EE 589 INTRODUCTION TO ARTIFICIAL NETWORK

REPORT OF THE TERM PROJECT

REAL TIME ODOR RECOGNATION SYSTEM

FATMA ÖZYURT SANCAR

1. Introduction

The aim of this study is to develop a novel fuzzy clustering neural network (FCNN) algorithm as pattern classifiers for real-time odor recognition system.

In this type of FCNN, the input neurons activations are derived through fuzzy c mean clustering of the input data, so that the neural system could deal with the statistics of the measurement error directly.

Then the performance of FCNN network is compared with the other network which is well-known algorithm, named multilayer perceptron (MLP), for the same odor recognition system.

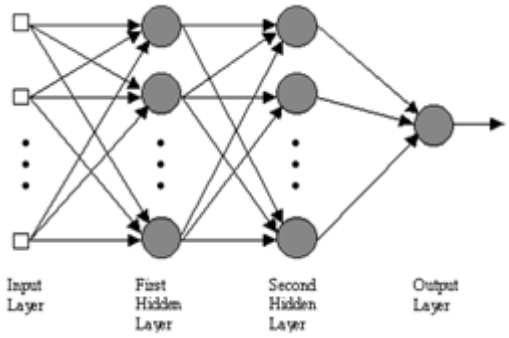
Experimental results show that both FCNN and MLP provided high recognition probability in determining various learn categories of odors, however, the FCNN neural system has better ability to recognize odors more than the MLP network.

****Fuzzy Clustering Neural Networks for Real-Time Odor Recognition System**

Bekir Karlık[1] and Kemal Y¨ uksek[2]

2.Multi Layer Perceptron

The most common neural network model is the multi layer perceptron (MLP). This type of neural network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown. A graphical representation of an MLP is shown below:



The MLP and many other neural networks learn using an algorithm called **backpropagation**. With backpropagation, the input data is repeatedly presented to the neural network. With each presentation the output of the neural network is compared to the desired output and an error is computed. This error is then fed back (backpropagated) to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural model gets closer and closer to producing the desired output. This process is known as "**training**".

Neural networks have been successfully applied to broad spectrum of data-intensive applications, such as:

- **Process Modeling and Control** - Creating a neural network model for a physical plant then using that model to determine the best control settings for the plant.
- **Machine Diagnostics** - Detect when a machine has failed so that the system can automatically shut down the machine when this occurs.
- **Portfolio Management** - Allocate the assets in a portfolio in a way that maximizes return and minimizes risk.
- **Target Recognition** - Military application which uses video and/or infrared image data to determine if an enemy target is present.
- **Medical Diagnosis** - Assisting doctors with their diagnosis by analyzing the reported symptoms and/or image data such as MRIs or X-rays.
- **Credit Rating** - Automatically assigning a company's or individuals credit rating based on their financial condition.
- **Targeted Marketing** - Finding the set of demographics which have the highest response rate for a particular marketing campaign.
- **Voice Recognition** - Transcribing spoken words into ASCII text.
- **Financial Forecasting** - Using the historical data of a security to predict the future movement of that security.
- **Quality Control** - Attaching a camera or sensor to the end of a production process to automatically inspect for defects.
- **Intelligent Searching** - An internet search engine that provides the most relevant content and banner ads based on the users' past behavior.
- **Fraud Detection** - Detect fraudulent credit card transactions and automatically decline the charge.

**http://www.mu-sigma.com/analytics/thought_leadership/cafe-cerebral-neural-network.html

3.Fuzzy C-Means Clustering

Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method (developed by Dunn in 1973 and improved by Bezdek in 1981) is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \left\| x_i - c_j \right\|^2 \quad , \quad 1 \le m < \infty$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster, and ||*|| is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}}, \qquad c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

This iteration will stop when $\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \varepsilon$, where ε is a termination criterion between 0 and 1, whereas k are the iteration steps. This procedure converges to a local minimum or a saddle point of J_m.
The algorithm is composed of the following steps:

1. *Initialize* $U=[u_{ij}]$ *matrix,* $U^{(0)}$
2. *At k-step: calculate the centers vectors* $C^{(k)}=[c_j]$ *with* $U^{(k)}$

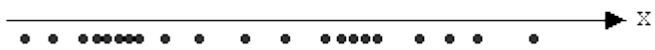
$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

