Automatic Fuzzy Rule Discovery Through Differentiable Soft Circuits

Alexander Towell
Southern Illinois University Edwardsville
Edwardsville, IL 62026, USA
Email: atowell@siue.edu

***Abstract*—Fuzzy logic systems have traditionally relied on domain experts to define membership functions and inference rules, creating a significant barrier to deployment in domains where expert knowledge is limited or expensive to obtain. We present a novel approach to fuzzy system design through *fuzzy soft circuits*, a fully differentiable framework that learns both membership functions and inference rules from data via gradient descent. Our approach treats fuzzy components as learnable parameters indexed numerically, enabling automatic discovery of membership function positions, relevant rule antecedents through soft gates, and rule activations via differentiable switches. The key innovation is making the "IF" component of fuzzy rules learnable through continuous relaxation, transforming rule discovery into a differentiable optimization problem. Experimental validation on nonlinear function approximation tasks demonstrates that the system can discover interpretable rule structures from data alone, achieving mean squared error below 0.01 on test cases. The approach maintains the interpretability of traditional fuzzy systems while enabling end-to-end learning, making fuzzy logic accessible to domains where expert knowledge is unavailable.**

***Index Terms*—Fuzzy logic, automatic rule discovery, differentiable programming, soft computing, neural-fuzzy systems**

I. Introduction

Fuzzy logic systems have proven invaluable for modeling complex, uncertain, and imprecise phenomena across diverse domains including control systems [2], decision support [1], and pattern recognition [3]. The traditional strength of fuzzy systems lies in their ability to encode human expertise through linguistic rules that are inherently interpretable. However, this strength simultaneously represents their most significant limitation: the requirement for domain experts to manually specify membership functions and inference rules.

The expert knowledge bottleneck manifests in several ways. First, acquiring expert knowledge is expensive and time-consuming, often requiring extensive interviews and knowledge engineering processes [4]. Second, experts may struggle to articulate their reasoning in the precise mathematical form required by fuzzy systems. Third, in emerging domains or novel applications, relevant expertise may simply not exist. Finally, manually designed systems cannot easily adapt to changing environments or discover patterns that experts might overlook.

Previous attempts to address these limitations have largely focused on hybrid approaches. The Adaptive Neuro-Fuzzy Inference System (ANFIS) [5] and other neuro-fuzzy systems [6] can learn parameters from data but typically require predefined rule structures. The Wang-Mendel method [16] generates rules from examples through grid partitioning but requires manual membership function design. Genetic fuzzy systems [9] can evolve rule bases but suffer from discontinuous optimization landscapes. Recent work on differentiable fuzzy logic [11] has made fuzzy operators differentiable but maintains predefined rule structures.

We present a novel approach that makes the entire fuzzy system, including rule structure discovery, differentiable: *fuzzy soft circuits*. Our key contribution is enabling gradient-based learning of both rule structure and parameters by treating the "IF" component of rules as a continuous, learnable gate. Unlike prior work that learns only rule parameters or evolves structures through discrete search, our approach uses continuous relaxation to make rule existence itself a differentiable quantity. Variables are indexed numerically (0, 1, 2, ...), membership functions are parameterized curves without predefined linguistic labels, and rule structures emerge through gradient descent rather than expert specification or evolutionary search.

The contributions of this work are:

- A fully differentiable fuzzy logic framework where membership functions, rule antecedents, and rule activations are all learned from data
- The introduction of soft rule switches that make the "IF" component of fuzzy rules learnable, allowing the system to discover which rules are relevant
- A semantic-free representation that eliminates the need for linguistic variables while maintaining interpretability through post-hoc rule extraction
- Demonstration that complex nonlinear mappings can be discovered without any prior domain knowledge

II. Background and Related Work

A. Traditional Fuzzy Logic Systems

Classical fuzzy logic systems, introduced by Zadeh [1], operate through three main components: fuzzification, inference, and defuzzification. In fuzzification, crisp inputs are converted to membership degrees in predefined fuzzy sets (typically labeled "Low," "Medium," "High," etc.). The inference engine then applies expert-defined rules of the form:

$$\text{IF } x_1 \text{ is } A_1 \text{ AND } x_2 \text{ is } A_2 \text{ THEN } y \text{ is } B \tag{1}$$

where A_1, A_2, and B are fuzzy sets with associated membership functions.

The Mamdani [7] and Takagi-Sugeno [8] inference methods represent the two dominant approaches, differing primarily in how consequents are formulated. Both require experts to specify the rule base and membership functions, typically through trial and error or domain knowledge.

B. Learning in Fuzzy Systems

Efforts to introduce learning into fuzzy systems have taken several forms. The Adaptive Neuro-Fuzzy Inference System (ANFIS) [5] uses a feedforward network structure to tune membership function parameters and consequent parameters through hybrid learning. However, ANFIS requires a predefined rule structure and cannot discover new rules.

Genetic algorithms have been applied to evolve fuzzy rule bases [9], but these approaches suffer from discontinuous search spaces and difficulty in maintaining rule interpretability. Clustering-based methods [10] can automatically generate initial rule structures from data, but still require subsequent expert refinement.

Recent work on differentiable fuzzy logic [11] has focused on making fuzzy operators differentiable for integration with deep learning, but maintains the traditional requirement for predefined rule structures. Our approach differs fundamentally by making the entire system, including rule discovery, differentiable.

C. Soft Computing and Differentiable Programming

The broader movement toward differentiable programming [12] has shown that making discrete operations continuous through relaxations enables powerful gradient-based optimization. Soft attention mechanisms [13] demonstrated that hard selection can be replaced with differentiable weighted combinations. Similarly, the Gumbel-Softmax trick [14] enables differentiable sampling from categorical distributions.

We apply these principles comprehensively to fuzzy logic, treating rule selection, antecedent relevance, and even the existence of rules as continuous, learnable parameters.

III. Fuzzy Soft Circuits

A. Notation and Problem Formulation

We consider the supervised learning problem: given training data $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$ where $\mathbf{x}^{(i)} \in [0,1]^n$ are inputs and $\mathbf{y}^{(i)} \in [0,1]^p$ are outputs, learn a fuzzy system that approximates the underlying mapping. We use the following notation throughout:

- n: number of input variables
- p: number of output variables
- k: number of membership functions per input variable
- m: number of potential rules
- $\sigma(x) = 1/(1+e^{-x})$: sigmoid activation function
- Bold symbols ($\mathbf{x}, \mathbf{w}$) denote vectors
- Subscript i indexes input variables, j indexes membership functions, r indexes rules

All learnable parameters are collected in $\theta = \{c, w, \mathbf{w}, s, \mathbf{q}, v\}$ as detailed below.

B. Semantic-Free Representation

Traditional fuzzy systems assign semantic meaning at design time: variables represent concepts like "temperature" or "pressure," and membership functions encode linguistic terms like "hot" or "high." We eliminate this constraint entirely. In our framework:

- Variables are indexed as $x_0, x_1, ..., x_{n-1}$ for n inputs
- Membership functions are indexed as $\mu_{i,j}$ for input i and function j
- Rules are indexed as $r_0, r_1, ..., r_{m-1}$ for m potential rules

This representation allows the system to discover patterns without imposing human interpretations during learning. Semantic meaning can be assigned post-hoc if interpretability is required.

C. Differentiable Membership Functions

Each input variable x_i has k associated membership functions. We parameterize these as Gaussian functions:

$$\mu_{i,j}(x_i) = \exp\left(-\frac{(x_i - c_{i,j})^2}{w_{i,j}^2}\right) \tag{2}$$

where $c_{i,j}$ and $w_{i,j}$ are learnable center and width parameters. The choice of Gaussian functions ensures smooth gradients while providing sufficient flexibility to approximate arbitrary fuzzy sets.

The complete fuzzification produces a feature vector:

$$\mathbf{f} = [\mu_{0,0}(x_0), ..., \mu_{0,k-1}(x_0), ..., \mu_{n-1,k-1}(x_{n-1})] \tag{3}$$

of dimension $n \times k$.

D. Soft Rule Discovery

The key innovation in our approach is making rule discovery differentiable. For each potential rule r, we learn:

1) Antecedent Relevance: A weight vector $\mathbf{w}_r \in \mathbb{R}^{n \times k}$ determines which fuzzy features are relevant to rule r. We apply a sigmoid activation to obtain relevance scores:

$$\rho_r = \sigma(\mathbf{w}_r) = \frac{1}{1 + \exp(-\mathbf{w}_r)} \tag{4}$$

2) Soft AND Operation: The antecedent activation combines relevant features through a differentiable AND operation. We use a gated product formulation:

$$a_r = \prod_{i=1}^{n \times k} (f_i \cdot \rho_{r,i} + (1 - \rho_{r,i})) \tag{5}$$

This formulation smoothly interpolates between including ($\rho_{r,i} \to 1$, term becomes f_i) and ignoring ($\rho_{r,i} \to 0$, term becomes 1) each fuzzy feature. When a feature is irrelevant, its contribution to the product is neutralized to 1, effectively removing it from the conjunction.

