Chapter

Fuzzy Perceptron Learning for Non-Linearly Separable Patterns

Raja Kishor Duggirala

Abstract

Perceptron learning has its wide applications in identifying interesting patterns in the large data repositories. While iterating through their learning process perceptrons update the weights, which are associated with the input data objects or data vectors. Though perceptrons exhibit their robustness in learning about interesting patterns, they perform well in identifying the linearly separable patterns only. In the real world, however, we can find overlapping patterns, where objects may associate with multiple patterns. In such situations, a clear-cut identification of patterns is not possible in a linearly separable manner. On the other hand, fuzzy-based learning has its wide applications in identifying non-linearly separable patterns. The present work attempts to experiment with the algorithms for fuzzy perceptron learning, where perceptron learning and fuzzy-based learning techniques are implemented in an interfusion manner.

Keywords: perceptron learning, fuzzy-based learning, fuzzy C-means, interfusion, weighted distances, pattern recognition, sum of squared errors, clustering fitness

1. Introduction

A learning system could be thought as a collection of methods that are brought together in order to create an environment to facilitate different learning processes. The learning systems will provide various types of learning resources and descriptions of procedures for obtaining quality results [1]. The learning systems find their applications in the areas like, image recognition, speech recognition, traffic prediction, e-mail spam and malware filtering, automatic language translation, medical diagnosis, etc. [2].

As the data increases in large volumes in the digital repositories, it has become essential to look for alternative approaches to yield better results in extracting interesting patterns from the repositories. Intelligent learning systems are gaining attention from a wide range of researchers in the recent years in extracting patterns from the data repositories. The learning systems have three kinds of approaches. They are supervised, unsupervised, and semi-supervised learning approaches [3].

The concept of perceptron learning plays a critical role in pattern recognition, which has become a challenging problem in the data science research. In the recent years, perceptron learning algorithms are exhibiting their robust performance in identifying interesting patterns from large data repositories when compared to the traditional supervised learning approaches [4]. A perceptron can be thought as a computational prototype of a neuron. As a supervised learning approach, perceptron learning is used for linear classification of patterns. This learning approach uses the already available labelled data to classify the future data by predicting the class labels.

In the literature, it is studied that many researchers experimented with perceptron learning for identifying interesting patterns from the data. A novel autonomous perceptron model (APM) was proposed to address the issues of complexity of traditional perceptron architectures [4]. APM is a nonlinear supervised learning model, which has the architecture using the computational power of the quantum bits (qubits). The researchers [5], using biophysical perceptron (BP), tried to simulate the pyramidal cells in the brain with a wide variety of active dendritic channels. The BP, here, explores the ability of real neurons with extended non-linear dendritic trees to effectively perform the classification task in identifying interesting patterns from the data. Many researchers have experimented with perceptron learning in a wide variety of ways. However, the perceptron learning suffers several limitations. It works well for linearly separable patterns. Though some researchers experimented for identifying non-linearly separable patterns, the perceptron learning produced best results for binary separation of patterns only [5]. Also that perceptron learning suffers poor performance in case of overlapping patterns, that is, when patterns are not having sharp boundaries.

Fuzzy-based learning, on the other hand, is found to show its ability in performing well for overlapping patterns [6]. As a fuzzy-based learning approach, fuzzy C-means (FCM) is widely used by researchers for pattern recognition. A weighted local fuzzy regression model showed a better efficiency than the least squares regression for non-linear and high-dimensional pattern recognition of transport system in China [7]. The new kernelized fuzzy C-means clustering algorithm [8] uses a kernel-induced distance function as a similarity measure showed improved performance in identifying the patterns when compared to the conventional fuzzy C-means technique. In many research findings, it is observed that the fuzzy-based learning approach was used in a wide variety of ways to achieve better results in extracting non-linear and overlapping patterns.

The present work attempts to experiment with fuzzy perceptron learning, which implements the perceptron learning and fuzzy-based learning techniques in an interfusion manner. In the research literature, we can find a good amount of work related to the combination of fuzzy logic with perceptron learning. The fuzzy neural network (FNN) was proposed for pattern classification, which uses supervised fuzzy clustering and pruning algorithm to determine the precise number of clusters with proper centroids representing the patterns to be recognised [9]. In the fuzzy neural integrated networks [10], the researchers attempted to integrate the concept of fuzzy sets and neural networks to deal with pattern recognition problems. In an enhanced algorithm for fuzzy lattice reasoning (FLR) classifier, a new nonlinear positive valuation function was defined to produce better results for pattern classification [11]. Along with these, however, many other research experiments of fuzzy perceptron learning are supervised learning approaches only. Therefore, the present work focuses on experimenting with effective implementation of some techniques involved in the perceptron and fuzzy-based learning systems for unsupervised learning to identify interesting patterns in large datasets. As part of the present work, five algorithms are developed, two of which are related to perceptron learning, one is the standard fuzzy C-means (FCM) algorithm. The remaining two algorithms are proposed by the present work, which implement the perceptron learning and fuzzy-based learning in an interfusion manner using weights and weighted distances respectively. All the algorithms are implemented using three benchmark datasets. The CPU time, clustering fitness (CF), and sum of squared errors (SSE) are taken into consideration for performance evaluation of the algorithms.

2. Perceptron learning

Nowadays, the perceptron learning model can be thought as a more general computational model in identifying interesting patterns in a dataset. It takes an input, aggregates it along with the weights and produces the result. A perceptron is used to learn patterns and relationships in data. Patterns help us knowing about the interesting features around which objects may be grouped in a given population of data.

A perceptron may be configured for a specific application, such as pattern recognition and data classification through some learning process [12]. Perceptrons are information processing devices, which are built from interconnected elementary processing units. These units are called neurons. The perceptrons are robust in exhibiting their ability in distributed representation and computation, learning, generalisation, adaptivity, inherent contextual information processing, and fault tolerance [13].

