Experiment 5: Build a simple MLP and implement forward propagation and Train the MLP on a small dataset

Aim: Building and Training a Simple Multi-Layer Perceptron (MLP)

Objective:

1. To construct a simple Multi-Layer Perceptron (MLP).

2. To implement forward propagation in the MLP.

3. To train the MLP on a small dataset.

4. To understand the learning process of an artificial neural network.

Code:

```
import numpy as np
import pandas as pd

from google.colab import drive
drive.mount('/content/drive')
!ls "/content/drive/My Drive/ANN/Student_dataset.xlsx"
```

```
# Initializing parameters
def init_parameters(layer_dimension):
  # print(f"layer_dimension : {layer_dimension}")
  np.random.seed(42)
  parameters = {}
  L = len(layer_dimension) # Total no. of layes in NN
  # print(f"L : {L}")
  for i in range(1, L):
    # print(f"i : {i}")
    parameters['w'+ str(i)] = np.random.randn(layer_dimension[i-1],
layer_dimension[i])*0.1
    # print("Weight parameters : ",parameters['w'+ str(i)])
    parameters['b'+ str(i)] = np.zeros((layer_dimension[i],1))
    # print("Bias parameters : ",parameters['b'+ str(i)])
  return parameters
```

```
# Forward propagation
def linear_forward(A_prev, W, B):
  print(f"W = {W}")
  print(f"A_prev : {A_prev}")
  print(f"np.dot(W.T,A_prev) : {np.dot(W.T,A_prev)}")
  print(f"B : {B}")
  print(f"np.dot(W.T,A_prev) + B : {np.dot(W.T,A_prev) + B}")
  z = np.dot(W.T,A_prev) + B
  return z
```

```
def relu(z):
  return np.maximum(0,z)

def sigmoid(z):
  return 1/(1 + np.exp(-z))

def L_layer_forward(X, parameters):
  A = X
  temp = []
  L = len(parameters) // 2
  for i in range(1,L):
    A_prev = A
    W = parameters['w' + str(i)]
    B = parameters['b' + str(i)]

    Z = linear_forward(A_prev, W, B )
    A = relu(Z)
    cache = [A_prev, W, B, Z]
    temp.append(cache)

  # Output layer
  W_out = parameters['w' + str(L)]
  B_out = parameters['b' + str(L)]
  Z_out = linear_forward(A, W_out, B_out)
  AL = sigmoid(Z_out)

  return AL, temp
```

```
# Example
import pandas as pd
file_path = "/content/drive/My Drive/ANN/Student_dataset.xlsx"
df = pd.read_excel(file_path)
print(f" Dataframe = \n{df}")
X = df[['CGPA', '10th Score', '12th Score', 'IQ']].values.T
print(f"X = \n{X}")
X = X / np.max(X, axis=1, keepdims=True)
parameters = init_parameters([4,2,1])
print(f"parameters length: {len(parameters)}")
y_cap, temp = L_layer_forward(X, parameters)
# print(temp)
print("Output = ",y_cap)
```

Output:


Experiment-5.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Connect
print(f"X = \n{X}")
X = X / np.max(X, axis=1, keepdims=True)
parameters = init_parameters([4,2,1])
print(f"parameters length: {len(parameters)}")
y_cap, temp = L_layer_forward(X, parameters)
# print(temp)
print("Output = ",y_cap)
Dataframe =
CGPA 10th Score 12th Score IQ Placement
0 8.5 85 88 120 1
1 7.2 78 74 110 0
2 9.1 90 92 130 1
3 6.8 70 65 105 0
4 7.5 75 78 115 0
5 8.0 80 81 118 1
6 7.9 79 77 113 1
7 8.3 83 85 125 1
8 6.5 65 60 100 0
9 9.0 92 95 135 1
X =
[[8.5 7.2 9.1 6.8 7.5 8. 7.9 8.3 6.5 9.]
[85. 78. 90. 70. 75. 80. 79. 83. 65. 92.]
[88. 74. 92. 65. 78. 81. 77. 85. 60. 95.]
[120. 110. 130. 105. 115. 118. 113. 125. 100. 135.]]
parameters length: 4
Output = [[0.49972471 0.49980193 0.49968288 0.49975641 0.49970206 0.49971128
0.49973745 0.49968452 0.4997432 0.49968828]]