MLP_tensorflow (1)

June 8, 2025

[1]:
```
import numpy as np
import pandas as pd
```

[3]:
```
from sklearn.datasets import load_iris
iris = load_iris()
x = iris.data
y = iris.target
```

[5]:
```
df = pd.DataFrame(x)
df
```

[5]:

	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]

[13]:
```
df.columns=['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

[15]:
```
x = df[['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']]
x = np.array(x)
y = np.array(y)
```

[17]:
```
y
```

[17]:
```
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, 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])
```

[31]:
```
import tensorflow as tf
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import tensorflow.keras as models
import tensorflow.keras.layers as layers
```

[33]:
```
model = models.Sequential([
 layers.Dense(16, activation='relu', input_shape=(4,)),
 layers.Dropout(0.3),
 layers.Dense(8, activation='relu'),
 layers.Dropout(0.2),
 layers.Dense(3, activation='softmax')
 ])

# Compile the model
model.compile(optimizer='adam',
 loss='sparse_categorical_crossentropy',
 metrics=['accuracy'])
```

```
C:\Users\sajee\anaconda3\Lib\site-packages\keras\src\layers\core\dense.py:87:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When
using Sequential models, prefer using an `Input(shape)` object as the first
layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

[35]:
```
xtrain , xtest , ytrain , ytest = train_test_split(x,y,test_size=0.
 ↪2,random_state=42)
ytrain
```

[35]:
```
array([0, 0, 1, 0, 0, 2, 1, 0, 0, 0, 2, 1, 1, 0, 0, 1, 2, 2, 1, 2, 1, 2,
       1, 0, 2, 1, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 1, 2, 0, 1, 2, 0, 2, 2,
       1, 1, 2, 1, 0, 1, 2, 0, 0, 1, 1, 0, 2, 0, 0, 1, 1, 2, 1, 2, 2, 1,
       0, 0, 2, 2, 0, 0, 0, 1, 2, 0, 2, 2, 0, 1, 1, 2, 1, 2, 0, 2, 1, 2,
       1, 1, 1, 0, 1, 1, 0, 1, 2, 2, 0, 1, 2, 2, 0, 2, 0, 1, 2, 2, 1, 2,
```

```
       1, 1, 2, 2, 0, 1, 2, 0, 1, 2])
```

