Article

Improving Classification Performance of Fully Connected Layers by Fuzzy Clustering in Transformed Feature Space

Tolga Ahmet Kalaycı * and Umut Asan

Department of Industrial Engineering, Istanbul Technical University, Maçka, Istanbul 34367, Turkey; asanu@itu.edu.tr
* Correspondence: kalaycit@itu.edu.tr

Abstract: Fully connected (FC) layers are used in almost all neural network architectures ranging from multilayer perceptrons to deep neural networks. FC layers allow any kind of symmetric/asymmetric interaction between features without making any assumption about the structure of the data. However, success of convolutional and recursive layers and findings of many studies have proven that the intrinsic structure of a dataset holds a great potential to improve the success of a classification problem. Leveraging clustering to explore and exploit this intrinsic structure in classification problems has been the subject of various studies. In this paper, we propose a new training pipeline for fully connected layers which enables them to make more accurate classification predictions. The proposed method aims to reflect the clustering patterns in the original feature space of the training dataset to the transformed feature space created by the FC layer. In this way, we intend to enhance the representation ability of the extracted features and accordingly increase the classification accuracy. The Fuzzy C-Means algorithm is employed in this study as the clustering tool. To evaluate the performance of the proposed method, 11 experiments were conducted on 9 benchmark UCI datasets. Empirical results show that the proposed method works well in practice and gives higher classification accuracies compared to a regular FC layer in most datasets.

Keywords: fully connected layer; classification; fuzzy clustering; neural networks; machine learning; deep learning

1. Introduction

Classification and clustering are two widely used methods in machine learning. While classification relies on supervised learning, clustering follows an unsupervised approach. Although these techniques are usually used in fundamentally different problem types, they both follow similar principles in pattern identification. In classification, the separation of classes is expected to follow the patterns in the data. Clustering, on the other hand, aims to discover implicit patterns in a dataset. At this point, it is reasonable to assert that these two methods are expected to share the same goals in a symmetric way [1].

Various studies in the literature have addressed the issue of using clustering to increase the success of classification predictions. Making use of clustering to improve the representation ability of extracted features is a foremost topic in this field. Gupta and Kumar [2] proposed using the Fuzzy C-Means algorithm together with an empirical wavelet transform (EWT) to extract stronger features in the mental task classification problem. They employed the Fuzzy C-Means algorithm to avoid overlapping segments that are created by the EWT. Experiments showed that when combined with the Fuzzy C-Means algorithm, the EWT resulted in better classification performances. Li et al. [3] used the Fuzzy C-Means clustering and principal component analysis in combination with the support vector machine method for multi-label classification in spacecraft electrical fault detection problems. Increased fault diagnosis accuracy and shortened computing time have been reported as the main contributions of their study. Srivastava et al. [4] came up with ExpertNet architecture which
combines an autoencoder with cluster-specific classifiers. In their method, the autoencoder is responsible for creating lower dimensional representations of observations in a way that preserves their clustering structures. On the other hand, observations are clustered based on these representations and forwarded to related classifiers. Experiments showed that their method improves both the classification performance and generalizability. Kalaycı and Asan [5] proposed a new regularization method that benefits from inherent clusters in training datasets to increase classification performance of neural networks. Their work depends on assigning hidden nodes of a neural network to specific clusters via a matrix which holds membership degrees of each observation to each cluster. Alternatingly using this matrix with a random binary matrix, they created a regularization method which performs better than the dropout in their experiments. Simultaneous learning frameworks which basically aim to enhance the objective function of clustering methods by embedding classification error in it, is another area where studies on combining clustering and classification are produced. Cai et al. [6] suggested a framework which uses the Bayesian theory to model cluster posterior probabilities of classes that represent the relations between clusters and classes. With the help of these posterior probabilities, they came up with a single objective function which includes both classification and clustering quality. To optimize the objective function, a particle swarm optimizer is employed in their method. The result of the experiments, a robust framework which is capable of clustering and classifying a dataset simultaneously has been asserted as the output of their study. Additionally, in Cai et al. [7], the authors advanced their framework to employ multiple objective functions for clustering and classification rather than a single one. Qian et al. [8] addressed the problem of high complexity in the objective function that is suggested by [6,7] and proposed a new framework that exploits cluster structure representations instead of cluster posterior probabilities of classes to relate clusters and classes. Thanks to decreased complexity and continuously differentiable forms of the objective function, a block coordinate descent algorithm was employed as the optimizer in their method, resulting in a more efficient framework. In another study by Hebboul et al. [9] an incremental self-organizing map model that was designed for simultaneous clustering and classification was proposed. Their method improves and accelerates the self-organizing map structure by combining it with SVM. Semi-supervised classification is another major field in which clustering has been proven to make significant contributions to classification success. Fang et al. [10] proposed a semi-supervised learning framework that incorporates a convolutional neural network with an approximate rank-order clustering algorithm for hyperspectral image classification (HSI). In their study, pseudo labels are created for the unlabeled data by clustering the features extracted by a lightweight CNN architecture and then the CNN is fine-tuned using both types of labels on a dual-loss cost function. Compared to state-of-the-art deep learning-based and traditional HSI classification methods, higher accuracy results have been stated as the result of these experiments. In another study, Sellars et al. [11] benefited from clustering to improve decision boundaries and construct more generic features in the semi-supervised classification task. In their study, with the help of clustering and graph construction methods, they iteratively use the extracted features using a base CNN model to generate supervised and unsupervised pseudo labels. By feeding these generated pseudo labels back to the training process they improve the accuracy of pseudo labels and subsequently classification accuracy of the base CNN model on several benchmark datasets. Making use of clustering in imbalanced datasets is also one of the areas to embed clustering into classification processes. Huang et al. [12] addressed low classification accuracy and high complexity problems in imbalanced binary classification and came up with an algorithm that combines clustering with SVM. Their method was mainly based on under-sampling from the majority class and an effective outlier elimination by taking into account the characteristics of the clusters formed by the minority class. Reduced feature dimensions and improved precision on the minor class decisions were reported as the main contributions of the proposed method. Prototype-based learning is another concept that makes clustering work in harmony with classification to yield better
accuracy. Chaudhuri et al. [13] proposed a guided clustering-based network to improve classification performance. They remark that the success of classification mainly depends on separability of features. They also assert that for a challenging classification task, distribution of training data in the feature space may still remain inseparable. At this point, the intuition behind their work is that rather than constructing the classification features from scratch and unaided, a simultaneous training process with another well separable but not necessarily labeled guide dataset may increase the classification accuracy by leveraging the cluster-wise dissociating capability of the guide set. Within this process, each class is arbitrarily assigned to a cluster in the guide set. In case there is not such a guide set, the authors proposed to use manually created guide vectors with the same number as the classes in the classification task. Results show that their proposed pipeline outperforms most state-of-the-art methods in classification accuracy. Ma et al. [14] suggested to provide classification models with high-level cluster information which they call semantic priors so that the models have the ability to deduce high-level semantic expressions. In their work, it is proposed to feed the models with both positive and negative semantic priors to assure the smoothness of semantic clustering and the robustness of classification. They also added that their proposed model can be used as a plug-in module for various deep learning applications.

