Program no. 4

Multipliclass Classification

```
import pandas
from keras.models import Sequential
from keras.layers import Dense
from keras.utils import np_utils
from sklearn.preprocessing import LabelEncoder
```

```
df=pandas.read_csv('Flower.csv',header=None)
print(df)
```

```
       0    1    2    3               4
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
..   ...  ...  ...  ...             ...
145  6.7  3.0  5.2  2.3  Iris-virginica
146  6.3  2.5  5.0  1.9  Iris-virginica
147  6.5  3.0  5.2  2.0  Iris-virginica
148  6.2  3.4  5.4  2.3  Iris-virginica
149  5.9  3.0  5.1  1.8  Iris-virginica

[150 rows x 5 columns]
```

```
X=df.iloc[:,0:4].astype(float)
y=df.iloc[:,4]
```

```
print(X)
```

```
       0    1    2    3
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
..   ...  ...  ...  ...
145  6.7  3.0  5.2  2.3
146  6.3  2.5  5.0  1.9
147  6.5  3.0  5.2  2.0
148  6.2  3.4  5.4  2.3
149  5.9  3.0  5.1  1.8

[150 rows x 4 columns]
```

```
print(y)
```

```
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
Name: 4, Length: 150, dtype: object
```

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

```
y
```

```
array([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])
```

```
from keras.utils import np_utils
encoded_Y=np_utils.to_categorical(y)
encoded_Y
```

```
array([[1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
```

```
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
```

```
model= Sequential()
model.add(Dense(8,input_dim=4,activation='relu'))
model.add(Dense(6,activation='relu'))
model.add(Dense(3,activation='softmax'))
```

```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics='accuracy')
```

```
model.fit(X,encoded_Y,epochs=100,batch_size=10)
```

```
Epoch 72/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4370 - accuracy: 0.
Epoch 73/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4378 - accuracy: 0.
Epoch 74/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4336 - accuracy: 0.
Epoch 75/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4324 - accuracy: 0.
Epoch 76/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4309 - accuracy: 0.
Epoch 77/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4308 - accuracy: 0.
Epoch 78/100
15/15 [==============================] - 0s 3ms/step - loss: 0.4278 - accuracy: 0.
Epoch 79/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4269 - accuracy: 0.
Epoch 80/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4243 - accuracy: 0.
Epoch 81/100
```

```
15/15 [==============================] - 0s 2ms/step - loss: 0.4228 - accuracy: 0.
Epoch 82/100
15/15 [==============================] - 0s 3ms/step - loss: 0.4210 - accuracy: 0.
Epoch 83/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4195 - accuracy: 0.
Epoch 84/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4177 - accuracy: 0.
Epoch 85/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4156 - accuracy: 0.
Epoch 86/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4140 - accuracy: 0.
Epoch 87/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4123 - accuracy: 0.
Epoch 88/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4108 - accuracy: 0.
Epoch 89/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4085 - accuracy: 0.
Epoch 90/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4058 - accuracy: 0.
Epoch 91/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4035 - accuracy: 0.
Epoch 92/100
15/15 [==============================] - 0s 2ms/step - loss: 0.4009 - accuracy: 0.
Epoch 93/100
15/15 [==============================] - 0s 2ms/step - loss: 0.3997 - accuracy: 0.
Epoch 94/100
15/15 [==============================] - 0s 2ms/step - loss: 0.3965 - accuracy: 0.
Epoch 95/100
15/15 [==============================] - 0s 2ms/step - loss: 0.3940 - accuracy: 0.
Epoch 96/100
15/15 [==============================] - 0s 2ms/step - loss: 0.3916 - accuracy: 0.
Epoch 97/100
15/15 [==============================] - 0s 2ms/step - loss: 0.3886 - accuracy: 0.
Epoch 98/100
15/15 [==============================] - 0s 2ms/step - loss: 0.3867 - accuracy: 0.
Epoch 99/100
15/15 [==============================] - 0s 2ms/step - loss: 0.3841 - accuracy: 0.
Epoch 100/100
```

```
predictions=model.predict(X)
```

```
for i in range(35,130,3):
  print(predictions[i],encoded_Y[i])
```

```
[9.9621320e-01 3.1072637e-03 6.7953131e-04] [1. 0. 0.]
[9.947212e-01 4.285797e-03 9.929605e-04] [1. 0. 0.]
[0.9666754  0.02592979 0.0073948 ] [1. 0. 0.]
[9.9751008e-01 2.1157451e-03 3.7422092e-04] [1. 0. 0.]
[9.9624169e-01 3.0868838e-03 6.7143043e-04] [1. 0. 0.]
[0.00116452 0.51556766 0.48326778] [0. 1. 0.]
[0.00139097 0.4846831  0.5139259 ] [0. 1. 0.]
[0.00142675 0.53951555 0.45905763] [0. 1. 0.]
[0.00383383 0.5546972  0.44146895] [0. 1. 0.]
[0.0013135  0.486363   0.51232344] [0. 1. 0.]
[0.0017461  0.52415586 0.474098  ] [0. 1. 0.]
[2.1960029e-04 3.8397127e-01 6.1580908e-01] [0. 1. 0.]
[0.00284184 0.53437746 0.46278074] [0. 1. 0.]
```

```
[0.00189672 0.52601695 0.4720863 ] [0. 1. 0.]
[2.9417503e-04 4.4207478e-01 5.5763108e-01] [0. 1. 0.]
[0.00335193 0.537967   0.45868105] [0. 1. 0.]
[3.4511302e-04 4.4645175e-01 5.5320311e-01] [0. 1. 0.]
[0.00075706 0.49829623 0.5009467 ] [0. 1. 0.]
[0.00211831 0.517938   0.47994366] [0. 1. 0.]
[0.00257358 0.5332306  0.46419576] [0. 1. 0.]
[0.01241769 0.6004778  0.38710442] [0. 1. 0.]
[0.033699   0.58668804 0.379613  ] [0. 1. 0.]
[2.1903153e-04 4.1097513e-01 5.8880579e-01] [0. 0. 1.]
[5.3068743e-05 3.5690138e-01 6.4304560e-01] [0. 0. 1.]
[2.7768167e-05 3.4501082e-01 6.5496141e-01] [0. 0. 1.]
[2.5964502e-04 4.3697530e-01 5.6276500e-01] [0. 0. 1.]
[1.3904169e-04 3.7176162e-01 6.2809938e-01] [0. 0. 1.]
[1.7360432e-04 4.2528740e-01 5.7453901e-01] [0. 0. 1.]
[1.6096994e-04 3.8296032e-01 6.1687875e-01] [0. 0. 1.]
[6.2491135e-06 2.7423811e-01 7.2575563e-01] [0. 0. 1.]
[8.048216e-05 4.051282e-01 5.947913e-01] [0. 0. 1.]
[5.7839919e-05 3.5093457e-01 6.4900756e-01] [0. 0. 1.]
```

✓ 0s completed at 21:28