

The screenshot shows a Jupyter Notebook interface running in a browser window. The top bar displays several tabs: Network Login, exp6.ipynb (auto-D), Inbox (3,125) - ds107, ChatGPT, 21AIC301J- Lab AI-A, and Welcome To Colab. The URL in the address bar is 10.1.38.19/user/ra2311047010033/lab/workspaces/auto-D/tree/DLT/exp6.ipynb.

The notebook interface includes a file browser on the left showing files in the DLT directory:

Name	Last Modified
exp2.ipynb	last month
exp3.ipynb	last month
exp4.ipynb	last month
exp5.ipynb	14 days ago
exp6.ipynb	13 days ago
exp7.ipynb	36 minutes ago
Untitled.ipynb...	13 days ago

The main area contains a code cell [1]:

```
[1]: import numpy as np

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

def sigmoid_derivative(A):
    return A * (1 - A)

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

def relu_derivative(z):
    return (z > 0).astype(float)

def initialize_parameters(layer_dims):
    np.random.seed(1)
    parameters = {}
    L = len(layer_dims)
    for l in range(1, L):
        parameters['W' + str(l)] = np.random.randn(layer_dims[l], layer_dims[l-1]) * 0.01
        parameters['b' + str(l)] = np.zeros((layer_dims[l], 1))
    return parameters

def linear_forward(A_prev, W, b):
    Z = np.dot(W, A_prev) + b
    return Z

def forward_propagation(X, parameters):
```

The status bar at the bottom shows "Simple" mode, 0 cells run, Python 3 (ipykernel) | Idle, Mem: 325.05 MB, Mode: Command, Ln 1, Col 1, exp6.ipynb, and a timestamp of 10:36 23-09-2025.

The screenshot shows a Jupyter Notebook interface with several tabs open in the background:

- Network Login
- exp6.ipynb (auto-D :)
- Inbox (3,125) - ds107
- ChatGPT
- 21AIC301J- Lab AI-A
- Welcome To Colab -

The main window displays a file browser on the left and a code editor on the right. The file browser shows a directory structure under "/DLT/" with the following files:

Name	Last Modified
exp2.ipynb	last month
exp3.ipynb	last month
exp4.ipynb	last month
exp5.ipynb	14 days ago
exp6.ipynb	13 days ago
exp7.ipynb	36 minutes ago
Untitled.ipynb	13 days ago

The code editor contains Python code for a neural network implementation:

```
for l in range(1, L):
    A_prev = A
    Z = linear_forward(A_prev, parameters['W' + str(l)], parameters['b' + str(l)])
    A = relu(Z)
    caches['Z' + str(l)] = Z
    caches['A' + str(l-1)] = A_prev

    ZL = linear_forward(A, parameters['W' + str(L)], parameters['b' + str(L)])
    AL = sigmoid(ZL)
    caches['Z' + str(L)] = ZL
    caches['A' + str(L-1)] = A
    return AL, caches

def compute_cost(AL, Y):
    m = Y.shape[1]
    cost = -1/m * np.sum(Y * np.log(AL + 1e-8) + (1 - Y) * np.log(1 - AL + 1e-8))
    return np.squeeze(cost)

def linear_backward(dZ, A_prev, W):
    m = A_prev.shape[1]
    dW = 1/m * np.dot(dZ, A_prev.T)
    db = 1/m * np.sum(dZ, axis=1, keepdims=True)
    dA_prev = np.dot(W.T, dZ)
    return dA_prev, dW, db

def backward_propagation(AL, Y, caches, parameters):
    grads = {}
    L = len(parameters) // 2
    m = Y.shape[1]
```

The status bar at the bottom indicates "Mode: Command" and "Ln 1, Col 1". The taskbar at the bottom shows various application icons.

