University of Pisa
Master Degree in Computer Engineering

Intelligent Systems Project: Estimation of Bike rentals in Washington D.C.

Author:
Piscione Pietro

Index

The part of this document is divided is this three part: neural network, fuzzy and forecasting.
Each part is further divided in two subpart. Each subpart (except for forecasting) has two subpart containing the study, evaluation and results referred to the two datasets: day and hour.

The table below clarifies the subdivision.

Assignment part	Subpart	Dataset	
		day	hour
Neural network	MLP neural network	1.1.1	1.1.2
	RBF neural network	1.2.1	1.2.2
Fuzzy	Mamdami fuzzy system	2.1.1	2.1.2
	ANFIS fuzzy system	2.2.1	2.2.2
Forecasting	Open loop strategy	3.1	-
	Closed loop strategy	3.2	-

Part 1

The first part of the project consists to fit the number of bikes rental in Washington D.C. based on a subset of the information listed in the text. More precisely, the main tasks of the project are the following:

1) Neural fitting development and features selection

i) Chose the best subset of provided features for estimated purposes. To this aim, use NN Fitting Tool and, for each subset taken into account , develop an MLP neural network to estimate the corresponding number of bike rentals. The optimal set will be determined based on the MSE values on the test set;

Part 1.1.1

Briefly, it consists to estimate the bike rentals dayly count based on best subset features. The performance parameter is the MSE: lower is better.

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
4	2011-01-04	1	0	1	0	2	1	1	0.2	0.212122	0.590435	0.160296	108	1454	1562
5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.22927	0.436957	0.1869	82	1518	1600
6	2011-01-06	1	0	1	0	4	1	1	0.204348	0.233209	0.518261	0.0895652	88	1518	1606
7	2011-01-07	1	0	1	0	5	1	2	0.196522	0.208839	0.498696	0.168726	148	1362	1510
8	2011-01-08	1	0	1	0	6	0	2	0.165	0.162254	0.535833	0.266804	68	891	959
9	2011-01-09	1	0	1	0	0	0	1	0.138333	0.116175	0.434167	0.36195	54	768	822
10	2011-01-10	1	0	1	0	1	1	1	0.150833	0.150888	0.482917	0.223267	41	1280	1321
11	2011-01-11	1	0	1	0	2	1	2	0.169091	0.191464	0.686364	0.122132	43	1220	1263
12	2011-01-12	1	0	1	0	3	1	1	0.172727	0.160473	0.599545	0.304627	25	1137	1162
13	2011-01-13	1	0	1	0	4	1	1	0.165	0.150883	0.470417	0.301	38	1368	1406
14	2011-01-14	1	0	1	0	5	1	1	0.16087	0.188413	0.537826	0.126548	54	1367	1421
15	2011-01-15	1	0	1	0	6	0	2	0.233333	0.248112	0.49875	0.157963	222	1026	1248
16	2011-01-16	1	0	1	0	0	0	1	0.231667	0.234217	0.48375	0.188433	251	953	1204
17	2011-01-17	1	0	1	1	1	0	2	0.175833	0.176771	0.5375	0.194017	117	883	1000
18	2011-01-18	1	0	1	0	2	1	2	0.216667	0.232333	0.861667	0.146775	9	674	683
19	2011-01-19	1	0	1	0	3	1	2	0.292174	0.298422	0.741739	0.208317	78	1572	1650
20	2011-01-20	1	0	1	0	4	1	2	0.261667	0.25505	0.538333	0.195904	83	1844	1927
21	2011-01-21	1	0	1	0	5	1	1	0.1775	0.157833	0.457083	0.353242	75	1468	1543

In order to find the best subset is useful to reduce the initial number of features. In the *daily.csv* file there are 14 columns as we can see in above image.

The preliminary study consists to remove some features considered unuseful for this part of assignment. Some of these columns (features) are correlated and redundant for determine the best subset of feature. The ignored features will be the following:

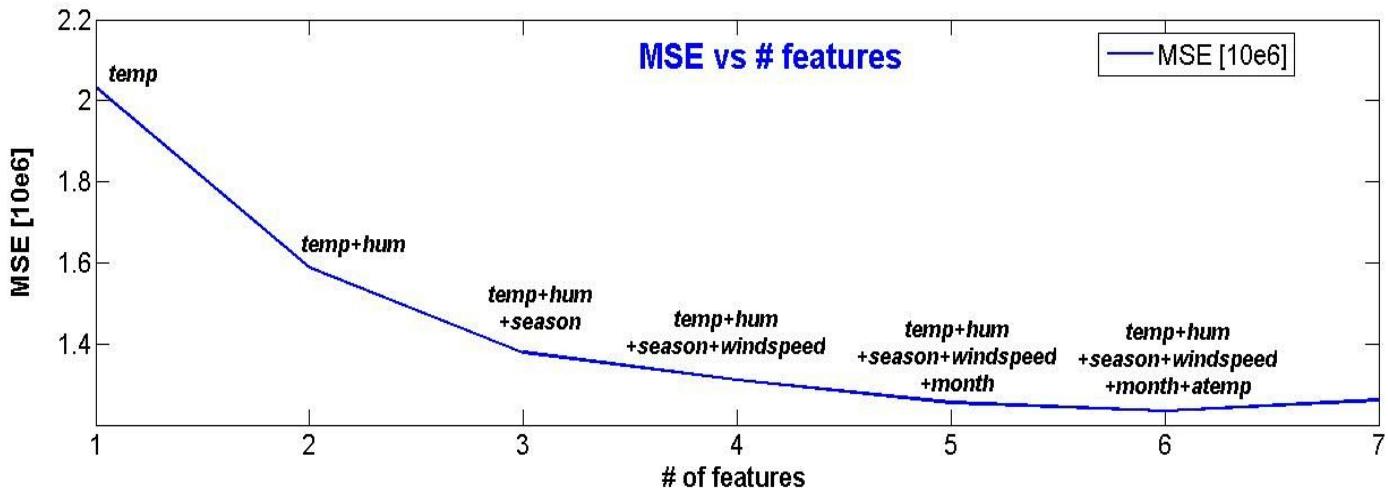
- **instant** because is a counter and it doesn't influence the number of bikes rental.
- **dteday**: for the same reason of instant feature and also because the month and year features are already present as features.
- **casual** and **registered**: they will be ignored because their sum gives cnt, thus the estimated variable.
- **cnt** because is the variable to estimate.

From 16 initial features (columns) we have 11 features now. This initial features filtering reduce the neural network training number.
In order to verify which feature influences further the estimated value cnt, we have to compute the testing MSE. The modus operandi is to build a code in Matlab by which the MSE for each specific feature is estimated.
The tool used for training the neural network is the *Neural Network Fitting Tool*. Each training is repeated five times because the testing MSE has a low variance. The MSE obtained is the average of MSEs in each training. For each training the number of neurons is set to 10, the standard value. The training set, validation set and test set are chosen 70%,15% and 15% of data set respectively.

The pseudocode is the following:

1. Load dataset into workspace.
2. Train the network with 10 hidden neurons.
3. Compute MSE.
4. Repeat step 2 and 3 five times.
5. Repeat step 2,3 and 4 for each selected feature.
6. Compute average MSE for each selected feature.

For this specific purpose is written a source code in *NN_training_single_feature.m* source code.
All results about this part are reported in *MSE part I project.xlsx* file.



The procedure has been repeated until is found the subset of features that give the lowest MSE.
This below graph summarizes the MSE decreasing step by step.
The MSE in the first steps decreases more than the last steps. The MSE reaches a stable values when the number of feature is either 5 or 6. When the number of feature is five and six, there is not substantial difference between MSE.
For computational cost reason the best subset feature has been chosen for this part are the following: **temp, hum, season, wind speed and month**. The MSE obtained is **1.255e6,** i.e. the mean error is 1120,27.
After have found the best subset of features, the network has been trained with different hidden neurons number in order to evaluate other considerations. The results are reported in the below table.

# neurons	10	20	50
Average MSE (10^6)	1.25	1.18	1.19

The different between first and second configuration is low. From the 20 neurons configuration we have a mean error 1086. We can choose one of them equivalently.
The 50 neurons configuration makes worse, i.e. the MSE is larger than 20 neurons configuration. However has been chosen the 10 neurons configuration in order to avoid overfitting risk.

Part 1.1.2 (Hour dataset)
This second part of first point of first part consists to estimate the bike rentals hourly count based on best subset features. The cardinality of sample space is the principal difference between this sample space and the previous one: in this sample space there are around 17300 samples instead of 731. The performance also in this case is the parameter testing MSE: lower is better.
Initially, in order to reduce the number of features, it is useful to discard those that are redundant. In the *hourly.csv* file there are 14 columns as we can see in the following image.

instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0	3	13	16
2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.2727	0.8	0	8	32	40
3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.2727	0.8	0	5	27	32
4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0	3	10	13
5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0	0	1	1
6	2011-01-01	1	0	1	5	0	6	0	2	0.24	0.2576	0.75	0.0896	0	1	1
7	2011-01-01	1	0	1	6	0	6	0	1	0.22	0.2727	0.8	0	2	0	2
8	2011-01-01	1	0	1	7	0	6	0	1	0.2	0.2576	0.86	0	1	2	3
9	2011-01-01	1	0	1	8	0	6	0	1	0.24	0.2879	0.75	0	1	7	8
10	2011-01-01	1	0	1	9	0	6	0	1	0.32	0.3485	0.76	0	8	6	14
11	2011-01-01	1	0	1	10	0	6	0	1	0.38	0.3939	0.76	0.2537	12	24	36
12	2011-01-01	1	0	1	11	0	6	0	1	0.36	0.3333	0.81	0.2836	26	30	56
13	2011-01-01	1	0	1	12	0	6	0	1	0.42	0.4242	0.77	0.2836	29	55	84
14	2011-01-01	1	0	1	13	0	6	0	2	0.46	0.4545	0.72	0.2985	47	47	94
15	2011-01-01	1	0	1	14	0	6	0	2	0.46	0.4545	0.72	0.2836	35	71	106
16	2011-01-01	1	0	1	15	0	6	0	2	0.44	0.4394	0.77	0.2985	40	70	110
17	2011-01-01	1	0	1	16	0	6	0	2	0.42	0.4242	0.82	0.2985	41	52	93
18	2011-01-01	1	0	1	17	0	6	0	2	0.44	0.4394	0.82	0.2836	15	52	67

Some features are equal as previous part. The new features is hr and the feature to estimate is cnt, which is the number of bikes rental hourly and not daily as in the previous part.
Also in this case, some of these features are redundant to determine the best subset of feature. The discarded features will be the following:

- **instant** because is a counter and it doesn't influence the estimated variable cnt.
- **dteday**: for the same reason of instant feature. The month and year features are already present.
- **casual** and **registered**: they will be ignored because their sum gives cnt, i.e. the variable to estimate.
- **cnt** because is the variable to estimate, i.e. the target.