Table 1 summarizes all studies mentioned so far, together with our proposed method in terms of their main contribution areas. It seems reasonable to collect main contributions of these studies under eight items, namely enhancing feature extraction, enhancing pseudo labels in semi-supervised classification, handling data imbalance problems, proposing a new regularization approach, exploiting a combined cost function for classification and clustering, centroid learning through backpropagation, applicability to different NN architectures and classification problems and enhancing clustering algorithms to serve as a classifier. In this paper, we propose a new training method for fully connected layers which enables them to make more accurate classification predictions. The proposed method basically relies on enhancing feature extraction capabilities of fully connected layers by equipping them with clustering abilities along with their classification abilities. To fulfill this purpose, a training process, which incorporates a combined cost function of classification and clustering costs, is suggested. As seen in Table 1, the proposed method differs from other studies in terms of the four different contribution areas it combines.

Table 1. Studies in the literature and the proposed method.

| Enhancing Feature Extraction | Enhancing Pseudo Labels in Semi-Supervised Classification | Handling Data Imbalance Problem | Proposing a New Regularization Approach | Exploiting a Combined Cost Function for Classification and Clustering | Centroid Learning through Backpropagation | Applicability to Different NN Architectures and Classification Problems | Enhancing Clustering Algorithms to Serve as a Classifier |
|-----------------------------|-----------------------------------------------------------|---------------------------------|---------------------------------------|---------------------------------------------------------------|------------------------------------------|----------------------------------------------------------|----------------------------------------------------------|
| Gupta & Kumar (2017)         | ✓                                                         |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Li et al., (2017)            | ✓                                                         |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Cai et al., (2009 & 2010)    |                                                           |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Qian et al., (2012)          |                                                           |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Habibud et al., (2015)       |                                                           |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Fang et al., (2020)          |                                                           |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Sellas et al., (2020)        |                                                           |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Huang et al., (2020)         |                                                           |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Chaudhuri et al., (2020)     |                                                           |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Srivastava et al., (2022)    | ✓                                                         |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Ma et al., (2022)            |                                                           |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Kalayci & Almaz (2020)       |                                                           |                                 |                                       |                                                                |                                          |                                                          |                                                          |
| Proposed Method              |                                                           |                                 |                                       |                                                                |                                          |                                                          |                                                          |

The main contributions of this paper can be summarized as follows: (i) it introduces a new training process which makes fully connected layers take advantage of the clustering information generated by the training dataset; (ii) it proposes an enhanced fully connected layer which is capable of classifying and clustering a dataset simultaneously; (iii) it learns cluster centroids through backpropagation; (iv) it provides experimental results indicating better prediction performances of the proposed method in most benchmark datasets compared to regular fully connected layers; and (v) it is applicable in any type of neural network architectures and classification problems in which fully connected layers are employed.
The rest of this paper is structured as follows. In the following section a brief overview of fully connected layers and the Fuzzy C-Means algorithm is given. In Section 3, the intuition of the proposed method, algebraic details of forward propagation and cost function calculations, and guidelines for the extension of the proposed method to the case of multiple fully connected layers are provided. Section 4 presents the results of 11 different experiments conducted on nine benchmark datasets. Finally, conclusions and future directions are given in the last section.

2. Background
2.1. Fully Connected Layers

Fully connected layers are the most general purpose layers in neural networks and they are used in almost all types of architectures. In a fully connected layer, each node is connected to each node in the previous and next layer [15]. The main task of a fully connected layer is to transform the feature space in order to make the problem more malleable [16]. During this transformation process, the number of dimensions may increase, decrease or stay fixed. In each case, the new dimensions are linear combinations of dimensions in the previous layer. Then, with the help of an activation function, the new dimensions are given non-linearity. In Figure 1, a fully connected layer, which transforms a five-dimensional feature space into a three-dimensional space, is seen.