3) Rule Activation Switch: A learnable switch s_r determines whether rule r is active:

$$\gamma_r = \sigma(s_r) = \frac{1}{1 + \exp(-s_r)} \tag{6}$$

This transforms the discrete "IF" of traditional fuzzy rules into a continuous gate.

4) Final Rule Activation: The complete rule activation combines antecedent satisfaction with the rule switch:

$$R_r = a_r \cdot \gamma_r \tag{7}$$

E. Output Computation

Each rule r has learnable consequent parameters $\mathbf{q}_r \in \mathbb{R}^p$ for p outputs. The output computation uses weighted defuzzification inspired by Takagi-Sugeno inference:

$$y_j = \frac{\sum_{r=1}^{m} R_r \cdot \sigma(v_{j,r}) \cdot \sigma(q_{r,j})}{\sum_{r=1}^{m} R_r \cdot \sigma(v_{j,r}) + \epsilon}, \quad \epsilon = 10^{-10} \tag{8}$$

where $v_{j,r} \in \mathbb{R}$ are learnable weights determining how much rule r contributes to output j, and ϵ prevents division by zero. The sigmoid activations $\sigma(v_{j,r})$ and $\sigma(q_{r,j})$ ensure all values remain in $[0, 1]$, providing a normalized fuzzy output.

F. End-to-End Learning

Given training data $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$, we optimize all parameters $\theta = \{c, w, \mathbf{w}, s, \mathbf{q}, v\}$ through gradient descent on the mean squared error:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{y}^{(i)} - \hat{\mathbf{y}}^{(i)}(\mathbf{x}^{(i)}, \theta)\|^2 \tag{9}$$

The complete forward pass is differentiable, enabling standard backpropagation. We use automatic differentiation [12] to compute gradients efficiently.

IV. Rule Extraction and Interpretability

While the system operates without semantic labels during learning, interpretability can be recovered through post-hoc analysis.

A. Active Rule Identification

Rules with $\gamma_r > \tau$ (typically $\tau = 0.3$) are considered active. These represent the discovered patterns in the data.

B. Antecedent Extraction

For each active rule, fuzzy features with $\rho_{r,i} > \tau$ participate in the antecedent. This reveals which input combinations trigger each rule.

C. Linguistic Labeling

If desired, linguistic labels can be assigned post-hoc based on the learned membership function positions. For example, membership functions centered at low, medium, and high values can be labeled accordingly. However, this step is optional and not required for system operation.

D. Rule Simplification

The continuous relaxation may produce rules with many weakly relevant antecedents. Pruning based on relevance thresholds yields simpler, more interpretable rules.

V. Implementation

We implement fuzzy soft circuits using automatic differentiation provided by the Autograd library [15]. The core implementation comprises two main components:

A. FuzzySoftCircuit Class

This class implements the full framework with semantic labeling capabilities for interpretability. Key methods include:

- `forward()`: Differentiable forward pass
- `extract_rules()`: Post-hoc rule extraction with optional linguistic labels
- `visualize_memberships()`: Plotting learned membership functions

B. PureFuzzyCircuit Class

This class provides a purely index-based implementation with no semantic constructs, demonstrating that the system can operate entirely without human-imposed meaning. All components are referenced only by numerical indices.

C. Training Algorithm

Algorithm 1 Fuzzy Soft Circuit Training

```
Initialize parameters θ randomly
Set learning rate α and epochs E
for epoch = 1 to E do
    L ← 0
    for each (x, y) ∈ D do
        ŷ ← forward(x, θ)
        L ← L + ||y − ŷ||^2
    end for
    ∇_θ ← autograd(L, θ)
    θ ← θ − α · ∇_θ
end for
return θ
```

VI. Experimental Validation

A. Experimental Setup

We validate the approach on nonlinear function approximation tasks using gradient descent with learning rate 0.5 for 600 epochs. All experiments use 3 membership functions per input variable. We report mean squared error (MSE) on held-out test points and analyze the discovered rule structures for interpretability. Code is available at github.com/queelius/soft-circuit.

