Function Approximation with Type-1 Fuzzy Rule-Based System (FRBS)

Mohammad Mahdi Salehi Morgani

40330481

1. Project Definition

The goal of this project is to approximate an unknown single-input, single-output function using a Type-1 Fuzzy Rule-Based System (FRBS). The system is trained on a given dataset (xtrain, ytrain) and tested on (xtest, ytest) to evaluate its performance. The approximation quality is measured using the Root Mean Squared Error (RMSE) on the test data. Additionally, the project includes visualizing the membership functions, the approximated function, and the rule base.

---

2. Methodology Used to Get the Rules

The FRBS uses a zero-order Takagi–Sugeno–Kang (TSK) fuzzy model. The rule base is derived as follows:

- **Rule Generation:** A specified number of rules (n_rules) are uniformly distributed across the input domain. The centers of the membership functions define the rule antecedents.
- **Consequent Calculation:** For each rule, the consequent parameter is calculated as the weighted average of the training outputs, where the weights are the membership degrees of the training samples.

Formulation:

For each rule i, the consequent parameter P_i is computed as:

$$p_i = \frac{\sum_{j=1}^{N} \mu_i(x_j) y_j}{\sum_{j=1}^{N} \mu_i(x_j)}$$

where:

- $\mu_i(x_j)$: Membership degree of input x_j to the i^{th} rule.
- y_j: Output corresponding to x_j.
- N: Number of training samples.

---

3. How the Fuzzy Sets Are Derived

The fuzzy sets are defined using **triangular membership functions**. Each membership function has three parameters: left point aa, center point bb, and right point cc. The centers are uniformly spaced over the input domain. The width is calculated as:

$$\text{width} = \frac{|x_{max} - x_{min}|}{n_{rules} - 1}$$

Triangle Membership Function:

$$\mu(x; a, b, c) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x \leq b \\ \frac{c-x}{c-b} & b < x < c \\ 0 & x \geq c \end{cases}$$

---

4. Obtained Results

4.1. RMSE on Test Data

The model was evaluated on the test dataset, and the **Root Mean Squared Error (RMSE)** was computed as:

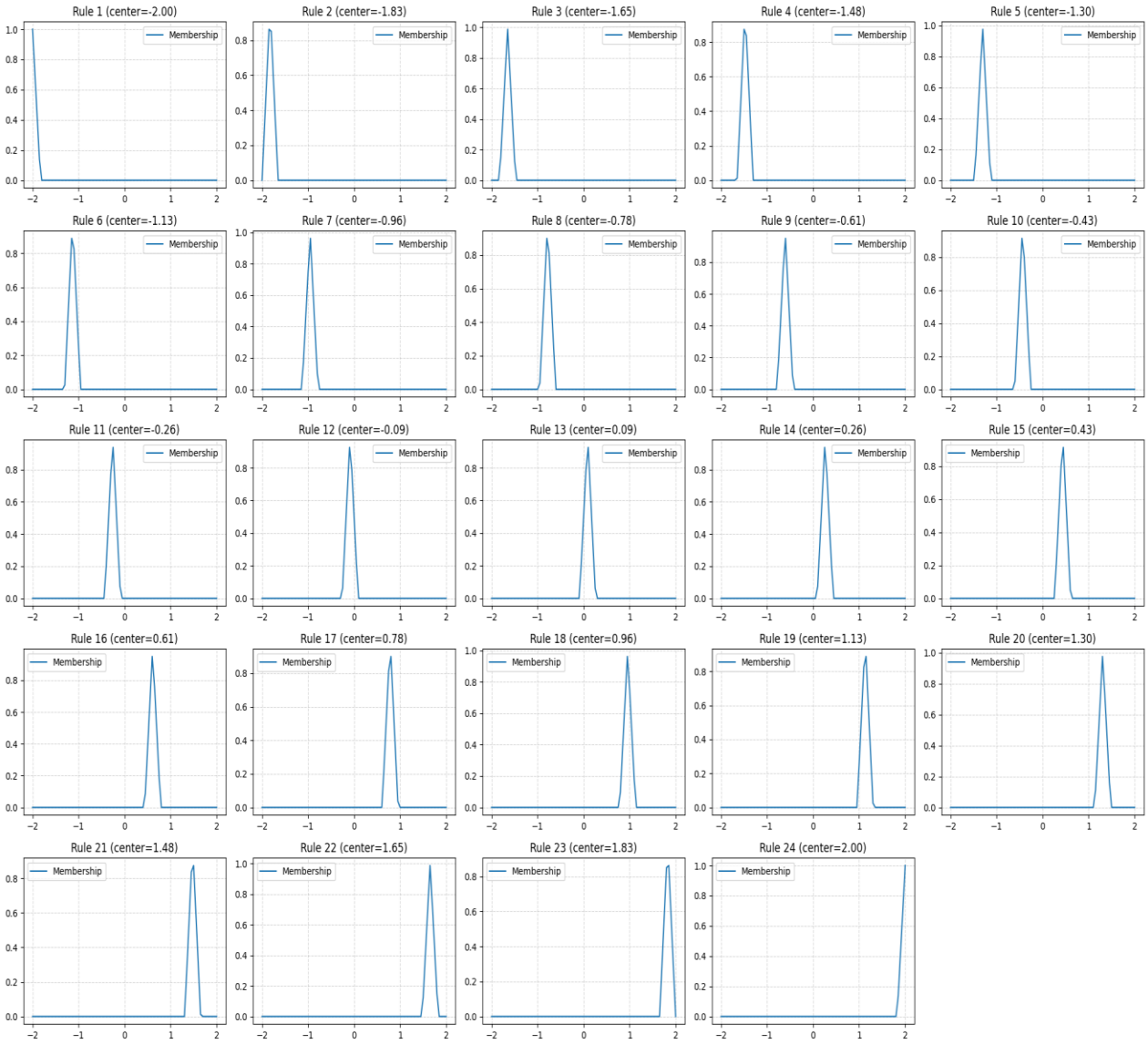
$$\text{RMSE} = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (y_i - \hat{y}_i)^2}$$