The modus operandi is the same as previous point:
- 1st step: train the neural network with only one feature and obtain the MSE.

- 2nd step: once obtained the lowest MSE from a single feature, another one feature is added iteratively, train the neural network and choose the couple of feature that give the lowest MSE.

- The i-th step is to add another one feature is added iteratively from the i-th that give the lowest MSE.

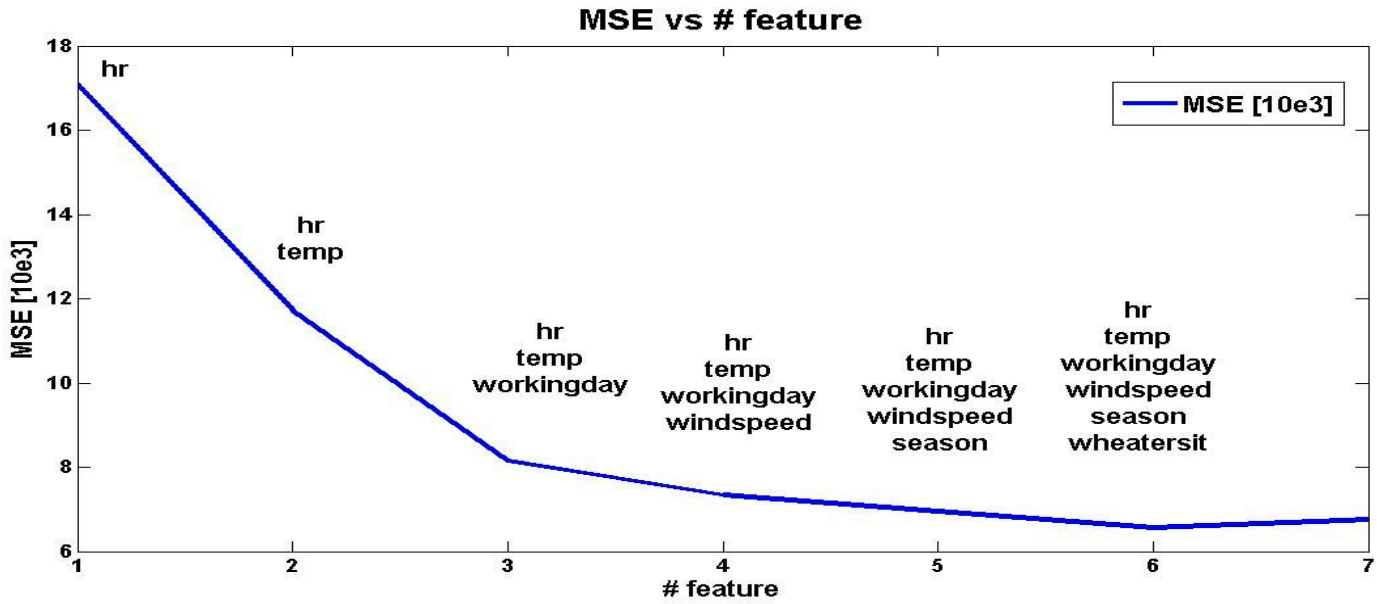
- The algorithm stops when at i-th step the MSE is larger than at i-1-th step.

The pseudo code is the following:

1. Load dataset into workspace.
2. Giving in input the feature, train the network with 10 hidden neurons.
3. Compute MSE.
4. Repeat step 2 and 3 five times.
5. Repeat step 2,3 and 4 for each selected feature.
6. Compute average MSE for each selected feature.

The code is written in *NN_training_single_feature_n_times.m* function.
The results are summarized in the below graph:

The MSE in the first steps decreases more than the last steps. The MSE reaches the lowest values when the number of feature is either five or six.
When the number of feature is five and six, there is not substantial difference between MSE.
For



computational cost reason, the best subset feature has been chosen for this part are the following: **hr, temp,working day, windspeed, season.**
The MSE obtained is **6.96e3,** thus the mean error is 83.43.

Part 1.2.1
This part of assignment consists to evaluate and compare the performance in terms of MSE with a neural network with RBF instead of MLP. The steps are the same as part 1.1, but the differences are the following:

- train the network with RBF *(Radial Basis Functions)* and not MLP *(Multilayer Perceptron)*
- repeat the training with different hidden neurons number
- repeat the training with different spread values
- The features have been normalized

In order to evaluate and compare the performance, the features chosen for the training are the best subset obtained in the previous part.
From an implementation point of view the training of RBF neural network has to know these parameters:

- spread
- goal
- input vector for training
- target vector for training
- max neurons
- how many neurons to add for each training

(For more details see newrb function documentation)

Instead, the testing has to know these parameters:
- the net obtained from the training
- the subsample test

In order to evaluate the neural network the *perform* function has to be executed.
The required parameters are:
- the network
- the subsample target test (30% of overall sample target)
- the output obtained from previous testing

The training and test subsample are chosen (pseudo)randomly, in particular the 70% and 30% of dataset,

The implementation is present in *NN_RBF_train.m* source code.

The parameters set for the neural network training for the dataset day are the following:

NN_RBF_train(chosen_features,cnt,1.2e6,0.2,0.2,1,50,5,3);

The first parameter contains a matrix in which there are the values of best subset of feature obtained in the Part 1.1.1.
- More precisely, the first parameter is the best subset of features is formed by below features:**hum ,season, temp, wind speed ,month** referred to day dataset**.**
- The second parameter is the target vector, which is cnt vector.
- The third parameter is the MSE obtained from the best subset of features. In this case is equal to **1.15e6.**
- The fourth, fifth and sixth parameters are the start spread, step spread and end spread respectively. In this specific case the initial spread is 0.2, with step 0.2 and with ending spread equal to 1. So, we have 6 values of spread: 0.2, 0.4, 0.6, 0.8, 1.

- The seventh parameter is the maximum of number neurons that neural network can have. We've chosen 50 because they were enough for our purposes.
- The eighth parameter is the neuron step. At each iteration 5 neurons will be added to neural network.

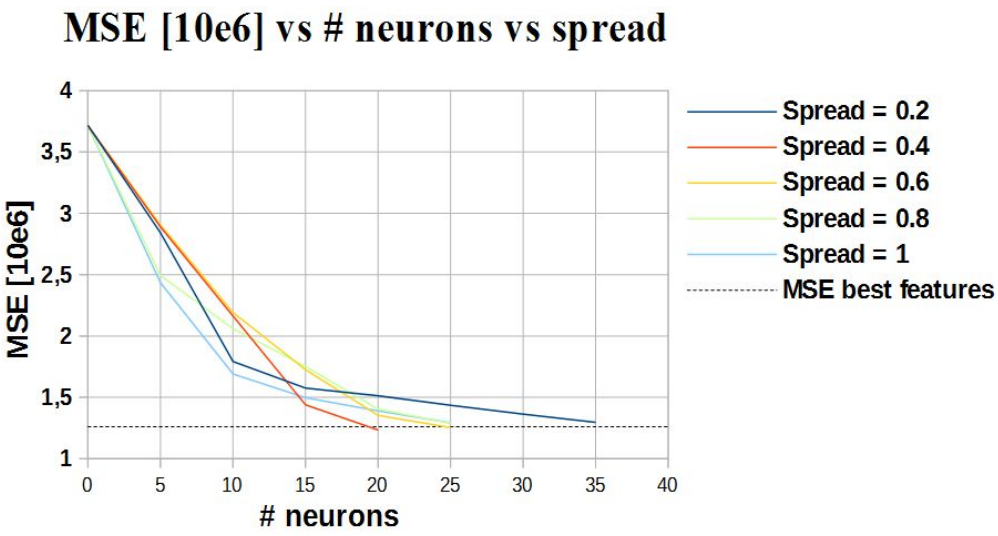
- The last parameter is the number of time by which the same configuration of neural network has to be repeated. We've chosen 3 for computational cost reason and because the test MSE has low variance.

The results obtained are stored in *NN RBF results.xlxs* file.

The MSE, generally, is better than MLP case.

In particular, for low values of spread (0.2), it is necessary to have 35 neurons to reduce the MSE below the MLP case.

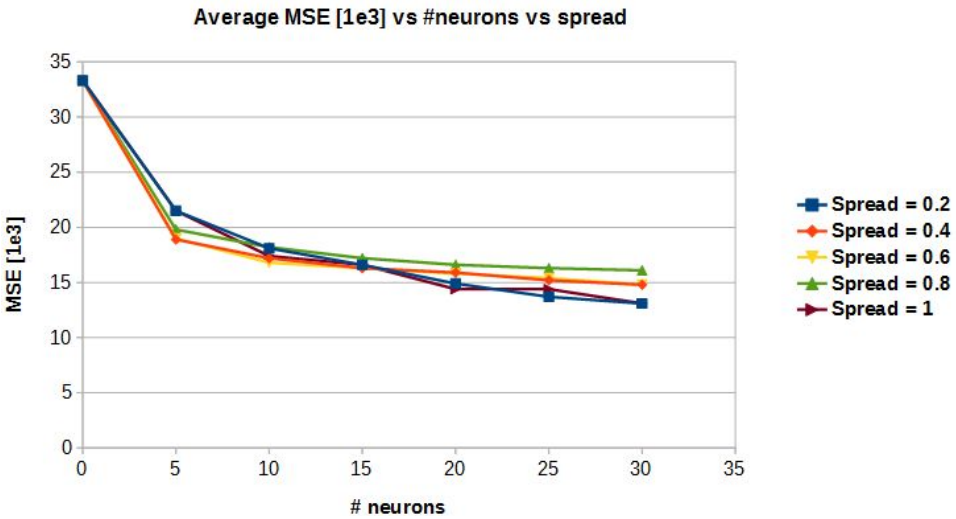
The spread value 0.4 is the better than others because all its points are under the curve spread value 0.6 and it requires only 20 neurons to obtain a MSE lower than test MSE in MLP case.