[37]:
```
# Train the model with early stopping
early_stopping = tf.keras.callbacks.EarlyStopping(
 monitor='val_loss',
 patience=20,
 restore_best_weights=True
 )
history = model.fit(xtrain, ytrain,
 epochs=200,
 batch_size=32,
 validation_split=0.2,
 callbacks=[early_stopping],
 verbose=1)
```

```
Epoch 1/200
3/3             2s 148ms/step -
accuracy: 0.3307 - loss: 1.3145 - val_accuracy: 0.2917 - val_loss: 1.0615
Epoch 2/200
3/3             0s 47ms/step -
accuracy: 0.3099 - loss: 1.3406 - val_accuracy: 0.2917 - val_loss: 1.0532
Epoch 3/200
3/3             0s 34ms/step -
accuracy: 0.3216 - loss: 1.3511 - val_accuracy: 0.2917 - val_loss: 1.0464
Epoch 4/200
3/3             0s 49ms/step -
accuracy: 0.3464 - loss: 1.2808 - val_accuracy: 0.2917 - val_loss: 1.0404
Epoch 5/200
3/3             0s 54ms/step -
accuracy: 0.2969 - loss: 1.3570 - val_accuracy: 0.2917 - val_loss: 1.0341
Epoch 6/200
3/3             0s 45ms/step -
accuracy: 0.2630 - loss: 1.3399 - val_accuracy: 0.2917 - val_loss: 1.0288
Epoch 7/200
3/3             0s 56ms/step -
accuracy: 0.3971 - loss: 1.2119 - val_accuracy: 0.2917 - val_loss: 1.0252
Epoch 8/200
3/3             0s 48ms/step -
accuracy: 0.3490 - loss: 1.2537 - val_accuracy: 0.2917 - val_loss: 1.0234
Epoch 9/200
3/3             0s 47ms/step -
accuracy: 0.3281 - loss: 1.2466 - val_accuracy: 0.2917 - val_loss: 1.0232
Epoch 10/200
3/3             0s 37ms/step -
accuracy: 0.4310 - loss: 1.1262 - val_accuracy: 0.2917 - val_loss: 1.0236
Epoch 11/200
3/3             0s 51ms/step -
accuracy: 0.3542 - loss: 1.1451 - val_accuracy: 0.2917 - val_loss: 1.0232
```

```
Epoch 12/200
3/3             0s 42ms/step -
accuracy: 0.3776 - loss: 1.1916 - val_accuracy: 0.2917 - val_loss: 1.0229
Epoch 13/200
3/3             0s 41ms/step -
accuracy: 0.2943 - loss: 1.1963 - val_accuracy: 0.2917 - val_loss: 1.0227
Epoch 14/200
3/3             0s 51ms/step -
accuracy: 0.3633 - loss: 1.1437 - val_accuracy: 0.2917 - val_loss: 1.0225
Epoch 15/200
3/3             0s 43ms/step -
accuracy: 0.3307 - loss: 1.0897 - val_accuracy: 0.2917 - val_loss: 1.0221
Epoch 16/200
3/3             0s 40ms/step -
accuracy: 0.3672 - loss: 1.1377 - val_accuracy: 0.2917 - val_loss: 1.0208
Epoch 17/200
3/3             0s 54ms/step -
accuracy: 0.3229 - loss: 1.1442 - val_accuracy: 0.2917 - val_loss: 1.0197
Epoch 18/200
3/3             0s 43ms/step -
accuracy: 0.3151 - loss: 1.1314 - val_accuracy: 0.2917 - val_loss: 1.0190
Epoch 19/200
3/3             0s 59ms/step -
accuracy: 0.4154 - loss: 1.0458 - val_accuracy: 0.2917 - val_loss: 1.0180
Epoch 20/200
3/3             0s 28ms/step -
accuracy: 0.3424 - loss: 1.1305 - val_accuracy: 0.2917 - val_loss: 1.0168
Epoch 21/200
3/3             0s 39ms/step -
accuracy: 0.3242 - loss: 1.1324 - val_accuracy: 0.2917 - val_loss: 1.0155
Epoch 22/200
3/3             0s 42ms/step -
accuracy: 0.3034 - loss: 1.1722 - val_accuracy: 0.2917 - val_loss: 1.0135
Epoch 23/200
3/3             0s 49ms/step -
accuracy: 0.3151 - loss: 1.0614 - val_accuracy: 0.2917 - val_loss: 1.0119
Epoch 24/200
3/3             0s 37ms/step -
accuracy: 0.3607 - loss: 1.0916 - val_accuracy: 0.3333 - val_loss: 1.0105
Epoch 25/200
3/3             0s 44ms/step -
accuracy: 0.2734 - loss: 1.1082 - val_accuracy: 0.3750 - val_loss: 1.0095
Epoch 26/200
3/3             0s 48ms/step -
accuracy: 0.3372 - loss: 1.0995 - val_accuracy: 0.3750 - val_loss: 1.0086
Epoch 27/200
3/3             0s 57ms/step -
accuracy: 0.3542 - loss: 1.0972 - val_accuracy: 0.4167 - val_loss: 1.0076
```

```
Epoch 28/200
3/3            0s 48ms/step -
accuracy: 0.4362 - loss: 1.1023 - val_accuracy: 0.4167 - val_loss: 1.0064
Epoch 29/200
3/3            0s 49ms/step -
accuracy: 0.5091 - loss: 1.0448 - val_accuracy: 0.4167 - val_loss: 1.0045