Network Login exp6.ipynb (auto-D) Inbox (3,125) - ds107 ChatGPT 21AIC301J- Lab AI-A Welcome To Colab -

Not secure 10.1.38.19/user/ra2311047010033/lab/workspaces/auto-D/tree/DLT/exp6.ipynb

File Edit View Run Kernel Tabs Settings Help

exp2.ipynb exp3.ipynb exp4.ipynb exp5.ipynb exp6.ipynb exp7.ipynb Untitled.ipynb

Code

Cost after iteration 2600: 0.3486
Cost after iteration 2700: 0.3485
Cost after iteration 2800: 0.3484
Cost after iteration 2900: 0.3483
Cost after iteration 3000: 0.3481
Cost after iteration 3100: 0.3482
Cost after iteration 3200: 0.3480
Cost after iteration 3300: 0.3488
Cost after iteration 3400: 0.3478
Cost after iteration 3500: 0.3479
Cost after iteration 3600: 0.3478
Cost after iteration 3700: 0.3477
Cost after iteration 3800: 0.3477
Cost after iteration 3900: 0.3476
Cost after iteration 4000: 0.3475
Cost after iteration 4100: 0.3476
Cost after iteration 4200: 0.3475
Cost after iteration 4300: 0.3475
Cost after iteration 4400: 0.3475
Cost after iteration 4500: 0.3474
Cost after iteration 4600: 0.3474
Cost after iteration 4700: 0.3474
Cost after iteration 4800: 0.3473
Cost after iteration 4900: 0.3473
Cost after iteration 5000: 0.3473
Cost after iteration 5100: 0.3473
Cost after iteration 5200: 0.3473
Cost after iteration 5300: 0.3472
Cost after iteration 5400: 0.3472
Cost after iteration 5500: 0.3472
Cost after iteration 5600: 0.3472
Cost after iteration 5700: 0.3472
Cost after iteration 5800: 0.3471
Cost after iteration 5900: 0.3471
Cost after iteration 6000: 0.3472
Cost after iteration 6100: 0.3471
Cost after iteration 6200: 0.3471
Cost after iteration 6300: 0.3471
Cost after iteration 6400: 0.3471
Cost after iteration 6500: 0.3471
Cost after iteration 6600: 0.3471
Cost after iteration 6700: 0.3471
Cost after iteration 6800: 0.3470
Cost after iteration 6900: 0.3470
Cost after iteration 7000: 0.3470
Cost after iteration 7100: 0.3470
Cost after iteration 7200: 0.3470
Cost after iteration 7300: 0.3470
Cost after iteration 7400: 0.3470
Cost after iteration 7500: 0.3470

Python 3 (ipykernel) | Idle Mem: 325.03 MB Mode: Edit ENG 10:37 US 23-09-2025

Type here to search

Network Login exp6.ipynb (auto-D :) Inbox (3,125) - ds107 ChatGPT 21AIC301J- Lab AI-A Welcome To Colab - +

Not secure 10.1.38.19/user/ra2311047010033/lab/workspaces/auto-D/tree/DLT/exp6.ipynb

File Edit View Run Kernel Tabs Settings Help

+ ☰ Filter files by name / DLT /

Name	Last Modified
exp2.ipynb	last month
exp3.ipynb	last month
exp4.ipynb	last month
exp5.ipynb	14 days ago
exp6.ipynb	13 days ago
exp7.ipynb	37 minutes ago
Untitled.ipynb...	13 days ago

Code

```
layers = [2, 4, 1]

params = train(X, Y, layers, learning_rate=0.1, num_iterations=10000)
predictions = predict(X, params)
print("Predictions:", predictions)
print("True labels:", Y)

Cost after iteration 0: 0.6931
Cost after iteration 100: 0.6929
Cost after iteration 200: 0.6920
Cost after iteration 300: 0.6867
Cost after iteration 400: 0.6614
Cost after iteration 500: 0.5870
Cost after iteration 600: 0.5045
Cost after iteration 700: 0.4449
Cost after iteration 800: 0.4027
Cost after iteration 900: 0.3792
Cost after iteration 1000: 0.3681
Cost after iteration 1100: 0.3621
Cost after iteration 1200: 0.3580
Cost after iteration 1300: 0.3556
Cost after iteration 1400: 0.3540
Cost after iteration 1500: 0.3527
Cost after iteration 1600: 0.3518
Cost after iteration 1700: 0.3512
Cost after iteration 1800: 0.3508
Cost after iteration 1900: 0.3502
Cost after iteration 2000: 0.3498
Cost after iteration 2100: 0.3495
```