Part 1.2.2(hour dataset)

In this case, in average, the MSE of RBF neural network is slightly higher than in MLP case, but they are comparable.

The following graph summarizes the training of neural network with different values of spread and different values of number of neurons. A low spread (0.2) has, for the various configuration of MSE, in average the lower MSEs than in other cases. The inverse of spread is the selectivity: this means that with high selectivity the neural network works, in average, better than low selectivity.



The following table summarizes the MSE in the previous part. MLP neural network and RBF neural network are comparable because their average training MSE are close. The magnitude of MSE are the same in both cases.

dataset	MSE with MLP	MSE with RBF
day	1.25e6	1.17e6
hour	6.96e3	13.1e3

The fact to obtain different result between MLP and RBF is due to activation functions type. In the MLP is with a nonlinear activation function, while in RBF the tuned parameters was the spread value.We can't say which is better in general, but we can say that in the day dataset the results lead us to say that RBF neural network is better than MLP neural network. In the other hand we can say the opposite for the hour dataset.

Part 2 (Fuzzy)
The second part of assignment is divided in two parts.

The first one consists to develop a *Mamdami fuzzy system* to fit the number of bike rental on the basis of the optimal set of features previous selected.

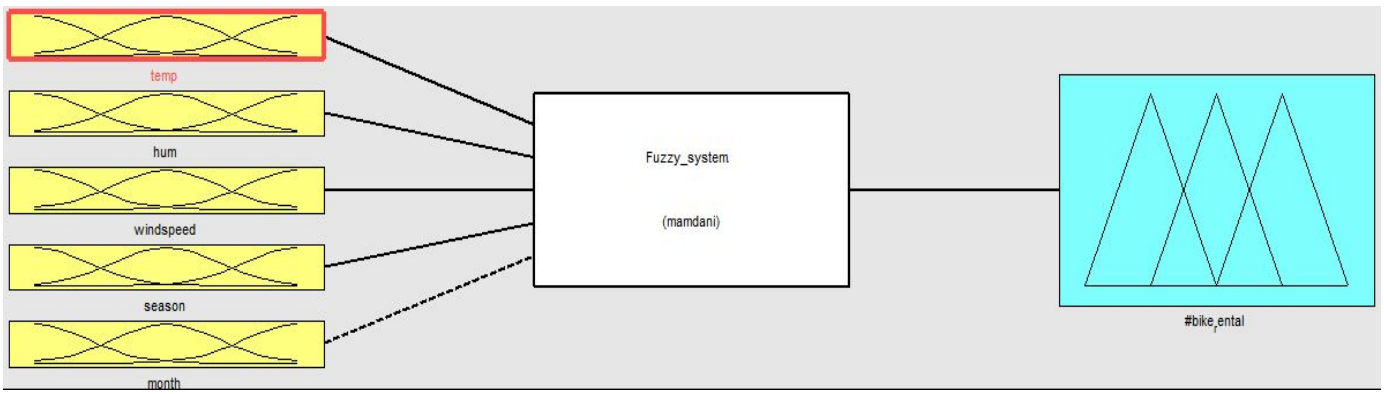
The second one consists to develop an *ANFIS fuzzy system* to fit the number of bike rental on the basis of the optimal set of features previous selected.

In both cases the number of selected features is maximum five for computational cost reasons.

For the two datasets there are chosen the two best subsets of features.
For the dataset daily the best subset is composed by **hum ,season, temp, wind speed ,month** features**.**
For the dataset hour the best subset is composed by **hr, temp, workingday, windspeed ,season** features**.**

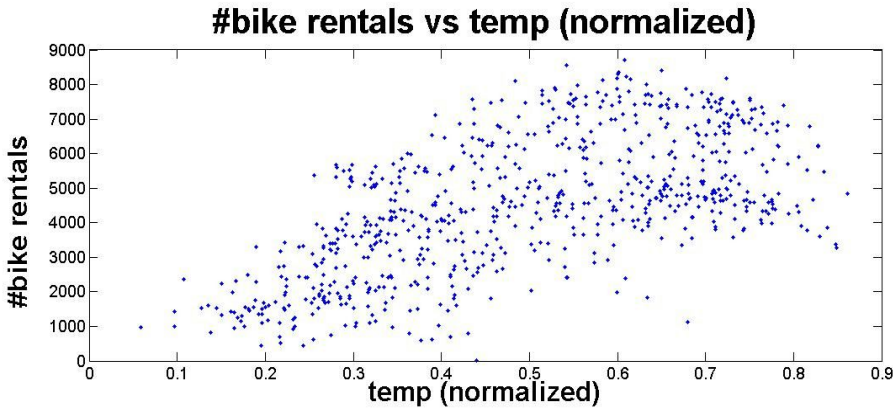
In this way, this part of assignment is further divided in two part. The first one is referred to day set and the second one is referred to hour data set.

Part 2.1.1 (Day dataset)
In order to develop a Mamdami fuzzy system is used the *Fuzzy Logic Design Tool* of Matlab, by which is possible to define the rules and the membership functions.
The membership functions are defined based on scatter plot features and the target. The Mamdami fuzzy system is shown on the right.

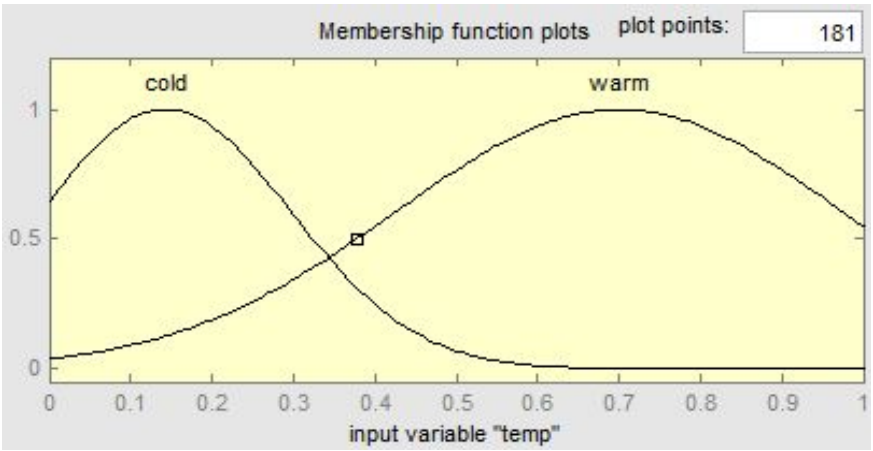


Temp membership functions

It is possible to note bigger is temp (normalized) bigger is, in average, the number of bikes rental. This is true until temp normalized is is equal to 0.6-0.7.
After these values the number of bikes rental decreases slightly.



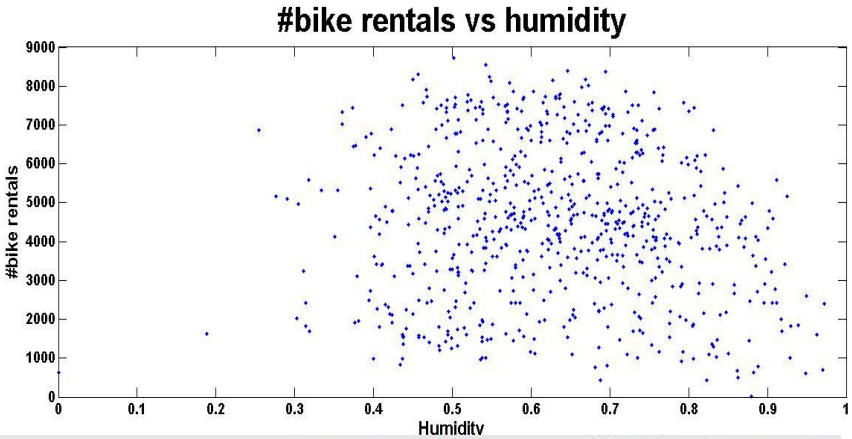
By this it is possible to build the membership functions for this feature.
The membership functions for temp are two gaussian shaped functions because the number of bikes rental increases\decreases softly. A certain number of bike rental can belong to both membership functions with a certain degree depending from membership functions values.

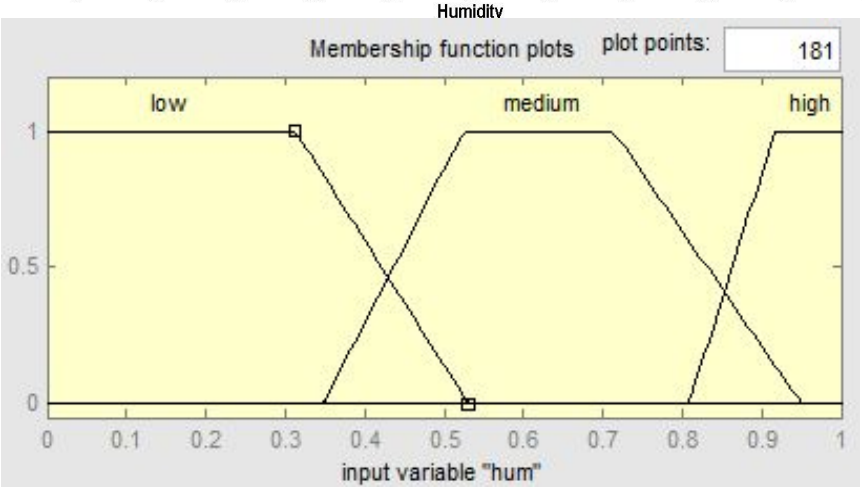


Hum membership functions

For the humidity feature there are three trapezoid shaped membership function.