Epoch 30/200
3/3            0s 43ms/step -
accuracy: 0.4831 - loss: 1.0839 - val_accuracy: 0.4167 - val_loss: 1.0034
Epoch 31/200
3/3            0s 41ms/step -
accuracy: 0.3893 - loss: 1.0881 - val_accuracy: 0.4583 - val_loss: 1.0030
Epoch 32/200
3/3            0s 41ms/step -
accuracy: 0.5104 - loss: 0.9901 - val_accuracy: 0.4583 - val_loss: 1.0017
Epoch 33/200
3/3            0s 39ms/step -
accuracy: 0.4375 - loss: 1.0763 - val_accuracy: 0.4583 - val_loss: 0.9993
Epoch 34/200
3/3            0s 33ms/step -
accuracy: 0.4102 - loss: 1.0884 - val_accuracy: 0.4583 - val_loss: 0.9963
Epoch 35/200
3/3            0s 33ms/step -
accuracy: 0.4349 - loss: 1.0690 - val_accuracy: 0.5000 - val_loss: 0.9929
Epoch 36/200
3/3            0s 39ms/step -
accuracy: 0.4362 - loss: 1.0460 - val_accuracy: 0.5000 - val_loss: 0.9893
Epoch 37/200
3/3            0s 37ms/step -
accuracy: 0.3919 - loss: 1.0928 - val_accuracy: 0.5000 - val_loss: 0.9861
Epoch 38/200
3/3            0s 57ms/step -
accuracy: 0.4076 - loss: 1.0497 - val_accuracy: 0.5000 - val_loss: 0.9833
Epoch 39/200
3/3            0s 48ms/step -
accuracy: 0.4453 - loss: 1.0826 - val_accuracy: 0.5000 - val_loss: 0.9792
Epoch 40/200
3/3            0s 46ms/step -
accuracy: 0.3620 - loss: 1.0931 - val_accuracy: 0.5000 - val_loss: 0.9754
Epoch 41/200
3/3            0s 56ms/step -
accuracy: 0.4505 - loss: 1.0437 - val_accuracy: 0.5000 - val_loss: 0.9723
Epoch 42/200
3/3            0s 56ms/step -
accuracy: 0.5690 - loss: 1.0341 - val_accuracy: 0.5000 - val_loss: 0.9688
Epoch 43/200
3/3            0s 64ms/step -
accuracy: 0.5091 - loss: 1.0314 - val_accuracy: 0.5000 - val_loss: 0.9656
```

```
Epoch 44/200
3/3            0s 55ms/step -
accuracy: 0.3711 - loss: 1.0295 - val_accuracy: 0.5000 - val_loss: 0.9625
Epoch 45/200
3/3            0s 39ms/step -
accuracy: 0.5521 - loss: 0.9482 - val_accuracy: 0.5000 - val_loss: 0.9595
Epoch 46/200
3/3            0s 63ms/step -
accuracy: 0.4232 - loss: 1.0286 - val_accuracy: 0.5000 - val_loss: 0.9567
Epoch 47/200
3/3            0s 65ms/step -
accuracy: 0.4115 - loss: 1.0456 - val_accuracy: 0.5000 - val_loss: 0.9541
Epoch 48/200
3/3            0s 51ms/step -
accuracy: 0.4727 - loss: 1.0167 - val_accuracy: 0.5000 - val_loss: 0.9513
Epoch 49/200
3/3            0s 58ms/step -
accuracy: 0.5078 - loss: 1.0106 - val_accuracy: 0.5000 - val_loss: 0.9484
Epoch 50/200
3/3            0s 47ms/step -
accuracy: 0.5534 - loss: 0.9887 - val_accuracy: 0.5000 - val_loss: 0.9454
Epoch 51/200
3/3            0s 63ms/step -
accuracy: 0.5482 - loss: 1.0009 - val_accuracy: 0.5000 - val_loss: 0.9429
Epoch 52/200
3/3            0s 61ms/step -
accuracy: 0.5221 - loss: 1.0069 - val_accuracy: 0.5000 - val_loss: 0.9403
Epoch 53/200
3/3            0s 60ms/step -
accuracy: 0.4570 - loss: 1.0036 - val_accuracy: 0.5000 - val_loss: 0.9377
Epoch 54/200
3/3            0s 79ms/step -
accuracy: 0.5026 - loss: 0.9981 - val_accuracy: 0.5000 - val_loss: 0.9348
Epoch 55/200
3/3            0s 63ms/step -
accuracy: 0.5846 - loss: 0.9136 - val_accuracy: 0.5000 - val_loss: 0.9315
Epoch 56/200
3/3            0s 55ms/step -
accuracy: 0.6094 - loss: 0.8898 - val_accuracy: 0.5000 - val_loss: 0.9280
Epoch 57/200
3/3            0s 57ms/step -
accuracy: 0.5586 - loss: 0.9260 - val_accuracy: 0.5000 - val_loss: 0.9248
Epoch 58/200
3/3            0s 63ms/step -
accuracy: 0.5990 - loss: 0.9171 - val_accuracy: 0.5000 - val_loss: 0.9215
Epoch 59/200
3/3            0s 52ms/step -
accuracy: 0.6081 - loss: 0.9151 - val_accuracy: 0.5000 - val_loss: 0.9176
```

```
Epoch 60/200
3/3            0s 63ms/step -
accuracy: 0.5599 - loss: 0.9551 - val_accuracy: 0.5000 - val_loss: 0.9126
Epoch 61/200
3/3            0s 47ms/step -
accuracy: 0.6068 - loss: 0.9207 - val_accuracy: 0.5000 - val_loss: 0.9058