Simple 0 \$ 5 Python 3 (ipykernel) | Idle Mem: 325.04 MB Mode: Edit 0 Ln 94, Col 20 exp6.ipynb 0 Type here to search ENG 10:37 US 23-09-2025

The screenshot shows a Jupyter Notebook interface with several tabs open in the top bar:

- Network Login
- exp6.ipynb (auto-D :)
- Inbox (3,125) - ds107
- ChatGPT
- 21AIC301J- Lab AI-A
- Welcome To Colab -

The main area displays a file browser on the left and a code editor on the right. The code editor contains the following Python script:

```
def train(X, Y, layer_dims, learning_rate=0.01, num_iterations=1000, print_cost=True):
    parameters = initialize_parameters(layer_dims)
    for i in range(num_iterations):
        AL, caches = forward_propagation(X, parameters)
        cost = compute_cost(AL, Y)
        grads = backward_propagation(AL, Y, caches, parameters)
        parameters = update_parameters(parameters, grads, learning_rate)

        if print_cost and i % 100 == 0:
            print(f"Cost after iteration {i}: {cost:.4f}")

    return parameters

def predict(X, parameters):
    AL, _ = forward_propagation(X, parameters)
    return (AL > 0.5).astype(int)

if __name__ == "__main__":
    X = np.array([[0, 0, 1, 1], [0, 1, 0, 1]])
    Y = np.array([[0, 1, 1, 0]])

    layers = [2, 4, 1]

    params = train(X, Y, layers, learning_rate=0.1, num_iterations=10000)
    predictions = predict(X, params)
    print("Predictions:", predictions)
    print("True labels:", Y)
```

The file browser on the left shows a directory structure under "/DLT/". The "exp6.ipynb" file is currently selected.

Network Login exp6.ipynb (auto-D : 1) Inbox (3,125) - ds107 ChatGPT 21AIC301J- Lab AI-A Welcome To Colab - +

Not secure 10.1.38.19/user/ra2311047010033/lab/workspaces/auto-D/tree/DLT/exp6.ipynb

File Edit View Run Kernel Tabs Settings Help

exp2.ipynb exp3.ipynb exp4.ipynb exp5.ipynb exp6.ipynb Notebook Python 3 (ipykernel)

Filter files by name / DLT /

Name	Last Modified
exp2.ipynb	last month
exp3.ipynb	last month
exp4.ipynb	last month
exp5.ipynb	14 days ago
exp6.ipynb	13 days ago
exp7.ipynb	36 minutes ago
Untitled.ipynb	13 days ago

```
L = len(parameters) // 2
m = Y.shape[1]

dAL = -(np.divide(Y, AL + 1e-8) - np.divide(1 - Y, 1 - AL + 1e-8))

ZL = caches['Z' + str(L)]
A_prev = caches['A' + str(L-1)]
dZL = dAL * sigmoid_derivative(sigmoid(ZL))
dA_prev, dW, db = linear_backward(dZL, A_prev, parameters['W' + str(L)])
grads['dW' + str(L)] = dW
grads['db' + str(L)] = db

dA = dA_prev

for l in reversed(range(1, L)):
    Z = caches['Z' + str(l)]
    A_prev = caches['A' + str(l-1)]
    dZ = dA * relu_derivative(Z)
    dA_prev, dW, db = linear_backward(dZ, A_prev, parameters['W' + str(l)])
    grads['dW' + str(l)] = dW
    grads['db' + str(l)] = db
    dA = dA_prev

return grads

def update_parameters(parameters, grads, learning_rate):
    L = len(parameters) // 2
    for l in range(1, L+1):
        parameters['W' + str(l)] -= learning_rate * grads['dW' + str(l)]
```

Simple 0 \$ 5 Python 3 (ipykernel) | Idle Mem: 325.05 MB Mode: Command Ln 1, Col 1 exp6.ipynb 0 Type here to search ENG US 10:36 23-09-2025

Network Login exp6.ipynb (auto-D) Inbox (3,125) - ds107 ChatGPT 21AIC301J- Lab AI-A Welcome To Colab - +

Not secure 10.1.38.19/user/ra2311047010033/lab/workspaces/auto-D/tree/DLT/exp6.ipynb