![Figure 1. A regular fully connected layer.](image)

FC layers make any kind of interactions between the input variables possible. Thanks to this structure agnostic approach, with sufficient depth and width, fully connected layers have the theoretical ability to learn any function [15]. However, practical experience has revealed that this theoretical potential is often not realized. Researchers have addressed this problem by developing more specialized layers like convolutional and recurrent layers. These layers basically take advantage of the inductive bias based on spatial or sequential structures of specific data types such as text, image, video etc. In this study, we came up with a proposal to add a kind of inductive bias to fully connected layers by feeding the inherent cluster information of the input samples.

2.2. Fuzzy Clustering

Fuzzy clustering is a clustering approach which separates a dataset into fuzzy partitions. In contrast to hard clustering, fuzzy clustering allows the elements of a dataset to share some fraction of membership among multiple clusters. This “fraction of membership” is called “membership degree” and in different fuzzy clustering algorithms it is represented as a Type-I, Type-II or Intuitionistic fuzzy set [17]. Throughout the years, various fuzzy clustering algorithms were developed to tackle problems such as outlier handling, sparsity, high dimensionality and nonlinear cluster separation. Type-I Fuzzy C-Means, Type-II Fuzzy C-Means, Possibilistic C-Means and Noise Clustering may be given as examples to prominent techniques in this field. For more information on the technical details of these
algorithms and their strengths and weaknesses under diverse circumstances the reader may refer to [17–22]. In this study, for the clustering process of the proposed method, we used the Type-I Fuzzy C-Means algorithm (for a detailed introduction refer to [23]), which is one of the most widely used fuzzy clustering techniques. The algorithm aims to minimize the objective function in Equation (1).

\[ J_p = \sum_{i=1}^{m} \sum_{j=1}^{c} \mu_{ij}^p ||x_i - w_j||^2 \]  

where \( p \) is a real number bigger than one, \( m \) is the number of observations, \( c \) is the number of clusters, \( \mu_{ij} \) is membership degree of observation \( i \) to cluster \( j \), \( x_i \) is the \( i \)th observation in \( n \) dimensional sample data, \( w_j \) is the centroid of \( j \)th cluster and \( ||*|| \) is the Euclidian norm.

The centroid of cluster \( j \) (\( w_j \)) is computed as in Equation (2) and the membership degree of observation \( i \) to cluster \( j \) (\( \mu_{ij} \)) is computed based on Equation (3).

\[ w_j = \frac{\sum_{i=1}^{m} \mu_{ij}^p x_i}{\sum_{i=1}^{m} \mu_{ij}^p} \]  

\[ \mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{||x_i - w_j||}{||x_i - w_k||} \right)^{\frac{p-1}{2}}} \]  

Reasons for preferring this algorithm are its efficiency, robustness, and ability to deal with overlapping data [17,19,24]. Moreover, results of a recent experimental study [5], where a fuzzy cluster-aware regularization method for feedforward neural networks is presented, have shown that the Fuzzy C-Means algorithm outperforms the k-means version. Indeed, the selection of the clustering algorithm is a kind of parametric choice and therefore a potential topic for further research.

3. Proposed Method

3.1. Motivation

Fully connected layers are designed to extract features to fulfill a classification task without having any prior information about the structure of the data. In fact, this lack of prior information constitutes the main difference of fully connected layers from other deep learning layers such as convolutional and recursive layers which are designed to exploit the intrinsic structure of training data. In addition to the differences made by these specialized layers, many studies have shown that this intrinsic data structure has an important potential to improve classification performance [5,25–27]. Having prior knowledge about the formed clusters in a dataset gives an opportunity to gain a good grasp of its nature [6]. Also, preserving the cluster structure of a dataset while creating its representations in a feature space has a proven importance for the performance of a classifier [4,14]. This potential led us to search for a way to add an inductive bias, that represents the clustering structure of the training dataset, to fully connected layers.

Each fully connected layer applies a transformation to the feature space in which the problem is represented [16]. Intuitively, in order that the observations preserve their clustering patterns in the new feature space, prior information about the original clustering structure should be fed to the layer. At this point, our assumption is that if a fully connected layer is fed by such prior information, the representation ability of the features constructed by this layer will increase in a way that improves classification performance. With this purpose, we propose a new training pipeline for fully connected layers in which the extracted features are expected to have the ability to cluster the dataset in the same way as in the original feature space. Next, we will discuss the algorithmic details of our proposed method.
3.2. Algorithmic Details of the Proposed Method

The proposed method consists of two main stages which are pre-training and training. In the pre-training stage, we cluster the dataset using the Fuzzy C-Means algorithm and come up with a matrix that contains the fuzzy membership degrees of each observation to each cluster. Here, the number of fuzzy clusters to choose is a hyperparameter of our method. The resulting fuzzy membership degrees matrix becomes an input to the second main stage of the proposed method. In the training stage, we aim to train a fully connected layer in a way to minimize a combined cost function that includes both classification and clustering costs aggregated in a weighted manner. The weighting between clustering and classification costs is another hyperparameter of our method. As the first step in the training process, a centroid matrix of size $n_h \times c$, where $n_h$ denotes the number of hidden nodes in the fully connected layer and $c$ denotes the number of fuzzy clusters, is randomly initialized. This matrix holds the cluster centroids in the transformed feature space created by the fully connected layer. These centroids are trainable and they are learned through the backpropagation process. The second step of the training process is to calculate the Euclidian distance between the activation values of the fully connected layer and the cluster centroids for each observation (see Equation (4)). Subsequently, these distances are transformed into predicted fuzzy membership degrees using the standard formula employed by the Fuzzy C-Means algorithm (see Equation (5)). Then, for each observation, the mean squared error (MSE) between these predicted fuzzy membership degrees and the target fuzzy membership degrees computed in the pre-training stage is calculated (see Equation (6)). Afterwards, the clustering cost is obtained as the binary cross entropy loss between the MSE values and a zero vector (see Equation (8)). Here the point is that these MSE values are all expected to be zero so as to get the same clustering structure of observations in the new feature space. Thus, the cross-entropy results against a zero vector are employed as the clustering cost of the fully connected layer. The reason why we use the cross-entropy result rather than the MSE result itself as the clustering cost, is to prevent any possible scale related bias that may happen when averaged with the classification cost. On the other hand, the classification cost is computed in exactly the same manner as the traditional training process of a fully connected layer. As the final step, the total cost on which the backpropagation will be executed is obtained as the weighted average of the classification and clustering costs (see Equation (9)). All these steps of the training stage of the proposed method are visualized in Figure 2 for a single fully connected layer. Also, related computational details are given in Algorithm 1 “Proposed Algorithm for a Single Fully Connected Layer” part below.

