

```
In [72]: import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
from sklearn.preprocessing import LabelEncoder
import warnings
warnings.filterwarnings('ignore')
from sklearn.metrics import classification_report, confusion_matrix
```

```
In [73]: df = pd.read_csv('iris.csv')
df.head()
```

Out[73]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
In [74]: df.shape
```

Out[74]: (150, 5)

```
In [75]: X = df[['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width']]
y = df[['Species']]

encoder = LabelEncoder()
y1 = encoder.fit_transform(y)
Y = pd.get_dummies(y1).values
```

```
In [76]: X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_
```

```
In [77]: print(X_train.shape, y_train.shape, X_test.shape, y_test.shape)

(120, 4) (120, 3) (30, 4) (30, 3)
```

```
In [78]: model = Sequential()

model.add(Dense(4,input_shape=(4,),activation='relu'))
#model.add(Dense(8,activation='tanh'))
#model.add(Dense(6,activation='tanh'))
model.add(Dense(3,activation='softmax'))

model.compile(Adam(lr=0.01),'categorical_crossentropy',metrics=['accuracy'])

model.summary()
```

Model: "sequential_11"

Layer (type)	Output Shape	Param #
dense_13 (Dense)	(None, 4)	20
dense_14 (Dense)	(None, 3)	15
Total params: 35		
Trainable params: 35		
Non-trainable params: 0		

```
In [79]: model.fit(X_train,y_train,epochs=100)
uracy: 0.4917
Epoch 17/100
4/4 [=====] - 0s 866us/step - loss: 1.0255 - a
ccuracy: 0.5500
Epoch 18/100
4/4 [=====] - 0s 1ms/step - loss: 1.0047 - acc
uracy: 0.6167
Epoch 19/100
4/4 [=====] - 0s 800us/step - loss: 0.9826 - a
ccuracy: 0.6250
Epoch 20/100
4/4 [=====] - 0s 1ms/step - loss: 0.9578 - acc
uracy: 0.5917
Epoch 21/100
4/4 [=====] - 0s 783us/step - loss: 0.9282 - a
ccuracy: 0.5333
Epoch 22/100
4/4 [=====] - 0s 1ms/step - loss: 0.8983 - acc
uracy: 0.4583
Epoch 23/100
```

```
In [80]: y_pred = model.predict(X_test)
y_test_class = np.argmax(y_test,axis=1)
y_pred_class = np.argmax(y_pred,axis=1)
```

```
In [81]: print(classification_report(y_test_class,y_pred_class))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	1.00	0.92	0.96	13
2	0.86	1.00	0.92	6
accuracy			0.97	30
macro avg	0.95	0.97	0.96	30
weighted avg	0.97	0.97	0.97	30

```
In [82]: print(confusion_matrix(y_test_class,y_pred_class))
```

```
[[11  0  0]
 [ 0 12  1]
 [ 0  0  6]]
```

```
In [83]: acc = accuracy_score(y_true = y_test_class,y_pred =y_pred_class)*100
print("Accuracy: ", acc, "%")
```

```
Accuracy:  96.66666666666667 %
```

```
In [ ]:
```

```
In [84]: # colsX = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width']
# colsy = ['Species']
# X = df[colsX]
# y = df[colsy]
```

```
In [85]: # train_x, test_x, train_y, test_y = train_test_split(X, y, test_size=0.20)
```

```
In [38]: # model = Sequential()
# model.add(Dense(10, input_dim=4, activation='relu', name='fc1'))
# model.add(Dense(10, activation='relu', name='fc2'))
# model.add(Dense(3, activation='softmax', name='output'))
```

```
In [39]: # optimizer = Adam(lr=0.001)
# model.compile(optimizer, loss='categorical_crossentropy', metrics=['accur
```

```
In [40]: # print(model.summary())
```

```
In [41]: # model.fit(train_x, train_y, batch_size=5, epochs=200)
```

```
In [ ]:
```