The reason is the following: as we can see we can identify three regions. The first one in which there is a low number of samples, the second one in which the number increases and then the third one in which the samples number decreases. In the first and the third region it is possible to say that the number of bikes rental depends from humidity. In the second one the scatterplot is cloudy, so we can't say anything.

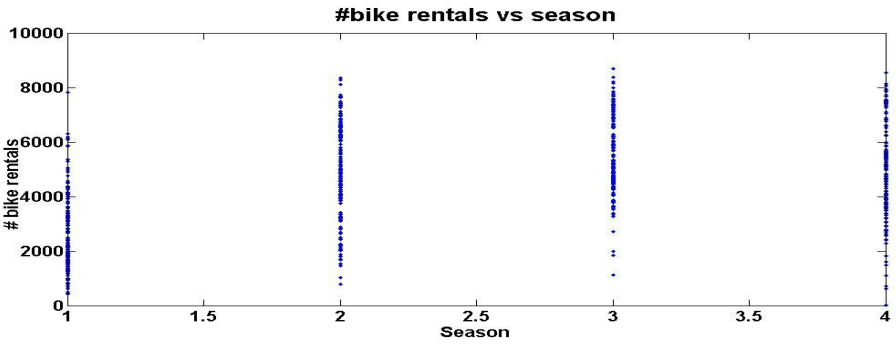


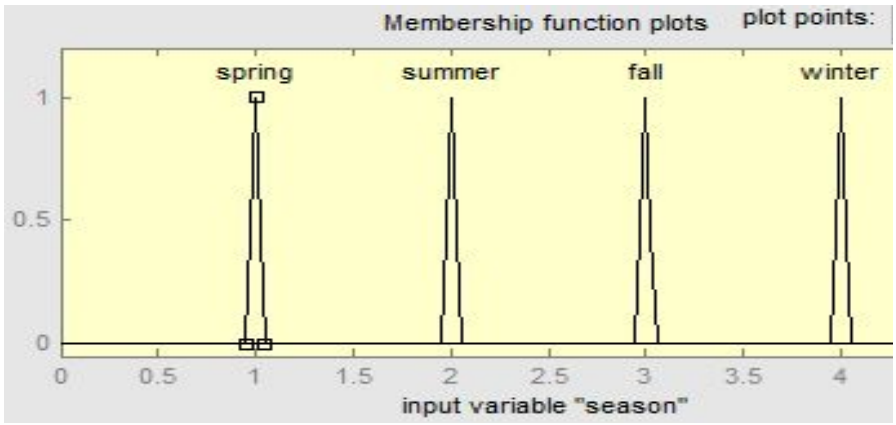


Season membership functions

The season input has four triangular shaped functions as many membership functions. The center of each of them is to the corresponding season value. It has been used triangular shaped function because the singleton function in Matlab Toolbox is not present. The number of bikes rental is “low” when the season is equal to 1 (spring) and increase until season reaches the values 3 (fall). Then it slightly decreases.

(Low is between inverted commas because means low with a certain degree specified from membership function).



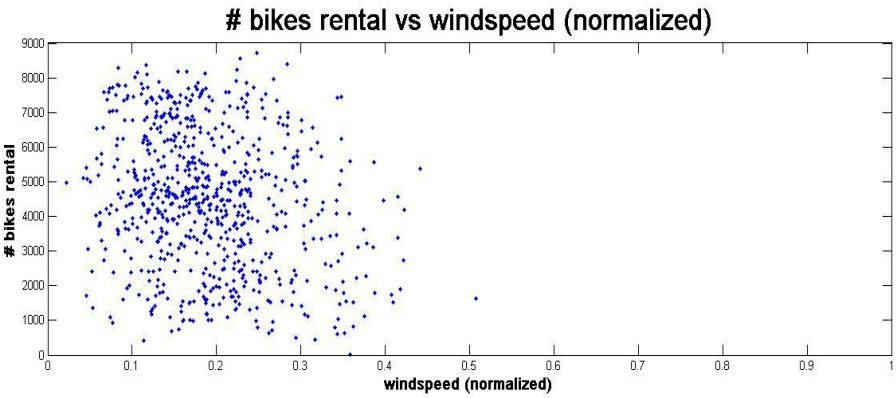


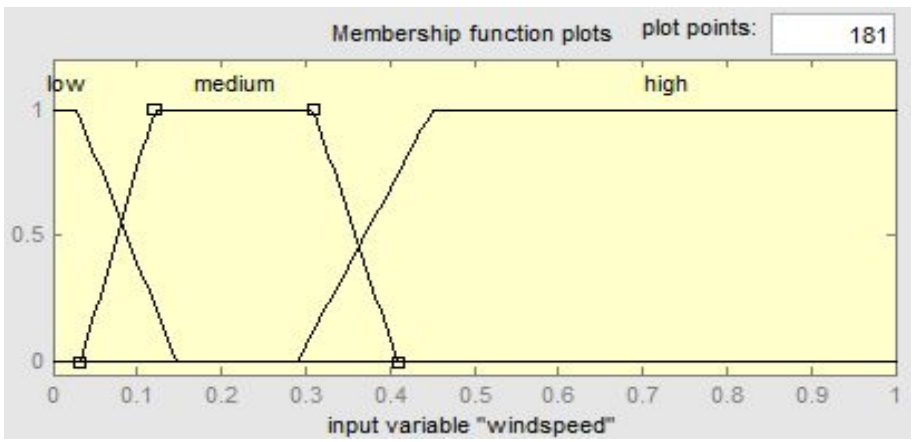
Wind speed membership functions

Also in this case we can see three different regions. The first one from 0 to around 0.05 in which the samples number is low and the numbers of bikes rental is “medium” and “low”. After this values there is a second region which the interval is around 0.05 and around 0.25. The numbers of bikes rental is a lot more variable but there is a bigger density in the middle zone. From the wind speed value 0.25 the values decrease.

The membership functions are three trapezoid shaped functions. We say that in the second region the number bikes rental is “independent” from wind speed. Instead, in the third region bigger is the wind speed lower is the number of bikes rental.

(Low and medium are between inverted commas because means

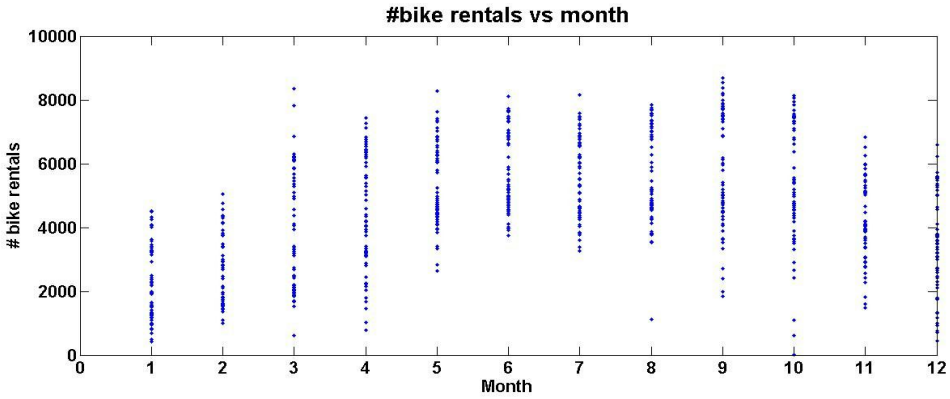


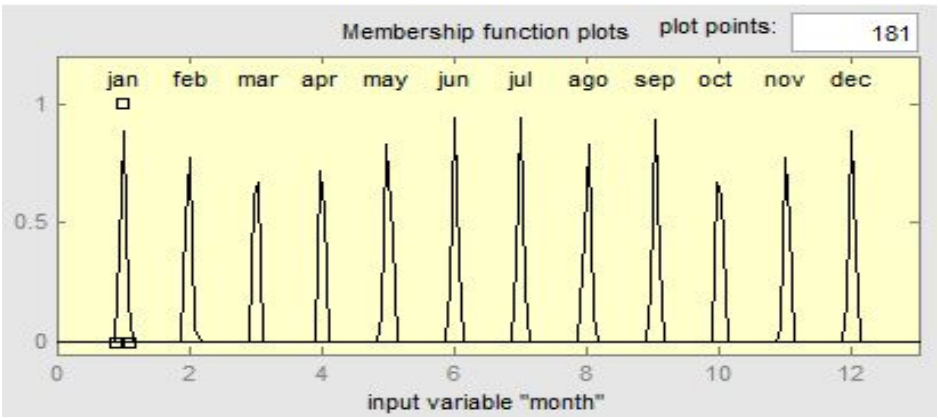


low with a certain degree specified from membership function).

Month membership functions

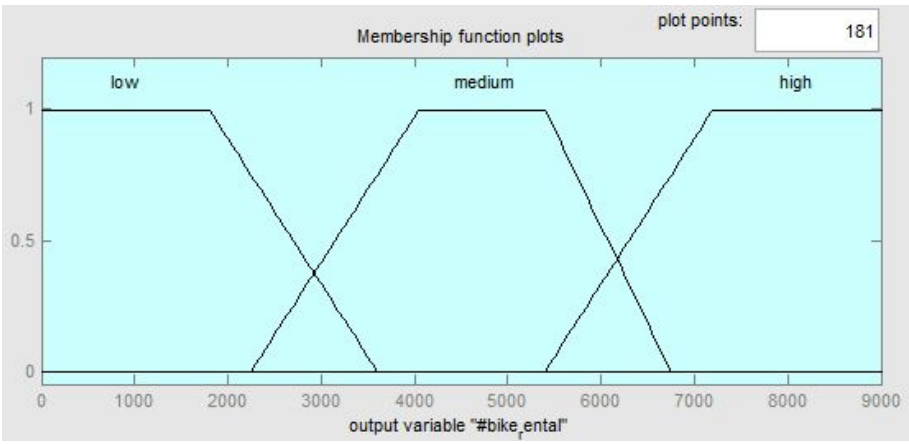
The membership functions are twelve (as many the months) triangular shaped functions. What we can notice is that the (average) number of bikes rental increases in the summer months, while in the winter months decreases.

