**Experiment No. 6 -To implement basic unsupervised neural network learning rules for a problem.**

**Learning Objective:** Student should be able to understand how to use learning rules for unsupervised neural network

**Tools:** Python 3 language and Jupyter Notebook

**Theory:**

Unsupervised neural networks learn patterns from data without relying on predefined labels or categories. They focus on discovering the underlying structure of the data. A common unsupervised learning rule is Hebbian Learning.

**Hebbian Learning Rule:**

Hebbian Learning is based on the principle that if two neurons are activated together frequently, the connection between them strengthens. In other words, connections between neurons that fire together become stronger. This helps the network learn associations and patterns in the data.

**How it works:**

1. **Initialization:** Start with random initial weights for connections between neurons.
2. **Input Presentation:** Feed an input pattern into the network.
3. **Weight Adjustment:** Adjust the weights based on the activity of the neurons. If two neurons are activated together, their connection strength increases.
4. **Repetition:** Repeat the process with various input patterns to refine the weights over time

**Application:**

Hebbian learning is often used in models like self-organizing maps (SOMs) or principal component analysis (PCA). For example, in a network where Hebbian learning is applied, frequently co-activated neurons will form stronger connections, allowing the network to better recognize and respond to similar input patterns in the future.

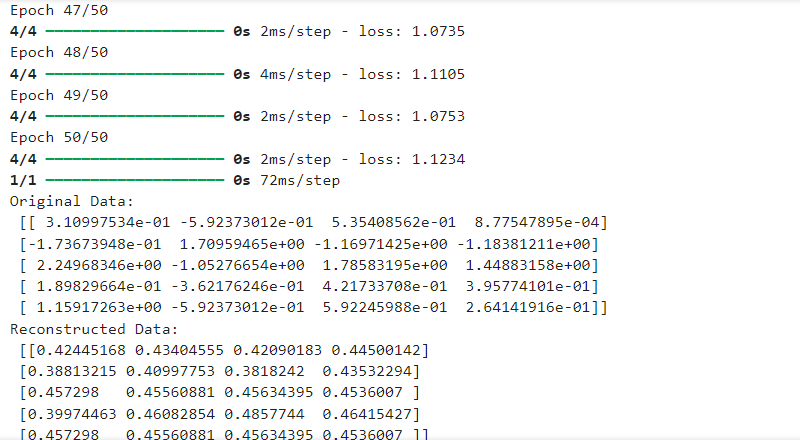
**Example:**

Suppose you have two neurons AAA and BBB with an initial weight www. If neuron AAA and neuron BBB are activated together, the weight between them will be increased according to the Hebbian rule. If this pattern is repeated, the connection between AAA and BBB becomes stronger, effectively learning the correlation between their activations.

**Code:**



**Output:**



**Learning Outcomes:** The student should have the ability to:

LO6.1 Explain what is unsupervised neural network & how to apply learning rules on unsupervised neural network.

LO6.2 Explain  what are the steps of learning rule using python.

LO6.3 Develop an understanding of different neural network architectures used in unsupervised learning

LO6.4 Learn to implement and experiment with basic unsupervised learning rules, such as Hebbian learning

**Course Outcomes:** Upon completion of the course students will be able to Learn to implement and experiment with basic unsupervised learning rules, such as Hebbian learning, k-means clustering. Understand how these rules help in adjusting the network's parameters based on the input data.

**Conclusion:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction** | **Formative** | **Timely** | **Attendance /** |  |
| **Parameters** | **Assessment** | **completion of** | **Learning** |
|  | **[40%]** | **Practical [ 40%]** | **Attitude** |
|  |  |  | **[20%]** |
| **Marks** |  |  |  |
| **Obtained** |

For Faculty Use: