

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
model = Sequential()
```

```
model.add(Dense(6, input_dim=13, activation='relu'))
model.add(Dense(6, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(500, activation='relu'))
model.add(Dense(200, activation='relu'))
model.add(Dense(50, activation='relu'))
model.add(Dense(2, activation='softmax'))
```

```
model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 6)	84
dense_1 (Dense)	(None, 6)	42
dense_2 (Dense)	(None, 100)	700
dense_3 (Dense)	(None, 500)	50500
dense_4 (Dense)	(None, 200)	100200
dense_5 (Dense)	(None, 50)	10050
dense_6 (Dense)	(None, 2)	102
Total params: 161678 (631.55 KB)		
Trainable params: 161678 (631.55 KB)		
Non-trainable params: 0 (0.00 Byte)		

```
num_params = model.count_params()
print("Number of parameters in the model:", num_params)
```

```
Number of parameters in the model: 161678
```

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.utils import to_categorical
```

```
df = pd.read_csv('/content/heart_disease_dataset_UCI.csv')
```

```
X = df.drop('target', axis=1)
y = df['target']
```

```
y = to_categorical(y, num_classes=2)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

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

```
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_split=0.2)
```

```
Epoch 1/10
7/7 [=====] - 2s 55ms/step - loss: 0.6629 - accuracy: 0.6166 - val_loss: 0.5718 - val_accuracy: 0.8163
Epoch 2/10
7/7 [=====] - 0s 14ms/step - loss: 0.6183 - accuracy: 0.6736 - val_loss: 0.5153 - val_accuracy: 0.7959
Epoch 3/10
7/7 [=====] - 0s 10ms/step - loss: 0.6122 - accuracy: 0.6943 - val_loss: 0.5401 - val_accuracy: 0.7755
```

```
Epoch 4/10
7/7 [=====] - 0s 11ms/step - loss: 0.6088 - accuracy: 0.6891 - val_loss: 0.5912 - val_accuracy: 0.7755
Epoch 5/10
7/7 [=====] - 0s 11ms/step - loss: 0.5843 - accuracy: 0.7254 - val_loss: 0.5214 - val_accuracy: 0.8367
Epoch 6/10
7/7 [=====] - 0s 13ms/step - loss: 0.5710 - accuracy: 0.7202 - val_loss: 0.4862 - val_accuracy: 0.8367
Epoch 7/10
7/7 [=====] - 0s 14ms/step - loss: 0.5462 - accuracy: 0.7565 - val_loss: 0.4908 - val_accuracy: 0.7755
Epoch 8/10
7/7 [=====] - 0s 10ms/step - loss: 0.5353 - accuracy: 0.7565 - val_loss: 0.4759 - val_accuracy: 0.7959
Epoch 9/10
7/7 [=====] - 0s 10ms/step - loss: 0.5230 - accuracy: 0.7513 - val_loss: 0.4556 - val_accuracy: 0.8163
Epoch 10/10
7/7 [=====] - 0s 11ms/step - loss: 0.5054 - accuracy: 0.7617 - val_loss: 0.4701 - val_accuracy: 0.8163
<keras.src.callbacks.History at 0x7bd4fcb50e80>

test_loss, test_acc = model.evaluate(X_test, y_test)

2/2 [=====] - 0s 11ms/step - loss: 0.5951 - accuracy: 0.7541

print(f'Test Accuracy: {test_acc * 100:.2f}%')

Test Accuracy: 75.41%
```