

```
import numpy as np
import tensorflow as tf
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from tensorflow import keras
from tensorflow.keras import layers

# Load the Breast Cancer dataset
data = load_breast_cancer()
X = data.data
y = data.target

# Convert labels to one-hot encoding
y_one_hot = tf.keras.utils.to_categorical(y, num_classes=2)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y_one_hot, test_size=0.2,
random_state=42)

# Build the neural network model
model = keras.Sequential([
    layers.Input(shape=(30,)),
    layers.Dense(128, activation='relu'),
    layers.Dense(64, activation='relu'),
    layers.Dense(2, activation='softmax')
])

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

# Train the model
batch_size = 32
epochs = 50
model.fit(X_train, y_train, batch_size=batch_size, epochs=epochs,
validation_split=0.2)

Epoch 23/50
12/12 [=====] - 0s 5ms/step - loss: 0.2811 - accuracy: 0.9093 - val_loss: 1.9171 - val_accuracy: 0.7912
```

```

12/12 [=====] - 0s 8ms/step - loss: 0.6764 - accuracy: 0.8874 - val_loss: 0.7125 - val_accuracy: 0.9121
Epoch 47/50
12/12 [=====] - 0s 11ms/step - loss: 0.6394 - accuracy: 0.8956 - val_loss: 0.3513 - val_accuracy: 0.9011
Epoch 48/50
12/12 [=====] - 0s 8ms/step - loss: 0.3453 - accuracy: 0.9176 - val_loss: 0.3208 - val_accuracy: 0.9011
Epoch 49/50
12/12 [=====] - 0s 14ms/step - loss: 0.2934 - accuracy: 0.9286 - val_loss: 0.6414 - val_accuracy: 0.8571
Epoch 50/50
12/12 [=====] - 0s 8ms/step - loss: 0.9091 - accuracy: 0.8269 - val_loss: 0.8989 - val_accuracy: 0.8462
<keras.src.callbacks.History at 0x794c7c12ddb0>

```

# Evaluate the model on the test data

```

test_loss, test_accuracy = model.evaluate(X_test, y_test)
print(f"Test Loss: {test_loss:.4f}, Test Accuracy: {test_accuracy:.4f}")

```

```

4/4 [=====] - 0s 4ms/step - loss: 0.5013 - accuracy: 0.9123
Test Loss: 0.5013, Test Accuracy: 0.9123

```

## After Optimization

```

model = keras.Sequential([
    layers.Input(shape=(30,)),
    layers.Dense(128, activation='relu',
kernel_regularizer=keras.regularizers.l2(0.01)),
    layers.Dropout(0.5), # Adding dropout with a rate of 0.5
    layers.Dense(64, activation='relu',
kernel_regularizer=keras.regularizers.l2(0.01)),
    layers.Dropout(0.5), # Adding dropout with a rate of 0.5
    layers.Dense(2, activation='softmax')
])

```

# Compile the model

```

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

```

# Implement early stopping

```

early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

```

# Train the model with early stopping

```

batch_size = 32
epochs = 100
model.fit(X_train, y_train, batch_size=batch_size, epochs=epochs,
validation_split=0.2, callbacks=[early_stopping])