3.3. Extension to Multiple Fully Connected Layers

Extension of the proposed method to multiple fully connected layers is fairly natural and easy. The only difference compared to the single fully connected layer structure is the computation of clustering cost, which in this case should be the weighted average of “v” layer clustering costs where “v” denotes the number of fully connected layers. The computation of clustering costs for each layer are executed independent of each other and in exactly the same manner as in the single layer case. This new structure adds “v-1” new hyperparameters to the proposed method, each representing the weight of the clustering cost incurred by the layer it belongs to. Figure 3 describes the multiple layer structure for two fully connected layers. In this figure, the weights of different layers’ clustering costs are denoted by $\alpha_v$, where $\sum_{i=1}^{v} \alpha_i = 1$. 
Figure 2. Forward propagation steps of the proposed method for single fully connected layer.

Figure 3. Forward propagation steps of the proposed method for multiple fully connected layers.
Algorithm 1 Proposed Algorithm for a Single Fully Connected Layer

**Pre-Training Stage:**

**Input(s):** Training dataset $X$, a matrix of size $n_x \times m$, where $n_x$ is the number of features and $m$ is the number of observations.

Perform Fuzzy C-Means clustering on the training dataset. Number of clusters to be used is a hyperparameter of the proposed algorithm.

**Output(s):** A matrix $F$ of size $m \times c$, where $c$ is the number of fuzzy clusters. This matrix contains the fuzzy membership degrees of observation to each cluster.

**Training Stage:**

**Input(s):** $F$ matrix and training dataset $X$.

**Centroid Initialization Step:** Randomly initialize the predicted centroids matrix $PC$ of size $n_y \times c$ where $n_y$ is the number of hidden nodes in the fully connected layer. This matrix contains the coordinates of fuzzy centroids in $n_y$ dimensional space. Make the variables in this matrix trainable, since the centroids in the transformed feature space will be learned during the backpropagation process.

**Distance Calculation Step:** Compute the Euclidian distance of fully connected layer activation values of each observation ($a_i$) to each cluster centroids computed in the previous step, where $a_i$ is a vector of size $n_y$ which holds the activation values of the $i$th observation in the batch (see Equation (4)). This calculation ends up with a matrix $D$ of size $batch\_size \times c$, where $batch\_size$ is the number of observations in the current batch. Each item in matrix $D$ (i.e., $d_{ij}$ where $i \in [1 \ldots batch\_size]$ and $j \in [1 \ldots c]$) holds the Euclidian distance of each observation to each cluster centroids in the transformed feature space:

$$d_{ij} = ||a_i - pc_{ij}||^2$$  \hspace{1cm} (4)

**Clustering Output Step:** Compute predicted membership degrees of each observation to each fuzzy cluster by using the membership degree formulation of the Fuzzy C-Means algorithm. $u_{ij}$ is the predicted membership degree of observation $i$ to cluster $j$. This step results in a matrix $U$ of size $batch\_size \times c$. Equation (5) transforms the Euclidian distances calculated in distance calculation step into membership degrees. In this equation, $p$ stands for the fuzziness index and it is taken as 2 in this study.

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( ||a_i - pc_{ik}||^p \right)^{1/p}}$$  \hspace{1cm} (5)

**MSE Calculation Step:** Compute the mean squared error between predicted membership degrees and original membership degrees that are computed in the pre-training steps. This calculation results in a vector $e$ of size $batch\_size$. Note that since both predicted membership degrees and original membership degrees are between 0–1, the resulting MSE values also end up between 0–1. In Equation (6), $f_{ij}$ corresponds to the element of matrix $F$ which holds the membership degree of the $i$th observation in the current batch to the $j$th fuzzy cluster. The MSE values in vector $e$ define how compatible is the clustering done by the fully connected layer with the clustering conducted in the pre-training phase.

$$e_i = \frac{1}{c} \sum_{j=1}^{c} (u_{ij} - f_{ij})^2$$  \hspace{1cm} (6)

**Clustering Cost Step:** Compute binary cross entropy loss between MSE values that are computed in the previous step and a zero vector ($y$) of the same shape. For the reason of preferring a cross entropy loss rather than directly using the MSE loss as the clustering cost, please refer to "Algorithmic Details of the Proposed Method" part of this study. Cross entropy calculation is given in Equation (7). Since all elements of $y$ are zero, Equation (7) simplifies to Equation (8). This calculation finalizes the clustering cost that will contribute to the total cost.

$$\text{Clustering Cost} = \frac{1}{batch\_size} \sum_{i=1}^{batch\_size} y_i \log(e_i) + (1 - y_i) \log(1 - e_i)$$  \hspace{1cm} (7)

$$\text{Clustering Cost} = \frac{1}{batch\_size} \sum_{i=1}^{batch\_size} \log(1 - e_i)$$  \hspace{1cm} (8)

**Classification Cost Step:** Compute the classification cost in the same way as the traditional training process of a fully connected layer.