3. *Update* $U^{(k)}$, $U^{(k+1)}$

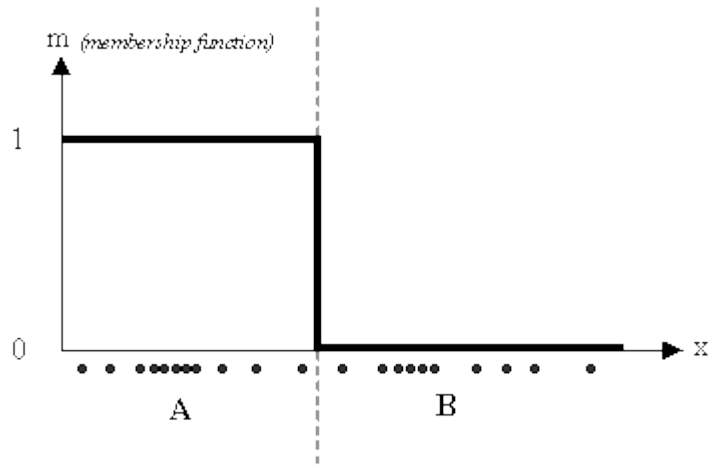
$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}}$$

4. *If* $\| U^{(k+1)} - U^{(k)} \| < \varepsilon$ *then STOP; otherwise return to step 2.*

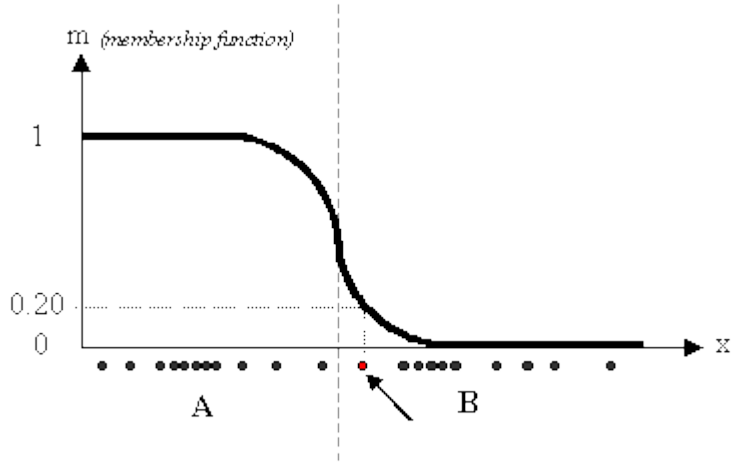
Remarks
As already told, data are bound to each cluster by means of a Membership Function, which represents the fuzzy behaviour of this algorithm. To do that, we simply have to build an appropriate matrix named U whose factors are numbers between 0 and 1, and represent the degree of membership between data and centers of clusters.
For a better understanding, we may consider this simple mono-dimensional example. Given a certain data set, suppose to represent it as distributed on an axis. The figure below shows this:



Looking at the picture, we may identify two clusters in proximity of the two data concentrations. We will refer to them using ‘A’ and ‘B’. In the first approach shown in this tutorial - the k-means algorithm - we associated each datum to a specific centroid; therefore, this membership function looked like this:



In the FCM approach, instead, the same given datum does not belong exclusively to a well defined cluster, but it can be placed in a middle way. In this case, the membership function follows a smoother line to indicate that every datum may belong to several clusters with different values of the membership coefficient.



In the figure above, the datum shown as a red marked spot belongs more to the B cluster rather than the A cluster. The value 0.2 of 'm' indicates the degree of membership to A for such datum. Now, instead of using a graphical representation, we introduce a matrix U whose factors are the ones taken from the membership functions:

$$U_{MC} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ .. & .. \\ 0 & 1 \end{bmatrix} \qquad U_{MC} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ .. & .. \\ 0.9 & 0.1 \end{bmatrix}$$

(a) (b)