The perceptron learning uses an iterative weight adjustment for the enhanced retrieval of patterns from a dataset. The iterative process converges to the weights, which produce the patterns that represent the different groups of data objects in the dataset uniquely. While operating for learning on patterns, the perceptrons use weights in connection to every input vector. A weight represents the information used by the perceptron to solve a problem [14].

The perceptron with multiple neurons is shown in **Figure 1**.

In **Figure 1**, $X_1, X_2, \ldots, X_n$ are the n input vectors and $Y_1, Y_2, \ldots, Y_m$ are the m neurons. The input vector X_1 is connected to neurons $Y_1, Y_2, \ldots, Y_m$ with weights $W_{11}, W_{12}, \ldots, W_{1m}$, respectively, the input vector X_2 is connected to the neurons with weights $W_{21}, W_{22}, \ldots, W_{2m}$, respectively so on and the input vector X_n is connected to the neurons with the weights $W_{n1}, W_{n2}, \ldots, W_{nm}$, respectively. The weights of all input vectors for all neurons will be formulated as the weight matrix as shown below.

$$\begin{bmatrix} W_{11} & W_{12} & \ldots & \ldots & W_{1m} \\ W_{21} & W_{22} & \ldots & \ldots & W_{2m} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ W_{n1} & W_{n2} & \ldots & \ldots & W_{nm} \end{bmatrix}$$

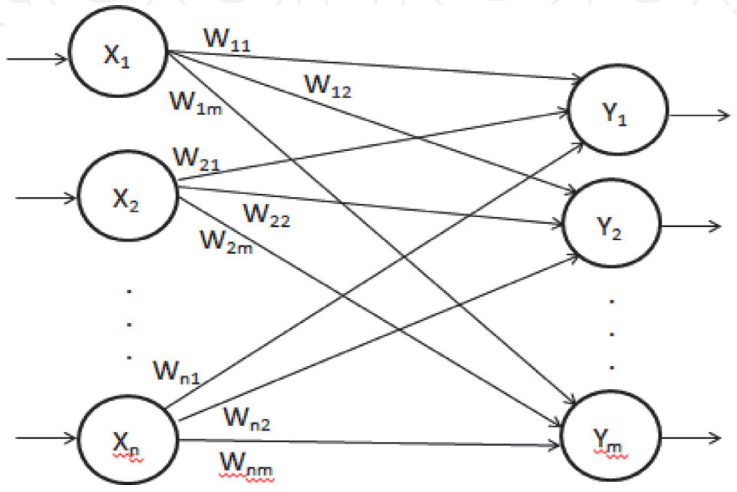

Figure 1.
A perceptron with a multiple neurons.

Though the perceptron learning exhibits its robustness in identifying the patterns in the data repositories, it works well for linearly separable patterns, that is, the patterns with sharp boundaries only. However, in the real time world, we may find overlapping patterns, that is, non-linearity in pattern associativity, where data objects may associate with multiple patterns. In such situations, the perceptron learning approach may suffer in identifying the patterns clearly. On the other hand, fuzzy-based learning has its wide applications in identifying patterns in the overlapping scenario. In the present work, two algorithms are implemented for perceptron learning. They are discussed in the following sub-sessions.

2.1 Perceptron learning using weights (PLW)

This algorithm implements the perceptron learning using weights [12]. With each input data vector, a weight is associated corresponding to each pattern. To generate the initial weights, one iteration of K-means algorithm is performed. The results of K-means iteration are used to compute the weight matrix. This weight matrix will be repeatedly updated in the subsequent iterations. For each input data vector weights are computed corresponding to every pattern. The input data vector is associated with the pattern corresponding to which the weight is maximum. This process is repeated for every iteration. The algorithm terminates when there is no change in the association of data vectors to the patterns. The algorithm for perceptron learning using weights is given below.

2.1.1 Algorithm PLW

Step 1: Determine the number of patterns, *k*, to be recognised from the dataset.
Step 2: Select *k* points randomly from the dataset and set them as cluster seeds to correspond the patterns to be recognised.
Step 3: Perform one iteration of K-means algorithm.
Step 4: Using the results of K-means iteration, compute cluster wise initial weights.
Step 5: Repeat steps 6–8 until the stopping condition.
Step 6: Generate weight matrix, where each element W_{ij} is computed as:

$$W_{ij}(i+1) = W_{ij}(i) + lr * (\|X_i\| - W_{ij}(i)) \quad (1)$$

Here, $W_{ij}(i+1)$ is the weight of ith data point X_i for jth cluster for the iteration $(i+1)$, $W_{ij}(i)$ is the weight of *i*th data point for *j*th cluster for the iteration *i*, $\|X_i\|$ is the norm of data point X_i, and *lr* is the learning rate. The *lr* may assume a value ranging between 0 and 1. To avoid possible biasedness in the computations, *lr* is assumed to be 0.5.
Step 7: Assign points to clusters using weights.
Step 8: Update cluster means, that is, refine patterns.
[End of step 5 loop]
Step 9: [End of algorithm]

2.2 Perceptron learning using weighted distances (PLWD)

This algorithm implements the perceptron learning using weighted distances [15]. With each input data vector, a weighted distance is associated corresponding to each pattern. To generate the initial weighted distances, one iteration of K-means algorithm is performed. Using the results of K-means the weight matrix is computed. This weight matrix is used to compute the weighted distances for each input data vector. The data vector is associated with the pattern corresponding to which the weighted distance is minimum. This weight matrix will be repeatedly updated in the subsequent iterations to compute the new weighted distances. This process repeats for every iteration. The algorithm terminates when there is no change in the association of data vectors to the patterns. The algorithm for perceptron learning using weighted distances is given below.

