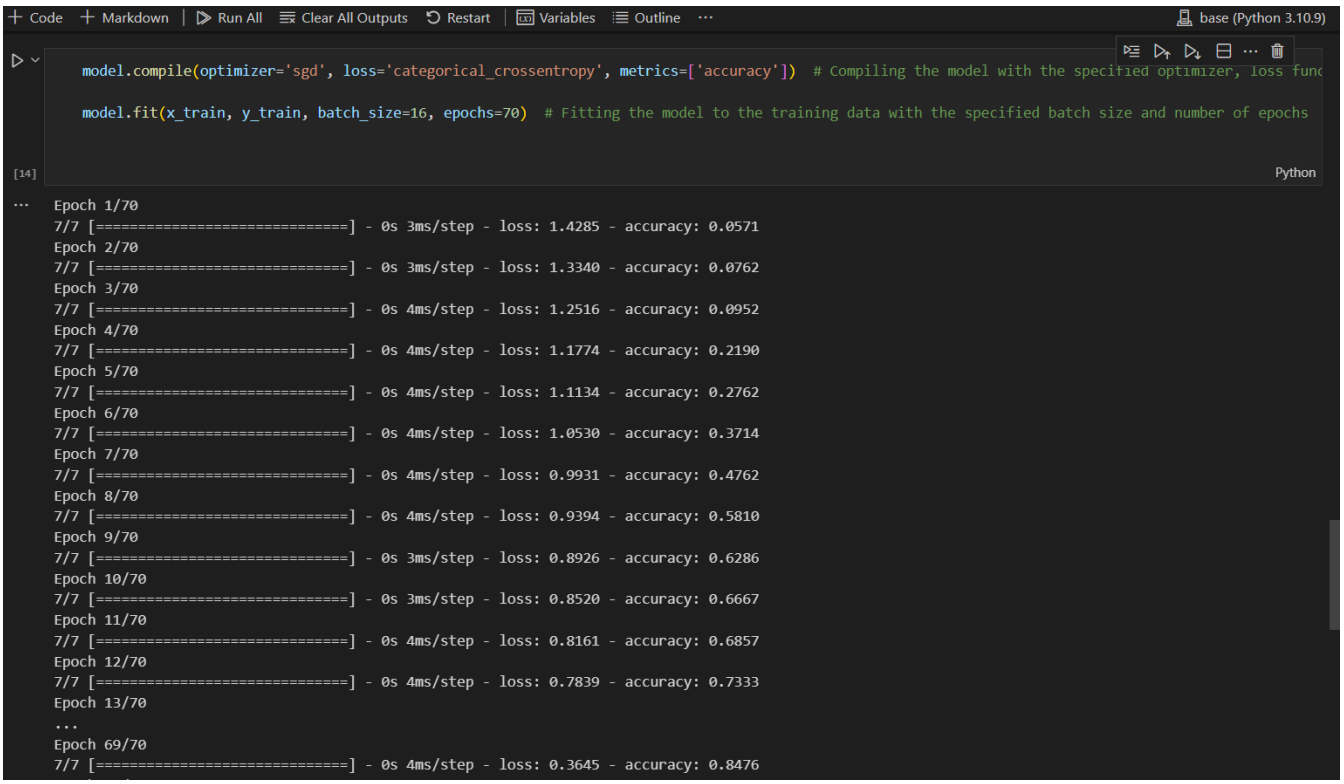
```
# Splitting the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42)
```