where M is the number of test samples, y_i is the actual output, and $\hat{y}_i$ is the predicted output.

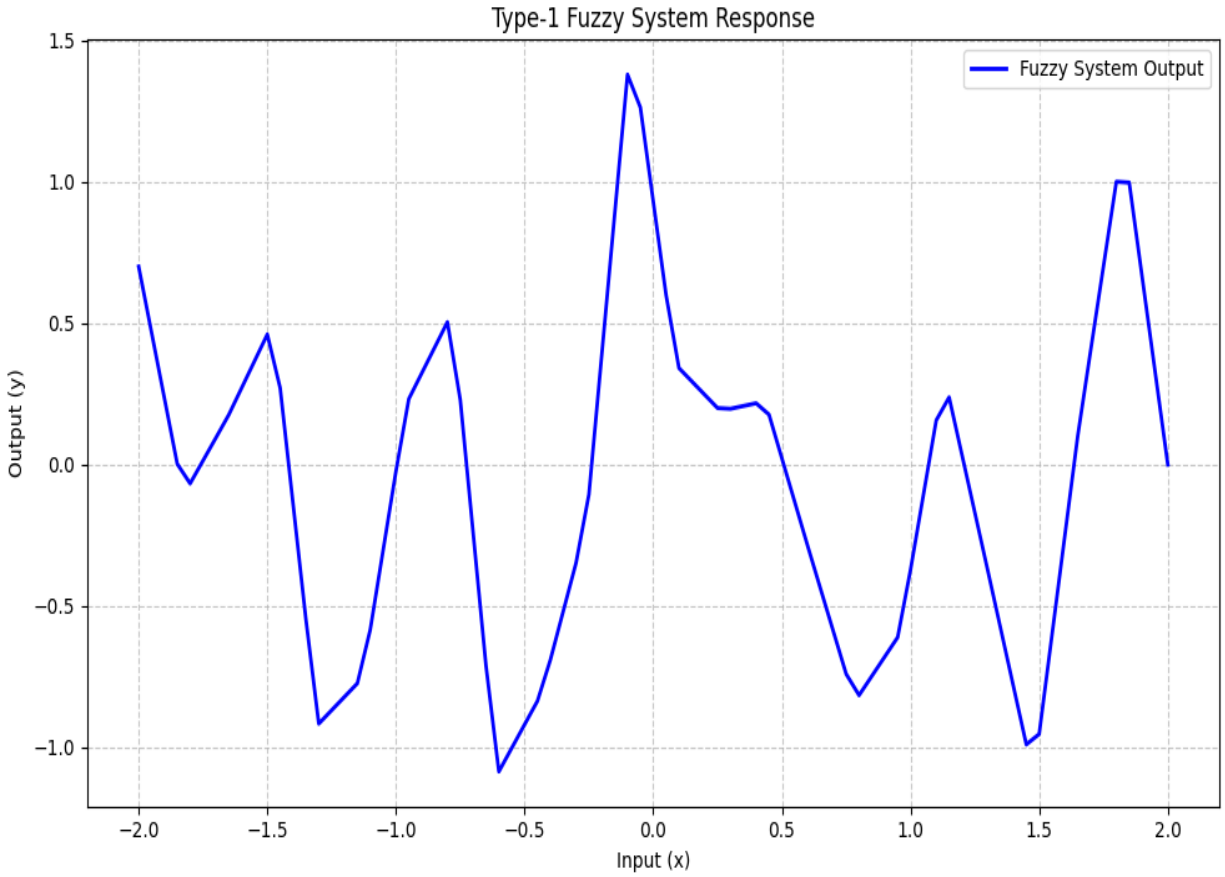
Test RMSE: *[0.4306]*

4.2. Visual Results

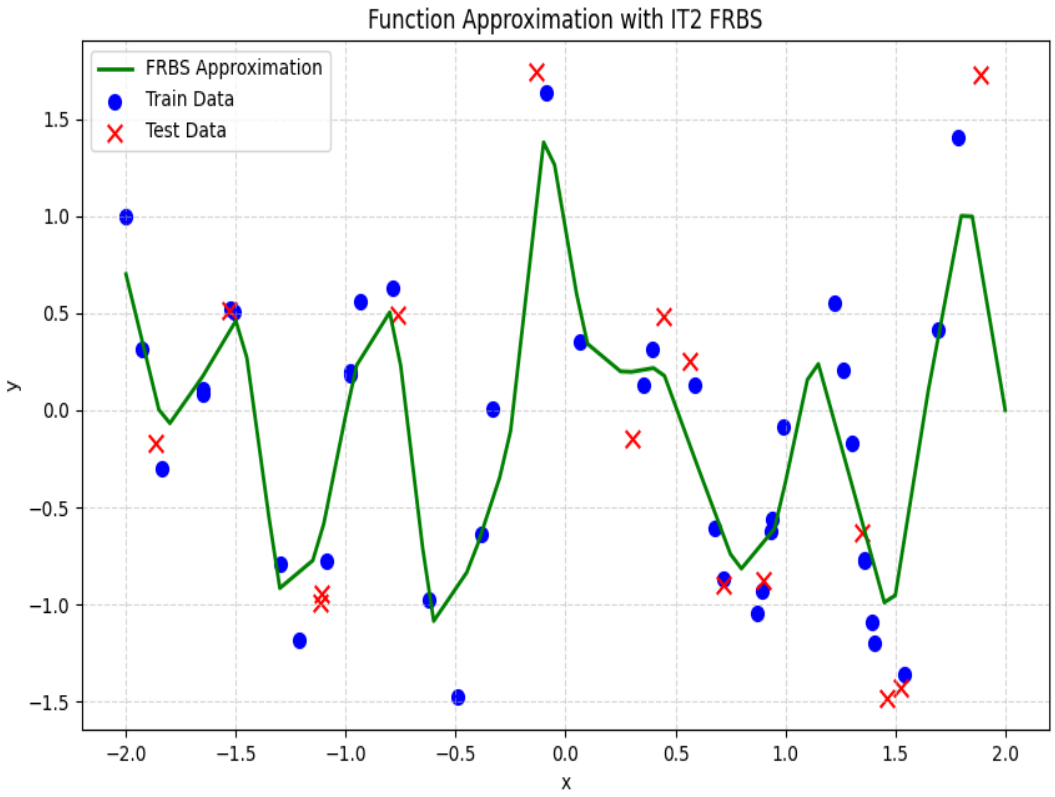
- Membership functions for all fuzzy rules were plotted.