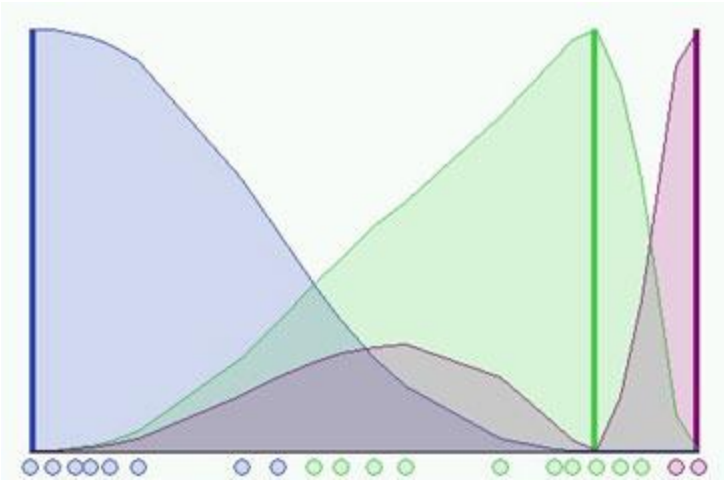
The number of rows and columns depends on how many data and clusters we are considering. More exactly we have C = 2 columns (C = 2 clusters) and N rows, where C is the total number of clusters and N is the total number of data. The generic element is so indicated: u_{ij}.

In the examples above we have considered the k-means (a) and FCM (b) cases. We can notice that in the first case (a) the coefficients are always unitary. It is so to indicate the fact that each datum can belong only to one cluster. Other properties are shown below:

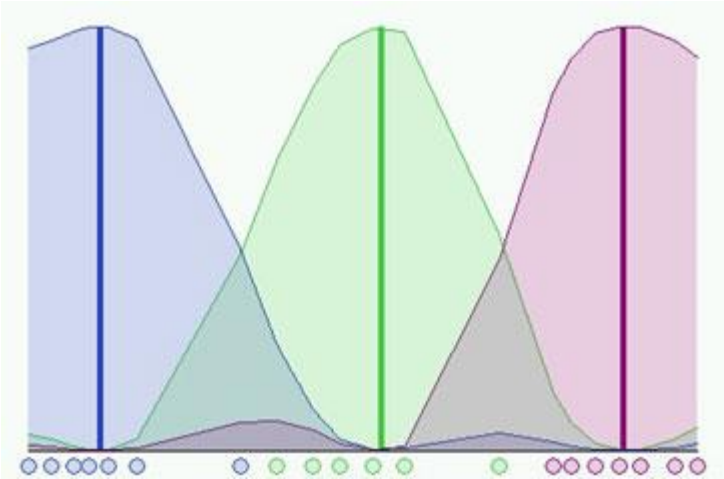
- $u_{ij} \in [0,1] \qquad \forall i, j$
- $\sum_{j=1}^{C} u_{ik} = 1 \qquad \forall i$
- $0 < \sum_{i=1}^{N} u_{ij} < N \quad \forall N$

Example

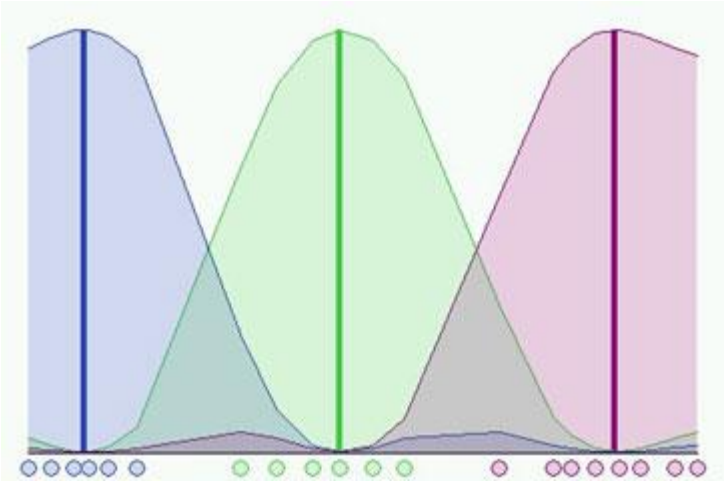
Here, we consider the simple case of a mono-dimensional application of the FCM. Twenty data and three clusters are used to initialize the algorithm and to compute the U matrix. Figures below show the membership value for each datum and for each cluster. The color of the data is that of the nearest cluster according to the membership function.