**Total Cost Step:** Take weighted average of classification and clustering costs to compute the total cost of which backpropagation will take the derivatives (see Equation (9)). The weights $\beta_1$ and $\beta_2$ are hyperparameters for the proposed algorithm where $\beta_1 \in [0,1]$ and $\beta_2 \in [0,1]$. Note that since $\beta_1$ and $\beta_2$ aim to allow for taking the weighted average of two costs, their sum should be equal to 1.

$$\text{Total Cost} = \beta_1 \times \text{Classification Cost} + \beta_2 \times \text{Clustering Cost}$$  \hspace{1cm} (9)

**Output(s):** Total cost.
4. Experiments

In this section, the performance of the proposed method is evaluated over 9 different UCI datasets [28]. These datasets were chosen as a way to allow for a diversity in experiments in terms of number of observations, dimensions and classes. References [6,8,29,30] are related example studies which also preferred the same datasets in their experiments. Descriptions of the datasets are given in Table 2. Since the “frogs” dataset contains three different levels of class definitions which are family, genus and species, experiments were run separately for each of the class levels. Thus, 11 different experiments were run on 9 different datasets to validate the performance of the proposed method. The proposed method and other parts of the experimental setup were coded in TensorFlow 2 [31]. All datasets were divided into training and test sets with a ratio of 4:1. In all experiments, the classification accuracy of a regular single fully connected layer is compared against the classification accuracy of a single fully connected layer which is trained by the proposed method. For each dataset, 50 trainings each with a different random weight initialization were executed for both the proposed method and a regular fully connected layer. In order to allow for a valid comparison, the same random initialization seeds were employed for both methods at each repetition. So as to determine whether a statistically significant accuracy improvement is achieved compared to the regular fully connected layer, a Wilcoxon signed-rank test was conducted between 50 test set accuracy values of both methods.

Table 2. Descriptions of UCI datasets used in the experiments.

| Dataset             | # of Observations | # of Dimensions | # of Classes |
|---------------------|-------------------|-----------------|--------------|
| ionosphere          | 351               | 34              | 2            |
| sonar               | 208               | 60              | 2            |
| new thyroid         | 215               | 5               | 3            |
| vehicle silhouettes | 846               | 18              | 4            |
| ecoli               | 336               | 7               | 8            |
| default credit card | 30,000            | 33              | 2            |
| frogs_family        | 7195              | 22              | 4            |
| frogs_genus         | 7195              | 22              | 8            |
| frogs_species       | 7195              | 22              | 10           |
| wdbc                | 569               | 30              | 2            |
| image segmentation  | 2310              | 19              | 7            |

*In case of categorical variables, total number of dimensions after one-hot encoding are given.

In each experiment, the number of fuzzy clusters in the pre-training steps of the proposed method were taken as equal to the number of classes in the dataset. Due to the known deficiencies of the Fuzzy C-Means algorithm in high dimensional spaces [32], for “sonar” and “default credit card” datasets fuzzy clustering was executed using the principal component scores instead of the original features. First 9 and 6 principal components were used for “sonar” and “default credit card” datasets, respectively. Considering the weights of clustering and classification costs in the total cost, it is essential to note that since the main task of the fully connected layer is classification, a higher weight of classification cost compared to clustering cost is intuitive. Accordingly, these weights were chosen as in Table 3.

It should also be pointed out that since the weights of classification and clustering costs in the total cost were taken as hyperparameters in this study, they are subject to be selected following the regular hyperparameter optimization techniques. However, approaching these values as “variables” rather than “hyperparameters” and allowing them to be learned through the backpropagation process is another option which deserves to be a further research topic. The remaining hyperparameters were selected in a way that assured a smooth decrease and stabilization in the training cost for both the proposed method and regular fully connected layer. All hyperparameter values for each experiment are presented in Table 3. In the next section, results of the experiments are presented, also an explanatory analysis on a multiple consecutive fully connected layer case is given.
Table 3. Hyperparameter values used in 11 different experiments.

| Dataset            | Learning Rate | Batch Size | Epochs | FC Layer Hidden Unit Count | Fuzzy Cluster Number | Total Cost Weights | Repetition Count |
|--------------------|---------------|------------|--------|----------------------------|----------------------|--------------------|------------------|
| ionosphere         | 0.001         | 64         | 100    | 10                         | 2                    | “classification”: 0.9,”clustering”: 0.1 | 50               |
| sonar *            | 0.001         | 64         | 100    | 10                         | 2                    | “classification”: 0.9,”clustering”: 0.1 | 50               |
| new thyroid        | 0.01          | 64         | 100    | 10                         | 3                    | “classification”: 0.9,”clustering”: 0.1 | 50               |
| vehicle silhouettes| 0.001         | 64         | 150    | 10                         | 4                    | “classification”: 0.7,”clustering”: 0.3 | 50               |
| ecoli              | 0.001         | 64         | 250    | 20                         | 8                    | “classification”: 0.8,”clustering”: 0.2 | 50               |
| default credit card*| 0.001         | 64         | 5      | 10                         | 2                    | “classification”: 0.9,”clustering”: 0.1 | 50               |
| frogs_family       | 0.001         | 64         | 20     | 20                         | 4                    | “classification”: 0.9,”clustering”: 0.1 | 50               |
| frogs_genus        | 0.001         | 64         | 20     | 20                         | 8                    | “classification”: 0.9,”clustering”: 0.1 | 50               |
| frogs_species      | 0.001         | 64         | 20     | 20                         | 10                   | “classification”: 0.9,”clustering”: 0.1 | 50               |
| wdbc               | 0.001         | 64         | 50     | 10                         | 2                    | “classification”: 0.9,”clustering”: 0.1 | 50               |
| image segmentation | 0.01          | 64         | 100    | 20                         | 7                    | “classification”: 0.7,”clustering”: 0.3 | 50               |

* In pre-training steps, Fuzzy C-Means conducted using first 9 and 6 principal component scores for “sonar” and “default credit card” datasets respectively.