2.2.1 Algorithm PLWD

Step 1: Determine the number of patterns, k, to be recognised from the dataset.
Step 2: Select k points randomly from the dataset and set them as cluster seeds μ_j (j = 1, 2, …, m) to correspond the patterns to be recognised.
Step 3: Perform one iteration of K-means algorithm.
Step 4: Using the results of K-means iteration, compute cluster wise initial weights.
Step 5: Repeat steps 6–10 until the stopping condition.
Step 6: Generate weight matrix W using Eq. (1).
Step 7: For each data point X_i, compute the Euclidean distance $d(X_i, \mu_j)$ as follows:

$$d\left(X_i, \mu_j\right) = \sqrt{\sum_{l=1}^{d} \left(x_{il} - \mu_{jl}\right)^2} \tag{2}$$

Here, X_i is the ith data point, μ_j is the mean vector of the cluster j.
Step 8: For each data point compute the weighted distances as follows:

$$Wd_j = W_{ij}(i+1) \cdot d\left(X_i, \mu_j\right) \tag{3}$$

Step 9: Assign points to clusters using weights.
Step 10: Update cluster means, that is, refine patterns.
[End of step 5 loop]
Step 11: [End of algorithm]
Though the perceptron learning algorithms are experimented widely by many researchers, they exhibit their robustness in identifying linearly separable patterns only.

3. Fuzzy-based learning

Fuzzy-based learning is used to handle the concept of partial truth, where the truth value may range between completely true and completely false [16]. It is an approach that allows for multiple possible truth values to be processed through the same data object. In fuzzy-based learning, the data objects are assumed being associated with multiple patterns. For each data object, the degree of association is measured in membership. This membership value may range between 0 and 1 (1 being high similarity and 0 being no similarity with the pattern).

Fuzzy-based learning techniques focus on modelling uncertain and vague information that is found in the real world situations. These techniques deal with the patterns whose boundaries cannot be defined sharply [17, 18]. By fuzzy-based learning, one can know if data objects fully or partially associate with the patterns that are under consideration based on their memberships of association [19]. Among the techniques of fuzzy-based learning, fuzzy C-means (FCM) is the most well-known one as it has the advantage of robustness for obscure information about the patterns [20, 21]. FCM is widely studied and applied in geological shape analysis [22], medical diagnosis [23], automatic target recognition [24], meteorological data [20], pattern recognition, image analysis, image segmentation and image clustering [25–27], agricultural engineering, astronomy, chemistry [28], detection of polluted sites [29], etc. The following section presents a brief discussion of FCM algorithm.

3.1 Fuzzy C-means (FCM)

The fuzzy C-means (FCM) is a technique that uses degree of membership for natural interpretation of patterns recognised [30]. The FCM associates the data vectors among k patterns. Each data vector may associate with each pattern with a membership degree. The membership of a data vector towards a pattern can range between 0 and 1.

The FCM algorithm is given below [31]. Here, U is the $k \times N$ membership matrix. While computing the cluster means and updating the membership matrix at each iteration, the FCM uses the fuzzifier factor, m. For most cases, m ranging between 1.5 and 3.0 gives good results [32]. In the present work, in all the experiments, m is set to 1.5.

3.1.1 Algorithm FCM

Step 1: Determine the number of patterns, k, to be recognised from the dataset.

Step 2: Select k points randomly from the dataset and set them as cluster seeds μ_j ($j = 1, 2, \ldots, m$) to correspond the patterns to be recognised.

Step 3: Perform one iteration of K-means algorithm. Set $t = 0$.

Step 4: Using the results of K-means iteration, compute membership matrix $U^{(0)}_{k \text{ X } N}$.

Step 5: Repeat steps 6–9 until the stopping condition.

Step 6: [Refine patterns] Update the mean of jth cluster μ_j as follows:

$$\mu_j = \frac{\sum_{i=1}^{N} \left(u_{ij}\right)^m X_i}{\sum_{i=1}^{N} \left(u_{ij}\right)^m} \tag{4}$$

Here, u_{ij} is the membership degree of the data point X_i w.r.t. jth pattern and m is the fuzzifier factor.

Step 7: Compute the new membership matrix using:

$$u_{ij}^{t+1} = \left[\sum_{l=1}^{k} \left(\frac{\left\| X_i - \mu_j{}^t \right\|^2}{\left\| X_i - \mu_l{}^t \right\|^2} \right)^{1/m-1} \right]^{-1} \tag{5}$$

Step 9: Assign points to clusters using membership degrees. Set $t = t + 1$.

[End of step 5 loop]

Step 10: [End of algorithm]

4. Fuzzy perceptron learning

The fuzzy perceptron learning works in an interfusion manner, where the fuzzy logic is combined with perceptron learning for identifying non-linear and overlapping patterns. Much research work may be found in the literation where fuzzy perceptron learning is experimented in different applications [33, 34]. However, those experiments are confined to supervised learning only. The present work attempts to experiment with fuzzy perceptron learning for unsupervised cases. The present work proposes two algorithms, one is for fuzzy perceptron learning using weights and the other is for fuzzy perceptron learning using weighted distances.

4.1 Fuzzy perceptron learning using weights (FPLW)

This algorithm implements the perceptron learning using weights and FCM techniques in an interfusion manner. These techniques are performed in alternative iterations until the termination condition. Initially, one iteration of K-means algorithm is performed. Using the results of K-means, initial weights are computed as mentioned in the Section 2.1. Using these weights, weight matrix is generated to perform one iteration of perceptron learning algorithm to associate the input data vectors to the patterns. Using the results of perceptron learning step, membership matrix is computed to perform one iteration of FCM algorithm as mentioned in Section 3.1. The results of FCM step are used to update weight matrix to perform perceptron learning step. In this way the perceptron learning and FCM algorithms are repeated in alternative iterations until termination condition. The algorithm for fuzzy perceptron learning using weights (FPLW) is given below.

