TÉCNICO LISBOA

Intelligent Systems Assignment 2

Group nr: 2	**Student 1:** Álvaro Lopes	**IST nr:** 96148
	Student 2: André Lopes	**IST nr:** 96351

Dataset

The dataset used in this assignment is the same used in assignment 1. It consists of results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. Each sample has measured 13 associated features: 1) Alcohol; 2) Malic acid; 3) Ash; 4) Alcalinity of ash; 5) Magnesium; 6) Total phenols; 7) Flavanoids; 8) Nonflavanoid phenols; 9) Proanthocyanins; 10) Color intensity; 11) Hue; 12) OD280/OD315 of diluted wines; 13) Proline.

Neural model using a multi-layer perceptron

First, it was proposed to develop a neural model by using a multi-layer perceptron. This was done using the library *scikit-learn*. In order to compare the results with the previously developed first-order Sugeno fuzzy system, only 50% of the data was used for training, with the rest being equally split for test and validation.

A study to evaluate the quantity of neurons and the number of layers was also conducted, however, the results were inconclusive. The number of neurons ranged from 1 to 20 and the number of layers from 1 to 3, with every layer having the same number of neurons. One reason why this study was inconclusive is the fact that the dataset is very small and the number of parameters would quickly surpass the number of the samples used in training making it not suitable for such a study.

Below is presented the accuracy results for said study.

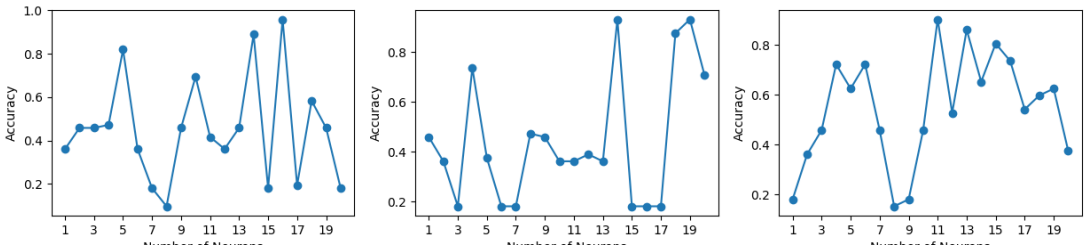

Figure 1: Effect of number of neurons and number of layers on the MLP model accuracy (left - 1 layer, center - 2 layers, right - 3 layers

Additionally a study to compare the loss in training and in validation was realized to check for any signs of overfitting. The model used for this consists of two layers of 14 neurons each. The comparison of the training and validation losses is presented below:

