

August 15, 2023

# 1 Báo cáo: Phân loại iris bằng mạng Perceptron đơn tầng

- Họ tên: Kim Minh Thắng
- Mã số sinh viên: B2007210

## 1.1 Bài 1

### 1.1.1 Yêu cầu

Xây dựng mô hình: - 1 Tầng (Dense) gồm 1 neuron - input\_shape: (4,) - use\_bias: True - activation: sigmoid

Biên dịch mô hình: - optimizer = adam - loss: binary\_crossentropy - metrics: ['accuracy']

### 1.1.2 Bài làm

**Đọc và xử lý dữ liệu** Load dữ liệu từ file Iris.csv và giữ lại 100 dòng đầu tiên.

```
[ ]: import pandas as pd
import numpy as np
```

```
[ ]: df = pd.read_csv("./Iris.csv")
```

```
[ ]: df = df.head(100)
df
```

```
[ ]:
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
0      1             5.1             3.5             1.4             0.2
1      2             4.9             3.0             1.4             0.2
2      3             4.7             3.2             1.3             0.2
3      4             4.6             3.1             1.5             0.2
4      5             5.0             3.6             1.4             0.2
...    ...             ...             ...             ...             ...
95     96             5.7             3.0             4.2             1.2
96     97             5.7             2.9             4.2             1.3
97     98             6.2             2.9             4.3             1.3
98     99             5.1             2.5             3.0             1.1
99    100             5.7             2.8             4.1             1.3
```

Species

```

0      Iris-setosa
1      Iris-setosa
2      Iris-setosa
3      Iris-setosa
4      Iris-setosa
..      ""
95     Iris-versicolor
96     Iris-versicolor
97     Iris-versicolor
98     Iris-versicolor
99     Iris-versicolor

[100 rows x 6 columns]

```

Tách các cột đặc trưng và nhãn

```
[ ]: X = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
X = np.asarray(X)
```

Chuyển nhãn về dạng số

```
[ ]: d = dict()
d['Iris-setosa'] = 0
d['Iris-versicolor'] = 1

Y = df['Species']
Y = [d[y] for y in Y]
Y = np.asarray(Y)
```

Tách dataset thành dữ liệu train và test với tỉ lệ 80:20

```
[ ]: from sklearn.model_selection import train_test_split
```

```
[ ]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
```

Tạo model và biên dịch

```
[ ]: from keras.models import Sequential
from keras.layers import Dense

model = Sequential()
model.add(Dense(1, input_shape=(4,), activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy',
              metrics=['accuracy'])
```

Train model

```
[ ]: his = model.fit(X_train, Y_train, epochs=150, batch_size=20,
                    validation_data=(X_test, Y_test), verbose=0)
```

### Đánh giá model

```
[ ]: score = model.evaluate(X_test, Y_test, verbose=0)
```

```
[ ]: print('Loss: ', score[0])  
     print('Accuracy: ', score[1])
```

