

Indian Institute of Technology, Indore
Computer Science & Engineering
CS 354N: Assignment VI - Backpropagation
Date- 11-02-2025

Some general instructions:

- Name your file in "Assignment6_yourRollno.pdf" format.
- You are not allowed to use in-built libraries related to the topic. Code everything from scratch.
- Submission of the assignment should be made using the Google Classroom platform only.
- Plagiarism in any form will not be tolerated.
- You are allowed to submit only one before the deadline. Avoid multiple submissions. In such a case, only the last submitted file will be used for evaluation.
- Last date for submission of the assignment: **17-02-2025**
- Submit a single file (report) containing the procedure (screenshot of main procedures/code/Results).

Q.1 Implement a feed-forward neural network with one layer of hidden nodes and a single output node. Use the sigmoid function as the activation function for all the hidden nodes and the output node. Train and test the neural network with the dataset using the **Diabetes Dataset** uploaded on the Google Classroom. Since the output node uses a sigmoid activation function that returns a number between 0 and 1, associate any output greater than 0.5 to one class and any output less than or equal to 0.5 to the other class. Use the following parameters:

- learning rate: 0.0001
- number of hidden nodes: 5 to 10
- number of Epochs: 30 Epochs
- You can initialize the weights randomly

Q.2 Implement a feed-forward neural network to classify the **Iris dataset**. Use the sigmoid function as the activation function for all the hidden nodes and the output node. Train and test the neural network using the dataset that has been uploaded to Google Classroom. Since the output node uses a sigmoid activation function that returns a number between 0 and 1, associate any output greater than 0.5 to one class and any output less than or equal to 0.5 to the other class.

Use class label 0 for the setosa class:

Use class label 1 for the versicolor class:

Use Back Propagation as a learning algorithm

- learning rate: 0.0001

- You can choose more than one hidden layer, and each hidden layer will have 5-15 hidden nodes.
- Number of Epochs: 35 Epochs
- You can initialize the weights randomly.

Note: All the programs must be user-friendly and user-interactive.