4.1.1 Algorithm FPLW

```
Step 1: Determine the number of patterns, $k$, to be recognised from the dataset.
Step 2: Select $k$ points randomly from the dataset and set them as cluster seeds $\mu_j$ ($j = 1, 2, \dots, m$) to correspond the patterns to be recognised.
Step 3: Perform one iteration of K-means algorithm.
Step 4: Using the results of K-means iteration, compute cluster wise initial weights.
Step-5: Update cluster means $\mu_j$ ($j = 1, 2, \dots, m$).
Step 6: Repeat steps 7–13 until the stopping condition.
Step 7: Compute the weight matrix using Eq. (1).
Step 8: Assign points to clusters using weights.
Step 9: If there is no change in cluster assignment then go to step 14.
Step 10: Update cluster means using Eq. (4).
Step 11: Generate membership matrix $U^{(0)}_{k\ \mathrm{X}\ N}$ using Eq. (5).
Step 12: Assign points to clusters using membership matrix.
Step 13: If there is no change in cluster assignment then go to step 14.
[End of Step 6 loop]
Step 14: [End of Algorithm]
```

4.2 Fuzzy perceptron learning using weighted distances (FPLWD)

This algorithm implements the perceptron learning using weighted distances and FCM techniques in an interfusion manner. These techniques are performed in alternative iterations until the termination condition. Initially, one iteration of K-means technique is performed. Using the results of K-means, initial weights are computed as mentioned in the Section 2.2. Now, one iteration of perceptron learning algorithm is performed where the weight matrix is generated using the initial weights. Using this weight matrix, weighted distances are computed for every input vector X_i with respect to each pattern using the Eq. (3). The weighted distances are used to associate the input vectors to the patterns. Using the results of perceptron learning step, membership matrix is computed to perform one iteration of FCM algorithm as mentioned in Section 3.1. The results of FCM step are used to compute weight matrix for perceptron learning step. In this way the perceptron learning and FCM steps are repeated in alternative iterations until termination condition. The algorithm for fuzzy perceptron learning using weighted distances (FPLWD) is given below.

4.2.1 Algorithm FPLWD

```
Step 1: Determine the number of patterns, $k$, to be recognised from the dataset.
Step 2: Select $k$ points randomly from the dataset and set them as cluster seeds $\mu_j$ ($j = 1, 2, \dots, m$) to correspond the patterns to be recognised.
Step 3: Perform one iteration of K-means algorithm.
Step 4: Using the results of K-means iteration, compute cluster wise initial weights.
Step-5: Update cluster means $\mu_j$ ($j = 1, 2, \dots, m$).
Step 6: Repeat steps 7–15 until the stopping condition.
Step 7: Compute the weight matrix using Eq. (1).
Step 8: For each data point compute the Euclidean distance using Eq. (2).
Step 9: For each data point compute weighted distances using Eq. (3).
Step 10: Assign points to clusters using weighted distances.
Step 11: If there is no change in cluster assignment then go to step 16.
Step 12: Update cluster means using Eq. (4).
Step 13: Generate membership matrix $U^{(0)}_{k \times N}$ using Eq. (5).
Step 14: Assign points to clusters using membership matrix.
Step 15: If there is no change in cluster assignment then go to step 16.
[End of Step 6 loop]
Step 16: [End of Algorithm]
```

5. Performance evaluation

For performance evaluation of algorithms, CPU time in seconds, sum of squared errors [35] and clustering fitness (CF) [36] are taken into consideration and are calculated for all the algorithms.

5.1 Sum of squared errors

The objective of pattern learning is to minimise the intra-cluster sum of squared errors (SSE). The lesser the SSE, the better the goodness of fit is. The SSE for the results of each algorithm is computed using Eq. (6).

$$SSE = \sum_{j=1}^{k} \sum_{X_i \in C_j} \left(X_i - \mu_j\right)^2 \tag{6}$$

Here, X_i is the ith data point in the dataset, μ_j ($j = 1, \dots, k$) is the mean of the cluster C_j, and k is the number of patterns to be recognised.

5.2 Cluster fitness

While achieving high intra-cluster similarity, it is also important to achieve well separation of patterns.

So, it is also important to consider inter-cluster similarity while evaluating the performance of the algorithms. For this, the present work, computes the clustering fitness (CF) as a performance criterion, which requires the calculation of both intra-cluster similarity and inter-cluster similarity. The computation of CF also requires the experiential knowledge, λ. The computation of CF results in higher value when the inter-cluster similarity is low and results in lower value for when the inter-cluster similarity is high. Also that to make the computation of CF unbiased, the value of λ is taken as 0.5 [36].

5.2.1 Intra-cluster similarity for the cluster C_j

It can be quantified via a function of the reciprocals of intra-cluster radii within each of the resulting clusters. The intra-cluster similarity of a cluster C_j $(1 = j = k)$, denoted as $S_{tra}(C_j)$ [36], is defined by:

$$S_{tra}(C_j) = \frac{1+n}{1+\sum_1^n dist(I_l, Centroid)} \tag{7}$$

Here, n is the number of items in cluster C_j, I_j $(1 = j = n)$ is the jth item in cluster C_j, and dist(I_j, *Centroid*) calculates the distance between I_j and the centroid of C_j, which is the intra-cluster radius of C_j. To smooth the value of $S_{tra}(C_j)$ and allow for possible singleton clusters, 1 is added to the denominator and numerator.