Epoch 62/200
3/3            0s 50ms/step -
accuracy: 0.5885 - loss: 0.9417 - val_accuracy: 0.5000 - val_loss: 0.9005
Epoch 63/200
3/3            0s 54ms/step -
accuracy: 0.5846 - loss: 0.9305 - val_accuracy: 0.5000 - val_loss: 0.8941
Epoch 64/200
3/3            0s 47ms/step -
accuracy: 0.4870 - loss: 0.9129 - val_accuracy: 0.5000 - val_loss: 0.8892
Epoch 65/200
3/3            0s 72ms/step -
accuracy: 0.5495 - loss: 0.9238 - val_accuracy: 0.5000 - val_loss: 0.8839
Epoch 66/200
3/3            0s 69ms/step -
accuracy: 0.6263 - loss: 0.8929 - val_accuracy: 0.5000 - val_loss: 0.8783
Epoch 67/200
3/3            0s 64ms/step -
accuracy: 0.5586 - loss: 0.9314 - val_accuracy: 0.5000 - val_loss: 0.8679
Epoch 68/200
3/3            0s 51ms/step -
accuracy: 0.4935 - loss: 0.9554 - val_accuracy: 0.5000 - val_loss: 0.8597
Epoch 69/200
3/3            0s 48ms/step -
accuracy: 0.6341 - loss: 0.8794 - val_accuracy: 0.5000 - val_loss: 0.8525
Epoch 70/200
3/3            0s 47ms/step -
accuracy: 0.5703 - loss: 0.9031 - val_accuracy: 0.5000 - val_loss: 0.8476
Epoch 71/200
3/3            0s 43ms/step -
accuracy: 0.5521 - loss: 0.9189 - val_accuracy: 0.5000 - val_loss: 0.8452
Epoch 72/200
3/3            0s 65ms/step -
accuracy: 0.6055 - loss: 0.8946 - val_accuracy: 0.5000 - val_loss: 0.8416
Epoch 73/200
3/3            0s 80ms/step -
accuracy: 0.5312 - loss: 0.9723 - val_accuracy: 0.5000 - val_loss: 0.8324
Epoch 74/200
3/3            0s 50ms/step -
accuracy: 0.5990 - loss: 0.9301 - val_accuracy: 0.5417 - val_loss: 0.8181
Epoch 75/200
3/3            0s 55ms/step -
accuracy: 0.5456 - loss: 0.8958 - val_accuracy: 0.6250 - val_loss: 0.8024
```

```
Epoch 76/200
3/3             0s 64ms/step -
accuracy: 0.5964 - loss: 0.9057 - val_accuracy: 0.7083 - val_loss: 0.7907
Epoch 77/200
3/3             0s 71ms/step -
accuracy: 0.5495 - loss: 0.9422 - val_accuracy: 0.8333 - val_loss: 0.7775
Epoch 78/200
3/3             0s 65ms/step -
accuracy: 0.6836 - loss: 0.8628 - val_accuracy: 0.9583 - val_loss: 0.7589
Epoch 79/200
3/3             0s 65ms/step -
accuracy: 0.5312 - loss: 0.8896 - val_accuracy: 0.9167 - val_loss: 0.7406
Epoch 80/200
3/3             0s 82ms/step -
accuracy: 0.6146 - loss: 0.8606 - val_accuracy: 0.9167 - val_loss: 0.7269
Epoch 81/200
3/3             0s 49ms/step -
accuracy: 0.7044 - loss: 0.7923 - val_accuracy: 0.7917 - val_loss: 0.7150
Epoch 82/200
3/3             0s 68ms/step -
accuracy: 0.5703 - loss: 0.8722 - val_accuracy: 0.7500 - val_loss: 0.7048
Epoch 83/200
3/3             0s 62ms/step -
accuracy: 0.6992 - loss: 0.8391 - val_accuracy: 0.7500 - val_loss: 0.6950
Epoch 84/200
3/3             0s 59ms/step -
accuracy: 0.6003 - loss: 0.8216 - val_accuracy: 0.7500 - val_loss: 0.6897
Epoch 85/200
3/3             0s 48ms/step -
accuracy: 0.7266 - loss: 0.7481 - val_accuracy: 0.7917 - val_loss: 0.6849
Epoch 86/200
3/3             0s 50ms/step -
accuracy: 0.5521 - loss: 0.8927 - val_accuracy: 0.9167 - val_loss: 0.6798
Epoch 87/200
3/3             0s 56ms/step -
accuracy: 0.6380 - loss: 0.8707 - val_accuracy: 0.9167 - val_loss: 0.6751
Epoch 88/200
3/3             0s 54ms/step -
accuracy: 0.6341 - loss: 0.8707 - val_accuracy: 0.8750 - val_loss: 0.6682
Epoch 89/200
3/3             0s 48ms/step -
accuracy: 0.6523 - loss: 0.8054 - val_accuracy: 0.8750 - val_loss: 0.6618
Epoch 90/200
3/3             0s 56ms/step -
accuracy: 0.5286 - loss: 0.8892 - val_accuracy: 0.8750 - val_loss: 0.6567
Epoch 91/200
3/3             0s 51ms/step -
accuracy: 0.6393 - loss: 0.7895 - val_accuracy: 0.8750 - val_loss: 0.6483
```

```
Epoch 92/200
3/3             0s 40ms/step -
accuracy: 0.5794 - loss: 0.8999 - val_accuracy: 0.8333 - val_loss: 0.6420
Epoch 93/200
3/3             0s 73ms/step -
