Date : 16/9/25

```
TASK 9: Implement a QSVM on the Iris dataset using PennyLane
```

Aim: To implement a Quantum Support Vector Machine (QSVM) using PennyLane and scikit- learn, where the quantum kernel is constructed from a quantum feature map, and evaluate its performance on the Iris dataset for classification tasks.

Algorithm - QSVM Algorithm

1. Load dataset (Iris, 150 samples, 3 classes).
2. Preprocess
3. Quantum Feature Map
4. Quantum Kernel Construction
5. Train QSVM
6. Test QSVM
7. Evaluate performance

```
# Install required libraries (uncomment if needed)
!pip install -U scikit-learn pennylane qiskit-machine-learning pylatexenc seaborn matplotlib

import pennylane as qml
from pennylane import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt

# ------------------------------
# Load Iris dataset
# ------------------------------
df_iris = pd.read_csv("Iris.csv")

# Check actual column names
print("Columns in dataset:", df_iris.columns)

# Rename columns if needed
# The original column names are 'sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'
# We need to rename 'species' to 'variety' to match the expected column name in the original code
df_iris.rename(columns={
    'sepal_length': 'sepal.length',
    'sepal_width': 'sepal.width',
    'petal_length': 'petal.length',
    'petal_width': 'petal.width',
    'species': 'variety' # Corrected column name
}, inplace=True)

X = df_iris[['sepal.length', 'sepal.width', 'petal.length', 'petal.width']].values
y = df_iris['variety'].values

# Encode labels into integers
encoder = LabelEncoder()
y = encoder.fit_transform(y)

# Train-test split
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)

# ------------------------------
# Define Quantum Feature Map
# ------------------------------
n_qubits = 4
dev = qml.device("default.qubit", wires=n_qubits)

def feature_map(x):
    """Embedding classical features into quantum states"""
    for i in range(n_qubits):
        qml.Hadamard(wires=i)
        qml.RZ(x[i], wires=i)
    # Entanglement
    for i in range(n_qubits - 1):
        qml.CNOT(wires=[i, i+1])
        qml.RZ(x[i] * x[i+1], wires=i+1)
        qml.CNOT(wires=[i, i+1])
```

```
# Kernel evaluation circuit
@qml.qnode(dev)
def kernel_circuit(x1, x2):
    feature_map(x1)
    qml.adjoint(feature_map)(x2)
    return qml.state()

# Kernel function: fidelity between states
def kernel(x1, x2):
    state = kernel_circuit(x1, x2)
    return np.abs(state[0])**2

# Compute Gram (kernel) matrix
def compute_kernel_matrix(X1, X2):
    K = np.zeros((len(X1), len(X2)))
    for i, x1 in enumerate(X1):
        for j, x2 in enumerate(X2):
            K[i, j] = kernel(x1, x2)
    return K

K_train = compute_kernel_matrix(x_train, x_train)
K_test = compute_kernel_matrix(x_test, x_train)

# ------------------------------
# Train QSVM
# ------------------------------
qsvm_model = SVC(kernel="precomputed")
qsvm_model.fit(K_train, y_train)

# Predictions
y_pred = qsvm_model.predict(K_test)

print("\nConfusion Matrix")
print(confusion_matrix(y_test, y_pred))

print("\nClassification Report")
print(classification_report(y_test, y_pred, target_names=encoder.classes_))

# ------------------------------
# Test on a new input
# ------------------------------
new_point = np.array([[4.4, 4.4, 4.4, 4.4]])
K_new = compute_kernel_matrix(new_point, x_train)
pred_label = qsvm_model.predict(K_new)
print("\nPredicted flower type for (4.4, 4.4, 4.4, 4.4):",
      encoder.inverse_transform(pred_label)[0])

# ------------------------------
# Optional: Visualize circuits
# ------------------------------
sample_x = x_train[0]
sample_y = x_train[1]

@qml.qnode(dev)
def feature_map_circuit(x):
    feature_map(x)
    return qml.state()

print("\n--- Feature Map Circuit ---")
print(qml.draw(feature_map_circuit)(sample_x))
fig, ax = qml.draw_mpl(feature_map_circuit)(sample_x)
fig.set_size_inches(6, 3)
