Developing a Neural Network Classification Model

AIM

To develop a neural network classification model for the given dataset.

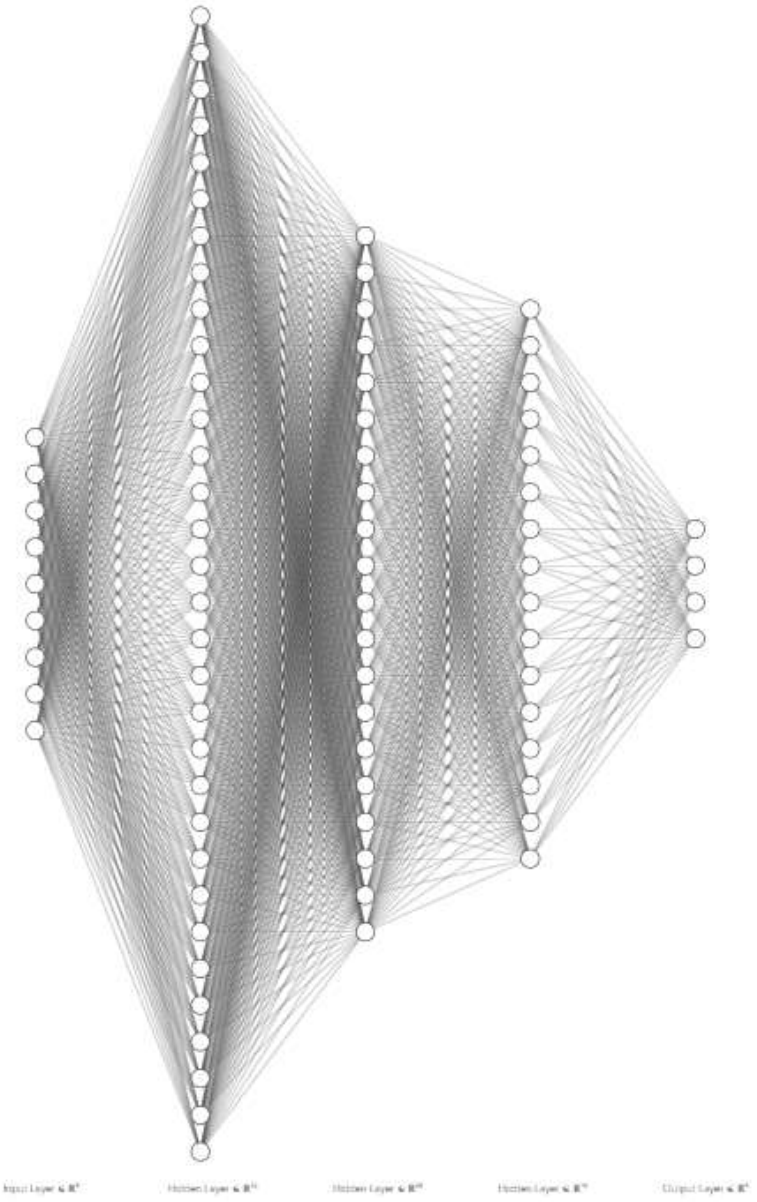
Problem Statement

An automobile company has plans to enter new markets with their existing products. After intensive market research, they’ve decided that the behavior of the new market is similar to their existing market.

In their existing market, the sales team has classified all customers into 4 segments (A, B, C, D). Then, they performed segmented outreach and communication for a different segment of customers. This strategy has work exceptionally well for them. They plan to use the same strategy for the new markets.

You are required to help the manager to predict the right group of the new customers.

Neural Network Model



DESIGN STEPS

STEP 1:

Import the required libraries for data handling and neural networks.

STEP 2:

Load the dataset and explore its structure.

STEP 3:

Clean the dataset and handle missing values if present.

STEP 4:

Encode categorical variables into numerical format.

STEP 5:

Normalize or scale the numerical features.

STEP 6:

Split the dataset into training and testing sets.

STEP 7:

Define the neural network architecture .

STEP 8:

Select CrossEntropyLoss as the loss function and Adam as the optimizer.

STEP 9:

Train the model using forward pass, loss calculation, backpropagation, and weight updates.

