```
!pip install pennylane
```

```
Collecting pennylane
  Downloading PennyLane-0.40.0-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: numpy<2.1 in /usr/local/lib/python3.11/dist-packages (from pennylane) (2.0.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from pennylane) (1.14.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages (from pennylane) (3.4.2)
Collecting rustworkx>=0.14.0 (from pennylane)
  Downloading rustworkx-0.16.0-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (10 kB)
Requirement already satisfied: autograd in /usr/local/lib/python3.11/dist-packages (from pennylane) (1.7.0)
Collecting tomlkit (from pennylane)
  Downloading tomlkit-0.13.2-py3-none-any.whl.metadata (2.7 kB)
Collecting appdirs (from pennylane)
  Downloading appdirs-1.4.4-py2.py3-none-any.whl.metadata (9.0 kB)
Collecting autoray>=0.6.11 (from pennylane)
  Downloading autoray-0.7.1-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: cachetools in /usr/local/lib/python3.11/dist-packages (from pennylane) (5.5.2)
Collecting pennylane-lightning>=0.40 (from pennylane)
  Downloading PennyLane_Lightning-0.40.0-cp311-cp311-manylinux_2_28_x86_64.whl.metadata (27 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from pennylane) (2.32.3)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.11/dist-packages (from pennylane) (4.13.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from pennylane) (24.2)
Collecting diastatic-malt (from pennylane)
  Downloading diastatic_malt-2.15.2-py3-none-any.whl.metadata (2.6 kB)
Collecting scipy-openblas32>=0.3.26 (from pennylane-lightning>=0.40->pennylane)
  Downloading scipy_openblas32-0.3.29.0.0-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (56 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.1/56.1 kB 4.6 MB/s eta 0:00:00
Requirement already satisfied: astunparse in /usr/local/lib/python3.11/dist-packages (from diastatic-malt->pennylane) (1.6.3)
Requirement already satisfied: gast in /usr/local/lib/python3.11/dist-packages (from diastatic-malt->pennylane) (0.6.0)
Requirement already satisfied: termcolor in /usr/local/lib/python3.11/dist-packages (from diastatic-malt->pennylane) (2.5.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->pennylane)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->pennylane) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->pennylane) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->pennylane) (2025.1
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from astunparse->diastatic-malt-
Requirement already satisfied: six<2.0,>=1.6.1 in /usr/local/lib/python3.11/dist-packages (from astunparse->diastatic-malt->pen
Downloading PennyLane-0.40.0-py3-none-any.whl (2.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 31.5 MB/s eta 0:00:00
Downloading autoray-0.7.1-py3-none-any.whl (930 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 930.8/930.8 kB 49.8 MB/s eta 0:00:00
Downloading PennyLane_Lightning-0.40.0-cp311-cp311-manylinux_2_28_x86_64.whl (2.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.4/2.4 MB 73.4 MB/s eta 0:00:00
Downloading rustworkx-0.16.0-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 81.2 MB/s eta 0:00:00
Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Downloading diastatic_malt-2.15.2-py3-none-any.whl (167 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 167.9/167.9 kB 11.7 MB/s eta 0:00:00
Downloading tomlkit-0.13.2-py3-none-any.whl (37 kB)
Downloading scipy_openblas32-0.3.29.0.0-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (8.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.6/8.6 MB 113.6 MB/s eta 0:00:00
Installing collected packages: appdirs, tomlkit, scipy-openblas32, rustworkx, autoray, diastatic-malt, pennylane-lightning, pe
Successfully installed appdirs-1.4.4 autoray-0.7.1 diastatic-malt-2.15.2 pennylane-0.40.0 pennylane-lightning-0.40.0 rustworkx
```

```
import torch
import torch.nn as nn
import torch.optim as optim
import pennylane as qml
import numpy as np



torch.manual_seed(42)
np.random.seed(42)

# Generate normally distributed data
num_samples = 1000
X_data = np.random.normal(0, 1, (num_samples, 4))
Y_data = np.random.normal(0, 1, (num_samples, 1))  # 1 output value

X_tensor = torch.tensor(X_data, dtype=torch.float32)
Y_tensor = torch.tensor(Y_data, dtype=torch.float32)



class ClassicalNN(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(ClassicalNN, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size)
        self.fc2 = nn.Linear(hidden_size, output_size)

    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

```
n_qubits = 4
dev = qml.device("default.qubit", wires=n_qubits)

@qml.qnode(dev, interface="torch", diff_method="parameter-shift")
def quantum_circuit(inputs, weights):
    inputs = inputs.float()
    weights = weights.float()

    for i in range(n_qubits):
        qml.RY(inputs[i], wires=i)

    for i in range(n_qubits):
        qml.RZ(weights[i], wires=i)
        qml.RY(weights[n_qubits + i], wires=i)
        qml.RZ(weights[2 * n_qubits + i], wires=i)

    for i in range(n_qubits - 1):
        qml.CNOT(wires=[i, i + 1])
    qml.CNOT(wires=[n_qubits - 1, 0])

    return qml.expval(qml.PauliZ(0))



class HybridModel(nn.Module):
    def __init__(self, classical_nn, n_qubits):
        super(HybridModel, self).__init__()
        self.classical_nn = classical_nn
        self.n_qubits = n_qubits
        self.q_weights = nn.Parameter(0.01 * torch.randn(3 * n_qubits, dtype=torch.float32, requires_grad=True))

    def forward(self, x):
        q_inputs = self.classical_nn(x)
        q_inputs = torch.tanh(q_inputs) * np.pi

        # Compute quantum outputs
        q_outs = [quantum_circuit(q_inputs[i], self.q_weights) for i in range(x.shape[0])]
        q_outs = torch.stack(q_outs).view(-1, 1)
        return q_outs



classical_nn = ClassicalNN(input_size=4, hidden_size=8, output_size=4)
hybrid_model = HybridModel(classical_nn, n_qubits)

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
hybrid_model.to(device)

optimizer = optim.Adam(hybrid_model.parameters(), lr=0.01)
loss_fn = nn.MSELoss()

X_tensor, Y_tensor = X_tensor.to(device), Y_tensor.to(device)

num_epochs = 10
batch_size = 32

for epoch in range(num_epochs):
    for i in range(0, len(X_tensor), batch_size):
        batch_x = X_tensor[i : i + batch_size]
        batch_y = Y_tensor[i : i + batch_size]

        optimizer.zero_grad()
        outputs = hybrid_model(batch_x)

        outputs = outputs.type(torch.float32)

        loss = loss_fn(outputs, batch_y)
        loss.backward()
        optimizer.step()

    print(f"Epoch {epoch+1}: Loss = {loss.item():.6f}")

print("Training complete!")
```

```
Epoch 1: Loss = 1.516780
Epoch 2: Loss = 1.468895
Epoch 3: Loss = 1.376399
Epoch 4: Loss = 1.263986
Epoch 5: Loss = 1.209350
Epoch 6: Loss = 1.152608
Epoch 7: Loss = 1.108297
Epoch 8: Loss = 1.066125
```

```
Epoch 9: Loss = 1.025168
Epoch 10: Loss = 1.009360
Training complete!
```

Start coding or generate with AI.