NAME : SAHIL SINGH

REG.NO : 20233240

SECTION : C

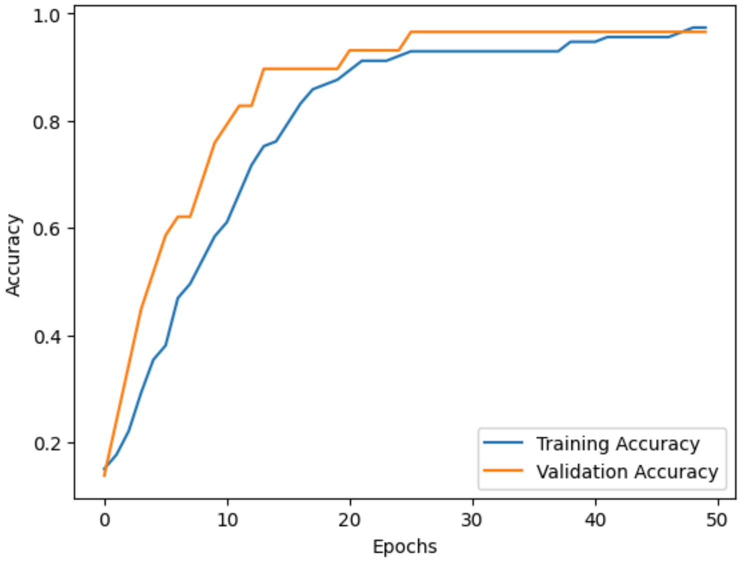
ASSIGNMENT : 6

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.utils import to_categorical

import torch
import torch.nn as nn
import torch.optim as optim

#TASK 1: DATA LOADING AND PREPROCESSING
#load
wine = load_wine()
x = wine.data
y = wine.target
#encode
y = to_categorical(y)
#split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
#normalise
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

#TASK 2: ANN MODEL DESIGN
model = Sequential()
model.add(Dense(units=16, activation='relu', input_shape=(x_train.shape[1],)))
model.add(Dense(units=3, activation='softmax'))

#TASK 3: COMPILATION AND TRAINING
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
history = model.fit(x_train, y_train, epochs=50, batch_size=16, validation_split=0.2)

#TASK 4: MODEL EVALUATION
loss, accuracy = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {accuracy * 100:.2f}%")

"""TRAINING PLOT:"""

import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

"""CONFUSION MATRIX :"""

from sklearn.metrics import confusion_matrix, classification_report
import numpy as np

y_pred_prob = model.predict(x_test)
y_pred = np.argmax(y_pred_prob, axis=1)
y_true = np.argmax(y_test, axis=1)

cm = confusion_matrix(y_true, y_pred)
print("Confusion Matrix:\n", cm)

print("\nClassification Report:\n", classification_report(y_true, y_pred))

"""REPORT :

Report: Wine Dataset Classification using ANN
✅ Dataset and Model Description

    Dataset: Wine dataset from sklearn.datasets
        Features: 13 chemical properties of wines
        Classes: 3 wine cultivars (Class 0, Class 1, Class 2)
```

```
        Samples: 178 total
    Model: Artificial Neural Network (ANN) using Keras Sequential API
        Architecture:
            Input Layer: 13 features
            Hidden Layer: 16 neurons, ReLU activation
            Output Layer: 3 neurons, Softmax activation
        Loss Function: Categorical Crossentropy
        Optimizer: Adam
        Epochs: 50
        Batch Size: 16

✅ Performance Metrics

    Test Accuracy: 97%
    Confusion Matrix:

        [[14  0  0]
        [ 1 13  0]
        [ 0  0  8]]

        Classification Report:
        Class Precision Recall  F1-Score  Support
        0 0.93  1.00  0.97  14
        1 1.00  0.93  0.96  14
        2 1.00  1.00  1.00  8