In the simulation shown in the figure above we have used a fuzzyness coefficient m = 2 and we have also imposed to terminate the algorithm when $\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < 0.3$. The picture shows the initial condition where the fuzzy distribution depends on the particular position of the clusters. No step is performed yet so that clusters are not identified very well. Now we can run the algorithm until the stop condition is verified. The figure below shows the final condition reached at the 8th step with m=2 and ε=0.3:



Is it possible to do better? Certainly, we could use an higher accuracy but we would have also to pay for a bigger computational effort. In the next figure we can see a better result having used the same initial conditions and ε=0.01, but we needed 37 steps!



It is also important to notice that different initializations cause different evolutions of the algorithm. In fact it could converge to the same result but probably with a different number of iteration steps.

- J. C. Dunn (1973): "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics* 3: 32-57
- J. C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algoritms", Plenum Press, New York

4.Real Time Odor Sensing by OMX-GR Sensors

In this study a “handheld odor meter, OMX-GR” is used to obtain odor data. This is completely manufactured by FiS as an OEM product. The OMX-GR sensor indicates two factors of odor, “strength” and “classification”, with numeric values.

This is very useful for various applications related to odor detection and measurement

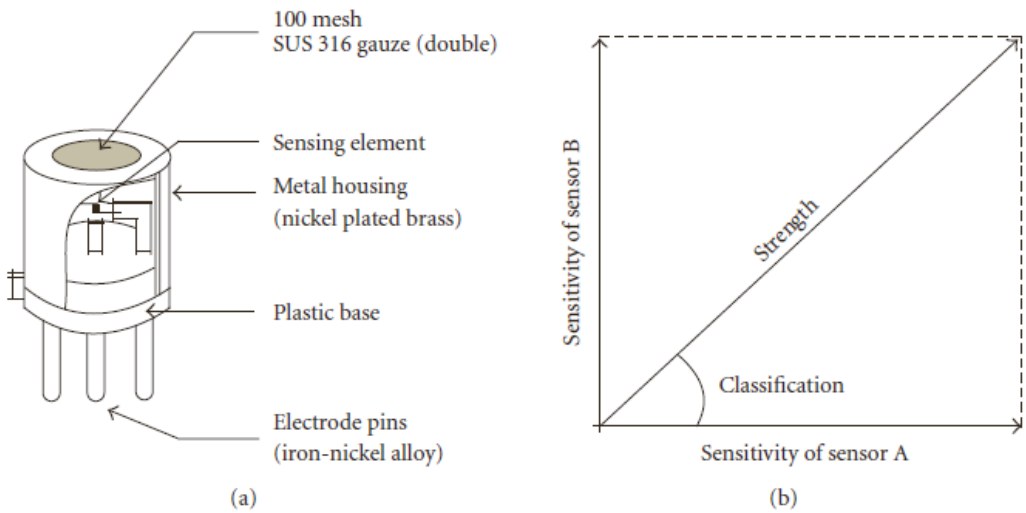


FIGURE 2: Sensor configuration and measurement principle.

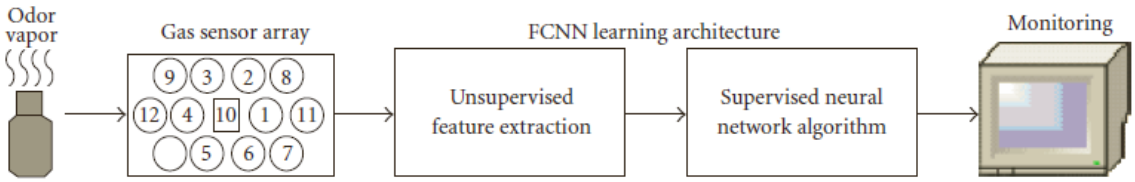


FIGURE 3: A prototype of a real-time odor recognition system.

Both of these gas sensors (OMX-GR) operate in the realtime sampling mode. The samples were delivered to a fuzzy c mean (FCM) clustering algorithm to obtain unsupervised feature extraction.

FCM is a fuzzy data clustering and partitioning algorithm in which each data point belongs to a cluster according to its degree of membership.

With FCM, an initial estimate of the number of clusters is needed so that the data set is split into *C* fuzzy groups. A cluster center is found for each group by minimizing a dissimilarity function.

Fuzzy clustering, essentiall y, deals with the task of splitting a set of patterns into a number of more or less homogeneous classes (clusters) with respect to a suitable similarity measure such that the patterns belonging to any one of the clusters are similar and the patterns of different clusters are as dissimilar as possible.

