**Batch: B4**

**Experiment Number:02**

**Roll Number: 16010421119**

**Name: Aarya Tiwari**

**Title of the Experiment: Feed Forward Neural Network**



**Program:**

import numpy as np

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

* Step 2: Download a simple dataset for classification iris = load\_iris()

X, y = iris.data, iris.target

* Convert labels to one-hot encoding

num\_classes = len(np.unique(y))

y\_one\_hot = np.eye(num\_classes)[y]

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_one\_hot, test\_size=0.2, random\_state=42)

# Step 3: Create sample weights for each layer

input\_layer\_weights = np.random.randn(X.shape[1], 5)

hidden\_layer\_weights = np.random.randn(5, num\_classes)

* Step 4: Propagate input signal through different layers def softmax(x):

exp\_x = np.exp(x - np.max(x, axis=-1, keepdims=True)) return exp\_x / np.sum(exp\_x, axis=-1, keepdims=True)

* Step 5: Forward propagate input signals to neurons in the hidden layer, use tanh function

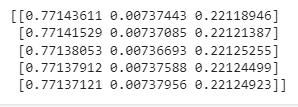
hidden\_layer\_input = np.dot(X\_train, input\_layer\_weights) hidden\_layer\_output = np.tanh(hidden\_layer\_input)

* Step 6: Forward propagate activation signals from the hidden layer to neurons in the output layer, use softmax function output\_layer\_input = np.dot(hidden\_layer\_output, hidden\_layer\_weights) output\_probs = softmax(output\_layer\_input)
* Print the output probabilities for the first few samples

print(output\_probs[:5])

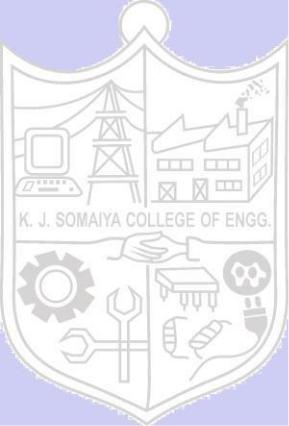


**Output:**



**Post Lab Question- Answers (If Any):**

1. **What is perceptron?**
   1. **a single layer feed-forward neural network with pre-processing**
   2. an auto-associative neural network
   3. a double layer auto-associative neural network
   4. a neural network that contains feedback
2. **A 4-input neuron has weights 1, 2, 3 and 4. The transfer function is linear with the constant of proportionality being equal to 2. The inputs are 4, 10, 5 and 20 respectively. What will be the output?**
   1. 76
   2. 119
   3. 123
   4. **238**
3. **A perceptron adds up all the weighted inputs it receives, and if it exceeds a certain value, it outputs a 1, otherwise it just outputs a 0.**
   1. True
   2. **False**
   3. Sometimes – it can also output intermediate values as well
   4. Can’t say



**CO2: Comprehend the Deep Network concepts.**



**Conclusion:** In this experiment we understood feed forward neural network and how it works. The same was implemented and displayed.