            Overall Accuracy: 97%
            Macro Avg F1-score: 0.98


✅ Observations

    - The training and validation accuracy plot shows smooth learning with no major overfitting. Both curves converge near 100%.
    - Class 2 was perfectly classified with 100% precision, recall, and F1-score.
    - Class 1 had one sample misclassified as Class 0.
    - Overall, the model demonstrates high accuracy and excellent generalization.
    - The ANN effectively learned from the wine dataset, making it reliable for predicting the wine classes.
"""
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim`
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 44ms/step - accuracy: 0.1427 - loss: 1.7507 - val_accuracy: 0.1379 - val_loss: 1.4773
Epoch 2/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.1573 - loss: 1.6030 - val_accuracy: 0.2414 - val_loss: 1.3743
Epoch 3/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - accuracy: 0.2115 - loss: 1.5415 - val_accuracy: 0.3448 - val_loss: 1.2765
Epoch 4/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.3473 - loss: 1.3043 - val_accuracy: 0.4483 - val_loss: 1.1872
Epoch 5/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.4089 - loss: 1.2194 - val_accuracy: 0.5172 - val_loss: 1.1072
Epoch 6/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 56ms/step - accuracy: 0.3891 - loss: 1.1966 - val_accuracy: 0.5862 - val_loss: 1.0307
Epoch 7/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 0.4779 - loss: 1.1578 - val_accuracy: 0.6207 - val_loss: 0.9609
Epoch 8/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - accuracy: 0.4551 - loss: 1.0974 - val_accuracy: 0.6207 - val_loss: 0.8971
Epoch 9/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step - accuracy: 0.4927 - loss: 1.0275 - val_accuracy: 0.6897 - val_loss: 0.8384
Epoch 10/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 0.5512 - loss: 0.9738 - val_accuracy: 0.7586 - val_loss: 0.7853
Epoch 11/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step - accuracy: 0.6010 - loss: 0.8991 - val_accuracy: 0.7931 - val_loss: 0.7338
Epoch 12/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step - accuracy: 0.6709 - loss: 0.8215 - val_accuracy: 0.8276 - val_loss: 0.6870
Epoch 13/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - accuracy: 0.7236 - loss: 0.7463 - val_accuracy: 0.8276 - val_loss: 0.6466
Epoch 14/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - accuracy: 0.7536 - loss: 0.7054 - val_accuracy: 0.8966 - val_loss: 0.6073
Epoch 15/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 0.7546 - loss: 0.6909 - val_accuracy: 0.8966 - val_loss: 0.5725
Epoch 16/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - accuracy: 0.8076 - loss: 0.6512 - val_accuracy: 0.8966 - val_loss: 0.5399
Epoch 17/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - accuracy: 0.8507 - loss: 0.5953 - val_accuracy: 0.8966 - val_loss: 0.5095
Epoch 18/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - accuracy: 0.8755 - loss: 0.5618 - val_accuracy: 0.8966 - val_loss: 0.4825
Epoch 19/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - accuracy: 0.8403 - loss: 0.5810 - val_accuracy: 0.8966 - val_loss: 0.4586
Epoch 20/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - accuracy: 0.8247 - loss: 0.5604 - val_accuracy: 0.8966 - val_loss: 0.4367
Epoch 21/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - accuracy: 0.8837 - loss: 0.4976 - val_accuracy: 0.9310 - val_loss: 0.4150
Epoch 22/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 0.8872 - loss: 0.5020 - val_accuracy: 0.9310 - val_loss: 0.3946
Epoch 23/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - accuracy: 0.9043 - loss: 0.4597 - val_accuracy: 0.9310 - val_loss: 0.3762
Epoch 24/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - accuracy: 0.9007 - loss: 0.4268 - val_accuracy: 0.9310 - val_loss: 0.3594
Epoch 25/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - accuracy: 0.8974 - loss: 0.4260 - val_accuracy: 0.9310 - val_loss: 0.3440
Epoch 26/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.9136 - loss: 0.4064 - val_accuracy: 0.9655 - val_loss: 0.3306
Epoch 27/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - accuracy: 0.9006 - loss: 0.4096 - val_accuracy: 0.9655 - val_loss: 0.3178
Epoch 28/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.8960 - loss: 0.3589 - val_accuracy: 0.9655 - val_loss: 0.3055
Epoch 29/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - accuracy: 0.9378 - loss: 0.3359 - val_accuracy: 0.9655 - val_loss: 0.2945
Epoch 30/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - accuracy: 0.9179 - loss: 0.3427 - val_accuracy: 0.9655 - val_loss: 0.2843
Epoch 31/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.9166 - loss: 0.3217 - val_accuracy: 0.9655 - val_loss: 0.2746
Epoch 32/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.9180 - loss: 0.3006 - val_accuracy: 0.9655 - val_loss: 0.2656
Epoch 33/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - accuracy: 0.9483 - loss: 0.2723 - val_accuracy: 0.9655 - val_loss: 0.2570
Epoch 34/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.9339 - loss: 0.2696 - val_accuracy: 0.9655 - val_loss: 0.2481
Epoch 35/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.9172 - loss: 0.2916 - val_accuracy: 0.9655 - val_loss: 0.2401
Epoch 36/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - accuracy: 0.9252 - loss: 0.2686 - val_accuracy: 0.9655 - val_loss: 0.2325
Epoch 37/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.9205 - loss: 0.2466 - val_accuracy: 0.9655 - val_loss: 0.2255
Epoch 38/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.9141 - loss: 0.2656 - val_accuracy: 0.9655 - val_loss: 0.2194
Epoch 39/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.9507 - loss: 0.2264 - val_accuracy: 0.9655 - val_loss: 0.2139
Epoch 40/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.9408 - loss: 0.2197 - val_accuracy: 0.9655 - val_loss: 0.2086
Epoch 41/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.9446 - loss: 0.2166 - val_accuracy: 0.9655 - val_loss: 0.2038
Epoch 42/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.9634 - loss: 0.2012 - val_accuracy: 0.9655 - val_loss: 0.1991
Epoch 43/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - accuracy: 0.9285 - loss: 0.2334 - val_accuracy: 0.9655 - val_loss: 0.1944
Epoch 44/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - accuracy: 0.9579 - loss: 0.2012 - val_accuracy: 0.9655 - val_loss: 0.1899
```

```
Epoch 45/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.9718 - loss: 0.1715 - val_accuracy: 0.9655 - val_loss: 0.1856
Epoch 46/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - accuracy: 0.9751 - loss: 0.1707 - val_accuracy: 0.9655 - val_loss: 0.1813
Epoch 47/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - accuracy: 0.9654 - loss: 0.1746 - val_accuracy: 0.9655 - val_loss: 0.1779
Epoch 48/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - accuracy: 0.9781 - loss: 0.1734 - val_accuracy: 0.9655 - val_loss: 0.1742
Epoch 49/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.9698 - loss: 0.1917 - val_accuracy: 0.9655 - val_loss: 0.1707
Epoch 50/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - accuracy: 0.9698 - loss: 0.1627 - val_accuracy: 0.9655 - val_loss: 0.1677
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - accuracy: 0.9711 - loss: 0.1620
Test Accuracy: 97.22%
```




```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
```