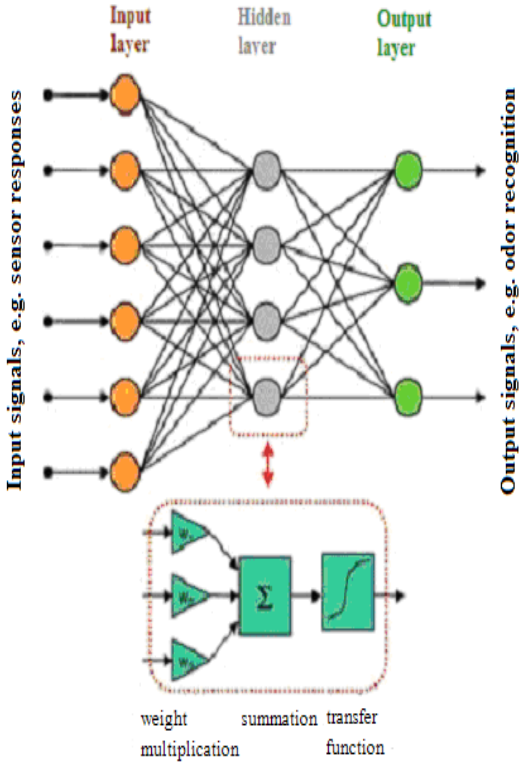
The similarity measure used has an important effect on the clustering results since it indicates which mathematical properties of the data set should be used in order to identify the clusters.

Fuzzy clustering provides partitioning results with additional information supplied by the cluster membership values, indicating different degrees of belongingness

Then the multiplexed time-series data, which belongs to 16 different odors of perfumes, are clustered and are inputs to the supervised neural network algorithm.

This neural network trained BP algorithm classifies the sensorarrayoutput patterns into odor categories. The system was trained to identify odors of 16 different perfumes with 20 samples for each.

This system allows users to obtain the desired data from a particular odorant (perfume).



5.Comparison Between MLP and FCNN in Odor Sensing

The sixteen different odors of perfume dataset were analyzed using two types ANN classifiers, namely the multilayer perceptron (MLP) and the proposed fuzzy clustering neural network (FCNN) structures.

The training of both ANN structures was performed with half of the whole data set. The
other half was used for testing both structures of neural networks.
As noted, the average mean square error (MSE) of FCNN is less than the MLP structure.

In other word, we can say that an average recognition accuracy of FCNN is better than MLP.

Moreover, it is noted, in the results above, that the FCNN converges to a determined error goal faster than the MLP.

In applications of matlab by MLP method train data has 6 iterations but in FCNN method 46 iterations had done.

As a result of this at the same time period FCNN Works more efficently according to MLP.FCNN had shown the clustering of data more than MLP .

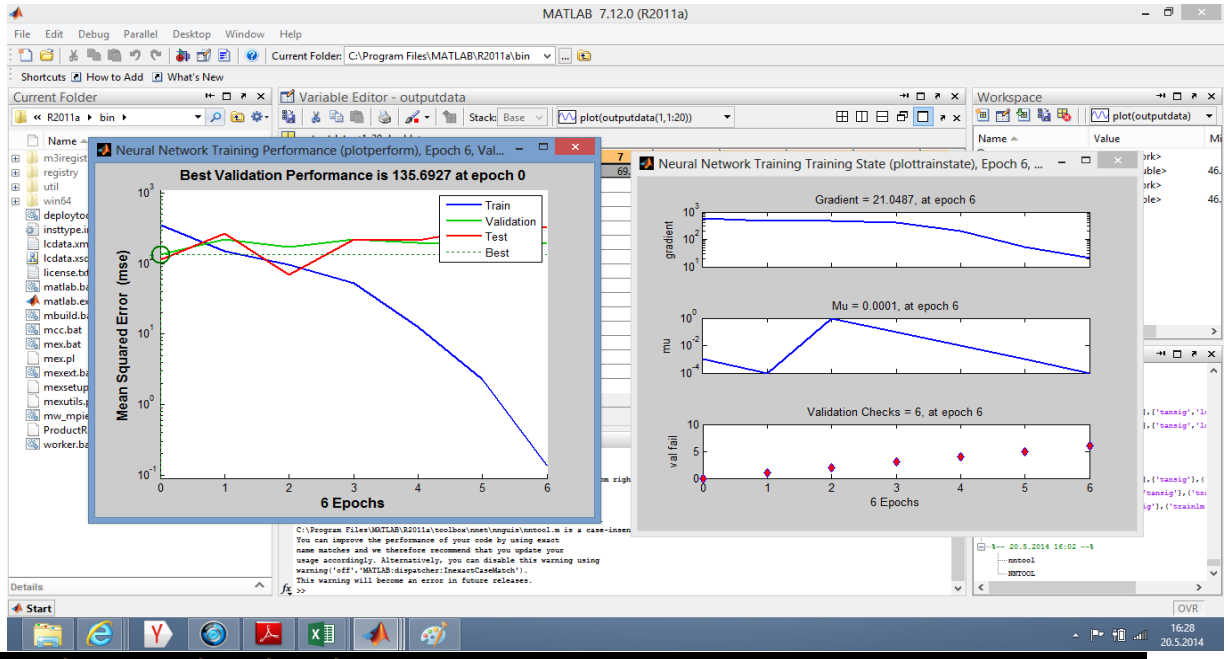
In the following section it will be seen the performance graphs of FCNN and MLP.