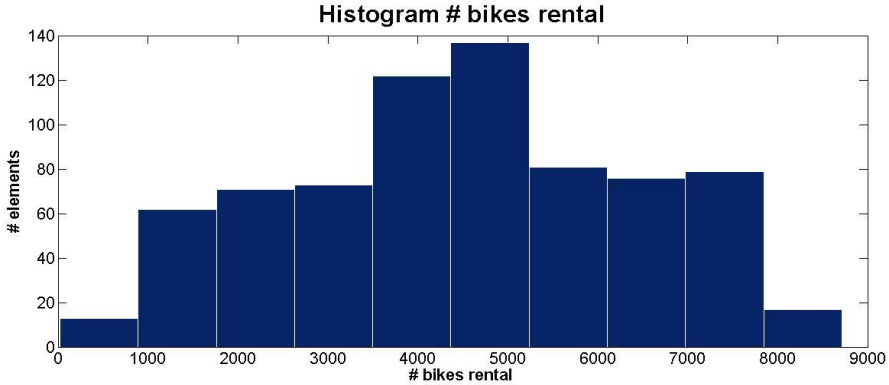




Output membership functions

An histogram is plotted in order to build the output membership functions. The histogram gave us the information about the numbers of samples. In this way we can identify three different region in which the number of bikes rental could be “low” and\or ”medium” and\or “high” with a certain degree depending from membership functions.





Mamdami rules
In order to build the Mamdami rules we have to focus on scatterplots and membership functions both.
Generally, the number of rental bikes depends a lot from weather situation, i.e from climatic conditions.
All information about mamdami rules, membership functions are stored in *Fuzzy_system.fis* file.

First rule
By the samples we can say that the people like to rent bikes when the weather situation is “good”. We can think about people that like either to enjoy a riding on bike or go to work in a “good” day. “Good” means that:

- temp is warm
- humidity is medium
- wind speed is medium
- season is fall

By these conditions we can build the first rule.
If temp is warm and humidity is medium and wind speed is medium and season is fall
the number of bikes rental is **high**.

Second rule
On the other hand, people don’t like rent bicycle when the weather situation is “not good”. We can think about windy and cold day in which the idea of rental bike is “not good”. “Not good” means that:

- temp is cold
- humidity is high
- wind speed is medium
- season is spring

If temp is cold and humidity is high and wind speed is medium and season is spring
the number of bikes rental is **low**.

Third rule
For example, when is **winter**, the weather is “**warm”**, the humidity is **medium, the** wind speed is **medium** then the number of rental bikes is **medium** because there are not the ideal condition to not have an high neither a low number of bikes rental.

If temp is warm and humidity is medium and season is winter
the number of bikes rental is **medium**.

Fourth rule
An influential parameter is the windspeed. We suppose to have a windy, and cold day. We suppose also to have a typical humidity and the season is winter. Only few people are going to rent a bike for previous reasons. By these consideration we can build the rule.

If temp is cold and wind speed is high and humidity is medium and season is winrter
the number of bikes rental is **low**.

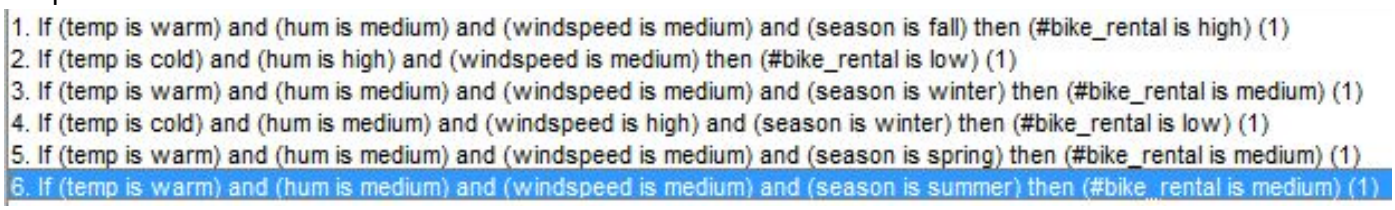
Fifth rule
Another example: in a **warm spring day** in which the humidity and wind speed are **medium** we can find a medium number of bikes rental.

If temp is warm and humidity is medium and wind speed is medium and season is spring

the number of bikes rental is **medium**.

Sixth rule
In a summer day where the temperature is warm, wind speed and humidity are medium when can find "medium" number of bikes rental.
If temp is warm and wind speed is medium and humidity is medium then
the number of bikes rental is **high**.

This picture summarizes all six rules above.



The rules chosen are six: low number of rule not necessary means low performance. The rules chosen have a meaningful due to relation between each features and number of bikes rental.

The fuzzy system has the rules and membership functions. In order to evaluate the fuzzy system
the following step are performed:

1. Compute the output Y of fuzzy system
2. Compute the error E=Y-T (T is the target vector)
3. Compute the MSE

These steps are performed in *compute_MSE_fuzzy.m* function, which gets the MSE of Mamdami fuzzy system.
In this specific case the MSE obtained is **4.072e06,** i.e the mean error is 2017.92.
This result is worse than RBF study which the MSE is 1.255e06.

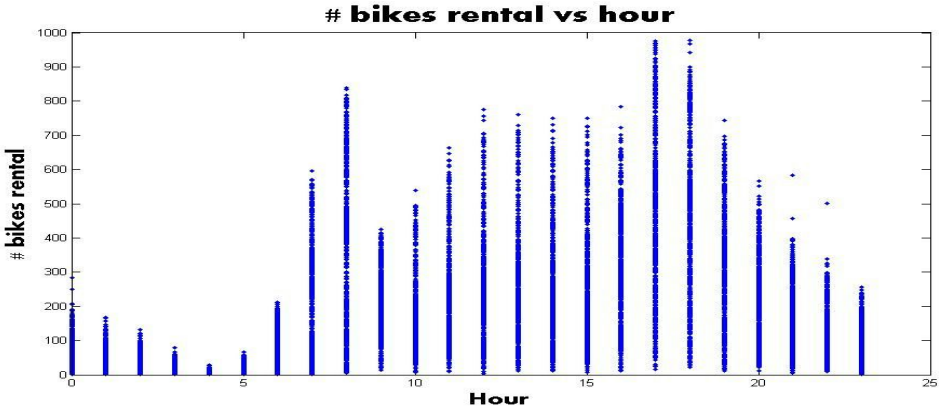
Part 2.1.2 (Hour dataset)

In order to develop a Mamdami fuzzy system is used the *Fuzzy Logic Design Tool* of Matlab, by which is possible to define membership functions and Mamdami rules. The approach used in this part is similar from the approach used in the part 2.1.1. The only different is the dataset and the best subset of feature.

The best subset of features is the following: hr, temp, working day, wind speed, season.
The dataset used is hour.

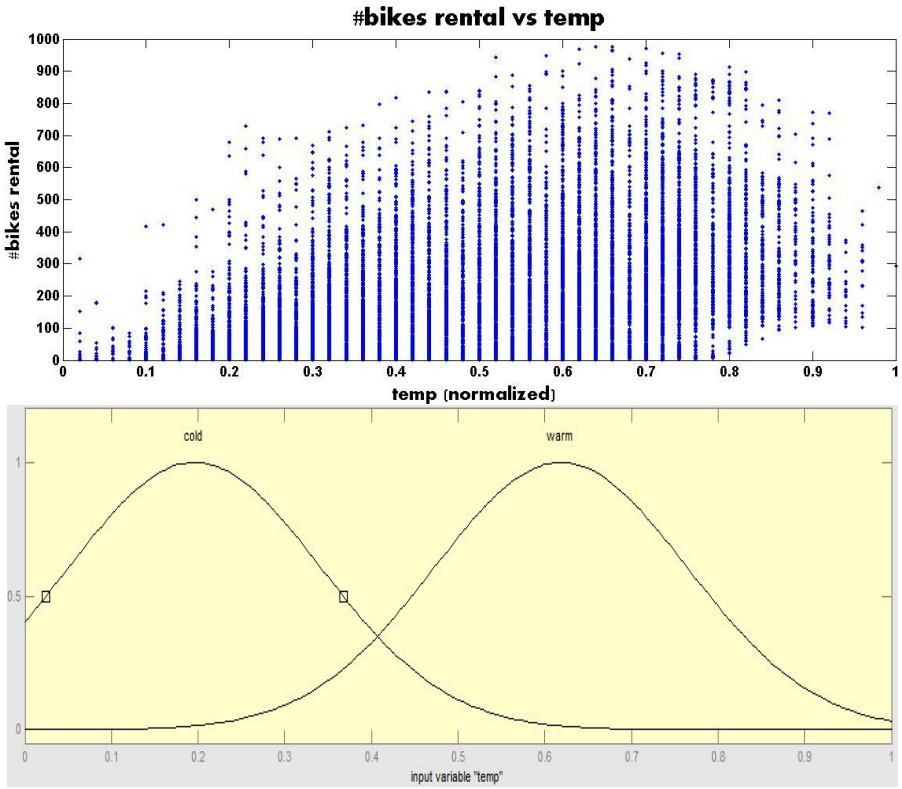
Hr feature

The hour feature is the most influential in this dataset as shown in MLP part. As we can see in the night hours the number of bikes rental is “low”, while it increases in the morning hours and then decrease in the evening hour. By this is possible to build the membership functions.



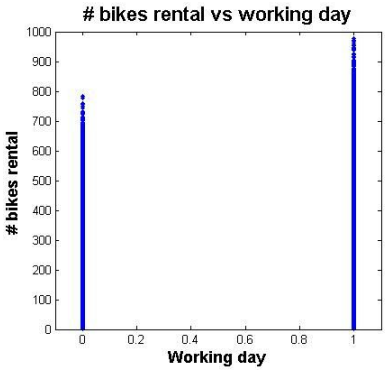
Temp feature

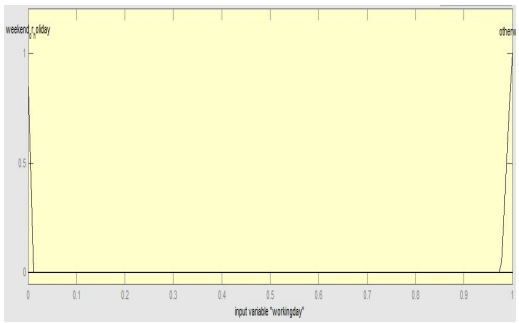
As in the previous part, the number of bikes rental is influenced by the temp. For values around 0.6 and 0.7 there is a “peak” of bikes rental. In the other ranges the number of bikes rental decreases slightly.



