9/9/25

6. Implement gradient decent & backpropagation in deep neural networks

AIM: To implement gradient descent and backpropagation in deep neural networks.

Objectives:

* To understand gradient descent in an optimized method.

* To implement back propogation in deep neural networks to update weights

* To implement a simple neural network for classification task

* To observe how loss decrease with iterations.

PSEUDOCODE:

BEGIN

Initialize weights and bias randomly

FOR epoch in range (max_epochs)

FOR each input sample :

*forward pass

compute $z = w * x_i + b$

apply activation to get A

compute output prediction

*compute loss:

loss = costly (y_true, y_pred)

*backward pass

compute gradients dw, db using chain rule

update parameters:

$w = w - a * dw$

$b = b - a * db$

END for

print after epoch

ENDFOR

END

OBSERVATION:

* Loss decreases as number of iterations increases
* weights and bias adjust to minimize error
* Backpropagation ensures errors are effectively distributed layer by layer
* Learning rate (η) greatly influences convergence speed.

Result:

Therefore implementation of gradient descent and backpropagation in neural network.

Output:

* Epoch 1/4:

  loss: 0.5954 training: 0.5607

* Epoch 2/4:

  loss: 0.5548 training: 0.5584

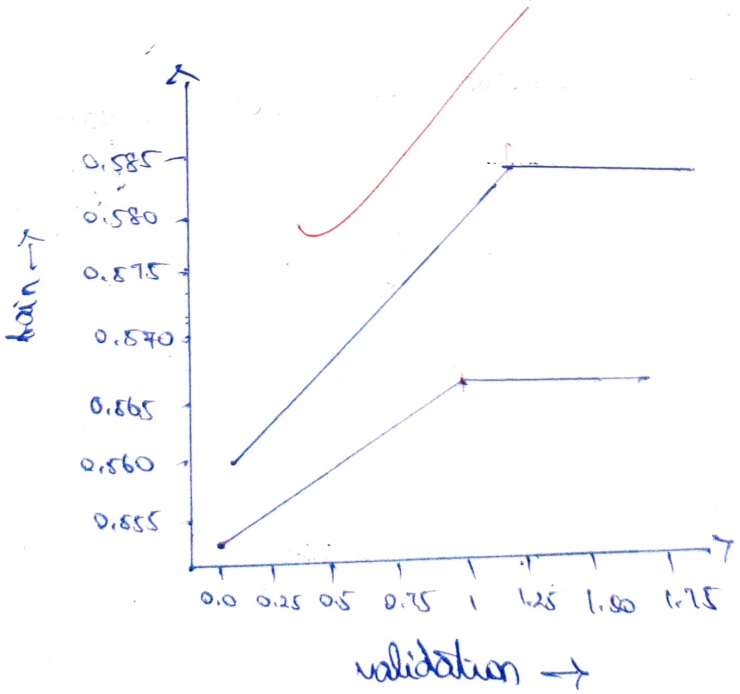
* Epoch 3/4:

  loss: 0.5441 training: 0.5887

* Epoch 4/4:

  loss: 0.5312 training: 0.6012

loss: 0.5312 val_loss: 0.5568



L6.ipynb (7) - JupyterLab

Not secure 10.1.38.19/user/ra2311047010019/lab/tree/DEEP%20LEARNING/L6.ipynb

File Edit View Run Kernel Tabs Settings Help

Logistic re | L_3.ipynb | L4.ipynb | jupyter-ra2 | L2.ipynb | EX1.ipynb | I5.ipynb | L6.ipynb | jupyter-ra2

Code Notebook Python 3 (ipykernel)

Filter files by name

/ DEEP LEARNING /

Name	Last Modified
data	25 days ago
EX1.ipynb	last month
L_3.ipynb	25 days ago
L2.ipynb	25 days ago
L3.ipynb	25 days ago
L4.ipynb	18 days ago
I5.ipynb	7 minutes ago
L6.ipynb	1 minute ago
Logistic reg...	last month

```
    z3 = a2 @ W3 + b3
    y_pred = torch.sigmoid(z3)
    predicted = (y_pred > 0.5).float()

print("Predictions:\n", predicted)
print("Ground Truth:\n", y)
```

```
Predictions:
 tensor([[0.],
         [1.],
         [1.],
         [0.]])
Ground Truth:
 tensor([[0.],
         [1.],
         [1.],
         [0.]])
```

[ ]:

Simple 2 7 Python 3 (ipykernel) | Idle Mem: 425.79 MB Mode: Command Ln 1, Col 1 L6.ipynb 0

Type here to search 9:13 AM 9/9/2025

L6.ipynb (7) - JupyterLab

Not secure 10.1.38.19/user/ra2311047010019/lab/tree/DEEP%20LEARNING/L6.ipynb

File Edit View Run Kernel Tabs Settings Help

Logistic re | L_3.ipynb | L4.ipynb | jupyter-ra2 | L2.ipynb | EX1.ipynb | I5.ipynb | L6.ipynb | jupyter-ra2

Code | Notebook | Python 3 (ipykernel)

Filter files by name

/ DEEP LEARNING /

Name	Last Modified
data	25 days ago
EX1.ipynb	last month
L_3.ipynb	25 days ago
L2.ipynb	25 days ago
L3.ipynb	25 days ago
L4.ipynb	18 days ago
I5.ipynb	6 minutes ago
L6.ipynb	1 minute ago
Logistic reg...	last month