STEP 10:

Evaluate the model using accuracy, confusion matrix, and classification report.

PROGRAM

Name: A.LAHARI

Register Number: 212223230111

```
class PeopleClassifier(nn.Module):
    def __init__(self, input_size):
        super(PeopleClassifier, self).__init__()
        self.fc1 = nn.Linear(input_size, 32)
        self.fc2 = nn.Linear(32, 20)
        self.fc3 = nn.Linear(20, 16)
        self.fc4 = nn.Linear(16, 4)

    def forward(self, x):
        x=F.relu(self.fc1(x))
        x=F.relu(self.fc2(x))
        x=F.relu(self.fc3(x))
```

```
        x=self.fc4(x)
        return x
```

```
model = PeopleClassifier(input_size=X_train.shape[1])
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(),lr=0.01)
```

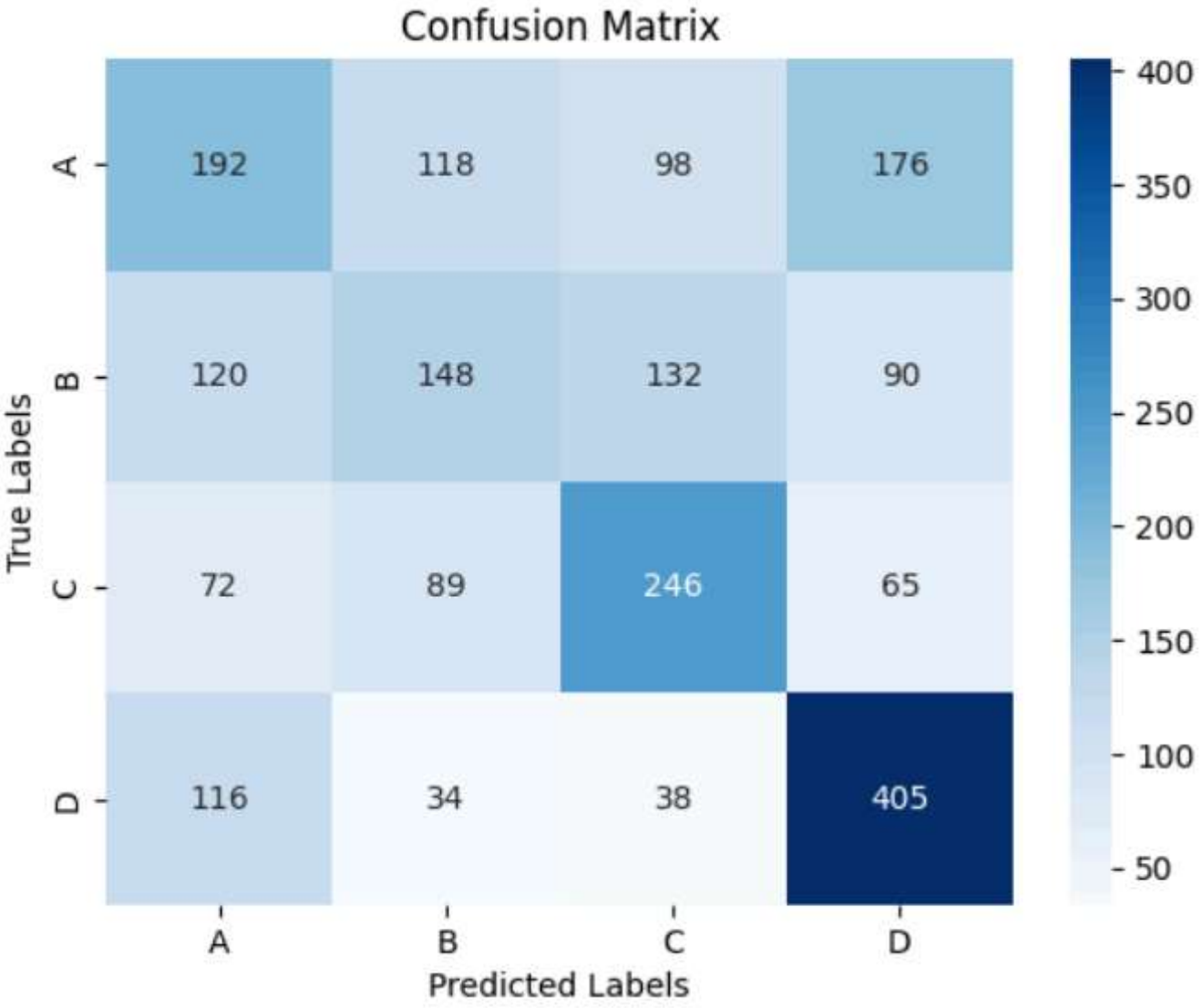
```
def train_model(model, train_loader,criterion,optimizer,epochs):
  for epoch in range(epochs):
    model.train()
    for X_batch, y_batch in train_loader:
      optimizer.zero_grad()
      output = model(X_batch)
      loss = criterion(output,y_batch)
      loss.backward()
      optimizer.step()

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

Dataset Information

	ID	Gender	Ever_Marri	Age	Graduated	Profession	Work_Expe	Spending_	Family_Siz	Var_1	Segmentation
2	462809	Male	No	22	No	Healthcare	1	Low	4	Cat_4	D
3	462643	Female	Yes	38	Yes	Engineer		Average	3	Cat_4	A
4	466315	Female	Yes	67	Yes	Engineer	1	Low	1	Cat_6	B
5	461735	Male	Yes	67	Yes	Lawyer	0	High	2	Cat_6	B
6	462669	Female	Yes	40	Yes	Entertainment		High	6	Cat_6	A
7	461319	Male	Yes	56	No	Artist	0	Average	2	Cat_6	C
8	460156	Male	No	32	Yes	Healthcare	1	Low	3	Cat_6	C
9	464347	Female	No	33	Yes	Healthcare	1	Low	3	Cat_6	D
10	465015	Female	Yes	61	Yes	Engineer	0	Low	3	Cat_7	D
11	465176	Female	Yes	55	Yes	Artist	1	Average	4	Cat_6	C