5.2.2 Intra-cluster similarity for one clustering result C

It is denoted as $S_{tra}(C)$ [36]. It is defined by:

$$S_{tra}(C) = \frac{\sum_1^k S_{tra}(C_j)}{k} \tag{8}$$

Here, k is the number of resulting clusters in C and $S_{tra}(C_j)$ is the intra-cluster similarity for the cluster C_j.

5.2.3 Inter-cluster similarity

It can be quantified via a function of the reciprocals of inter-cluster radii of the clustering centroids. The inter-cluster similarity for one of the possible clustering results C, denoted as $S_{ter}(C)$ [36] is defined by:

$$S_{ter}(C) = \frac{1+k}{1+\sum_1^k dist\left(Centroid_j, Centroid^2\right)} \tag{9}$$

Here, k is the number of resulting clusters in C, $1 = j = k$, $Centroid_j$ is the centroid of the jth cluster in C, $Centroid^2$ is the centroid of all centroids of clusters in C. We compute inter-cluster radius of $Centroid_j$ by calculating dist($Centroid_j$, $Centroid^2$), which is distance between $Centroid_j$, and $Centroid^2$. To smooth the value of $S_{ter}(C)$

and allow for possible all-inclusive clustering result, 1 is added to the denominator and the numerator.

5.2.4 Clustering fitness

The clustering fitness for one of the possible clustering results C, denoted as CF [36], is defined by:

$$CF = \lambda \text{ x } S_{tra}(C) + \frac{1-\lambda}{S_{ter}(C)} \tag{10}$$

Here, λ $(0 < \lambda < 1)$ is an experiential weight, $S_{tra}(C)$ is the intra-cluster similarity for the clustering result C and $S_{ter}(C)$ is the inter-cluster similarity for the clustering result C.

6. Experiments and results

Experimental work has been carried out on the system with Intel(R) Core(TM) i3-5005 U CPU@2.00GHz processor speed, 4GB RAM, Windows 7 OS (64-bit) and using JDK1.7.0_45. Separate modules are written for each of the above discussed methods to observe the CPU time for clustering any dataset by keeping the cluster seeds same for all methods. I/O operations are eliminated and the CPU time observed is strictly for clustering of the data.

Along with the proposed algorithms FPLW and FPLWD for fuzzy perceptron learning, experiments are also conducted with the algorithms PLW, PLWD and FCM for performance comparison. All the algorithms are executed using the benchmark datasets with varying number of patterns to be recognised. In the present work, Magic Gamma, Letter Recognition and Intrusion datasets are used from UCI ML data repository [37]. All the developed algorithms, PLW, PLWD, FCM, FPLW and FPLWD, are executed using these datasets for varying number of patterns to be recognised (k = 10, 11, 12, 13, 14, 15).

All the algorithms operate in an iterative manner and terminate when a stopping condition is met. The stopping condition is when there is no change in the pattern associativity of the data vectors. The termination condition is the same for all the algorithms.

Details of the datasets are available in **Table 1**.

6.1 Observations with Magic Gamma dataset

The results of all algorithms, using Magic Gamma dataset, with respect to CPU time in seconds, clustering fitness and sum of squared errors are shown in **Figures 2–4**, respectively.

S. No.	Dataset	No. of points	No. of dimensions
1	Magic Gamma data	19,020	10
2	Letter Recognition data	20,000	16
3	Intrusion data	4,94,019	35

Table 1.
Details of datasets.

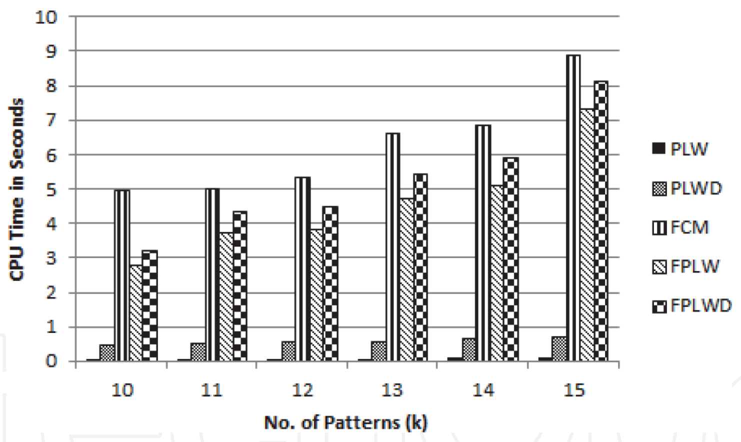

Figure 2.
CPU time of each clustering method (Magic Gamma dataset).

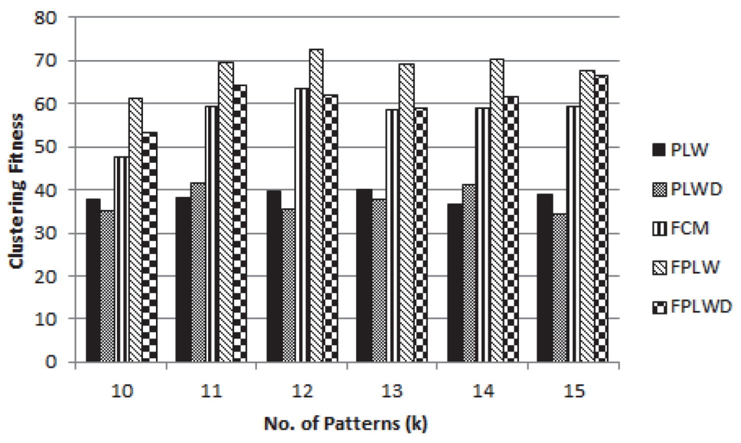

Figure 3.
Clustering fitness of each clustering method (Magic Gamma dataset).