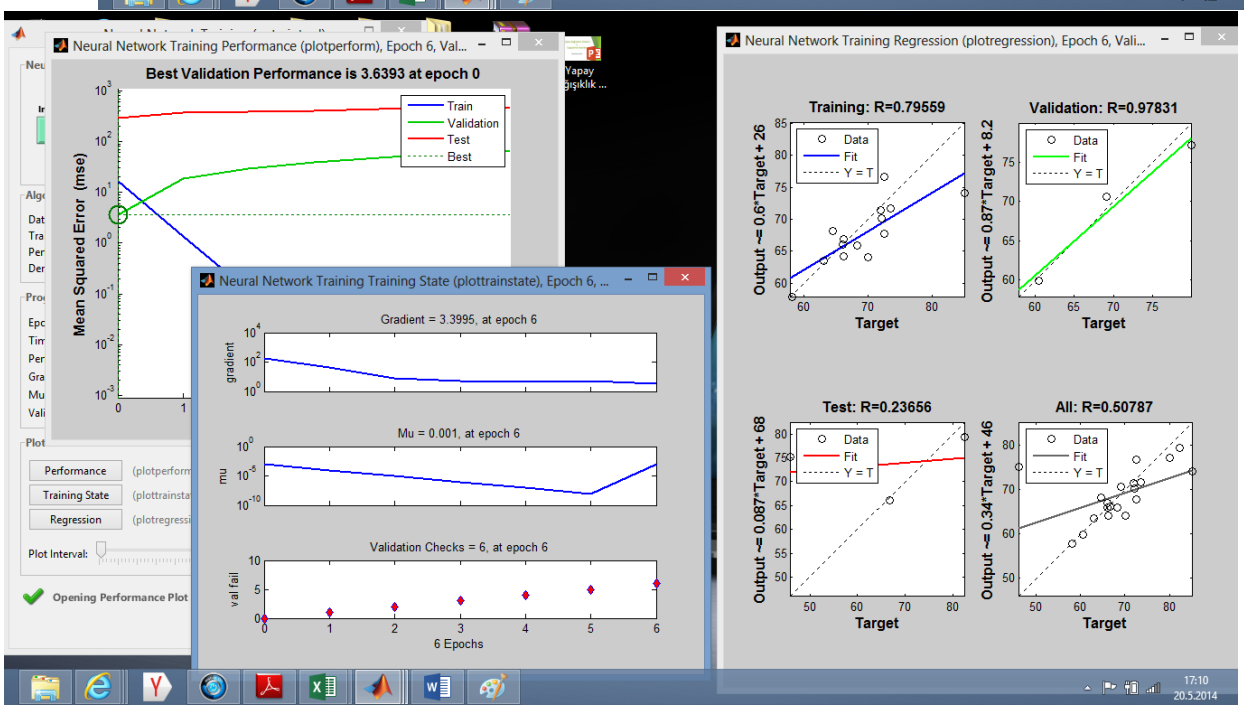
6.Matlab GUI Applications

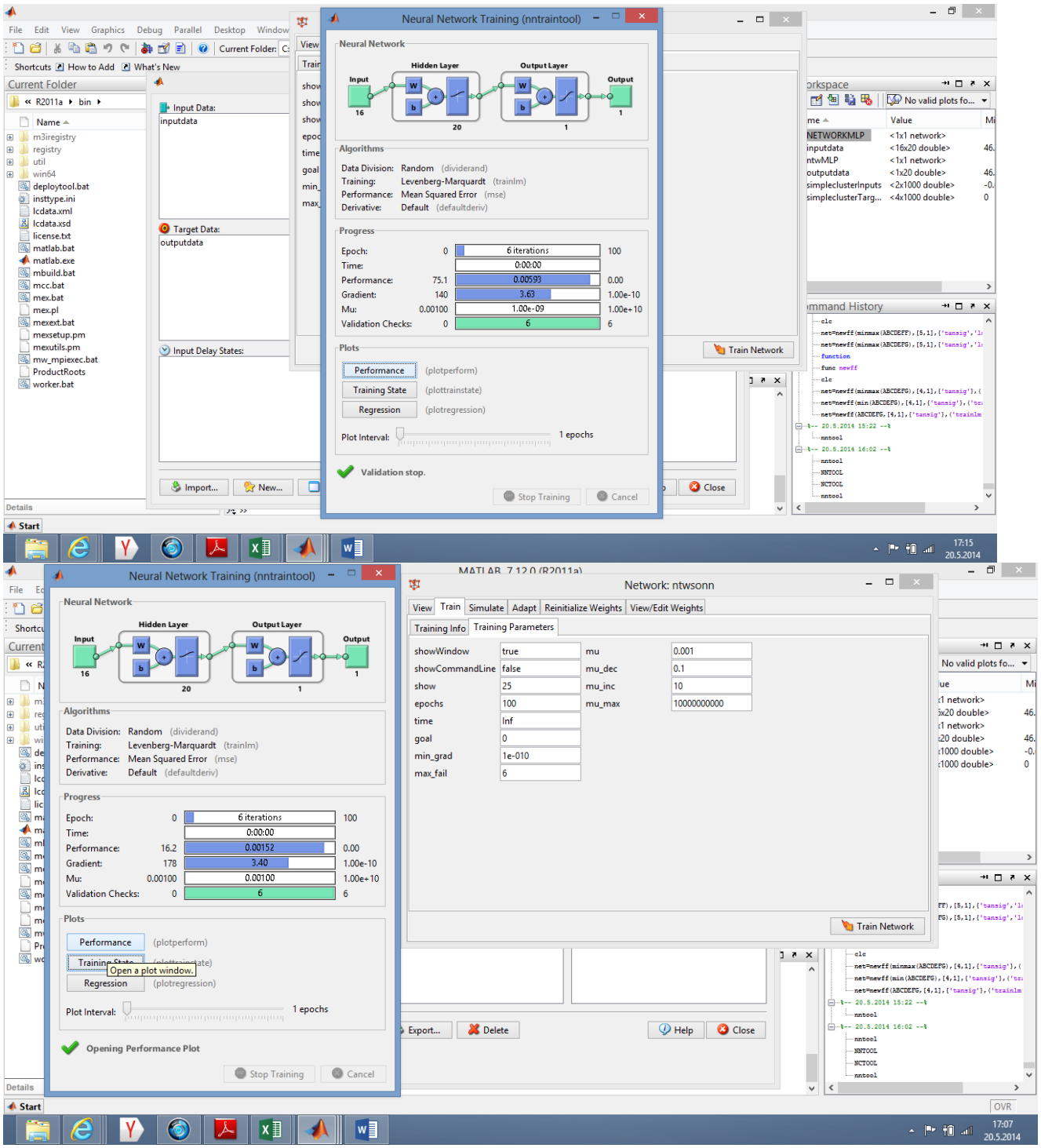
In this two pictures represent the graphics of MLP tool matlab simulation.By MLP model just six iterations had done.While calculating the weights similarity value is important but MLP can not use the similarity as efficent as FCNN .By back propagation algorithm it tries to get the similar data together.

In the second picture the regression values are seen.At R=0,50787 is the most suitable range for odor recognation.