12	464041	Female	No	26	Yes	Engineer	1	Low	3	Cat_6	A
13	464942	Male	No	19	No	Healthcare	4	Low	4	Cat_4	D
14	461230	Female	No	19	No	Executive	0	Low		Cat_3	D
15	459573	Male	Yes	70	No	Lawyer		Low	1	Cat_6	A
16	460849	Female	Yes	58	No	Doctor	0	Low	1	Cat_3	B
17	460563	Female	No	41	No	Healthcare	1	Low	2	Cat_1	C
18	466865	Female	No	32	No	Homemak	9	Low	5	Cat_3	D
19	461644	Male	No	31	No	Healthcare	1	Low	6	Cat_6	B
20	466772	Male	Yes	58	Yes	Entertainm	1	Average	4	Cat_6	B
21	464291	Female	Yes	79	Yes	Artist	0	High	1	Cat_6	C
22	466084	Male	Yes	49	Yes	Homemak	12	Low	1	Cat_3	A
23	459675	Female	No	18	No	Healthcare	3	Low	4	Cat_6	D
24	465602	Male	Yes	33	Yes	Artist	13	Low	2	Cat_3	A
25	459168	Female	No	36	Yes	Artist	5	Low	2	Cat_6	B
26	461021	Female		58	No	Executive	1	Average	3	Cat_3	B
27	465083	Male	Yes	56	No	Artist	1	Average	3	Cat_6	C

OUTPUT

Confusion Matrix



Classification Report

```
Name: A.LAHARI
Register No: 212223230111
Test Accuracy: 0.46%
Confusion Matrix:
 [[192 118  98 176]
 [120 148 132  90]
 [ 72  89 246  65]
 [116  34  38 405]]
Classification Report:
              precision    recall  f1-score   support

           A       0.38      0.33      0.35       584
           B       0.38      0.30      0.34       490
           C       0.48      0.52      0.50       472
           D       0.55      0.68      0.61       593

    accuracy                           0.46      2139
   macro avg       0.45      0.46      0.45      2139
weighted avg       0.45      0.46      0.45      2139
```

New Sample Data Prediction

```
Name: A.LAHARI
Register No: 212223230111
Predicted class for sample input: D
Actual class for sample input: A
```

RESULT

Thus a neural network classification model for the given dataset is executed successfully.