[1]:
```
import torch
import torch.nn.functional as F
import matplotlib.pyplot as plt
```

[4]:
```
X = torch.tensor([[0, 0],
                  [0, 1],
                  [1, 0],
                  [1, 1]], dtype=torch.float32)
y = torch.tensor([[0], [1], [1], [0]], dtype=torch.float32)
```

[5]:
```
torch.manual_seed(42)
input_size = 2
hidden1 = 4
hidden2 = 4
output_size = 1
```

[6]:
```
W1 = torch.randn(input_size, hidden1, requires_grad=True)
b1 = torch.zeros(hidden1, requires_grad=True)
```

[7]:
```
W2 = torch.randn(hidden1, hidden2, requires_grad=True)
b2 = torch.zeros(hidden2, requires_grad=True)
```

[8]:
```
W3 = torch.randn(hidden2, output_size, requires_grad=True)
b3 = torch.zeros(output_size, requires_grad=True)
```

[9]:
```
lr = 0.1
epochs = 5000
```

Simple 2 7 Python 3 (ipykernel) | Idle Mem: 425.84 MB Mode: Command Ln 1, Col 1 L6.ipynb 0

Type here to search 9:12 AM 9/9/2025

Logistic re | L_3.ipynb | L4.ipynb | jupyter-ra2 | L2.ipynb | EX1.ipynb | I5.ipynb | L6.ipynb | jupyter-ra2

Code Notebook Python 3 (ipykernel)

```
    z3 = a2 @ W3 + b3
    y_pred = torch.sigmoid(z3)
    loss = F.binary_cross_entropy(y_pred, y)
    loss.backward()
    with torch.no_grad():
        W1 -= lr * W1.grad
        b1 -= lr * b1.grad

        W2 -= lr * W2.grad
        b2 -= lr * b2.grad

        W3 -= lr * W3.grad
        b3 -= lr * b3.grad

        # Zero the gradients
        W1.grad.zero_()
        b1.grad.zero_()
        W2.grad.zero_()
        b2.grad.zero_()
        W3.grad.zero_()
        b3.grad.zero_()

    loss_history.append(loss.item())

    if epoch % 500 == 0:
        print(f"Epoch {epoch} - Loss: {loss.item():.4f}")
```

```
Epoch 0 - Loss: 0.0134
Epoch 500 - Loss: 0.0107
```

Filter files by name
/ DEEP LEARNING /

Name	Last Modified
data	25 days ago
EX1.ipynb	last month
L_3.ipynb	25 days ago
L2.ipynb	25 days ago
L3.ipynb	25 days ago
L4.ipynb	18 days ago
I5.ipynb	6 minutes ago
L6.ipynb	1 minute ago
Logistic reg...	last month

L6.ipynb (7) - JupyterLab

Not secure 10.1.38.19/user/ra2311047010019/lab/tree/DEEP%20LEARNING/L6.ipynb

File Edit View Run Kernel Tabs Settings Help

Logistic re | L_3.ipynb | L4.ipynb | jupyter-ra2 | L2.ipynb | EX1.ipynb | I5.ipynb | L6.ipynb | jupyter-ra2

Code

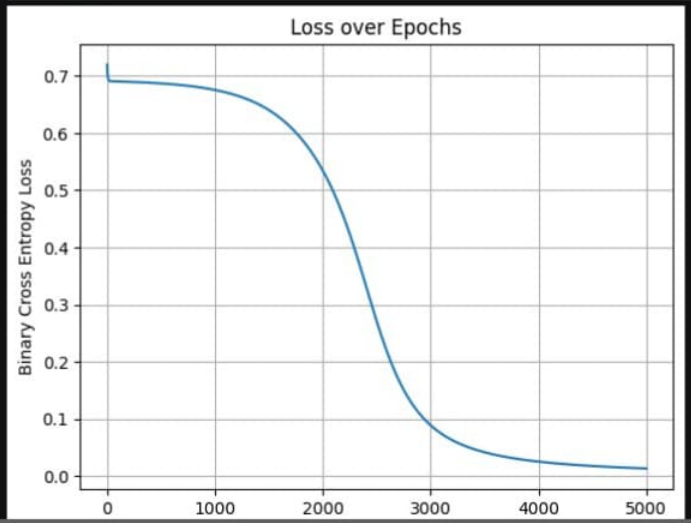
Notebook Python 3 (ipykernel)

Filter files by name

/ DEEP LEARNING /

Name	Last Modified
data	25 days ago
EX1.ipynb	last month
L_3.ipynb	25 days ago
L2.ipynb	25 days ago
L3.ipynb	25 days ago
L4.ipynb	18 days ago
I5.ipynb	6 minutes ago
L6.ipynb	1 minute ago
Logistic reg...	last month

```
plt.title("Loss over Epochs")
plt.xlabel("Epoch")
plt.ylabel("Binary Cross Entropy Loss")
plt.grid(True)
plt.show()
```



Simple 2 7 Python 3 (ipykernel) | Idle Mem: 425.80 MB Mode: Command Ln 1, Col 1 L6.ipynb 0

Type here to search 9:13 AM 9/9/2025