Results and Analysis

Comparison of the test set classification accuracy of a single fully connected layer, which is trained by the proposed method and a regular single fully connected layer, are summarized in Table 4. In each experiment, a statistically significant difference was sought between the mean accuracy values to reach the conclusion that one of the two methods outperforms the other. According to the results of the Wilcoxon signed-rank test with 0.05 significance level (see the last column in Table 4), in ten of the 11 experiments the proposed method achieves statistically significant higher accuracies in test sets. In the remaining experiment, which employ the “new thyroid” dataset, no statistically significant difference is observed between the two methods. Another measure that is worth reporting next to the \( p \)-value is the effect size that provides an estimation of the magnitude and thereby the importance of the results obtained. Except the dataset new thyroid, the results demonstrate an effect size estimate ranging between medium and high magnitude of a difference in accuracy (for more details on the benchmark values please refer to the recent studies by Gignac & Szodorai [33] and Funder and Ozer [34]). In the fourth column of Table 4, the number of repetitions in which the proposed method resulted in an accuracy value that is greater than or equal to the regular fully connected layer is presented for each dataset. Similarly, in the fifth column of the same table, the number of repetitions in which the regular fully connected layer resulted in an accuracy value that is greater than or equal to the proposed method is presented. These numbers reveal that the proposed method beats the regular fully connected layer in most of the repetitions consistently for each dataset. It is worth mentioning in particular, that for the “ecoli” dataset, the proposed method got beaten in none of the 50 repetitions. The results show that even for very low significance levels, the proposed method is superior compared to a regular fully connected layer in all datasets except “new thyroid”.

Figure 4 presents the box-plot diagrams for the test set accuracies of 50 repetitions in each experiment. The first thing to notice in these plots is that the proposed method does not perform worse than a regular fully connected layer in any of the experiments (in terms of the interquartile range of the result set). Additionally, in the datasets ionosphere, sonar, vehicle silhouettes, ecoli, default credit card, frogs_genus, frogs_species, wdbc and image segmentation, the results of the proposed method are distributed within a range resulting in smaller or at least equal standard deviations compared to the results of the regular fully connected layer. Also, a high correlation between the results of the two methods justifies our choice of Wilcoxon signed-rank test as the comparison tool.
Table 4. Comparison of average test accuracy values for 50 repetitions.

| Dataset              | Proposed Method * | Regular Fully Connected Layer * | # of Proposed Method ≥ Regular FCL | # of Proposed Method ≤ Regular FCL | Wilcoxon Signed-Rank Test p-Value ** |
|----------------------|-------------------|---------------------------------|-----------------------------------|-----------------------------------|-------------------------------------|
| ionosphere           | 86.54 (8.9)       | 84.54 (10.1)                    | 42                                | 24                                | 0.006 (0.272)                       |
| sonar                | 78.29 (8.1)       | 76.90 (8.7)                     | 40                                | 24                                | 0.024 (0.226)                       |
| new thyroid          | 88.74 (6.8)       | 88.62 (7.0)                     | 44                                | 41                                | 0.863 (0.017)                       |
| vehicle silhouettes  | 77.55 (4.0)       | 76.35 (4.6)                     | 40                                | 24                                | 0.004 (0.284)                       |
| ecoli                | 79.53 (10.8)      | 79.03 (10.8)                    | 50                                | 37                                | 0.001 (0.330)                       |
| default credit card  | 80.58 (1.4)       | 80.44 (1.4)                     | 37                                | 20                                | 0.002 (0.304)                       |
| frogs_family         | 95.62 (4.2)       | 95.26 (4.1)                     | 47                                | 12                                | 0.000 (0.531)                       |
| frogs_genus          | 93.63 (3.9)       | 93.15 (4.0)                     | 48                                | 8                                 | 0.000 (0.544)                       |
| frogs_species        | 91.61 (5.9)       | 91.01 (6.5)                     | 47                                | 8                                 | 0.000 (0.501)                       |
| wdbc                 | 96.56 (5.0)       | 96.30 (5.0)                     | 47                                | 35                                | 0.004 (0.288)                       |
| image segmentation   | 77.95 (11.0)      | 77.76 (11.0)                    | 38                                | 14                                | 0.005 (0.284)                       |
| **Average**          | **86.05**         | **85.4**                        |                                   |                                   |                                     |

* Values in brackets indicate the standard deviations of accuracy values; ** Values in brackets indicate the effect size.

Figure 4. Box plot diagrams of test set accuracies for each experiment.
Moreover, in order to observe the change in the clustering cost of each consecutive fully connected layer in multiple layer cases, we conducted another analysis using the “ionosphere” dataset. In this setup, unlike previous experiments, we used six consecutive fully connected layers, each consisting of ten hidden units, to perform the classification task. The number of fuzzy clusters were selected as two for each layer and the clustering cost of each layer was given equal weight in the total clustering cost. Weights of classification and clustering costs in the total cost were determined with a ratio of 9:1. Figure 5 shows an example course of each layer’s clustering cost during a 100-epoch training process.