[11] Python

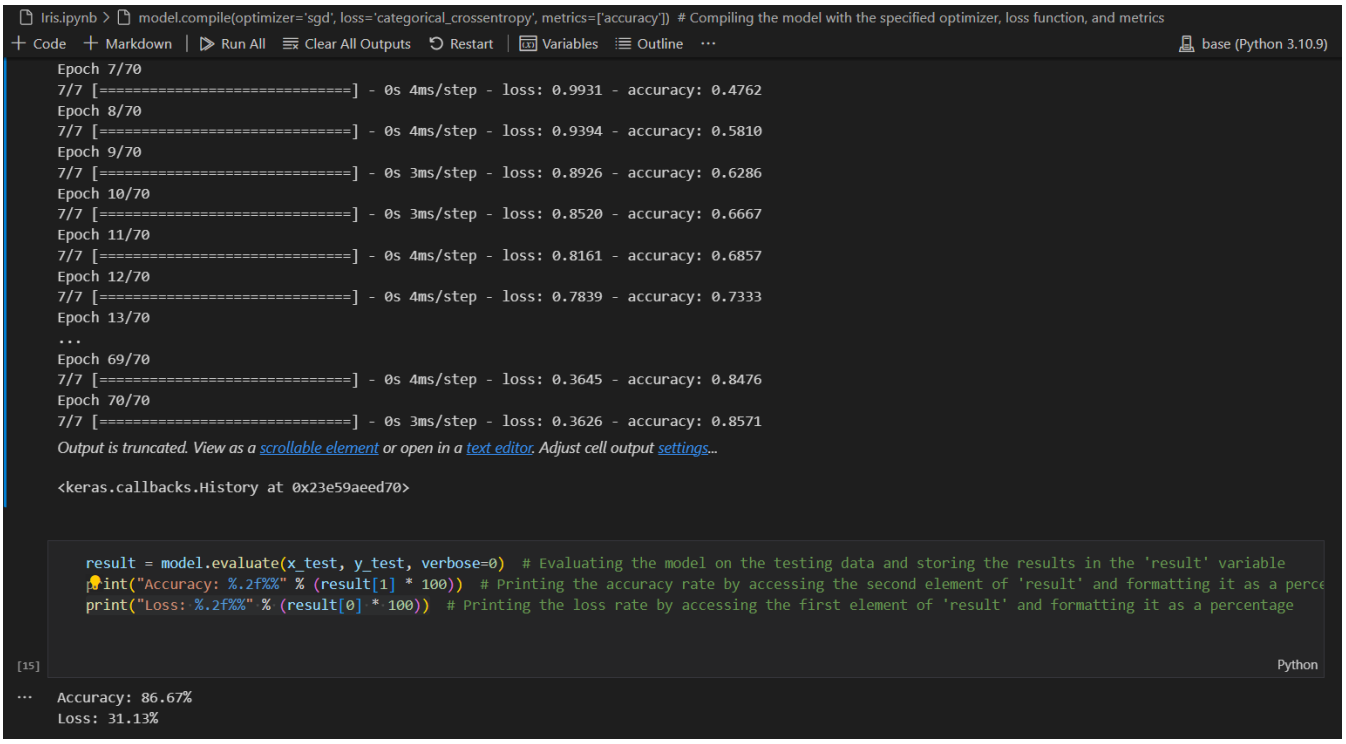
```
enrty_number = x_train.shape[1]  # Storing the number of input features in the variable 'enrty_number'
class_number = y_train.shape[1]  # Storing the number of output classes in the variable 'class_number'
```

Tarih:
11.06.2023





Tarih:
11.06.2023



Bu kod, veri kümesini okur, veri önişleme adımlarını uygular, veriyi eğitim ve test kümelerine böler, yapay sinir ağı modelini oluşturur ve eğitir, son olarak modelin performansını değerlendirir ve sonuçları yazdırır.

Bu makalede, yapay sinir ağları kullanarak çiçek türü sınıflandırması gerçekleştirmeyi öğrendik. Yapay sinir ağları, geniş bir uygulama yelpazesine sahip olduğundan, farklı problemleri çözmek için kullanılabilirler.

GİTHUB LİNK: https://github.com/Akcanbasri/yapay-sinir-aglarina-sinir-final-odevi

Hazırlayan: Hasan Bari Akcan

KTO Karatay Üniversitesi Bilgisayar Mühendisliği

Öğretmen adı: Muhammed Karaaltun