```
import tensorflow as tf
from tensorflow.keras import layers
import pandas as pd
import numpy as np
from tensorflow.keras import datasets, layers, models
from tensorflow.keras.utils import to_categorical
from sklearn.datasets import load_iris
```

```
dataset=load_iris()
```

```
df=pd.DataFrame(dataset['data'],columns=dataset['feature_names'])
df['Species']=dataset['target']
df['Species']=df['Species'].apply(lambda x:dataset['target_names'][x])
df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Species
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

```
X = df.iloc[:,0:3].values
y = df.iloc[:,4:].values
```

```
print(X[0:5])
print(y[0:5])
```

```
[[5.1 3.5 1.4]
 [4.9 3.  1.4]
 [4.7 3.2 1.3]
 [4.6 3.1 1.5]
 [5.  3.6 1.4]]
[['setosa']
 ['setosa']
```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and reopen the link.

```
 ['setosa']]
```

```
print(X.shape)
print(y.shape)
```

```
(150, 3)
(150, 1)
```

```
from sklearn.preprocessing import LabelEncoder
encoder =  LabelEncoder()
y1 = encoder.fit_transform(y)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_label.py:115: DataConversionWarning: A column-vector y wa
  y = column_or_1d(y, warn=True)
```

```
print(y1)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
```

```
Y = pd.get_dummies(y1).values
print(Y[0:5])
```

```
[[1 0 0]
 [1 0 0]
 [1 0 0]
 [1 0 0]
 [1 0 0]]
```

```
from sklearn.model_selection import train_test_split
```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and reopen the link. ×

```
print(X_train[0:5])
```

```
[[6.4 3.1 5.5]
 [5.4 3.  4.5]
 [5.2 3.5 1.5]
 [6.1 3.  4.9]
 [6.4 2.8 5.6]]
```

```
print(y_train[0:5])
```

```
[[0 0 1]
 [0 1 0]
 [1 0 0]
 [0 0 1]
 [0 0 1]]
```

```
print(X_test[0:5])
```

```
[[5.8 2.8 5.1]
 [6.  2.2 4. ]
 [5.5 4.2 1.4]
 [7.3 2.9 6.3]
 [5.  3.4 1.5]]
```

```
print(y_test[0:5])
```

```
[[0 0 1]
 [0 1 0]
 [1 0 0]
 [0 0 1]
 [1 0 0]]
```

```
model = tf.keras.Sequential([
    tf.keras.layers.Dense(10, activation='relu'),
    tf.keras.layers.Dense(10, activation='relu'),
```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and reopen the link.

```
model
```

```
     <keras.engine.sequential.Sequential at 0x7f3815c66250>
```

```
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

```
model.fit(X_train, y_train, batch_size=50, epochs=100)
```

```
     3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667
     Epoch 72/100
     3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667
     Epoch 73/100
     3/3 [==============================] - 0s 6ms/step - loss: 5.5070 - accuracy: 0.3667
     Epoch 74/100
     3/3 [==============================] - 0s 5ms/step - loss: 5.5070 - accuracy: 0.3667
     Epoch 75/100
     3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667

     Epoch 76/100
     3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667
     Epoch 77/100
     3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667
     Epoch 78/100
     3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667
     Epoch 79/100
     3/3 [==============================] - 0s 5ms/step - loss: 5.5070 - accuracy: 0.3667
     Epoch 80/100
     3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667
     Epoch 81/100
     3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667
     Epoch 82/100
     3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667
     Epoch 83/100
     3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667
```

```
    3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 84/100
    3/3 [==============================] - 0s 6ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 85/100
```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and reopen the link. ✕

```
    3/3 [==============================] - 0s 4ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 87/100
    3/3 [==============================] - 0s 8ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 88/100
    3/3 [==============================] - 0s 13ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 89/100
    3/3 [==============================] - 0s 6ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 90/100
    3/3 [==============================] - 0s 5ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 91/100
    3/3 [==============================] - 0s 7ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 92/100
    3/3 [==============================] - 0s 6ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 93/100
    3/3 [==============================] - 0s 6ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 94/100
    3/3 [==============================] - 0s 6ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 95/100
    3/3 [==============================] - 0s 5ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 96/100
    3/3 [==============================] - 0s 6ms/step - loss: 5.5070 - accuracy: 0.3667

    Epoch 97/100
    3/3 [==============================] - 0s 6ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 98/100
    3/3 [==============================] - 0s 7ms/step - loss: 5.5070 - accuracy: 0.3667
    Epoch 99/100
    3/3 [==============================] - 0s 5ms/step - loss: 5.5070 - accuracy: 0.3667
```

```
loss, accuracy = model.evaluate(X_test, y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', accuracy)
```

```
    Test loss: 2.1490793228149414
    Test accuracy: 0.20000000298023224
```

```
y_pred = model.predict(X_test)
y_pred
```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and reopen the link.

```
       [ 0.02374216, -0.35066772,  0.30138385],
       [-0.17621416, -0.03369495,  0.10298071],
       [ 0.22628416, -0.48364523,  0.27846602],
       [-0.17923597, -0.02651193,  0.08896062],
       [-0.11567868, -0.19617277,  0.26465827],
       [-0.15706937, -0.16489285,  0.26549232],
       [-0.15789397, -0.13094476,  0.23729965],
       [ 0.10432341, -0.36729804,  0.2607637 ],
       [-0.07732377, -0.21601218,  0.25583607],
       [-0.03181774, -0.24421287,  0.24843037],
       [-0.13169296, -0.17345953,  0.25568134],
       [-0.01883916, -0.2554856 ,  0.24941918],
       [-0.1398376 , -0.05923411,  0.09645239],
       [-0.03439499, -0.23676619,  0.24402863],
       [ 0.01098912, -0.25498226,  0.22842091],
       [-0.21070427,  0.00410924,  0.09626284],
       [-0.18578058, -0.03328779,  0.08792856],
       [ 0.1065345 , -0.3447903 ,  0.2405591 ],
       [ 0.06044795, -0.3026111 ,  0.23617634],
       [-0.10889591, -0.09399265,  0.13189495],
       [-0.16065791, -0.02610457,  0.09756337],
       [-0.03427277, -0.24675113,  0.2522307 ],
       [-0.12647216, -0.05211609,  0.06927569],
       [-0.09930379, -0.11140442,  0.13121043],
       [-0.11134873, -0.17711794,  0.24526899],
       [-0.12373573, -0.11138672,  0.1969123 ],
       [-0.16745298, -0.04345139,  0.11003551]], dtype=float32)
```

```
actual = np.argmax(y_test,axis=1)
predicted = np.argmax(y_pred,axis=1)
print(f"Actual: {actual}")
print(f"Predicted: {predicted}")
```

```
Actual: [2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0]
Predicted: [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and reopen the link. ✕

✓ 0s completed at 10:04 PM ● ✕