![Figure 5](image)

**Figure 5.** Fully connected layer clustering costs on training set.

According to Figure 5, as we move away from the original feature space, in other words as we move towards the last layers, the resulting clustering cost tends to increase. Intuitively, as we move towards the last layers it becomes difficult for the obtained features to catch the clustering pattern created by the original features. Exploring the behaviors and importance of each layer’s clustering capabilities in a multilayer architecture requires detailed analysis and it is the subject of future research however, based on Figure 5 we think it is fair to assert that last layers’ effects on the total network cost will be higher compared to previous ones. In the final section, concluding remarks, potential further research topics are summarized and possible improvement points of the proposed method are presented.

### 5. Conclusions and Suggestions for Future Research

In this paper, a new training method for fully connected layers has been presented. The key idea of the proposed algorithm is to enable the features extracted by the fully connected layer to cluster the dataset in accordance with the clustering structure in the original feature space. By using this algorithm, the representation ability of the extracted features is expected to increase in a way that improves classification performance. Achievements of other deep learning layers, like convolutional and recursive layers which basically contain an inductive bias that represents the intrinsic structure of specific data types, has been an inspiration point for this study. The proposed method mainly aims to add a kind of an inductive bias to fully connected layers by feeding the endogenous clustering information of the training samples. With this purpose, a new training pipeline consisting of pre-training and training stages was proposed for fully connected layers. This pipeline employs a combined cost function which is the weighted average of a regular classification cost and a clustering cost that incurs from the clustering abilities of the extracted features.
A total of 11 different experiments on nine UCI datasets were conducted to evaluate the classification accuracy of the proposed method against a regular fully connected layer. The results showed that, in all experiments except one, the proposed method results in statistically significant higher accuracy values compared to a regular fully connected layer. Moreover, in the case of multiple consecutive fully connected layers, it was observed that the last layers tend to have a bigger effect on the total network cost compared to prior layers.

The key contributions of this paper can be listed as follows: (i) it proposes a new training process which makes fully connected layers benefit from the clustering structure of the training dataset; (ii) it puts forward an enhanced fully connected layer which has the ability to classify and cluster a dataset simultaneously; (iii) it incorporates the learning process of cluster centroids into backpropagation; (iv) it conducts experiments that indicate superior prediction performances of the proposed method in various benchmark datasets compared to regular fully connected layers, and (v) it is ready to be employed, without any revision, in any classification architecture that uses fully connected layers.

There are a couple of aspects that can be further addressed as future research on this study. Firstly, hyperparameter optimization for the proposed method seems to have a high potential to improve current prediction performances. Also, selection of the clustering algorithm in the pre-training stage with respect to the structure of the dataset is another topic that can be further studied. Furthermore, it is fair to assert that in a classification problem, the task is more difficult around the boundary regions separating the classes. Considering this fact, revising the proposed method to focus only on cluster structures in these boundary regions may be another way to achieve bigger improvements in classification performances. Lastly, applying a similar methodology to improve prediction performances of recursive and convolutional layers that form the basis of many deep learning architectures may be another further research topic.

Author Contributions: Conceptualization, T.A.K.; methodology, T.A.K.; software, T.A.K.; validation, T.A.K. and U.A.; formal analysis, T.A.K. and U.A.; investigation, T.A.K. and U.A.; resources, T.A.K.; data curation, T.A.K.; writing—original draft preparation, T.A.K.; writing—review and editing, T.A.K. and U.A.; supervision, U.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets that are used in this study can be accessed from https://archive.ics.uci.edu/ml/index.php (accessed on 31 January 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Piernik, M.; Morzy, T. A study on using data clustering for feature extraction to improve the quality of classification. Knowl. Inf. Syst. 2021, 63, 1771–1805. [CrossRef]
2. Gupta, A.; Kumar, D. Fuzzy clustering-based feature extraction method for mental task classification. Brain Inform. 2017, 4, 135–145. [CrossRef]
3. Li, K.; Wu, Y.; Song, S.; Sun, Y.; Wang, J.; Li, Y. A novel method for spacecraft electrical fault detection based on FCM clustering and WPSVM classification with PCA feature extraction. Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng. 2017, 231, 98–108. [CrossRef]
4. Srivastava, S.; Kawaguchi, K.; Rajan, V. ExpertNet: A Symbiosis of Classification and Clustering. arXiv 2022, arXiv:2201.06344.
5. Kalayci, T.A.; Asan, U. A new fuzzy cluster-aware regularization of neural networks. J. Intell. Fuzzy Syst. 2020, 39, 6487–6496. [CrossRef]
6. Cai, W.; Chen, S.; Zhang, D. A simultaneous learning framework for clustering and classification. Pattern Recognit. 2009, 42, 1248–1259. [CrossRef]
7. Cai, W.; Chen, S.; Zhang, D. A multiobjective simultaneous learning framework for clustering and classification. IEEE Trans. Neural Netw. 2010, 21, 185–200. [CrossRef]
8. Qian, Q.; Chen, S.; Cai, W. Simultaneous clustering and classification over cluster structure representation. *Pattern Recognit.* 2012, 45, 2227–2236. [CrossRef]

9. Hebboul, A.; Hachouf, F.; Boulemadjiel, A. A new incremental neural network for simultaneous clustering and classification. *Neurocomputing* 2015, 169, 89–99. [CrossRef]

10. Fang, B.; Li, Y.; Zhang, H.; Chan, J.C.W. Collaborative learning of lightweight convolutional neural network and deep clustering for hyperspectral image semi-supervised classification with limited training samples. *ISPRS J. Photogramm. Remote Sens.* 2020, 161, 164–178. [CrossRef]

11. Sellars, P.; Aviles-Rivero, A.; Schönlieb, C.B. Two Cycle Learning: Clustering Based Regularisation for Deep Semi-Supervised Classification. *arXiv* 2020, arXiv:2001.05317.

