

LAB-02

Aim: To build a Multilayer Perceptron using PyTorch

```
from sklearn.datasets import load_iris
from sklearn import preprocessing
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import torch
import torch.nn as nn
import torch.optim as optim
from sklearn.metrics import accuracy_score

iris=load_iris()

x=iris.data
y=iris.target

xt,xte,yt,yte=train_test_split(x,y,test_size=0.3,random_state=27)

X_train_tensor = torch.tensor(xt, dtype=torch.float32)
X_test_tensor = torch.tensor(xte, dtype=torch.float32)
y_train_tensor = torch.tensor(yt, dtype=torch.long)
y_test_tensor = torch.tensor(yte, dtype=torch.long)

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

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

#hyperparameters
input_size = 4
hidden_size = 20
output_size = 3
learning_rate = 0.1
epochs = 100

#initialization
model = MLP(input_size, hidden_size, output_size)
```

```

criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=learning_rate)

for epoch in range(epochs):
    optimizer.zero_grad()
    outputs = model(X_train_tensor)
    loss = criterion(outputs, y_train_tensor)
    loss.backward()
    optimizer.step()

    if (epoch+1) % 10 == 0:
        print(f'Epoch [{epoch+1}/{epochs}], Loss: {loss.item():.4f}')

Epoch [10/100], Loss: 0.6134
Epoch [20/100], Loss: 0.4762
Epoch [30/100], Loss: 0.4400
Epoch [40/100], Loss: 0.4023
Epoch [50/100], Loss: 0.3701
Epoch [60/100], Loss: 0.3567
Epoch [70/100], Loss: 0.3389
Epoch [80/100], Loss: 0.3285
Epoch [90/100], Loss: 0.3196
Epoch [100/100], Loss: 0.3100

with torch.no_grad():
    outputs = model(X_test_tensor)
    _, predicted = torch.max(outputs, 1)
    accuracy = accuracy_score(yte, predicted.numpy())
    print(f'Accuracy on test set: {accuracy:.4f}')

Accuracy on test set: 0.8000

```