Working day feature

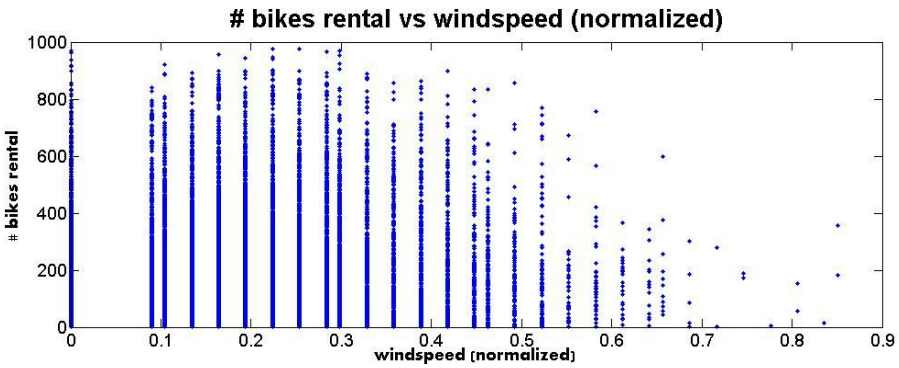
It is composed by only two values: 1 and 0. It is easy to determine which are the membership functions. As we can see when working day has value 1 the number of bikes rental increases.

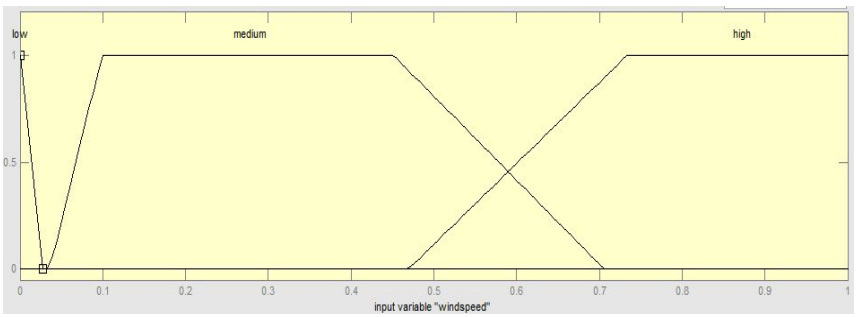




Wind speed feature

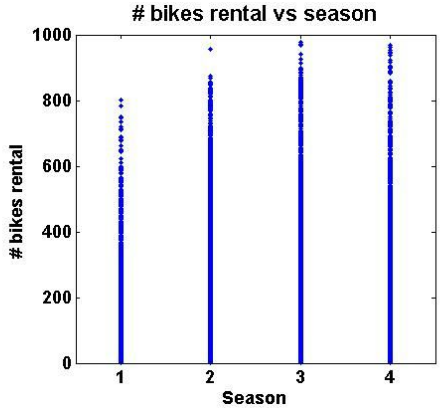
The wind speed is between 0 and 1. Zero value means no wind. In other cases the wind speed values is normalized. In order to build the membership functions, the strategy has been followed is the same in the previous part.

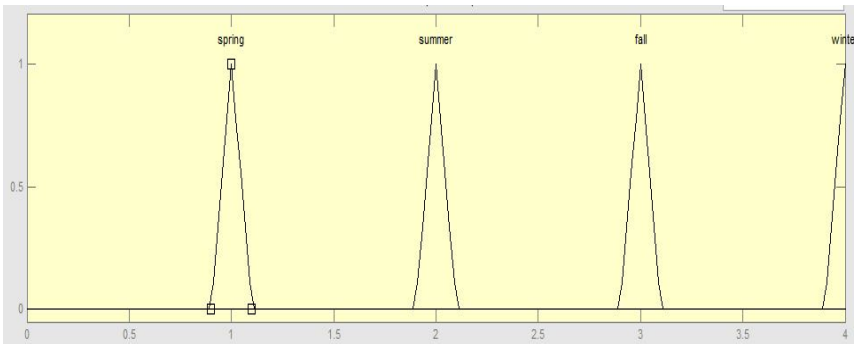




Season feature

Also in this case is easy to determine the membership functions: they are 4 triangular membership functions centred on season values.





Mamdami rules
The modus operandi to define the Mamdami rules is similar as previous part. As we can see, people “don’t like” to rent bikes during night hours. This is due because is too “early” to rent a bike. Then, around the 8 a.m. is possible to notice that the number of bikes rental increase. This fact is due, for example, to people go to work.
The same line of thinking can be adopted for temp, hum and windspeed features: as previous part we can define a “good” day when the climatic condition respect certain conditions mentioned in previous part.
In opposite we can also (re)define “not good” day when “opposite” climatic conditions occur.

Of course there are possible tradeoff between “good” and “not good”. In these conditions the number of bikes rental is not “high” neither “low”, i.e. is “medium”. This happens when the features take values between “good” climatic conditions and “not good” climatic condition.

The inverted comma are used because we are talking in fuzzy terms, i.e. “high”,”low” and “medium” are not defined as a single value but a membership functions which shape is shown above.

The rules are listed below.

1. If (temp is cold) and (windspeed is high) then (bikes_rental is low) (1)
2. If (temp is warm) and (workingday is weekend_or_holiday) and (windspeed is medium) then (bikes_rental is not high) (1)
3. If (hr is 21-23) and (temp is cold) and (workingday is weekend_or_holiday) and (windspeed is high) then (bikes_rental is low) (1)
4. If (hr is 17-20) and (temp is warm) and (windspeed is medium) then (bikes_rental is not high) (1)
5. If (hr is 0-6) then (bikes_rental is low) (1)

After evaluate the Mamdami Fuzzy system, we have been obtained an MSE of **6.65e4**, i.e. a mean error of around 250.

Part 2.2
An *Adaptive Neuro-Fuzzy Inference System* makes it possible to build a fuzzy inference system where the membership function parameters are automatically tuned using an adaptive learning method. Since we have two datasets (day and hour) the same modus operandi is applied to the two datasets.

To accomplish the task of building such a system, the following choices have been made:

- The algorithm ends after 30 epochs because the average training error reaches the steady state
- Grid partitioning because is possibile to choose the type of membership functions
- Optim method: hybrid
- Error tolerance:0
- MF type: constant

These choices have been chosen for day and hour dataset both.

2.2.1 (Day dataset)
For the day dataset are reported the five best features its number of membership functions.

feature	**temp**	**hum**	**windspeed**	**season**	**month**
# membership functions	**3**	**3**	**3**	**4 for the number of seasons**	**4 for computational cost reasons**

For each type of membership function is reported average train error, average testing error and checking error.

Membership functions type	**Average train error (MSE)**	**Average testing error**	**Average checking error**
tri	836.36 (0.7e6)	2221,04	2372,43
trap	1019.78(1.04e6)	2393.35	2546.91
gbell	819.65 (0.67e6)	2699.46	2956.86
gauss	**813.54 (0.66e6)**	**2207.62**	**2040.81**
gauss2	834.44 (0.696e6)	4472.90	3012.81
pinf	1077.21 (1.13e6)	2751	2783.75
dsig	867.3(0.75e6)	5305	3557.63
psig	875.93	5196.58	3422.81

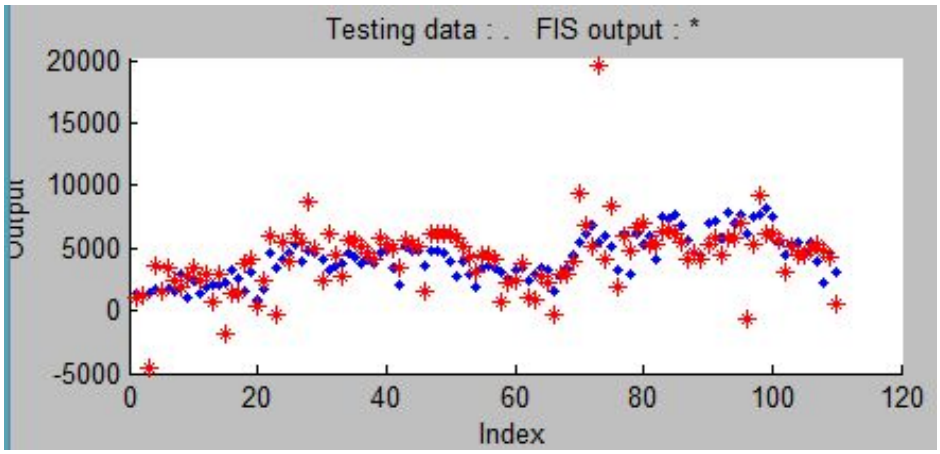
We can see the **gaussian membership functions** is the best membership functions in terms of MSE in the average error training, average testing error and average checking error.