12. Huang, B.; Zhu, Y.; Wang, Z.; Fang, Z. Imbalanced Data Classification Algorithm Based on Clustering and SVM. *J. Circuits Syst. Comput.* 2021, 30, 2150036. [CrossRef]

13. Chaudhuri, U.; Chaudhuri, S.; Chaudhuri, S. GuCNet: A guided clustering-based network for improved classification. In *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy, 10–15 January 2020; pp. 7335–7342. [CrossRef]

14. Ma, W.; Tu, X.; Luo, B.; Wang, G. Semantic clustering based deduction learning for image recognition and classification. *Pattern Recognit.* 2022, 124, 108440. [CrossRef]

15. Green, A. A guide to Deep Learning. *Nat. Med.* 2019, 25, 24–29.

16. Reza Bosagh Zadeh, B.R. 4. Fully Connected Deep Networks—TensorFlow for Deep Learning. Available online: https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html (accessed on 14 November 2021).

17. Gosain, A.; Dahiya, S. Performance Analysis of Various Fuzzy Clustering Algorithms: A Review. *Procedia Comput. Sci.* 2016, 79, 100–111. [CrossRef]

18. Baraldi, A.; Blonda, P. A survey of fuzzy clustering algorithms for pattern recognition—Part I. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 1999, 29, 778–785. [CrossRef]

19. Ruspini, E.H.; Bezdek, J.C.; Keller, J.M. Fuzzy clustering: A historical perspective. *IEEE Comput. Intell. Mag.* 2019, 14, 45–55. [CrossRef]

20. Miller, D.J.; Nelson, C.A.; Cannon, M.B.; Cannon, K.P. Comparison of Fuzzy Clustering Methods and Their Applications to Geophysics Data. *Appl. Comput. Intell. Soft Comput.* 2009, 2009, 876361. [CrossRef]

21. Almeida, R.J.; Sousa, J.M.C. Comparison of fuzzy clustering algorithms for classification. In *Proceedings of the 2006 International Symposium on Evolving Fuzzy Systems, Ambelside, UK, 7–9 September 2006*; pp. 112–117. [CrossRef]

22. Li, C.; Cerrada, M.; Cabrera, D.; Sanchez, R.V.; Pacheco, F.; Ulutagay, G.; Valente De Oliveira, J. A comparison of fuzzy clustering algorithms for bearing fault diagnosis. *J. Intell. Fuzzy Syst.* 2018, 34, 3565–3580. [CrossRef]

23. Ross, T.J. *Fuzzy Logic with Engineering Applications*, 3rd ed.; John Wiley & Sons Ltd.: Chichester, UK, 2010; ISBN 9780470743768.

24. Kannan, S.R.; Ramathilagam, S.; Chung, P.C. Effective fuzzy c-means clustering algorithms for data clustering problems. *Expert Syst. Appl.* 2012, 39, 6292–6300. [CrossRef]

25. Gan, H.; Luo, Z.; Fan, Y.; Sang, N. Enhanced manifold regularization for semi-supervised classification. *J. Opt. Soc. Am. A* 2016, 33, 1207. [CrossRef] [PubMed]

26. Gan, H.; Sang, N.; Huang, R.; Tong, X.; Dan, Z. Using clustering analysis to improve semi-supervised classification. *Neurocomputing* 2013, 101, 290–298. [CrossRef]

27. Wang, Y.; Chen, S.; Xue, H.; Fu, Z. Semi-supervised classification learning by discrimination-aware manifold regularization. *Neurocomputing* 2015, 147, 299–306. [CrossRef]

28. Dua, D.; Graff, C. UCI Machine Learning Repository. Available online: http://archive.ics.uci.edu/ml (accessed on 31 January 2022).

29. Gan, H.; Huang, R.; Luo, Z.; Xi, X.; Gao, Y. On using supervised clustering analysis to improve classification performance. *Inf. Sci.* 2018, 454–455, 216–228. [CrossRef]

30. Abpeykar, S.; Ghatee, M.; Zare, H. Ensemble decision forest of RBF networks via hybrid feature clustering approach for high-dimensional data classification. *Comput. Stat. Data Anal.* 2019, 131, 12–36. [CrossRef]

31. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Greg, S.; Corrado, A.D.; Dean, J.; Devin, M.; et al. TensorFlow: Large-scale machine learning on heterogeneous systems. *arXiv* 2015, arXiv:1603.04467.

32. Winkler, R.; Klawonn, F.; Kruse, R. Fuzzy C-means in high dimensional spaces. *Int. J. Fuzzy Syst. Appl.* 2011, 1, 16. [CrossRef]

33. Gignac, G.E.; Szodorai, E.T. Effect size guidelines for individual differences researchers. *Pers. Individ. Dif.* 2016, 102, 74–78. [CrossRef]

34. Funder, D.C.; Ozer, D.J. Evaluating Effect Size in Psychological Research: Sense and Nonsense. *Adv. Methods Pract. Psychol. Sci.* 2019, 2, 156–168. [CrossRef]