```

```

Epoch 1/100
12/12 [=====] - 3s 77ms/step - loss: 88.7117 - accuracy: 0.4423 - val_loss: 18.4126 - val_accuracy: 0.36
Epoch 2/100
12/12 [=====] - 0s 19ms/step - loss: 68.2886 - accuracy: 0.4533 - val_loss: 8.0658 - val_accuracy: 0.637
Epoch 3/100
12/12 [=====] - 0s 15ms/step - loss: 54.7058 - accuracy: 0.4533 - val_loss: 7.0623 - val_accuracy: 0.692
Epoch 4/100
12/12 [=====] - 0s 11ms/step - loss: 37.5114 - accuracy: 0.5110 - val_loss: 4.6710 - val_accuracy: 0.417
Epoch 5/100
12/12 [=====] - 0s 9ms/step - loss: 29.9192 - accuracy: 0.4945 - val_loss: 2.9180 - val_accuracy: 0.4945
Epoch 6/100
12/12 [=====] - 0s 14ms/step - loss: 21.1474 - accuracy: 0.5495 - val_loss: 2.0992 - val_accuracy: 0.637
Epoch 7/100
12/12 [=====] - 0s 10ms/step - loss: 17.7907 - accuracy: 0.5797 - val_loss: 4.5216 - val_accuracy: 0.362
Epoch 8/100
12/12 [=====] - 0s 10ms/step - loss: 16.0145 - accuracy: 0.5769 - val_loss: 3.7381 - val_accuracy: 0.362
Epoch 9/100
12/12 [=====] - 0s 16ms/step - loss: 13.4806 - accuracy: 0.5989 - val_loss: 1.7191 - val_accuracy: 0.648
Epoch 10/100
12/12 [=====] - 0s 10ms/step - loss: 12.3094 - accuracy: 0.6346 - val_loss: 1.6981 - val_accuracy: 0.648
Epoch 11/100
12/12 [=====] - 0s 8ms/step - loss: 10.6901 - accuracy: 0.5632 - val_loss: 2.5266 - val_accuracy: 0.4396
Epoch 12/100
12/12 [=====] - 0s 9ms/step - loss: 9.3414 - accuracy: 0.5604 - val_loss: 2.0052 - val_accuracy: 0.5934
Epoch 13/100
12/12 [=====] - 0s 14ms/step - loss: 7.2335 - accuracy: 0.6044 - val_loss: 1.1391 - val_accuracy: 0.8022
Epoch 14/100
12/12 [=====] - 0s 14ms/step - loss: 5.7486 - accuracy: 0.6731 - val_loss: 1.2031 - val_accuracy: 0.8791
Epoch 15/100
12/12 [=====] - 0s 11ms/step - loss: 6.3566 - accuracy: 0.6291 - val_loss: 1.0508 - val_accuracy: 0.8901
Epoch 16/100
12/12 [=====] - 0s 12ms/step - loss: 4.8384 - accuracy: 0.6456 - val_loss: 1.0156 - val_accuracy: 0.8681
Epoch 17/100
12/12 [=====] - 0s 11ms/step - loss: 4.0696 - accuracy: 0.6511 - val_loss: 1.0141 - val_accuracy: 0.8352
Epoch 18/100
12/12 [=====] - 0s 11ms/step - loss: 4.0235 - accuracy: 0.6786 - val_loss: 1.0041 - val_accuracy: 0.8681
Epoch 19/100
12/12 [=====] - 0s 10ms/step - loss: 3.8154 - accuracy: 0.6896 - val_loss: 0.9977 - val_accuracy: 0.8791

```

```
Epoch 20/100
12/12 [=====] - 0s 11ms/step - loss: 3.8155 - accuracy: 0.6703 - val_loss: 1.0014 - val_accuracy: 0.8791
Epoch 21/100
12/12 [=====] - 0s 14ms/step - loss: 3.2063 - accuracy: 0.6648 - val_loss: 0.9995 - val_accuracy: 0.8571
Epoch 22/100
12/12 [=====] - 0s 14ms/step - loss: 3.3921 - accuracy: 0.6209 - val_loss: 0.9837 - val_accuracy: 0.8681
Epoch 23/100
12/12 [=====] - 0s 11ms/step - loss: 2.4806 - accuracy: 0.6731 - val_loss: 1.0137 - val_accuracy: 0.8681
Epoch 24/100
12/12 [=====] - 0s 15ms/step - loss: 2.4355 - accuracy: 0.7253 - val_loss: 1.0362 - val_accuracy: 0.8571
Epoch 25/100
12/12 [=====] - 0s 9ms/step - loss: 2.4601 - accuracy: 0.7363 - val_loss: 1.0389 - val_accuracy: 0.8791
Epoch 26/100
12/12 [=====] - 0s 10ms/step - loss: 1.9377 - accuracy: 0.6923 - val_loss: 1.0574 - val_accuracy: 0.8791
Epoch 27/100
12/12 [=====] - 0s 11ms/step - loss: 2.1365 - accuracy: 0.7335 - val_loss: 1.1048 - val_accuracy: 0.7912
Epoch 28/100
12/12 [=====] - 0s 12ms/step - loss: 2.3039 - accuracy: 0.7115 - val_loss: 1.1062 - val_accuracy: 0.8132
```

```
test_loss, test_accuracy = model.evaluate(X_test, y_test)
print(f"Test Loss: {test_loss:.4f}, Test Accuracy: {test_accuracy:.4f}")
```

```
4/4 [=====] - 0s 4ms/step - loss: 0.8682 - accuracy: 0.9386
Test Loss: 0.8682, Test Accuracy: 0.9386
```