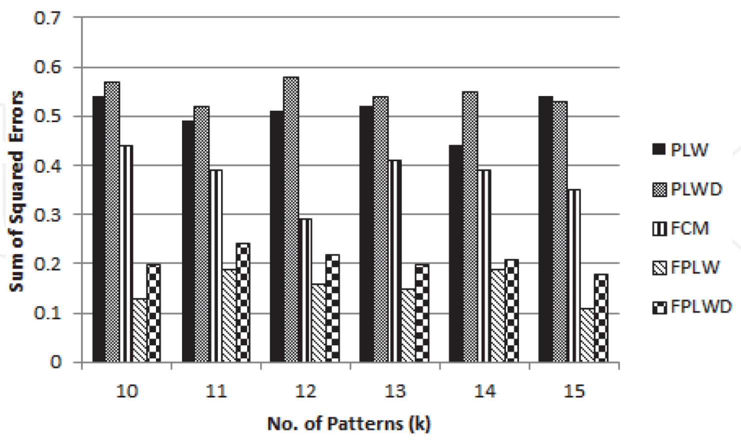

Figure 4.
SSE of each clustering method (Magic Gamma dataset).

6.2 Observations with Letter Recognition dataset

The results of all algorithms, using Letter Recognition dataset, with respect to CPU time in seconds, clustering fitness and sum of squared errors are shown in **Figures 5–7**, respectively.

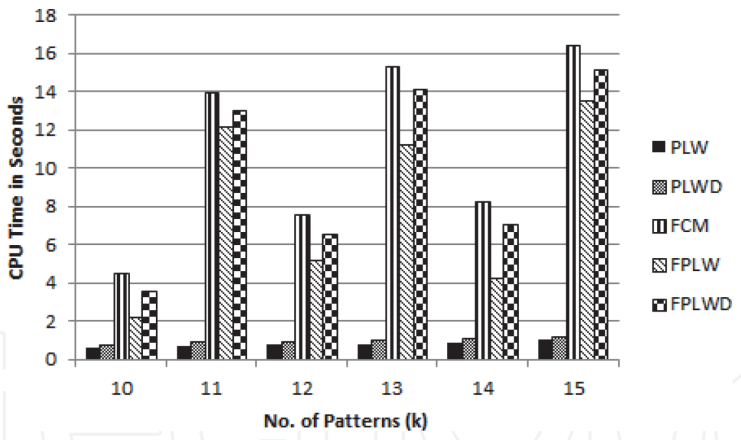

Figure 5.
CPU time of each clustering method (Letter Recognition dataset).

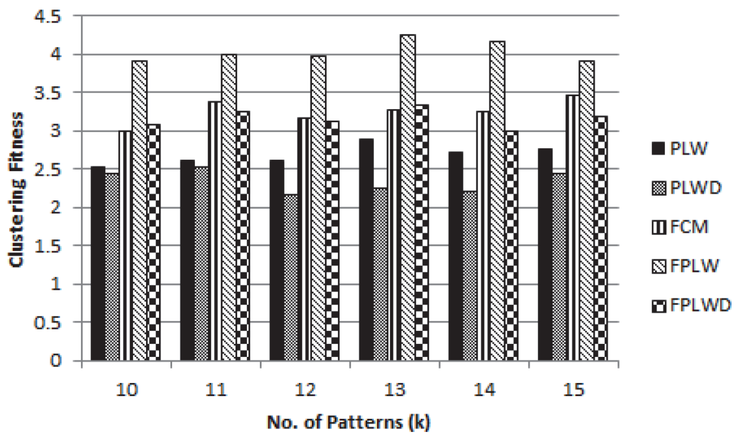

Figure 6.
Clustering fitness of each clustering method (Letter Recognition dataset).

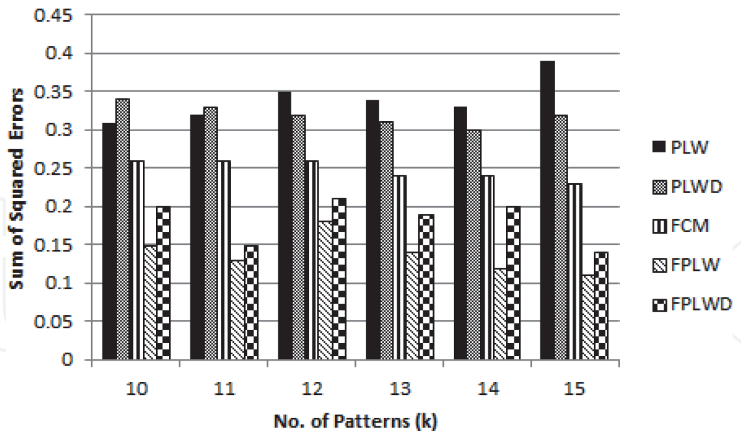

Figure 7.
SSE of each clustering method (Letter Recognition dataset).

6.3 Observations with Intrusion dataset

The results of all algorithms, using Intrusion dataset, with respect to CPU time in seconds, clustering fitness and sum of squared errors are shown in **Figures 8–10**, respectively.

In all the experiments, it is observed that the algorithm FPLW, which implements the perceptron learning using weights and the FCM techniques in an inter-fusion manner, is showing consistently better performance in terms of clustering fitness (CF) and SSE than the other algorithms.

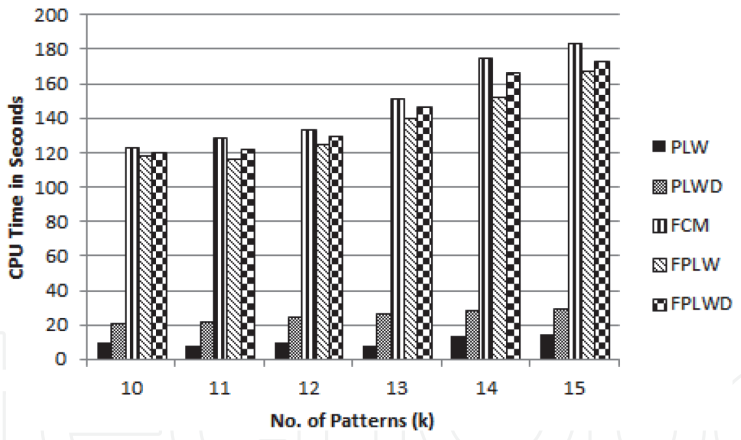

Figure 8.
CPU time of each clustering method (Intrusion dataset).

