Implementation of Clustering using Simple Competitive Learning

Aim: To implement and analyze unsupervised pattern clustering using the Simple Competitive Learning (SCL) algorithm.

Problem Definition: Design and implement a competitive neural network capable of grouping input data into clusters without prior class labels.

The system should:

1. Accept a dataset of where records are represented as unlabeled input vectors.
2. Use competitive learning to automatically group the data into distinct clusters.
3. Adjust the weight vectors of "winning neurons" to represent different clusters.

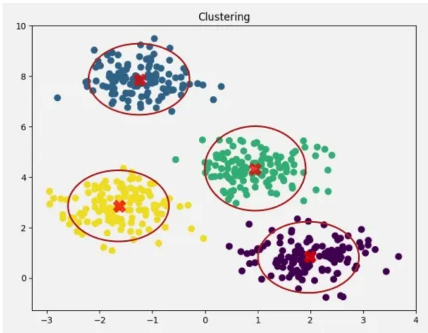
Theory: Clustering is an unsupervised machine learning technique that groups similar data points together into clusters based on their characteristics, without using any labeled data. The objective is to ensure that data points within the same cluster are more similar to each other than to those in different clusters, enabling the discovery of natural groupings and hidden patterns in complex datasets.

Goal: Discover the natural grouping or structure in unlabeled data without predefined categories.

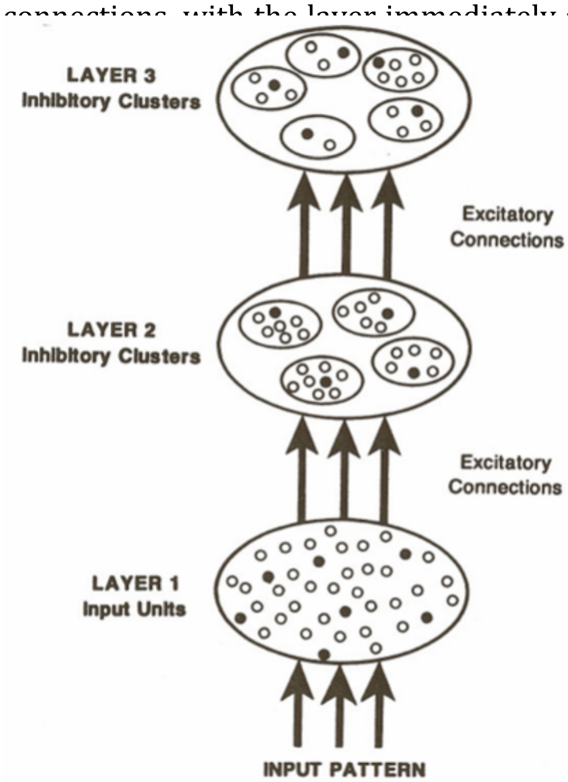
How: Data points are assigned to clusters based on similarity or distance measures.

Similarity Measures: Can include Euclidean distance, cosine similarity or other metrics depending on data type and clustering method.

Output: Each group is assigned a cluster ID, representing shared characteristics within the cluster.



The basic architecture of a competitive learning system is a common one. It consists of a set of hierarchically layered units in which each layer connects, via excitatory connections, with the layer immediately above it, and has inhibitory connections to



ase, each unit in a layer receives an input elow it and projects to each unit in the in a layer, the units are broken into a set within a cluster inhibit all other elements a cluster at one level compete with one ng on the layer below. The more strongly ng stimulus, the more it shuts down the

This version of competitive learning has the following properties:

The units in a given layer are broken into several sets of nonoverlapping clusters. Each unit within a cluster inhibits every other unit within a cluster. Within each cluster, the unit receiving the largest input achieves its maximum value while all other units in the cluster are pushed to their minimum value. We have arbitrarily set the maximum value to 1 and the minimum value to 0.

Every unit in every cluster receives inputs from all members of the same set of input units.

A unit learns if and only if it wins the competition with other units in its cluster.

A stimulus pattern S_j consists of a binary pattern in which each element of the pattern is either *active* or *inactive*. An active element is assigned the value 1 and an inactive element is assigned the value 0.

Each unit has a fixed amount of weight (all weights are positive) that is distributed among its input lines. The weight on the line connecting to unit i on the upper layer from unit j on the lower layer is designated w_{ij}. The fixed total amount of weight for unit j is designated $\sum_j w_{ij} = 1$. A unit learns by shifting weight from its inactive to its active input lines. If a unit does not respond to a particular pattern, no learning takes place in that unit. If a unit wins the competition, then each of its input lines gives up some portion ϵ of its weight and that weight is then distributed equally among the active input lines. Mathematically, this learning rule can be stated

$$\Delta w_{ij} = \begin{cases} 0 & \text{if unit } i \text{ loses on stimulus } k \\ \epsilon \frac{active_{jk}}{nactive_k} - \epsilon w_{ij} & \text{if unit } i \text{ wins on stimulus } k \end{cases}$$

where $active_{jk}$ is equal to 1 if in stimulus pattern S_k, unit j in the lower layer is active and is zero otherwise, and $nactive_k$ is the number of active units in pattern S_k (thus $nactive_k = \sum_j active_{jk}$).

There are several characteristics of a competitive learning mechanism that make it an interesting candidate for study, for example:

> Each cluster classifies the stimulus set into M groups, one for each unit in the cluster. Each of the units captures roughly an equal number of stimulus patterns. It is possible to consider a cluster as forming an M-valued feature in which every stimulus pattern is classified as having exactly one of the M possible values of this feature. Thus, a cluster containing two units acts as a binary feature detector. One element of the cluster responds when a particular feature is present in the stimulus pattern, otherwise the other element responds.
>
> If there is structure in the stimulus patterns, the units will break up the patterns along structurally relevant lines. Roughly speaking, this means that the system will find clusters if they are there.
>
> If the stimuli are highly structured, the classifications are highly stable. If the stimuli are less well structured, the classifications are more variable, and a given stimulus pattern will be responded to first by one and then by another member of the cluster. In our experiments, we started the weight vectors in random directions and presented the stimuli randomly. In this case, there is rapid movement as the system reaches a relatively stable configuration (such as one with a unit roughly in the center of each cluster of stimulus patterns). These configurations can be more or less stable. For example, if the stimulus points do not actually fall into nice clusters, then the configurations will be relatively unstable and the presentation of each stimulus will modify the pattern of responding so that the system will undergo continual evolution. On the other hand, if the stimulus patterns fall rather nicely into clusters, then the system will become very stable in the sense that the same units will always respond to the same stimuli.
>
> The particular grouping done by a particular cluster depends on the starting value of the weights and the sequence of stimulus patterns actually presented. A large number of clusters, each receiving inputs from the same

input lines can, in general, classify the inputs into a large number of different groupings or, alternatively, discover a variety of independent features present in the stimulus population. This can provide a kind of distributed representation of the stimulus patterns.

To a first approximation, the system develops clusters that minimize within-cluster distance, maximize between-cluster distance, and balance the number of patterns captured by each cluster. In general, tradeoffs must be made among these various forces and the system selects one of these tradeoffs.