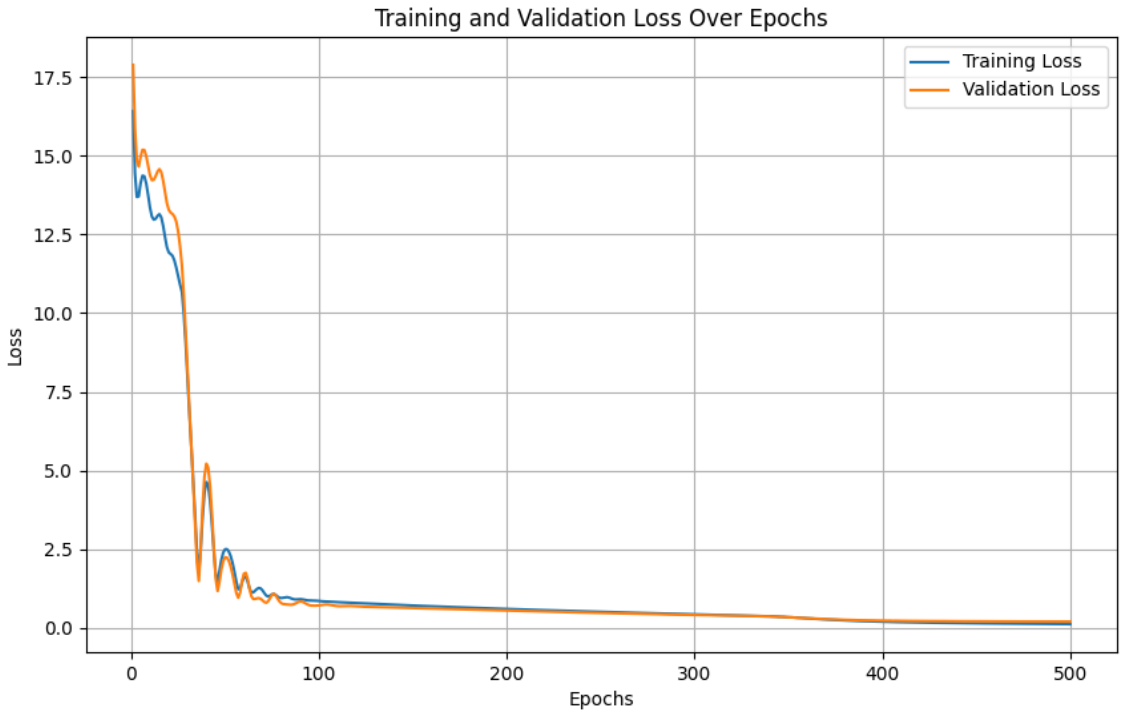

Figure 2: Training and validation loss

The absence of divergence between both losses indicates that the model is not overfitting. Additionally, looking back at figure 1, one can assume that the fact that the dataset is small is contributing to inconclusiveness of the first study.

The overall scores for the model were the following:

- Accuracy: 91.1%
- Precision: 91.2%
- Recall: 91.1%
- F1 Score: 91%
- Matthews Correlation Coefficient: 86.8%

Since the dataset is rather simple, even with relatively small training set, the model achieves good results. Beyond the accuracy, the Matthews correlation coefficient ensures good results for all 4 categories of the confusion matrix and the precision and recall corroborate that.

Neuro-fuzzy model

In this approach, we opted to use ANFIS in Matlab, instead of scikit-fuzzy for Python since the professor recommended it due to its faster nature.

First, we started by creating the Fuzzy Inference System, which was later tuned using ANFIS method. A loop was created in order to automate the creation of several models. These models are different as there is randomness both in

the train-test split and in the generation of the model. 80% of the data was used as test, just like in the assignment 1, so that we can better compare the results between assignments. The accuracy results were the following:

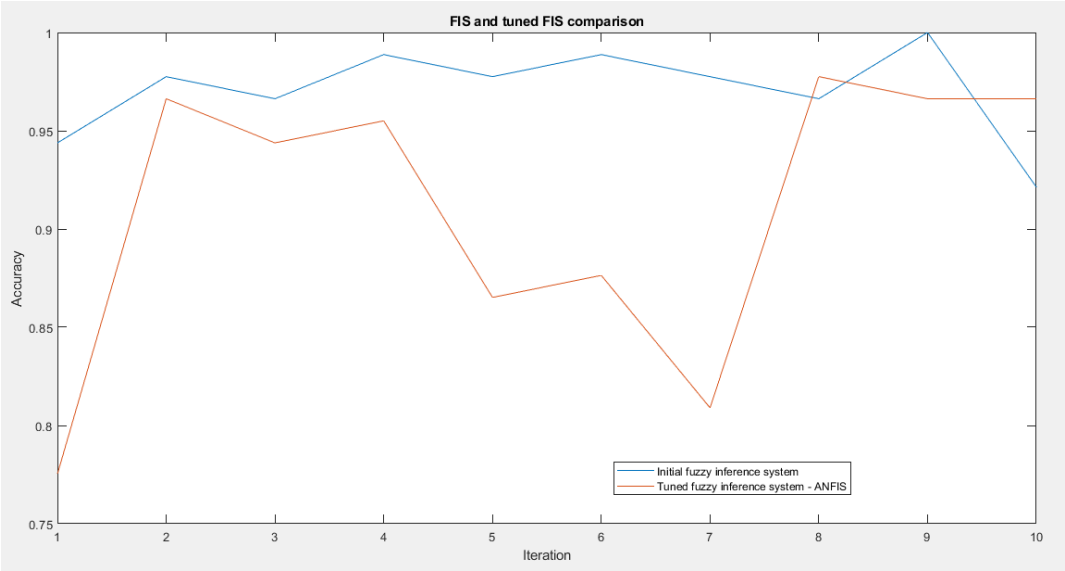

Figure 3: Comparison of accuracy metric in FIS and ANFIS

As we can observe, the initial model without the tuning not only gives better accuracy results, but they are much more consistent. This suggests that the data is better described by a simpler model, less prone to overfitting. This makes sense since our dataset is very simple, as we already had covered in assignment 1. By tuning it, the model may fit the training data better but becomes less generalized, explaining this results. Note that this is a particular case since our dataset has few samples.

Models Comparison

Although the neural model lacked convergence as the number of neurons and layers increased, this model could have shown improvements had other measures been taken, such as making a grid search to optimize the learning rate and the activating functions. Exploring different architectures (in this assignment only a rectangle shape - same number of neurons for every layer - was tried and tested) could have also shown signs of improvement.

Remembering the fuzzy inference system developed in the previous assignment, whose accuracy is displayed below, it performs a bit worse than the initial FIS developed in MATLAB (probably because of the way they were generated and the parameters associated with them).

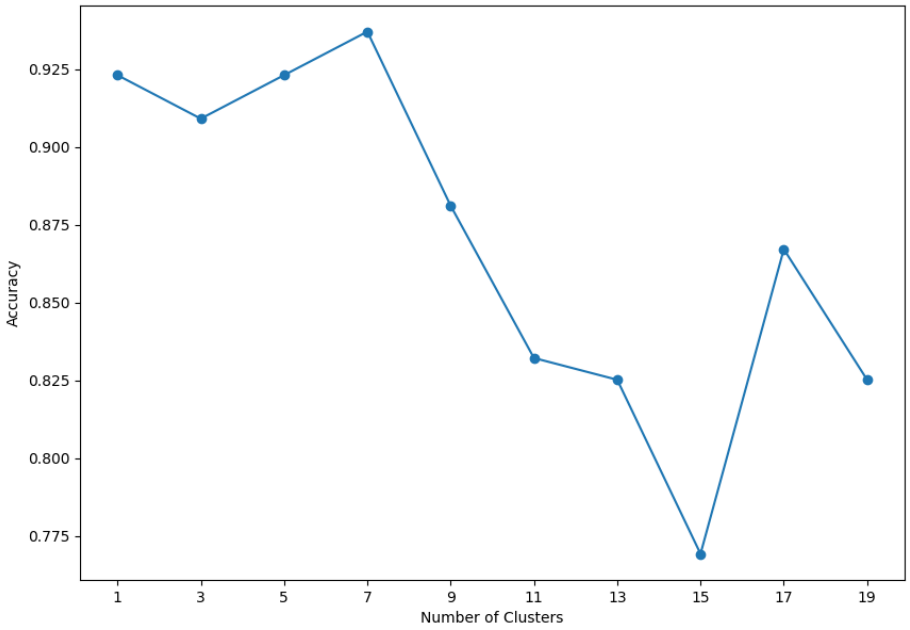

Figure 4: Assignment 1 - accuracy score for different numbers of clusters

However, it is when compared with the performance of the tuned model using ANFIS that the difference is a bit more pronounced, although not by much. The difference between the tuned ANFIS and the assignment 1 model is not much and could very well be attributed to different random states or a test subset that failed to be predicted correctly. It is easier to compare both lines in figure 3.

Link to group GitHub: `https://github.com/andrelopes2001/Intelligent_Systems`