File Edit View Run Kernel Tabs Settings Help

exp2.ipynb exp3.ipynb exp4.ipynb exp5.ipynb exp6.ipynb exp7.ipynb Untitled.ipynb

Code

Filter files by name / DLT/

Name

- exp2.ipynb
- exp3.ipynb
- exp4.ipynb
- exp5.ipynb
- exp6.ipynb
- exp7.ipynb
- Untitled.ipynb

Notebook Python 3 (ipykernel)

Cost after iteration 7900: 0.3478
Cost after iteration 8000: 0.3478
Cost after iteration 8100: 0.3469
Cost after iteration 8200: 0.3469
Cost after iteration 8300: 0.3469
Cost after iteration 8400: 0.3469
Cost after iteration 8500: 0.3469
Cost after iteration 8600: 0.3469
Cost after iteration 8700: 0.3469
Cost after iteration 8800: 0.3469
Cost after iteration 8900: 0.3469
Cost after iteration 9000: 0.3469
Cost after iteration 9100: 0.3469
Cost after iteration 9200: 0.3469
Cost after iteration 9300: 0.3469
Cost after iteration 9400: 0.3469
Cost after iteration 9500: 0.3469
Cost after iteration 9600: 0.3469
Cost after iteration 9700: 0.3469
Cost after iteration 9800: 0.3469

Training Loss over Iterations

Cost

Iterations (per hundreds)

Simple 0 5 Python 3 (ipykernel) Idle Mem: 325.01 MB Mode: Edit ENG 10:37 US 23-09-2025

Type here to search

b: Implement gradient descent and backpropagation in deep neural network.

Aim:- To implement gradient descent and back propagation algorithm in a simple deep neural network and study their role in training

objectives:-

- 1) To understand the working of gradient descent as an optimization method.
- 2) To implement back propagation for updating weight in a neural network.
- 3) To train a simple neural network for a classification task using these algorithms.
- 4) To observe how loss decreases with iteration.

Pseudo code:-

Start:-

- 1) Initialize dataset (x, y) for training (e.g.: XOR dataset)
- 2) Initialize neural network parameters.
 - Input layer, hidden layer, output layer sizes.
 - Random weights, biases
 - Learning rate (η)

3. Define forward propagation.

$$\Rightarrow z_1 = w_1 \cdot x + b_1$$

$\Rightarrow A_1 = \text{activation}(z_1)$ # sigmoid or ReLU

$$\Rightarrow z_2 = w_2 \cdot A_1 + b_2$$

$\Rightarrow A_2 = \text{activation}(z_2)$ # sigmoid / Softmax (output)

4) compute loss using mean squared error (or) cross entropy.

5) Backpropagation:-

\Rightarrow compute error at output ($dA_2 = A_2 - y$)

\Rightarrow calculate gradients for w_2, b_2

6) update weights and biases using gradient descent.

$$w = w - \eta \cdot dw$$

$$b = b - \eta \cdot db$$

7) Repeat the steps 3-6 for given number of epochs

8) observe cost reduction and accuracy improvement
End

Observation:-

- \Rightarrow loss decreases as the number of iteration (epochs) increases
- \Rightarrow weights and biases adjust to minimize error.
- \Rightarrow backpropagation ensures errors are efficiently distributed layer by layer
- \Rightarrow learning rate (η): greatly influences convergence.
Speed

