# Project 2 – Computational intelligence

Solve a regression problem using TSK models

The aim of this project is to investigate the ability of TSK models in modeling multivariable, non-linear functions. Specifically, two datasets are selected from the UCI repository to estimate the target variable from the available data, using fuzzy neural models. The first data set will be used for a simple investigation of the process of training and evaluating models of this kind, as well as to suggest ways of analyzing and interpreting the results. The second, more complex data set will be used for a more complete modeling process, which will include, among other things, pre-processing steps such as feature selection, as well as methods for optimizing the models through cross validation.

## 1. Simple Dataset

In the first phase of the project, the Airfoil Self-Noise dataset is selected from the UCI repository, which includes 1503 instances and 6 features. We follow the following steps:

- Split the dataset into training, validation, and test sets. In the first phase, it is necessary to divide the data set into three non-overlapping subsets $D_{irn}$, $D_{val}$, $D_{chk}$, of which the first will be used for training, the second for validation and avoiding the overtraining phenomenon and the last for performance control of our final model. It is suggested to use 60% of the total samples for the training subset and 20% of the total samples for each of the two remaining subsets.
- Training TSK models with different parameters. In this stage, various TSK models will be examined in terms of their performance on the control set. Specifically, 4 TSK models will be trained, in which the format of the output as well as the number of participation functions for each input variable will be changed. Table 1 is given for further explanation. All 4 models will be trained with the hybrid method, according to which the parameters of the membership functions are optimized through the backpropagation algorithm while the output parameters are optimized using the Least Squares Method. The membership functions should be bell-shaped and they should be initialized so the fuzzy inputs have an overlapping rate of 0.5.

| | Lots of membership functions | Output format |
|---|---|---|
| TSK_model_1 | 2 | Singleton |
| TSK_model_2 | 3 | Singleton |
| TSK_model_3 | 2 | Polynomial |
| TSK_model_4 | 3 | Polynomial |

Table 1: Classification of models to be trained.

To compare the performance of the models, use the following metrics:

- MSE (Mean squared error)
- $R^2$ (R squared error)
- NMSE
- NDEI

Problem requirements:

For each of the 4 TSI models described in the above table, make the appropriate initializations and then perform the training of the models with the parameters described above. The final model should always be the one that corresponds to the smallest error in the validation set. For each of the 4 cases:

- Present the corresponding diagrams depicting the final forms of the fuzzy sets obtained through the training process.
- Present the learning curve for each model.
- Present the error diagrams.
- Finally present the aforementioned metrics.

Comment on the results of the models in terms of both the shape of the output and the partitioning of the input space. The larger number of fuzzy sets per input in the case of the corresponding TSK models led to overtraining? Interpret any differences in the performance of the four models.

## 2. Dataset with high dimensionality

In the second phase of the project, a more systematic approach to the problem of modeling an unknown function will be followed. For this purpose, a dataset with a higher dimensionality will be selected. An obvious problem arising from this choice is the so-called "explosion" of the number of IF-THEN rules (rule explosion). As is known from theory, for the classic case of grid partitioning of the input space, the number of rules increases exponentially with respect to the number of inputs, which makes it very difficult to model through a TSI model even for medium-scale datasets.

The dataset chosen to demonstrate the above methods is the Superconductivty dataset from the UCI Repository, which includes 21263 samples each described by 81 variables/attributes. It is obvious that the size of the dataset makes a simple application of a TSK model, like that of the previous part of the project, prohibitive. The large number of variables makes it necessary to use methods to reduce the dimensionality as well as the number of IF-THEN rules (e.g. with 81 variables/predictors, we would divide the input space of each variable with two fuzzy sets, we would end up with 281 rules) . This goal will be achieved through feature selection and the use of variance partitioning. However, these two methods, despite the reduction of complexity they bring, add two free parameters to the problem, namely, the number of features to be selected and the number of groups to be created. The choice of these two parameters is up to the user and is essential to the final performance of the model. In this work, the grid search method will be implemented to find the optimal values of the parameters. In detail, the modeling of the problem will therefore follow the steps below:

1. Separation into training-validation-control sets: As in the first part of the work, it is necessary to divide the data set into three subsets Dtrn, $D_{val}$, $D_{chk}$, one of which will be used for training and the second for testing performance.

2. Selection of the optimal parameters: As mentioned above, our system includes two free parameters whose value we must choose. The most popular method by which this is achieved is grid search. Specifically, after obtaining a set of values for each parameter, we create a n-dimensional grid (in our case n = 2), where each point corresponds to a n-set of values for the parameters in question, and at each point we use an evaluation method to check the correctness of the specific values. An established option for this evaluation is cross validation. According to this method, and for selected values of the parameters, we divide the training set into two subsets, one of which will be used to train a model and the second to evaluate it. This process is repeated – usually five or ten times – where each time a different partition of the training set is used, and at the end we get the average of the model's error. The rationale behind multiple training and testing is that in this way, we get a good estimate of the model's performance, and indirectly the parameter values on

which the model was built. When the above procedure is performed for each point of the grid, we obtain as optimal values of the parameters, the values corresponding to the model that presented the minimum average error. These values are used to train our final model.

For the purposes of the project, we define the following parameters:

• Number of features: The number of features that will be used to train the models.

• Radius of the clusters $r_a$: The parameter that determines the radius of the flow of the clusters and, by extension, the number of rules that will be generated.

3. Based on the optimal parameter values selected from the previous step, we train a final TSK model and check its performance on the test set.

The above steps fully summarize the modeling process to be followed. Note that the parameter sets as presented above are optional, and one can override their values, especially if the cross-validation process proves particularly time-consuming. The following are required:

1. The data set should be divided as in the first part, with the training-validation-control sets comprising respectively 60% - 20% - 20%.

2. Perform a grid search and evaluation through 5-fold cross validation to select the optimal values of the parameters. At each iteration, the mean error should be stored. Split the data so that in each iteration, 80% of the data is used for training and the remaining 20% for validation (as inputs to the anfis function of MATLAB). The Subtractive Clustering (SC) algorithm is chosen as the clustering method for creating the IF-THEN rules, and the feature selection can be performed with one of the following algorithms (Relief, mRMR, FMI). Pre-processing of the data is deemed necessary. After the procedure, comment on the results in terms of the mean error as a function of the parameter values. Provide graphs depicting the curve of this error versus the number of rules and versus the number of selected features. What conclusions can be made?

3. Train the final TSK model with the optimal parameter values and with the same specifications as before (SC). Present the following diagrams:

- Charts showing the predictions of the models.
- Diagrams of the learning curves.
- Present some sets in their initial and final form.
- Present the tables containing the metrics.

Finally, comment on the results regarding the features selected and the number of IF-THEN rules of the fuzzy inference system. To compare with the corresponding number of rules if, for the same number of features, we had chosen grid partitioning with two or three fuzzy sets per input. What are the conclusions?