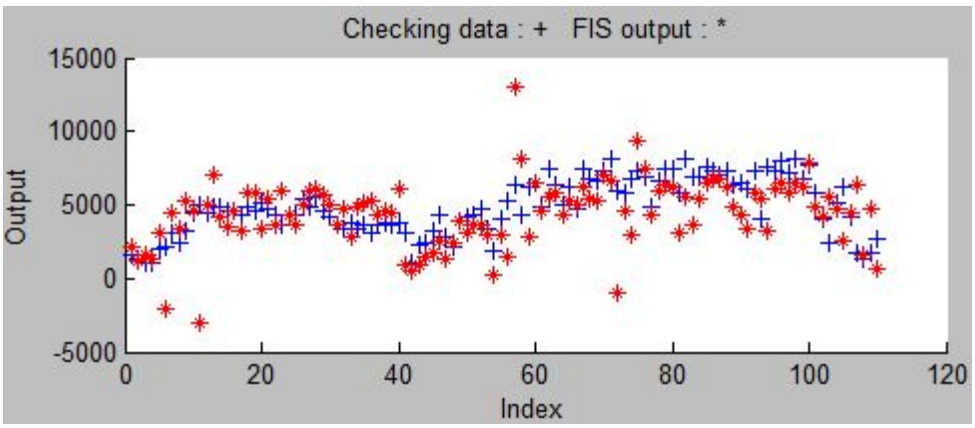
After a while the ANFIS training has terminated its work and these are the info about the ANFIS system:

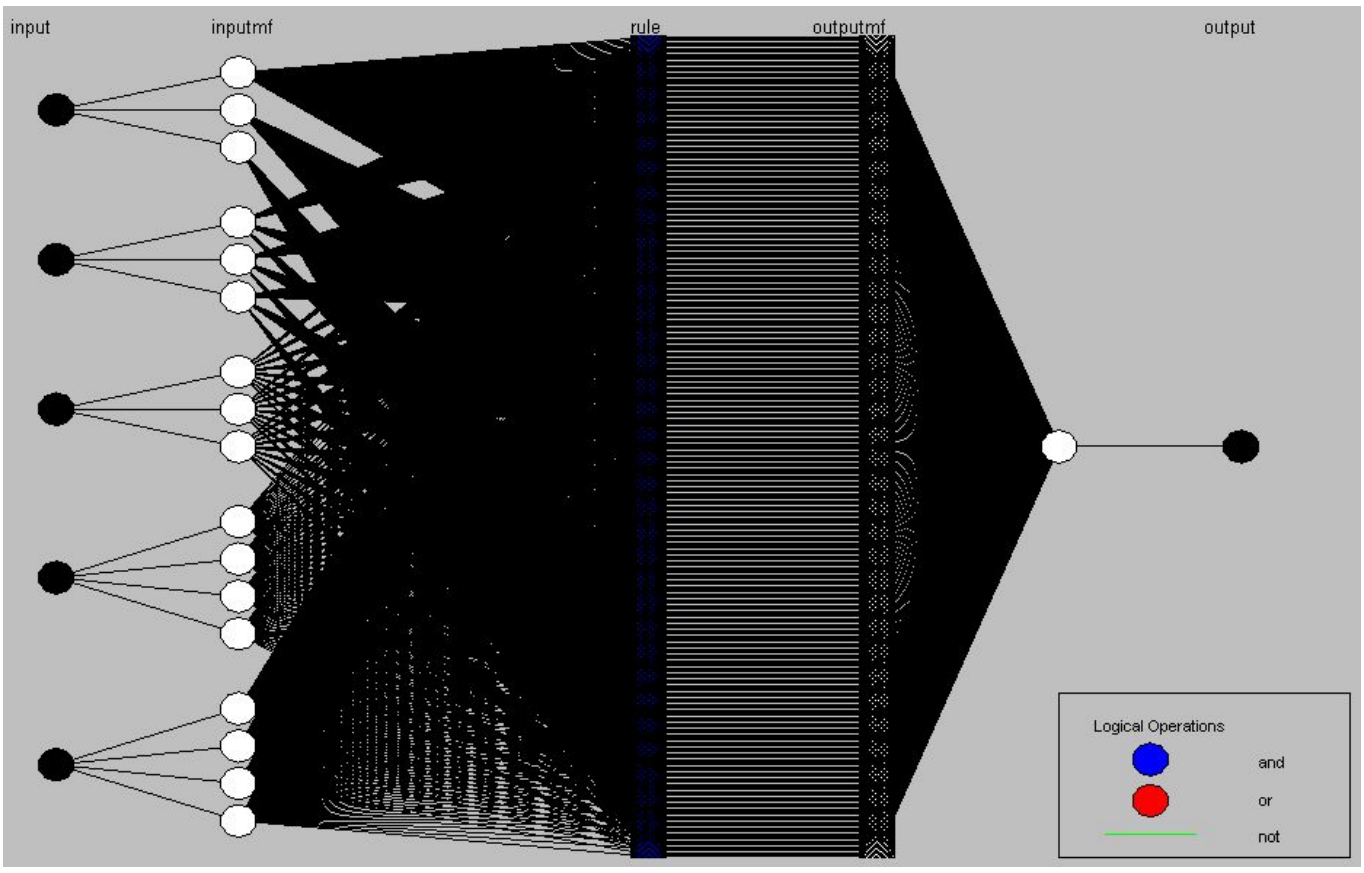
- **Number of nodes: 906**
- Number of linear parameters: 432
- Number of nonlinear parameters: 34
- Total number of parameters: 466
- Number of training data pairs: 511
- Number of checking data pairs: 110
- **Number of fuzzy rules: 432**

What we focus on is the high number of nodes and the higher number of rules than Mamdami Fuzzy system. Certainly with the ANFIS we have been obtained better performance than Mamdami Fuzzy system because in this case the membership functions parameters are tuned in way that minimize the testing error.

Below are reported the structure of neural network, the training, test and validation plot generated from ANFIS.







2.2.2 (Hour dataset)

The same strategy has been adopted for hour dataset.

The number of membership functions are chosen meaningfully and for computational cost reasons.

feature	**hr**	**temp**	**working day**	**windspeed**	**season**
# membership functions	4 for computational cost reason	3	2	3	4

The ANFIS has been trained for different type of membership functions.

Membership functions type	**Average train error (MSE)**	**Average testing error**	**Average checking error**
tri	120.81	127.7	130
trap	125.67(15.8ee)	130.81	143.61
gbell	121.89(14.8e3)	125.68	129.02
gauss	**120.85(14.6e3)**	**125.64**	**128.59**
gauss2	125.86(15.8e3)	128.32	128.8
pinf	126.17	131.44	174.46
dsig	126.03	130.38	129.12
psig	126.34	131.3	130.2

As in the day dataset, the gaussian is the best membership functions type because has the lowest MSE in the training, testing and checking (validation). All information about the network are stored in *ANFIS_hr .fis.*

The information about the rules are the following:

- **Number of nodes: 616**
- Number of linear parameters: 288
- Number of nonlinear parameters: 32
- Total number of parameters: 320
- Number of training data pairs: 12165
- Number of checking data pairs: 2607
- **Number of fuzzy rules: 288**

In this case the MSE is **14.6e3,** i.e. a mean error of around 120.

Sum up Part 2

The below table summarizes the results of part 2.

Dataset	**Mamdami Fuzzy MSE**	**ANFIS MSE**
day	4.07e6	0.66e6
hour	6.64e4	14.6e3

The ANFIS MSE is lower (and better) than Mamdami Fuzzy MSE.

The explanation could be the following: in the Mamdami Fuzzy the parameter and the membership functions are chosen meaningful and heuristically. When a “good” MSE is obtained no more rules adding and tuning membership functions parameters has been performed.

In the ANFIS part the network has been trained until the **best** combination of membership functions parameters and rules has been found. Moreover in this case the rules number are larger.

Part 3 Time series

Differently to the two previous parts, this part of assignment is performed only to day dataset. It is used the *NN Time Series Tool* of Matlab by which is possible to obtain the corresponding performance in terms of MSE. For open and closed loop strategies is used the 70% of set for the training, 15% for the validation and 15% for the test.

Part 3.1 Open loop strategy

This part of assignment consists to find heuristically the best combination of hidden neurons number and delays number in order to obtain the lowest MSE.

The data set used in this part is the day dataset. In particular are taken the data of 2011 to develop a forecasting model able to predict the number of bikes rental.

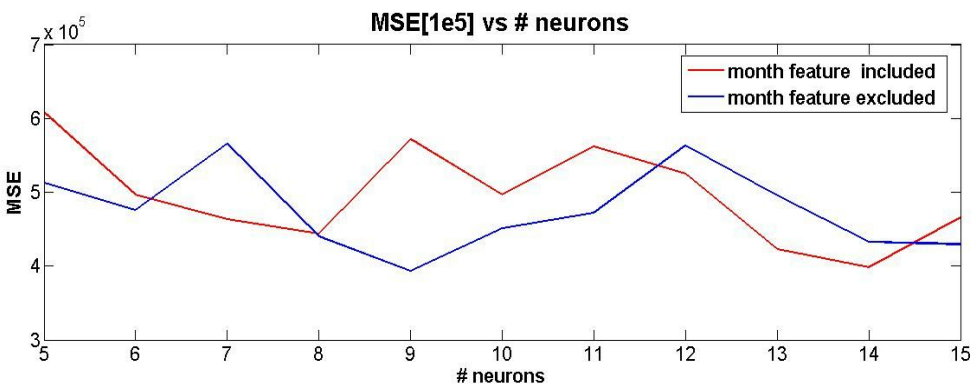
In order to determine which are the best combination of number of hidden neurons and number of delays, we have made some assumptions.

The features in input to the neural network are the best obtained in the Part 1 except for the month features. This feature has been excluded because makes worse the neural network performance, i.e. the average MSE is slightly higher.

To prove this assertion the open loop neural network has been trained through the same configuration with all features and all features except month feature both. The configuration is the following:

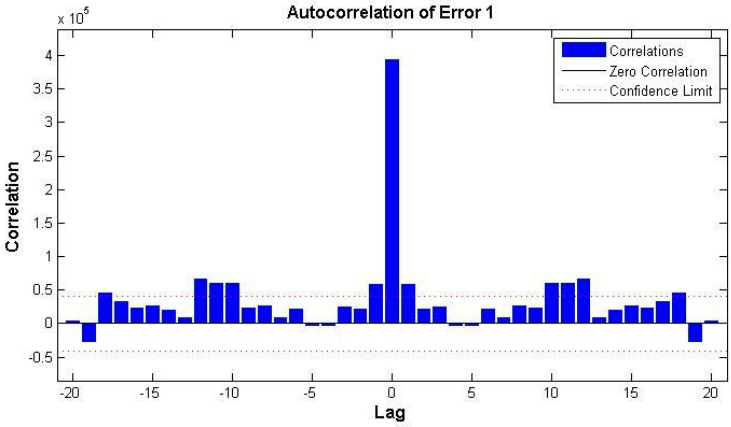
- number of hidden neurons: from 5 to 15 by step of 1
- number of delays: 2

The following graph demonstrate the previous assertion.