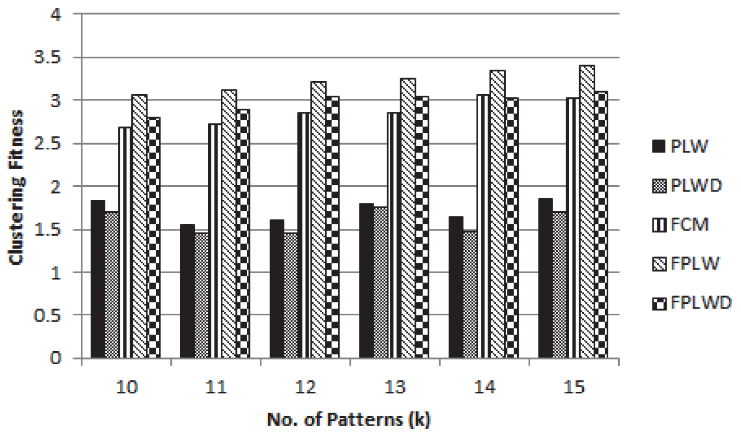

Figure 9.
Clustering fitness of each clustering method (Intrusion dataset).

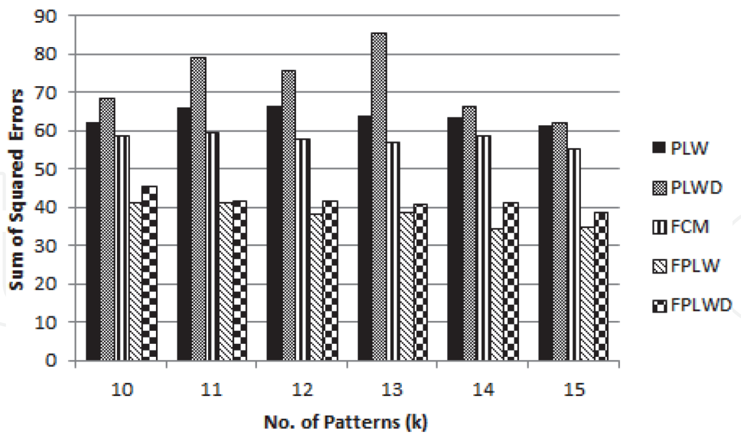

Figure 10.
SSE of each clustering method (Intrusion dataset).

7. Conclusion

The present experiment mainly focuses on the study fuzzy perceptron learning for recognising non-linear patterns in the datasets. Many researchers contributed greatly towards fuzzy perceptron learning. However, their experiments are confined to supervised learning only. So, the present work experimented with the fuzzy perceptron learning approaches for unsupervised learning. The work proposes two new algorithms, that is, FPLW and FPLWD. These algorithms are implemented using three benchmark datasets. Along with these algorithms, the algorithms for standard FCM and perceptron learning using weights and weighted distances are also implemented for performance comparison. For all the algorithms the CPU time in seconds, clustering fitness (CF) and sum of squared errors (SSE) are taken into consideration for performance evaluation. All the developed algorithms are experimented with varying number of patterns (k) to be recognised.

In all the experiments, it is observed that the proposed algorithm for fuzzy perceptron learning using weights (FPLW) is consistently showing better performance with respect to clustering fitness and SSE. Of course, the algorithm FPLW is taking a little more time for its execution than the other algorithms. However, it could be negligible, as the main concern is for clearly recognising the non-linear patterns in the datasets.

Author details

Raja Kishor Duggirala
Department of Computer Science Engineering, Dr. Lankapalli Bullayya College of Engineering, Visakhapatnam, Andhra Pradesh, India

*Address all correspondence to: rajakishor@gmail.com

References

[1] Satapathy SK et al. EEG Brain Signal Classification for Epileptic Seizure Disorder Detection, ScienceDirect, 2019, ISBN 978-0-12-817426-5, DOI: https://doi.org/10.1016/C2018-0-01888-5

[2] Benton WC, Hat R, Raleigh. Machine learning systems and intelligent applications. IEEE Software. 2020;**37**: 43-49. DOI: 10.1109/MS.2020.2985224

[3] Krendzelak M. Machine Learning and Its Applications in e-Learning Systems. Stary Smokovec, Slovakia: IEEE, 2014 IEEE 12th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA); 2014. pp. 267-269. DOI:10.1109/ICETA.2014.7107596

[4] Sagheer A, Zidan M, Abdelsamea MM. A novel autonomous perceptron model for pattern classification applications. Entropy. 2019;**21**(8):763. DOI: 10.3390/e21080763

[5] Moldwin T, Segev I. Perceptron learning and classification in a modeled cortical pyramidal cell. Frontiers in Computational Neuroscience. 2020. DOI: 10.3389/fncom.2020.00033. https://www.frontiersin.org/articles/10.3389/fncom.2020.00033/full

[6] Leoni Sharmila S, Dharuman C, Venkatesan P. A fuzzy based classification—An experimental analysis, International Journal of Innovative Technology and Exploring Engineering. 2019;**8**(10):4634-4638

[7] Tan Y, Chen S. Pattern recognition based on weighted fuzzy C-means clustering. In: 6th International Congress on Image and Signal Processing (CISP). 2013. pp. 1061-1065. DOI: 10.1109/*CISP*.2013.6745213