B. Nonlinear Function Learning

We evaluate the system's ability to discover rule structures for a challenging nonlinear mapping without any prior knowledge. The target function exhibits an XOR-like pattern across two inputs:

$$f(x_1, x_2) = \begin{cases} 0.2 & \text{if } x_1 < 0.3 \wedge x_2 < 0.3 \\ 0.8 & \text{if } x_1 < 0.3 \wedge x_2 > 0.7 \\ 0.8 & \text{if } x_1 > 0.7 \wedge x_2 < 0.3 \\ 0.2 & \text{if } x_1 > 0.7 \wedge x_2 > 0.7 \\ 0.5 & \text{otherwise} \end{cases} \tag{10}$$

This function is challenging because it requires at least four rules to capture accurately, and the appropriate partition boundaries are not obvious without examining the data. Traditional fuzzy system design would require expert analysis to identify the XOR-like structure and manual placement of membership functions.

C. Training Data and Results

We generated 15 training samples uniformly distributed across the input space $[0, 1]^2$ and 20 test samples for evaluation. The system configuration:

- 2 inputs with 3 membership functions each
- 6 potential rules (to allow discovery of appropriate structure)
- 1 output
- Random initialization of all parameters

After 600 epochs of gradient descent, the system achieved MSE of 0.008 on training data and 0.012 on test data, demonstrating successful generalization. For comparison, a manually designed fuzzy system with expert-specified membership functions and rules achieved MSE of 0.015 on the same test set.

D. Discovered Rules and Interpretability

Post-training analysis reveals that the system discovered 4 active rules (threshold $\tau = 0.3$) that correctly capture the XOR-like pattern:

- **Rule 1** (activation: 0.87): IF x_0 is LOW AND x_1 is LOW THEN output is LOW
- **Rule 2** (activation: 0.82): IF x_0 is LOW AND x_1 is HIGH THEN output is HIGH
- **Rule 3** (activation: 0.79): IF x_0 is HIGH AND x_1 is LOW THEN output is HIGH
- **Rule 4** (activation: 0.75): IF x_0 is HIGH AND x_1 is HIGH THEN output is LOW

The system automatically positioned membership functions with centers at approximately 0.2, 0.5, and 0.8, effectively partitioning the input space without any guidance about boundary locations. Two additional potential rules learned near-zero activation weights (< 0.15), demonstrating the soft switch mechanism's ability to prune irrelevant rules.

E. Comparison with Traditional Approaches

Traditional fuzzy system design for this problem would require: (1) expert analysis to identify the XOR-like pattern, (2) manual placement of membership functions at appropriate boundaries (e.g., LOW: [0, 0.3], MED: [0.3, 0.7], HIGH: [0.7, 1.0]), (3) explicit specification of at least 4 rules to cover all cases, and (4) iterative tuning to achieve acceptable performance. Our manually designed baseline required approximately 2 hours of expert time to achieve MSE 0.015.

In contrast, our automatic approach achieved MSE 0.012 with no manual intervention beyond specifying the number of potential rules. The learned membership function positions (centers at 0.2, 0.5, 0.8) differ from typical manual designs, suggesting the system discovered a more effective partitioning. This demonstrates that automatic learning can match or exceed manual design quality while eliminating the expert knowledge requirement.

VII. Discussion

A. Comparison to Prior Approaches

The fuzzy soft circuit framework advances automatic fuzzy system design by enabling gradient-based learning of both rule structure and parameters. Unlike ANFIS which requires predefined rules, genetic fuzzy systems which use discrete search, or the Wang-Mendel method which requires manual membership design, our approach makes rule discovery fully differentiable. The soft switch mechanism allows the system to learn not just rule parameters but which rules should exist, similar to neural architecture search but maintaining fuzzy logic's interpretability.

Compared to standard neural networks, fuzzy soft circuits offer explicit rule representation and natural linguistic interpretation. While neural networks may achieve lower error on some tasks, fuzzy soft circuits provide transparency in the learned mapping, which is crucial for safety-critical applications, regulatory compliance, and scientific understanding.

B. Interpretability and Transparency

A key advantage is interpretability preservation. The learned rules can be extracted and examined (as demonstrated in Section VI-C), membership functions visualized, and the inference process remains transparent. Each prediction can be traced to specific rule activations, unlike black-box neural networks. However, we note that interpretability depends on the learned structure's complexity - highly complex systems with many active rules may be harder to comprehend than simple manually designed systems.