accuracy: 0.5443 - loss: 0.9193 - val_accuracy: 0.7917 - val_loss: 0.6376
Epoch 94/200
3/3             0s 44ms/step -
accuracy: 0.7057 - loss: 0.7733 - val_accuracy: 0.7917 - val_loss: 0.6343
Epoch 95/200
3/3             0s 41ms/step -
accuracy: 0.7161 - loss: 0.7436 - val_accuracy: 0.7917 - val_loss: 0.6323
Epoch 96/200
3/3             0s 48ms/step -
accuracy: 0.6979 - loss: 0.8690 - val_accuracy: 0.7917 - val_loss: 0.6283
Epoch 97/200
3/3             0s 50ms/step -
accuracy: 0.6979 - loss: 0.7931 - val_accuracy: 0.7917 - val_loss: 0.6243
Epoch 98/200
3/3             0s 63ms/step -
accuracy: 0.6680 - loss: 0.7834 - val_accuracy: 0.7500 - val_loss: 0.6216
Epoch 99/200
3/3             0s 64ms/step -
accuracy: 0.6120 - loss: 0.8157 - val_accuracy: 0.7917 - val_loss: 0.6196
Epoch 100/200
3/3             0s 56ms/step -
accuracy: 0.6458 - loss: 0.7846 - val_accuracy: 0.7917 - val_loss: 0.6166
Epoch 101/200
3/3             0s 42ms/step -
accuracy: 0.6510 - loss: 0.7831 - val_accuracy: 0.7917 - val_loss: 0.6141
Epoch 102/200
3/3             0s 65ms/step -
accuracy: 0.6120 - loss: 0.8069 - val_accuracy: 0.7917 - val_loss: 0.6120
Epoch 103/200
3/3             0s 54ms/step -
accuracy: 0.7018 - loss: 0.7587 - val_accuracy: 0.7917 - val_loss: 0.6088
Epoch 104/200
3/3             0s 61ms/step -
accuracy: 0.6289 - loss: 0.8187 - val_accuracy: 0.7917 - val_loss: 0.6043
Epoch 105/200
3/3             0s 66ms/step -
accuracy: 0.7083 - loss: 0.7254 - val_accuracy: 0.7917 - val_loss: 0.6001
Epoch 106/200
3/3             0s 42ms/step -
accuracy: 0.6914 - loss: 0.7844 - val_accuracy: 0.7917 - val_loss: 0.5973
Epoch 107/200
3/3             0s 66ms/step -
accuracy: 0.6302 - loss: 0.7672 - val_accuracy: 0.8333 - val_loss: 0.5962
```

```
Epoch 108/200
3/3             0s 68ms/step -
accuracy: 0.7188 - loss: 0.7307 - val_accuracy: 0.8333 - val_loss: 0.5937
Epoch 109/200
3/3             0s 63ms/step -
accuracy: 0.7070 - loss: 0.7118 - val_accuracy: 0.8750 - val_loss: 0.5919
Epoch 110/200
3/3             0s 61ms/step -
accuracy: 0.6589 - loss: 0.7447 - val_accuracy: 0.8750 - val_loss: 0.5906
Epoch 111/200
3/3             0s 74ms/step -
accuracy: 0.7747 - loss: 0.6693 - val_accuracy: 0.8750 - val_loss: 0.5867
Epoch 112/200
3/3             0s 46ms/step -
accuracy: 0.7201 - loss: 0.6853 - val_accuracy: 0.8750 - val_loss: 0.5836
Epoch 113/200
3/3             0s 70ms/step -
accuracy: 0.7201 - loss: 0.7261 - val_accuracy: 0.9167 - val_loss: 0.5817
Epoch 114/200
3/3             0s 77ms/step -
accuracy: 0.7565 - loss: 0.6584 - val_accuracy: 0.9167 - val_loss: 0.5801
Epoch 115/200
3/3             0s 73ms/step -
accuracy: 0.7174 - loss: 0.7455 - val_accuracy: 0.9167 - val_loss: 0.5753
Epoch 116/200
3/3             0s 65ms/step -
accuracy: 0.7630 - loss: 0.6773 - val_accuracy: 0.9167 - val_loss: 0.5681
Epoch 117/200
3/3             0s 49ms/step -
accuracy: 0.7474 - loss: 0.6911 - val_accuracy: 0.9167 - val_loss: 0.5611
Epoch 118/200
3/3             0s 49ms/step -
accuracy: 0.7669 - loss: 0.7026 - val_accuracy: 0.8750 - val_loss: 0.5559
Epoch 119/200
3/3             0s 70ms/step -
accuracy: 0.7396 - loss: 0.6776 - val_accuracy: 0.8750 - val_loss: 0.5523
Epoch 120/200
3/3             0s 75ms/step -
accuracy: 0.6940 - loss: 0.6739 - val_accuracy: 0.8750 - val_loss: 0.5481
Epoch 121/200
3/3             0s 74ms/step -
accuracy: 0.7669 - loss: 0.6589 - val_accuracy: 0.8750 - val_loss: 0.5431
Epoch 122/200
3/3             0s 45ms/step -
accuracy: 0.7669 - loss: 0.6309 - val_accuracy: 0.8750 - val_loss: 0.5375
Epoch 123/200
3/3             0s 44ms/step -
accuracy: 0.7344 - loss: 0.6770 - val_accuracy: 0.8750 - val_loss: 0.5328
```

```
Epoch 124/200
3/3            0s 72ms/step -