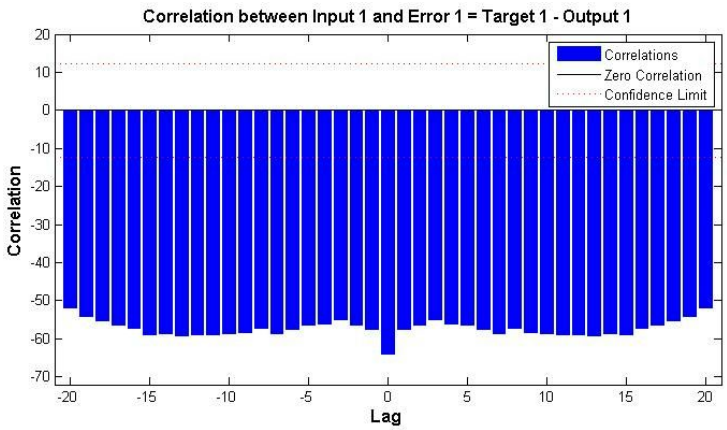
The code that executes the open loop neural network training is written in *Forecasting_training.m* source code.

By that, it is possible, heuristically, train the network with different values of number of delays and number of neurons. For each number of delays there are some combinations of hidden neurons number. Beyond the MSE we have also to consider the error autocorrelation and also input error correlation. We can have the lowest MSE but the values come from it has to be uncorrelated or less correlated possible.



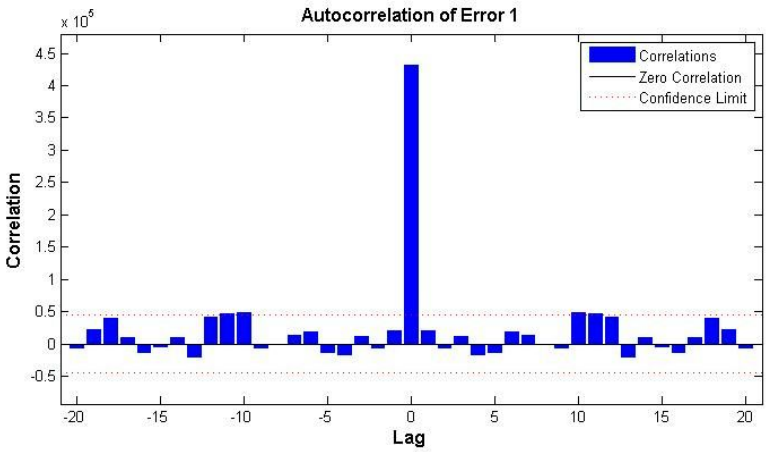
With combination 3 delays and 10 neurons the error autocorrelation and input error correlation are over the confidence limits. The following graph shows them.

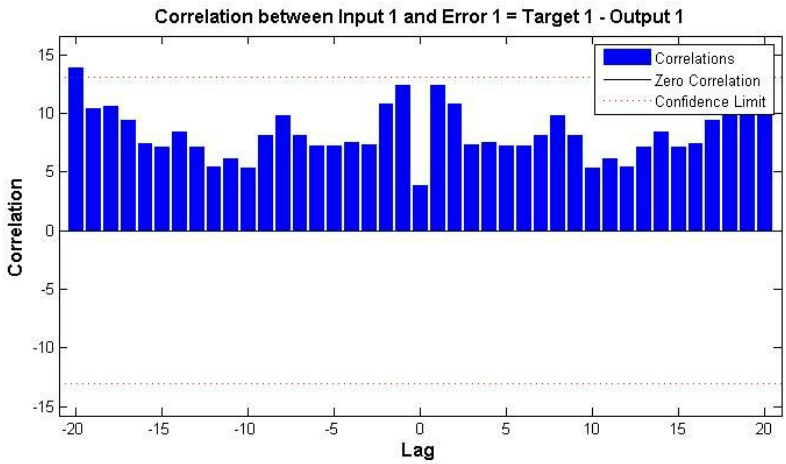
The vertical bars exceed the confidence limits in both cases.



The number of delays is fixed to two and the number of neuron has been changed.
We have been stopped to 11 neurons because the combination 2 number of delays and 11 neurons have been again exceed the confidence limits.

The best performance in terms of MSE, error autocorrelation and input-error correlation is given by the following combination: 2 delays and 10 neurons.
As shown in the following figures, in this particular cases we haven't the blue bars that exceed the confidence limits. the choice to exclude the feature month from the input to open loop was revealed correct, because for the previous configuration the MSE is lower than MSE of feature month included.





The MSE obtained with above combination is computed the MSE, which is **4.93e5,** i.e. a mean error of 701. The net obtained from the open loop training neural network has been saved. Its name is *net_openloop.dat.*

Part 3.2 Closed loop strategy

A neural network can also be simulated only in closed-loop form, so that given an external input series and initial conditions, the neural network performs as many predictions as the input series has time steps.

The previous open loop neural network has been saved in order to feedback itself through a loop. More precisely, closed loop strategy consists to evaluate the forecasting performance based on 2012 year datas. In particular, it consists to estimate the average forecasting performance for 1,2,7 and 10 days ahead.

The closed loop neural network accepts in input x(t) vector and y(t-1),y(t-2),...,y(t-d) inputs.

In order to estimate the forecasting performance some days are taken randomly from 2012 set in day dataset.

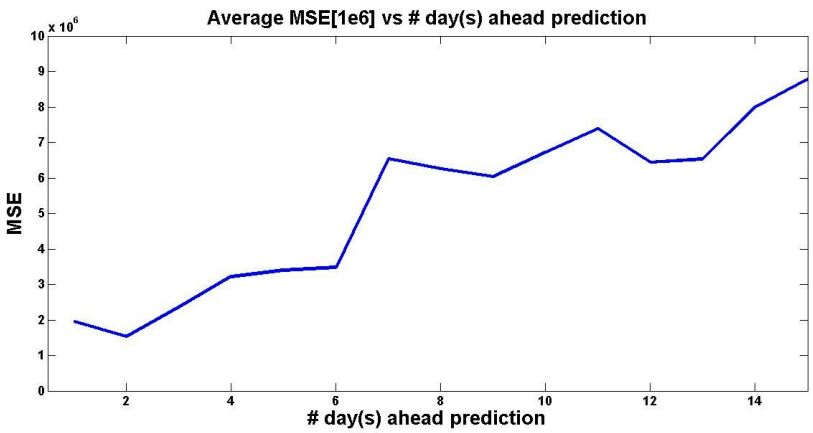
Day(s) before (Delays)			**Day(s) ahead (Prediction)**					
2	1	Today	1	2	...	7	....	10

In order to perform this operation some considerations have to be made. For instance, if we consider the testing MSE of prediction of two day ahead will be, in average, either equal or greater than of testing MSE one day ahead prediction. This is due because the prediction of two days ahead is based on prediction of one day ahead.

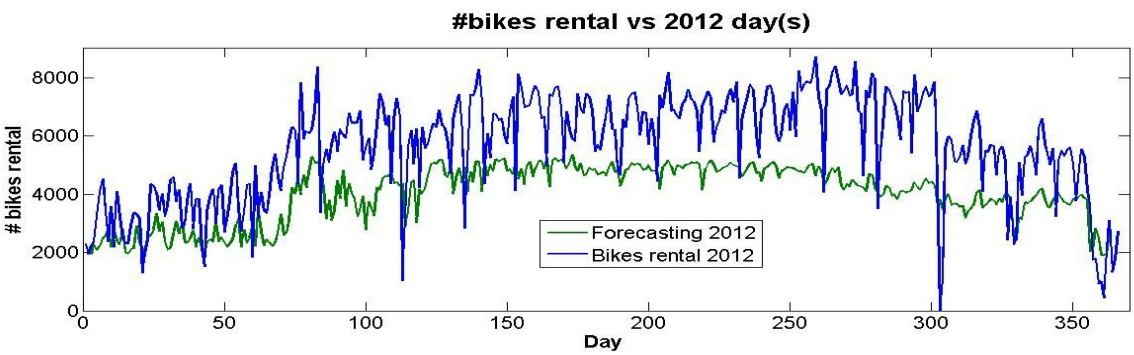
This modus operandi is repeated until we get the prediction from more distant day, i.e. the tenth day ahead.

The code for this part is written in *Forecasting_training_closed_loop.m* source code.

Giving in input the open loop net, the actual outputs, the input feature vector, the number of delays and which has to be the farest prevision is possible to compute the (average) MSE of the prevision. What we have been obtained is the following graph: more far is the prediction, bigger is the (average) MSE. This can be justified by this fact: the prediction in the 1 day ahead is more precise than prediction of 15 days ahead because the 15th day prediction ahead is based on previous prediction.



To generalize is reported the forecasting one day ahead of all 2012 year against the real



number of bikes rental.

Conclusions

The study of renal bikes in Washington D.C. has reported a lot of interesting results. In first place the features to consider “important” are less than we initially thought.

Moreover an important part of the study was the preliminar study: is basically to study the dataset and the point of assignment in order to make the right decision, e.g. study how the data are spreaded in the dataset and their correlation with the output.

More features doesn't mean better performance and higher number of neurons dosen’t means better performance too. Moreover the complexity of the network doesn’t mean better performance. The important thing is to find the best tradeoff in the neural network complexity because it leads to “good” results reported from testing MSE.

In the first dataset (day) we have been noticed a bigger MSE than second dataset (hour). This is due from two factor: the cardinality and the magnitude of target vector. Bigger is the dataset, bigger is the training dataset. In this way the neural network will be trained better than with smallest dataset.

About the fuzzy part: the performance obtained in the ANFIS are better in the Mamdami fuzzy system because the membership functions parameters are tuned by a neural network which gives the best combination of parameters. This has been caused the generation of a more complex system than Mamdami fuzzy system case.

At least, about the forecasting in the open loop strategy, we have not been focused only the MSE among all combinations of delays number and hidden neurons number. We have been focused on error autocorrelation

and input error cross-correlation. The first one shows us how the error is correlated with itself in the the predictions and the second one is the correlation between input and error. Is important to notice that the sample must be independent for a correct study of the system, i.e. means that the correlation (in general) has to be lowest possible otherwise the results are meaningless.