[8] Das S, Baruah H. A new kernelized fuzzy C-means clustering algorithm with enhanced performance. Semantic Scholar; 2014. Corpus ID: 212563388

[9] Kulkarni A, Kulkarni N. Fuzzy neural networks for pattern recognition. Procedia Computer Science. 2020;**167**: 2606-2616. DOI: 10.1016/j.procs.2020.03.321

[10] Baraldi A, Blonda P, Petrosino A. Fuzzy neural networks for pattern recognition. In: Marinaro M, Tagliaferri R, editors. Neural Nets WIRN VIETRI-97. Perspectives in Neural Computing. London: Springer; 1998. DOI: 10.1007/978-1-4471-1520-5_2

[11] Jamshidi Khezeli YJ, Nezamabadi-pour H. Fuzzy lattice reasoning for pattern classification using a new positive valuation function. Advances in Fuzzy Systems. 2012;**2012**:206121. DOI: 10.1155/2012/206121

[12] Sivanandam SN, Sumathi S, Deepa SN. Introduction to Neural Networks Using Matlab 6.0. India: Tata McGraw Hill; 2008

[13] Nadal J-P, Parga N. Information processing by a perceptron in an unsupervised learning task. Network: Computation in Neural Systems. 1993; **4**(3):295-312. DOI: 10.1088/0954-898X_4_3_004

[14] Haykin S. Neural Networks: A Comprehensive Foundation. 2nd ed. New Delhi, India: Pearson Education; 2007

[15] Nalaie K, Ghiasi-Shirazi K, Akbarzadeh-T M. Efficient implementation of a generalized convolutional neural networks based on weighted Euclidean distance. In: 2017 7th International Conference on Computer and Knowledge Engineering (ICCKE). 2017. pp. 211-216. DOI: 10.1109/*ICCKE*.2017.8167877

[16] Novák V, Perfilieva I, Močkoř J. Mathematical Principles of Fuzzy Logic. Dordrecht: Kluwer Academic; 1999

[17] Zadeh LA. Fuzzy sets. Information and Control. 1965;**8**(3):338-353

[18] Lemiare J. Fuzzy insurance. ASTIN Bulletin. 1990;**20**(1):33-55

[19] Das S. Pattern recognition using fuzzy C-means technique. International Journal of Energy Information and Communications. 2013;**4**(1):1-14

[20] Lu Y, Ma T, Yin C, Xie X, Tian W, Zhong SM. Implementation of the fuzzy C-means clustering algorithm in meteorological data. International Journal of Database Theory and Application. 2013;**6**(6):1-18

[21] Kaltri K, Mahjoub M. Image segmentation by Gaussian mixture models and modified FCM algorithm. The International Arab Journal of Information Technology. 2014;**11**(1):11-18

[22] Bezdek JC, Trivedi M, Ehrlich R, Full WE. Fuzzy clustering: A new approach for geostatistical analysis. International Journal of System Measurement and Decision. 1981;**1**:13-23

[23] Bezdek JC. Feature selection for binary data-medical diagnosis with fuzzy sets. In: Proc. Nat. Comput. Conf. AFIPS Press; 1972. pp. 1057-1068

[24] Cannon RL, Dave JV, Bezdek JC. Efficient implementation of fuzzy C-means clustering algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986;**8**(2):248-255

[25] Cannon RL, Jacobs C. Multispectral pixel classification with fuzzy objective functions. Technical Report CAR-TR-51. College Park: Center for Automation Research, University of Maryland; 1984

[26] Gong M, Liang Y, Ma W, Ma J. Fuzzy C-means clustering with local information and kernel metric for image segmentation. IEEE Transactions on Image Processing. 2013;**22**(2):573-584

[27] Krinidis S, Krinidis M, Chatzis V. Fast and robust fuzzy active contours. IEEE Transactions on Image Processing. 2010;**19**(5):1328-1337

[28] Yong Y, Chongxun Z, Pan L. A novel fuzzy C-means clustering algorithm for image thresholding. Measurement Science Review. 2004;**4**(1):11-19

[29] Hanesch M, Scholger R, Dekkers MJ. The application of fuzzy C-means cluster analysis and non-linear mapping to a soil data set for the detection of polluted sites. Physics and Chemistry of the Earth, Part A. 2001;**26** (11–12):885-891

[30] Ghosh S, Dubey SK. Comparative analysis of K-means and fuzzy C-means algorithms. International Journal of Advanced Computer Science and Applications. 2013;**4**(4):35-39

[31] Klir GJ, Yuan B. Fuzzy Sets and Fuzzy Logic: Theory and Applications. India: Prentice Hall of India Private Limited; 2005

[32] Bezdek JC, Ehrlich R, Full W. FCM: The fuzzy C-means clustering algorithm. Computers and Geosciences. 1984;**10**(2–3):191-203

[33] Auephanwiriyakul S, Dhompongsa S. An investigation of a linguistic perceptron in a nonlinear decision boundary problem. In: 2006 IEEE International Conference on Fuzzy Systems. 2006. pp. 1240-1246

[34] Yang J, Wu W, Shao Z. A new training algorithm for a fuzzy perceptron and its convergence. In: Advances in Neural Networks—ISNN 2005, Second International Symposium on Neural Networks, Chongqing, China; May 30–June 1 2005

[35] Han J, Kamber M. Data Mining Concepts and Techniques. 2nd ed. San Francisco, CA: Morgan Kaufmann Publishers, An Imprint of Elsevier; 2007

[36] Han X, Zhao T. Auto-K dynamic clustering algorithm. Journal of Animal and Veterinary Advances. 2005;**4**(5): 535-539

[37] Lichman M. UCI Machine Learning Repository. 2013. Available from: http://archive.ics.uci.edu/ml