accuracy: 0.7305 - loss: 0.6670 - val_accuracy: 0.9167 - val_loss: 0.5292
Epoch 125/200
3/3            0s 40ms/step -
accuracy: 0.6719 - loss: 0.7010 - val_accuracy: 0.9167 - val_loss: 0.5256
Epoch 126/200
3/3            0s 49ms/step -
accuracy: 0.7357 - loss: 0.7062 - val_accuracy: 0.9167 - val_loss: 0.5224
Epoch 127/200
3/3            0s 59ms/step -
accuracy: 0.7357 - loss: 0.6560 - val_accuracy: 0.9167 - val_loss: 0.5184
Epoch 128/200
3/3            0s 46ms/step -
accuracy: 0.6914 - loss: 0.7112 - val_accuracy: 0.8750 - val_loss: 0.5141
Epoch 129/200
3/3            0s 49ms/step -
accuracy: 0.7526 - loss: 0.6097 - val_accuracy: 0.8750 - val_loss: 0.5100
Epoch 130/200
3/3            0s 56ms/step -
accuracy: 0.7031 - loss: 0.7058 - val_accuracy: 0.8750 - val_loss: 0.5071
Epoch 131/200
3/3            0s 53ms/step -
accuracy: 0.6758 - loss: 0.6583 - val_accuracy: 0.9167 - val_loss: 0.5053
Epoch 132/200
3/3            0s 47ms/step -
accuracy: 0.6745 - loss: 0.7140 - val_accuracy: 0.9167 - val_loss: 0.5043
Epoch 133/200
3/3            0s 48ms/step -
accuracy: 0.7331 - loss: 0.6243 - val_accuracy: 0.8750 - val_loss: 0.5023
Epoch 134/200
3/3            0s 53ms/step -
accuracy: 0.7305 - loss: 0.6138 - val_accuracy: 0.8750 - val_loss: 0.4991
Epoch 135/200
3/3            0s 63ms/step -
accuracy: 0.6953 - loss: 0.6778 - val_accuracy: 0.8750 - val_loss: 0.4952
Epoch 136/200
3/3            0s 60ms/step -
accuracy: 0.7695 - loss: 0.6367 - val_accuracy: 0.8750 - val_loss: 0.4922
Epoch 137/200
3/3            0s 63ms/step -
accuracy: 0.7682 - loss: 0.6085 - val_accuracy: 0.8750 - val_loss: 0.4897
Epoch 138/200
3/3            0s 41ms/step -
accuracy: 0.7630 - loss: 0.6031 - val_accuracy: 0.8750 - val_loss: 0.4857
Epoch 139/200
3/3            0s 65ms/step -
accuracy: 0.6029 - loss: 0.6953 - val_accuracy: 0.8750 - val_loss: 0.4831
```

```
Epoch 140/200
3/3            0s 42ms/step -
accuracy: 0.7305 - loss: 0.6413 - val_accuracy: 0.8750 - val_loss: 0.4808
Epoch 141/200
3/3            0s 42ms/step -
accuracy: 0.6146 - loss: 0.7057 - val_accuracy: 0.9167 - val_loss: 0.4797
Epoch 142/200
3/3            0s 53ms/step -
accuracy: 0.7656 - loss: 0.5826 - val_accuracy: 0.9167 - val_loss: 0.4776
Epoch 143/200
3/3            0s 58ms/step -
accuracy: 0.7591 - loss: 0.6093 - val_accuracy: 0.9583 - val_loss: 0.4755
Epoch 144/200
3/3            0s 66ms/step -
accuracy: 0.6823 - loss: 0.6358 - val_accuracy: 0.9583 - val_loss: 0.4747
Epoch 145/200
3/3            0s 67ms/step -
accuracy: 0.7604 - loss: 0.5922 - val_accuracy: 0.9583 - val_loss: 0.4744
Epoch 146/200
3/3            0s 63ms/step -
accuracy: 0.7344 - loss: 0.6580 - val_accuracy: 0.9583 - val_loss: 0.4739
Epoch 147/200
3/3            0s 60ms/step -
accuracy: 0.7044 - loss: 0.6201 - val_accuracy: 0.9583 - val_loss: 0.4722
Epoch 148/200
3/3            0s 44ms/step -
accuracy: 0.6484 - loss: 0.6389 - val_accuracy: 1.0000 - val_loss: 0.4698
Epoch 149/200
3/3            0s 48ms/step -
accuracy: 0.7747 - loss: 0.6228 - val_accuracy: 1.0000 - val_loss: 0.4677
Epoch 150/200
3/3            0s 57ms/step -
accuracy: 0.7370 - loss: 0.6114 - val_accuracy: 1.0000 - val_loss: 0.4658
Epoch 151/200
3/3            0s 52ms/step -
accuracy: 0.6875 - loss: 0.6109 - val_accuracy: 1.0000 - val_loss: 0.4633
Epoch 152/200
3/3            0s 72ms/step -
accuracy: 0.6784 - loss: 0.6068 - val_accuracy: 1.0000 - val_loss: 0.4610
Epoch 153/200
3/3            0s 53ms/step -
accuracy: 0.7044 - loss: 0.6190 - val_accuracy: 1.0000 - val_loss: 0.4603
Epoch 154/200
3/3            0s 66ms/step -
accuracy: 0.7539 - loss: 0.6292 - val_accuracy: 1.0000 - val_loss: 0.4586
Epoch 155/200
3/3            0s 54ms/step -
accuracy: 0.7565 - loss: 0.5957 - val_accuracy: 0.9583 - val_loss: 0.4571
```