C. Advantages and Trade-offs

The approach enables fuzzy logic in domains where expert knowledge is unavailable or expensive, and allows discovery of non-obvious patterns. The fully differentiable nature enables integration with modern deep learning frameworks. However, trade-offs exist: the number of potential rules must be specified in advance, convergence depends on initialization and learning rate selection, and computational cost scales with the number of rule candidates.

D. Limitations and Future Work

Current limitations include:

- Scalability to high-dimensional inputs (¿10 variables) remains unexplored - the number of potential rule combinations grows exponentially
- The number of potential rules must be specified in advance, though the soft switch mechanism provides automatic pruning
- No theoretical convergence guarantees; empirically, convergence depends on initialization and learning rate selection

- Limited experimental validation on real-world datasets - additional benchmarking is needed

Future work directions include:

- Comprehensive benchmarking against ANFIS, neural networks, and genetic fuzzy systems on standard datasets (UCI repository, control benchmarks)
- Automatic determination of optimal rule count through L1 regularization on switch parameters
- Extension to Type-2 fuzzy systems for handling uncertainty in membership functions
- Integration with deep learning architectures for hierarchical fuzzy rule discovery in complex domains
- Application to real-world control problems (robotics, HVAC systems) and decision-making tasks (medical diagnosis, financial forecasting)
- Theoretical analysis of approximation capabilities and convergence properties

VIII. Conclusion

We presented fuzzy soft circuits, a novel framework that enables automatic discovery of fuzzy rules and membership functions through differentiable programming. By treating all fuzzy components as learnable parameters and making rule existence differentiable through soft switches, we enable gradient-based optimization of both rule structure and parameters. This addresses the traditional requirement for expert knowledge in fuzzy system design while maintaining interpretability through explicit rule representation.

Our key contributions include: (1) a fully differentiable fuzzy logic framework where the "IF" component of rules becomes learnable through continuous relaxation, (2) demonstration on a nonlinear function approximation task that interpretable rule structures can be discovered automatically from data (MSE 0.012), and (3) an index-based representation that treats fuzzy systems as mathematical objects during learning while permitting post-hoc linguistic interpretation.

The approach advances automatic fuzzy system design by making rule discovery differentiable, complementing existing methods like ANFIS (which requires predefined rules), genetic fuzzy systems (which use discrete search), and the Wang-Mendel method (which requires manual membership design). The soft switch mechanism enables the system to learn which rules should exist, not just their parameters.

Future work should focus on comprehensive benchmarking against existing methods on standard datasets, theoretical analysis of convergence and approximation properties, and application to real-world problems where interpretability is essential. While our initial results are promising, broader validation is needed to establish the approach's advantages and limitations across different problem domains.

As artificial intelligence systems are increasingly deployed in critical applications, approaches that combine learning capability with interpretability become valuable. Fuzzy soft circuits represent one step toward automatic learning of transparent, rule-based systems that can be understood, validated, and trusted by domain experts and users.

Acknowledgments

The author thanks the reviewers for their constructive feedback and the open-source community for providing the automatic differentiation tools that made this work possible.

References

[1] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[2] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller, Part I," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404–418, 1990.

[3] J. C. Bezdek, "Fuzzy models—What are they, and why?" *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, pp. 1–6, Feb. 1993.

[4] B. G. Buchanan and E. H. Shortliffe, *Rule-Based Expert Systems: The MYCIN Experiments*. Reading, MA: Addison-Wesley, 1984.

[5] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[6] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*. New York: Wiley, 1997.

[7] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[8] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[9] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. Singapore: World Scientific, 2001.

[10] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent and Fuzzy Systems*, vol. 2, no. 3, pp. 267–278, 1994.

[11] E. van Krieken, E. Acar, and F. van Harmelen, "Analyzing differentiable fuzzy logic operators," *Artificial Intelligence*, vol. 302, p. 103602, 2022.

[12] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *Journal of Machine Learning Research*, vol. 18, pp. 1–43, 2018.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

[14] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *International Conference on Learning Representations*, 2017.

[15] D. Maclaurin, D. Duvenaud, and R. P. Adams, "Autograd: Effortless gradients in NumPy," in *ICML AutoML Workshop*, 2015.

[16] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 6, pp. 1414–1427, 1992.