plt.show()

print("\n--- Kernel Circuit ---")
print(qml.draw(kernel_circuit)(sample_x, sample_y))
fig, ax = qml.draw_mpl(kernel_circuit)(sample_x, sample_y)
fig.set_size_inches(6, 3)
plt.show()
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.12/dist-packages (1.7.2)
Requirement already satisfied: pennylane in /usr/local/lib/python3.12/dist-packages (0.43.0)
Requirement already satisfied: qiskit-machine-learning in /usr/local/lib/python3.12/dist-packages (0.8.4)
Requirement already satisfied: pylatexenc in /usr/local/lib/python3.12/dist-packages (2.10)
Requirement already satisfied: seaborn in /usr/local/lib/python3.12/dist-packages (0.13.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.7)
Requirement already satisfied: numpy>=1.22.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (2.0.2)
Requirement already satisfied: scipy>=1.8.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (1.15.3)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (1.5.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from pennylane) (3.5)
Requirement already satisfied: rustworkx>=0.14.0 in /usr/local/lib/python3.12/dist-packages (from pennylane) (0.17.1)
Requirement already satisfied: autograd in /usr/local/lib/python3.12/dist-packages (from pennylane) (1.8.0)
Requirement already satisfied: appdirs in /usr/local/lib/python3.12/dist-packages (from pennylane) (1.4.4)
Requirement already satisfied: autoray==0.8.0 in /usr/local/lib/python3.12/dist-packages (from pennylane) (0.8.0)
Requirement already satisfied: cachetools in /usr/local/lib/python3.12/dist-packages (from pennylane) (5.5.2)
Requirement already satisfied: pennylane-lightning>=0.43 in /usr/local/lib/python3.12/dist-packages (from pennylane) (0.43.0)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from pennylane) (2.32.4)
Requirement already satisfied: tomlkit in /usr/local/lib/python3.12/dist-packages (from pennylane) (0.13.3)
Requirement already satisfied: typing_extensions in /usr/local/lib/python3.12/dist-packages (from pennylane) (4.15.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from pennylane) (25.0)
Requirement already satisfied: diastatic-malt in /usr/local/lib/python3.12/dist-packages (from pennylane) (2.15.2)
Requirement already satisfied: qiskit<2.0,>=1.0 in /usr/local/lib/python3.12/dist-packages (from qiskit-machine-learning) (1.4.5)
Requirement already satisfied: setuptools>=40.1 in /usr/local/lib/python3.12/dist-packages (from qiskit-machine-learning) (75.2.0)
Requirement already satisfied: dill>=0.3.4 in /usr/local/lib/python3.12/dist-packages (from qiskit-machine-learning) (0.3.8)
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.12/dist-packages (from seaborn) (2.2.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=3 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.2.5)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: scipy-openblas32>=0.3.26 in /usr/local/lib/python3.12/dist-packages (from pennylane-lightning>=0.43
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib) (1.17.0
Requirement already satisfied: sympy>=1.3 in /usr/local/lib/python3.12/dist-packages (from qiskit<2.0,>=1.0->qiskit-machine-learni
Requirement already satisfied: stevedore>=3.0.0 in /usr/local/lib/python3.12/dist-packages (from qiskit<2.0,>=1.0->qiskit-machine-
Requirement already satisfied: symengine<0.14,>=0.11 in /usr/local/lib/python3.12/dist-packages (from qiskit<2.0,>=1.0->qiskit-mac
Requirement already satisfied: astunparse in /usr/local/lib/python3.12/dist-packages (from diastatic-malt->pennylane) (1.6.3)
Requirement already satisfied: gast in /usr/local/lib/python3.12/dist-packages (from diastatic-malt->pennylane) (0.6.0)
Requirement already satisfied: termcolor in /usr/local/lib/python3.12/dist-packages (from diastatic-malt->pennylane) (3.2.0)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->pennylane) (3.4
```