- The approximated function was plotted over the range [-2, 2] with a step size of 0.05.



- Training and test data points were visualized alongside the approximated function.



---

5. Code Overview

5.1. Classes and Functions

Class: T1FRBS

Represents the Type-1 FRBS with zero-order TSK rules.

- **Attributes:**
  - n_rules: Number of fuzzy rules.
  - centers: Centers of membership functions.

- width: Width of the triangular membership functions.
- p: Consequent parameters.

- **Methods:**
  - **__init__(...):** Initializes the FRBS with uniformly distributed membership functions.
  - **compute_membership(x):** Calculates the membership degrees for an input.
  - **predict(x):** Predicts the output for a single input.
  - **fit(x_train, y_train):** Learns the consequent parameters using training data.
  - **predict_all(x_array):** Predicts outputs for an array of inputs.
  - **plot_rules(...):** Plots all membership functions.
  - **plot_function(...):** Plots the fuzzy system's output over a range.

Function: triangle_mf(x, a, b, c)

Computes the membership degree for a given input using the triangular membership function.

Function: rmse(y_true, y_pred)

Calculates the RMSE between actual and predicted outputs.

Function: plot_function_approximation(...)

Plots the approximated function, training data, and test data.

---

6. Appendix

6.1. List of All Rules

For a zero-order TSK model with n_{rules} = 24, the rules are of the form:

- **Rule 1:** If x is μ_1 (center = -2.000) then y = 0.999
- **Rule 2:** If x is μ_2 (center = -1.826) then y = -0.086
- **Rule 3:** If x is μ_3 (center = -1.652) then y = 0.094
- **Rule 4:** If x is μ_4 (center = -1.478) then y = 0.511
- **Rule 5:** If x is μ_5 (center = -1.304) then y = -0.987
- **Rule 6:** If x is μ_6 (center = -1.130) then y = -0.776
- **Rule 7:** If x is μ_7 (center = -0.957) then y = 0.314
- **Rule 8:** If x is μ_8 (center = -0.783) then y = 0.632
- **Rule 9:** If x is μ_9 (center = -0.609) then y = -1.475
- **Rule 10:** If x is μ_{10} (center = -0.435) then y = -0.638
- **Rule 11:** If x is μ_{11} (center = -0.261) then y = 1.638
- **Rule 12:** If x is μ_{12} (center = -0.087) then y = 1.638
- **Rule 13:** If x is μ_{13} (center = 0.087) then y = 0.351
- **Rule 14:** If x is μ_{14} (center = 0.261) then y = 0.126
- **Rule 15:** If x is μ_{15} (center = 0.435) then y = 0.310
- **Rule 16:** If x is μ_{16} (center = 0.609) then y = -0.610
- **Rule 17:** If x is μ_{17} (center = 0.783) then y = -1.046
- **Rule 18:** If x is μ_{18} (center = 0.957) then y = -0.620
- **Rule 19:** If x is μ_{19} (center = 1.130) then y = 0.553
- **Rule 20:** If x is μ_{20} (center = 1.304) then y = 0.209
- **Rule 21:** If x is μ_{21} (center = 1.478) then y = -1.483
- **Rule 22:** If x is μ_{22} (center = 1.652) then y = -1.358
- **Rule 23:** If x is μ_{23} (center = 1.826) then y = 0.410
- **Rule 24:** If x is μ_{24} (center = 2.000) then y = 1.406

6.2. How to Run the Project

1. Install the required libraries:
2. pip install numpy matplotlib
3. Run the main script:
4. python function.py
5. Output files (plots) will be saved in the project directory.

---

7. Conclusion

This project successfully approximated a single-input, single-output function using a Type-1 Fuzzy Rule-Based System with zero-order TSK rules. The methodology involved deriving fuzzy sets with triangular membership functions, fitting the model to training data, and validating it on test data. The obtained RMSE and visualizations demonstrate the system's approximation capabilities.