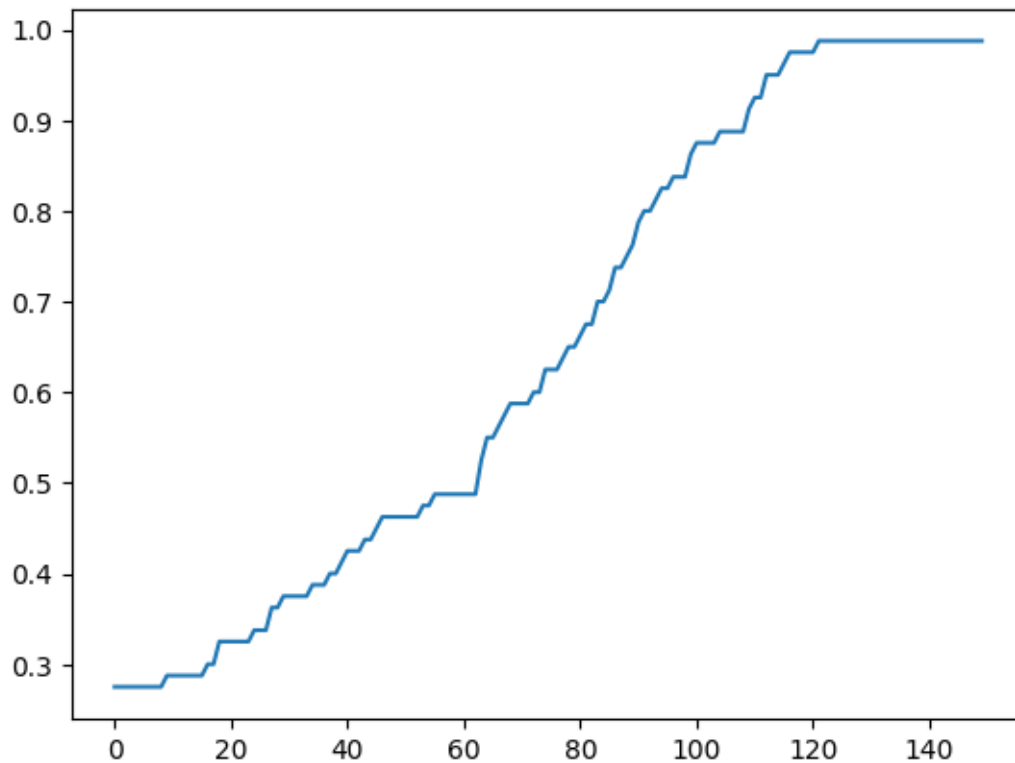
Loss: 0.4364587664604187

Accuracy: 1.0

```
[ ]: import matplotlib.pyplot as plt
```

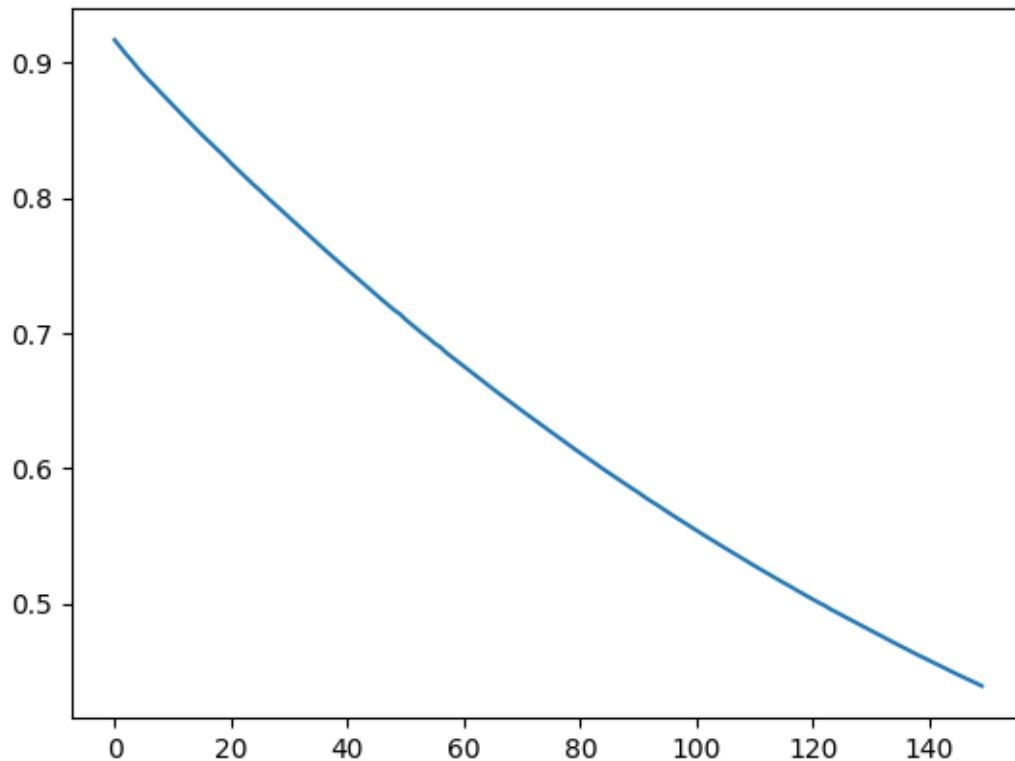
```
[ ]: plt.plot(hist.history['accuracy'])
```

```
[ ]: [ <matplotlib.lines.Line2D at 0x7f83398a6fb0>]
```



```
[ ]: plt.plot(hist.history['loss'])
```

```
[ ]: [ <matplotlib.lines.Line2D at 0x7f8339f41c30>]
```



## 1.2 Bài 2

### 1.2.1 Yêu cầu

Xây dựng mô hình: - 1 Tầng (Dense) gồm 3 neuron - input\_shape: (4,) - use\_bias: True - activation: softmax

Biên dịch mô hình: - optimizer = adam - loss: square\_categorical\_crossentropy - metrics: ['accuracy']

### 1.2.2 Bài làm

**Đọc và xử lý dữ liệu** Load dữ liệu từ file Iris.csv

```
[ ]: import pandas as pd
import numpy as np
```

```
[ ]: df = pd.read_csv("./Iris.csv")
df
```

```
[ ]:      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
0      1         5.1         3.5         1.4         0.2
1      2         4.9         3.0         1.4         0.2
2      3         4.7         3.2         1.3         0.2
```