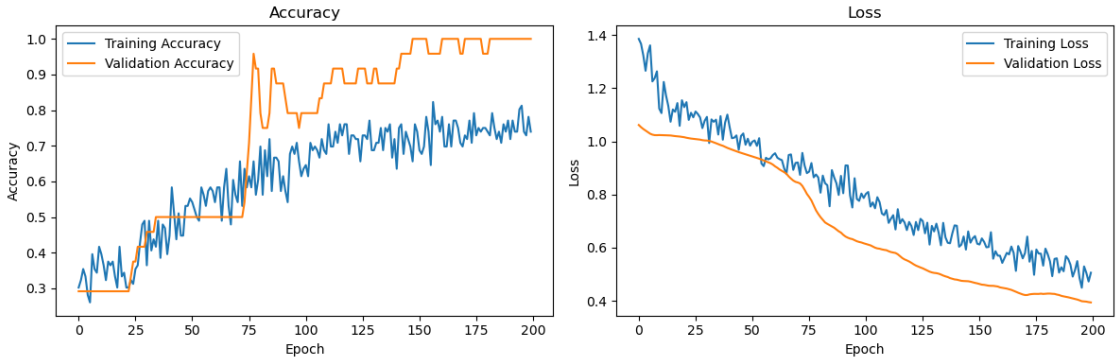
```
Epoch 156/200
3/3            0s 44ms/step -
accuracy: 0.6237 - loss: 0.6714 - val_accuracy: 0.9583 - val_loss: 0.4566
Epoch 157/200
3/3            0s 53ms/step -
accuracy: 0.8099 - loss: 0.5722 - val_accuracy: 0.9583 - val_loss: 0.4566
Epoch 158/200
3/3            0s 59ms/step -
accuracy: 0.7357 - loss: 0.6101 - val_accuracy: 0.9583 - val_loss: 0.4552
Epoch 159/200
3/3            0s 53ms/step -
accuracy: 0.7526 - loss: 0.5765 - val_accuracy: 0.9583 - val_loss: 0.4541
Epoch 160/200
3/3            0s 65ms/step -
accuracy: 0.7799 - loss: 0.5293 - val_accuracy: 0.9583 - val_loss: 0.4507
Epoch 161/200
3/3            0s 39ms/step -
accuracy: 0.7812 - loss: 0.5528 - val_accuracy: 1.0000 - val_loss: 0.4493
Epoch 162/200
3/3            0s 61ms/step -
accuracy: 0.7005 - loss: 0.5785 - val_accuracy: 1.0000 - val_loss: 0.4483
Epoch 163/200
3/3            0s 54ms/step -
accuracy: 0.6849 - loss: 0.5867 - val_accuracy: 1.0000 - val_loss: 0.4461
Epoch 164/200
3/3            0s 52ms/step -
accuracy: 0.7357 - loss: 0.5859 - val_accuracy: 1.0000 - val_loss: 0.4437
Epoch 165/200
3/3            0s 55ms/step -
accuracy: 0.6888 - loss: 0.6116 - val_accuracy: 1.0000 - val_loss: 0.4413
Epoch 166/200
3/3            0s 77ms/step -
accuracy: 0.7839 - loss: 0.5696 - val_accuracy: 1.0000 - val_loss: 0.4380
Epoch 167/200
3/3            0s 47ms/step -
accuracy: 0.8034 - loss: 0.4772 - val_accuracy: 1.0000 - val_loss: 0.4343
Epoch 168/200
3/3            0s 52ms/step -
accuracy: 0.7461 - loss: 0.6033 - val_accuracy: 1.0000 - val_loss: 0.4306
Epoch 169/200
3/3            0s 53ms/step -
accuracy: 0.7253 - loss: 0.5662 - val_accuracy: 0.9583 - val_loss: 0.4275
Epoch 170/200
3/3            0s 41ms/step -
accuracy: 0.6771 - loss: 0.5782 - val_accuracy: 0.9583 - val_loss: 0.4242
Epoch 171/200
3/3            0s 71ms/step -
accuracy: 0.7201 - loss: 0.6100 - val_accuracy: 1.0000 - val_loss: 0.4219
```

```
Epoch 172/200
3/3            0s 57ms/step -
accuracy: 0.7383 - loss: 0.6293 - val_accuracy: 1.0000 - val_loss: 0.4221
Epoch 173/200
3/3            0s 53ms/step -
accuracy: 0.7604 - loss: 0.5612 - val_accuracy: 1.0000 - val_loss: 0.4245
Epoch 174/200
3/3            0s 42ms/step -
accuracy: 0.7253 - loss: 0.5489 - val_accuracy: 1.0000 - val_loss: 0.4256
Epoch 175/200
3/3            0s 38ms/step -
accuracy: 0.8060 - loss: 0.4881 - val_accuracy: 1.0000 - val_loss: 0.4263
Epoch 176/200
3/3            0s 73ms/step -
accuracy: 0.7435 - loss: 0.5879 - val_accuracy: 1.0000 - val_loss: 0.4262
Epoch 177/200
3/3            0s 42ms/step -
accuracy: 0.7383 - loss: 0.5643 - val_accuracy: 1.0000 - val_loss: 0.4262
Epoch 178/200
3/3            0s 44ms/step -
accuracy: 0.7370 - loss: 0.5806 - val_accuracy: 1.0000 - val_loss: 0.4272
Epoch 179/200
3/3            0s 45ms/step -
accuracy: 0.7461 - loss: 0.5270 - val_accuracy: 0.9583 - val_loss: 0.4261
Epoch 180/200
3/3            0s 41ms/step -
accuracy: 0.7656 - loss: 0.5774 - val_accuracy: 0.9583 - val_loss: 0.4268
Epoch 181/200
3/3            0s 56ms/step -
accuracy: 0.7487 - loss: 0.5777 - val_accuracy: 0.9583 - val_loss: 0.4274
Epoch 182/200
3/3            0s 61ms/step -
accuracy: 0.6771 - loss: 0.5817 - val_accuracy: 1.0000 - val_loss: 0.4275
Epoch 183/200
3/3            0s 52ms/step -
accuracy: 0.8099 - loss: 0.4615 - val_accuracy: 1.0000 - val_loss: 0.4266
Epoch 184/200
3/3            0s 77ms/step -
accuracy: 0.7461 - loss: 0.5499 - val_accuracy: 1.0000 - val_loss: 0.4262
Epoch 185/200
3/3            0s 51ms/step -
accuracy: 0.7695 - loss: 0.5058 - val_accuracy: 1.0000 - val_loss: 0.4238
Epoch 186/200
3/3            0s 61ms/step -
accuracy: 0.7331 - loss: 0.5203 - val_accuracy: 1.0000 - val_loss: 0.4221
Epoch 187/200
3/3            0s 67ms/step -
accuracy: 0.6940 - loss: 0.5304 - val_accuracy: 1.0000 - val_loss: 0.4189
```

```
Epoch 188/200
3/3             0s 72ms/step -
accuracy: 0.7747 - loss: 0.5362 - val_accuracy: 1.0000 - val_loss: 0.4170
Epoch 189/200
3/3             0s 49ms/step -
accuracy: 0.7565 - loss: 0.5519 - val_accuracy: 1.0000 - val_loss: 0.4154
Epoch 190/200
3/3             0s 60ms/step -
accuracy: 0.7487 - loss: 0.5749 - val_accuracy: 1.0000 - val_loss: 0.4135
Epoch 191/200
3/3             0s 84ms/step -
accuracy: 0.7266 - loss: 0.5266 - val_accuracy: 1.0000 - val_loss: 0.4114
Epoch 192/200
3/3             0s 57ms/step -
accuracy: 0.8034 - loss: 0.4628 - val_accuracy: 1.0000 - val_loss: 0.4091
Epoch 193/200
3/3             0s 61ms/step -
accuracy: 0.6979 - loss: 0.5297 - val_accuracy: 1.0000 - val_loss: 0.4069
Epoch 194/200
3/3             0s 48ms/step -
accuracy: 0.7253 - loss: 0.5796 - val_accuracy: 1.0000 - val_loss: 0.4050
Epoch 195/200
3/3             0s 56ms/step -
accuracy: 0.7956 - loss: 0.5012 - val_accuracy: 1.0000 - val_loss: 0.4021
Epoch 196/200
3/3             0s 40ms/step -
accuracy: 0.7969 - loss: 0.4505 - val_accuracy: 1.0000 - val_loss: 0.3987
Epoch 197/200
3/3             0s 59ms/step -
accuracy: 0.6940 - loss: 0.5896 - val_accuracy: 1.0000 - val_loss: 0.3977
Epoch 198/200
3/3             0s 47ms/step -
accuracy: 0.6849 - loss: 0.5201 - val_accuracy: 1.0000 - val_loss: 0.3972
Epoch 199/200
3/3             0s 48ms/step -
accuracy: 0.7344 - loss: 0.5018 - val_accuracy: 1.0000 - val_loss: 0.3954
Epoch 200/200
3/3             0s 46ms/step -
accuracy: 0.7292 - loss: 0.5009 - val_accuracy: 1.0000 - val_loss: 0.3942
```