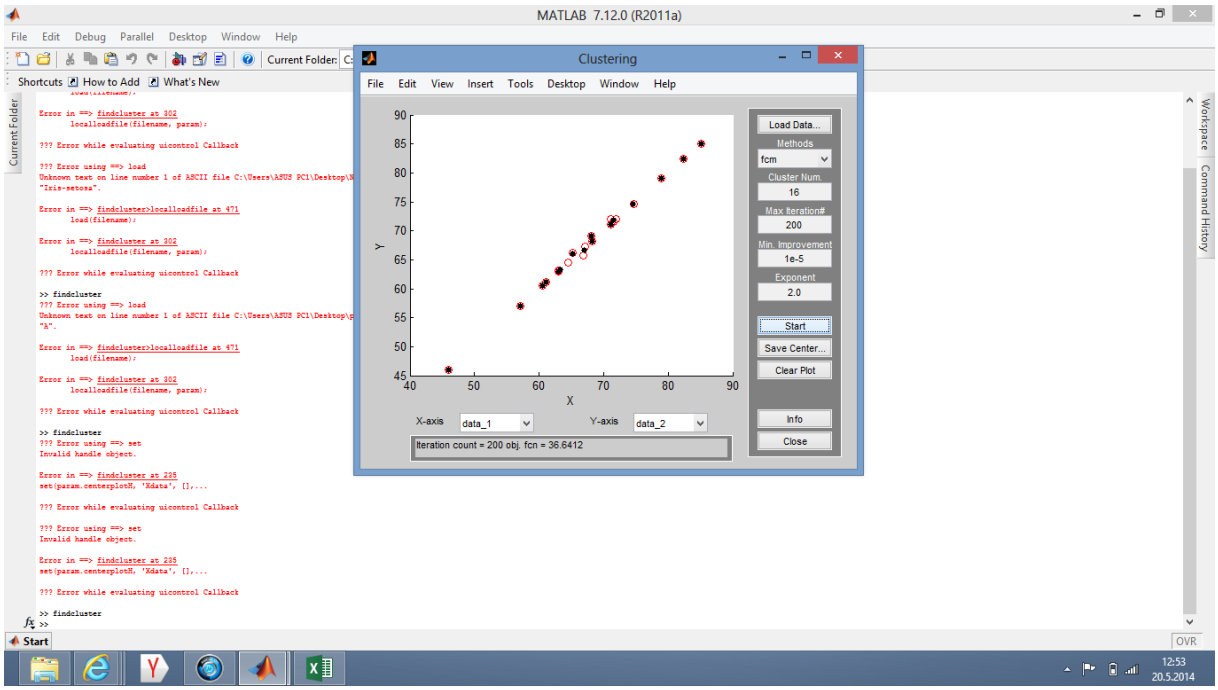
Last two pictures show the iterations and the trainning varibles of MLP applications.







In this picture FCNN algorithm clustering is represented.By the similarity between memberships FCNN clustered the data sub layers.As a result of that the train could done more iterations and get more faster and realiable results according to MLP.



7.Conclusion

In this works, a real-time odor recognition system, employing two classifiers, is described. It contains two phases for training and testing phases.

The training phase aims at localizing samples in their respective classes. It was shown that odors are identified very reliably and faster with FCNN than MLP.

These systems are designed for specific applications with a limited range of odors.

Training the ANN system, using the data we have collected during our study of the electronic nose, resulted in the following output of error.

Another advantage of the parallel processing nature of the ANN is the speed performance.

During development, ANNs are configured in a training mode. This involves a repetitive process of presenting data from known diagnoses to the training algorithm. This training mode often takes many hours using, especially, ordinary MLP.

The payback occurs in the field, where the actual odor identification is accomplished by propagating the data through the system which takes only a fraction of a second.

This proposed ANN program, named FCNN, is very useful for real-time odor record and odor recognition system, which has a various types of odor samples.

8.REFERENCES

**REAL TIME MONITORING ODOR SENSING SYSTEM USING OMX-GR SENSOR AND NEURAL NETWORK

BEKIR KARLIK AND YOUSIF AL-BASTAKI

Fuzzy Clustering Neural Networks for Real-Time Odor Recognition System
Bekir Karlık[1] and Kemal Y¨ uksek[2]