output:-

training

cost after iteration 0 :- 0.6931

cost after iteration 100 :- 0.6929

cost after iteration 200 :- 0.5870

cost after iteration 300 :- 0.4649

cost after iteration 400 :- 0.3792

cost after iteration 1100 :- 0.3621

cost after iteration 2000 :- 0.3498

cost after iteration 3000 :- 0.3481

cost after iteration 4000 :- 0.3475

cost after iteration 5000 :- 0.3473

cost after iteration 6000 :- 0.3472

cost after iteration 7000 :- 0.3470

cost after iteration 8000 :- 0.3470

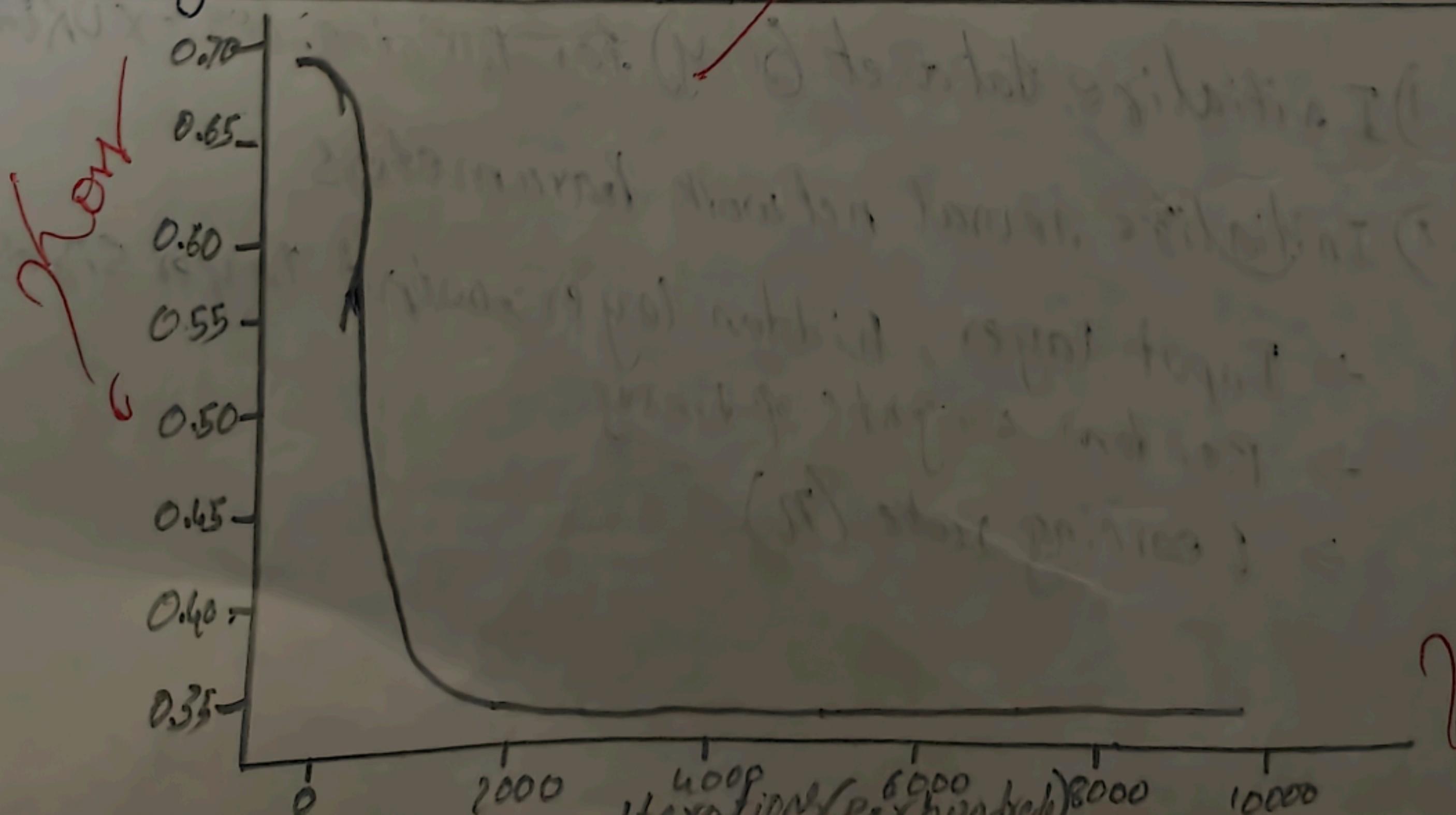
cost after iteration 9000 :- 0.3469

cost after iteration 9900 :- 0.3469

Prediction :- [0 0 1 0]

True labels :- [0 1 1 0] /

Training loss over Iterations.



2.000 bits stored in 1000 follow nodes. So 1000 bits stored in 1000 nodes.

batch delta

batch

connections



Forward propagation
Back propagation

Result :- Successfully implemented gradient descent
and back propagation in deep neural network