3	4	4.6	3.1	1.5	0.2
4	5	5.0	3.6	1.4	0.2
..	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3
146	147	6.3	2.5	5.0	1.9
147	148	6.5	3.0	5.2	2.0
148	149	6.2	3.4	5.4	2.3
149	150	5.9	3.0	5.1	1.8

	Species
0	Iris-setosa
1	Iris-setosa
2	Iris-setosa
3	Iris-setosa
4	Iris-setosa
..	...
145	Iris-virginica
146	Iris-virginica
147	Iris-virginica
148	Iris-virginica
149	Iris-virginica

[150 rows x 6 columns]

Tách các cột đặc trưng và nhãn

```
[ ]: X = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
X = np.asarray(X)
```

```
[ ]: d = dict()
d['Iris-setosa'] = 0
d['Iris-versicolor'] = 1
d['Iris-virginica'] = 2

Y = df['Species']
Y = [d[y] for y in Y]
Y = np.asarray(Y)
```

Tách dataset thành dữ liệu train và test với tỉ lệ 80:20

```
[ ]: from sklearn.model_selection import train_test_split
```

```
[ ]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
```

Tạo model và biên dịch

```
[ ]: from keras.models import Sequential
from keras.layers import Dense
```

```
model = Sequential()
model.add(Dense(3, input_shape=(4,), activation='softmax'))
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

### Train model

```
[ ]: his = model.fit(X_train, Y_train, epochs=150, batch_size=20,
                    validation_data=(X_test, Y_test), verbose=0)
```

### Đánh giá model

```
[ ]: score = model.evaluate(X_test, Y_test, verbose=0)
```

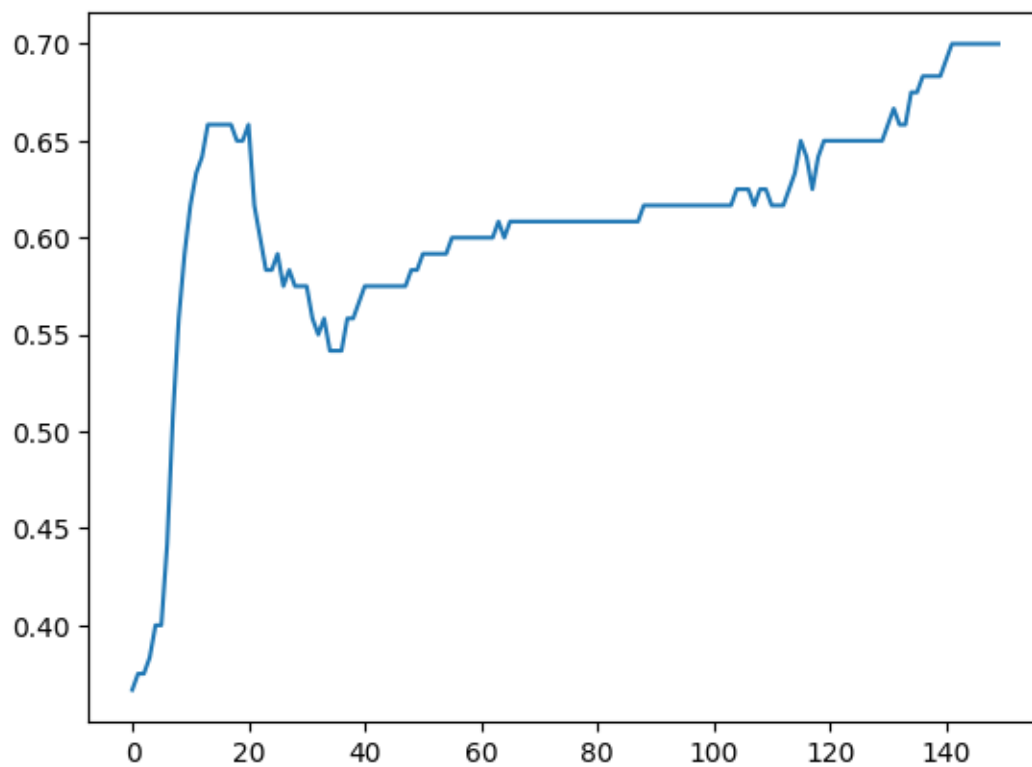
```
[ ]: print('Loss: ', score[0])
     print('Accuracy: ', score[1])
```

```
Loss:  0.4355047643184662
Accuracy:  0.7666666507720947
```

```
[ ]: import matplotlib.pyplot as plt
```

```
[ ]: plt.plot(his.history['accuracy'])
```

```
[ ]: [ <matplotlib.lines.Line2D at 0x7f8339ab3a90>]
```



```
[ ]: plt.plot(his.history['loss'])
```

```
[ ]: [ <matplotlib.lines.Line2D at 0x7f8339a6c7f0>]
```