[39]:
```python
def plot_training_history(history):
 plt.figure(figsize=(12, 4))

 # Plot accuracy
 plt.subplot(1, 2, 1)
 plt.plot(history.history['accuracy'], label='Training Accuracy')
 plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
```

```
plt.title('Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()

# Plot loss
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()

plt.tight_layout()
plt.show()


plot_training_history(history)
```



[41]:
```
from sklearn.metrics import classification_report,confusion_matrix
ypred = model.predict(xtest)
ypredcl= ypred.argmax(axis=1)
print(classification_report(ypredcl,ytest))
```

```
1/1             0s 106ms/step
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       0.89      1.00      0.94         8
           2       1.00      0.92      0.96        12

    accuracy                           0.97        30
   macro avg       0.96      0.97      0.97        30
```

```
weighted avg       0.97      0.97      0.97        30
```

[43]:
```
test_loss, test_accuracy = model.evaluate(xtest, ytest, verbose=0)
```

[45]:
```
print(test_loss, test_accuracy)
```

```
0.3915502429008484 0.9666666388511658
```

[47]:
```
model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 16)	80
dropout_2 (Dropout)	(None, 16)	0
dense_4 (Dense)	(None, 8)	136
dropout_3 (Dropout)	(None, 8)	0
dense_5 (Dense)	(None, 3)	27

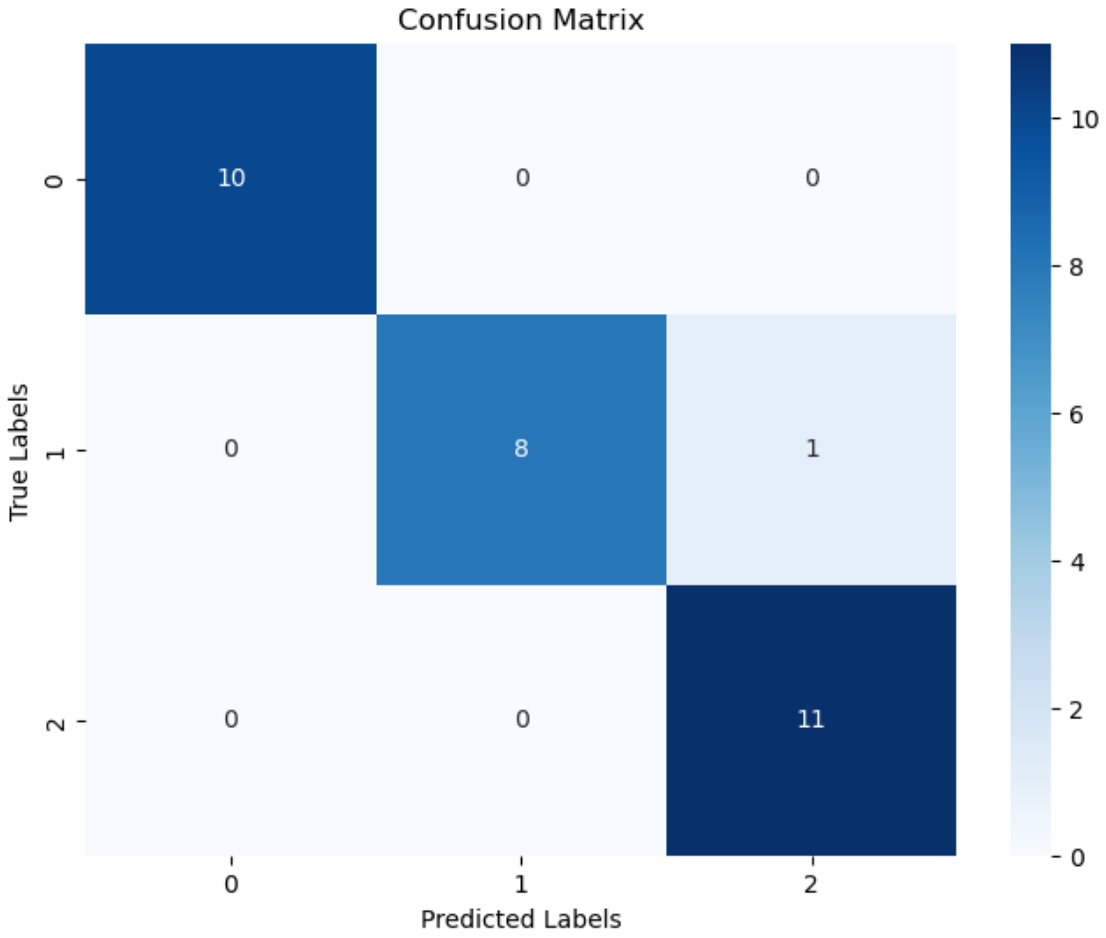
Total params: 731 (2.86 KB)

Trainable params: 243 (972.00 B)

Non-trainable params: 0 (0.00 B)

Optimizer params: 488 (1.91 KB)

[49]:
```
cm = confusion_matrix(ytest, ypredcl)
import seaborn as sns
# Plot confusion matrix using seaborn
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=np.unique(y),␣
 ↪yticklabels=np.unique(y))
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```


Confusion Matrix
10
0
0
0
8
1
0
0
11
0
1
2
True Labels
Predicted Labels
10
8
6
4
2
0