Possibilistic Clustering Enabled Neuro Fuzzy Logic

Blake Ruprecht, Wenlong Wu, Muhammad Aminul Islam, Derek Anderson, James Keller, Grant Scott, Curt Davis, Fred Petry, Paul Elmore, Kristen Nock, and Elizabeth Gilmour

***Abstract*—Artificial neural networks are a dominant force in our modern era of data-driven artificial intelligence. The adaptive neuro fuzzy inference system (ANFIS) is a neural network based on fuzzy logic versus a more traditional premise like convolution. Advantages of ANFIS include the ability to encode and potentially understand machine learned neural information in the pursuit of explainable, interpretable, and ultimately trustworthy artificial intelligence. However, real-world data is almost always imperfect, e.g., incomplete or noisy, and ANFIS is not naturally robust. Specifically, ANFIS is susceptible to over inflated uncertainty, poor antecedent (fuzzy set) data alignment, degenerate optimization conditions, and hard to interpret logic, to name a few factors. Herein, we explore the use of possibilistic clustering to identify outliers, specifically typicality degrees, to increase the robustness of ANFIS; or any fuzzy logic neuron/network at that. Experiments are presented that demonstrate the need and quality of the proposed solutions in the pursuit of robust interpretable machine learned neuro fuzzy logic solutions.**

TABLE I: Acronyms and Notation

ANFIS	Adaptive Neuro Fuzzy Inference System
SP1M	Sequential Possibilistic One-Means
N	Number of input samples
K	Number of input features
R	Number of rules in an ANFIS rule-base
$N \times K$	Dimensionality of input dataset
$\mathbf{x}_n$	Input of feature length K
$\mathbf{x}_n(k)$	Scalar feature of input vector
$\mathbf{w}_n$	Antecedent vector of the rule base, of length R
$\mathbf{z}_n$	Consequent vector of the rule base, of length R
$A_k^r(\cdot)$	Membership function of rule r
p_k^r	Component weight of rule r
y_n	Output of ANFIS for $\mathbf{x}_n$
c	Number of clusters
u_{ji}	Typicality of ith sample to cluster j
v_j	jth cluster prototype

I. Introduction

The world is once again fixated on neural nets, due in large part to their recent performance leaps across numerous application domains; computer vision, natural language processing, etc. On the other hand, modern deep learning has a list of equally deep concerns, e.g., are we really just engineering machines that discover desirable correlations versus underlying causations [1]. Regardless, in all this excitement the field has more-or-less converged into a single mathematical foundation, convolution; which powers more complicated constructs like residual and recurrent networks. Furthermore, the vast majority of these deep nets have given rise to black box solutions– that have little emphasis on explainability or interpretability. Herein, we focus on a non-convolutional contribution from the field of fuzzy set theory, the *adaptive neuro-fuzzy inference system* (ANFIS) [2]. Specifically, we focus on a first order (linear) *Takagi-Sugeno-Kang* (TSK) type ANFIS.

A benefit of a *fuzzy logic neuron/network* (FLN), e.g., TSK ANFIS, is it holds the potential to help realize more explainable, interpretable, and ultimately trustworthy AI. There are a number of ways in which this can occur. For one, it is possible to insert human knowledge as rules into a FLN. Furthermore, a FLN can be derived from data then opened to study what variables, rules, and output combination strategies were learned. However, as shown by Keller and Yager in [3], fuzzy logic can be achieved using a *multi layer perceptron* (MLP), as both are well-known universal function approximators. A benefit of an approach like ANFIS, versus [3], is the information is explicit and centralized versus implicit and distributed. While numerous approaches exist to learn a fuzzy inference system, we focus on neural to support "plug-and-play" into existing deep learners and to maintain homogeneity for the sake of optimization (e.g., backpropagation and gradient descent).

One challenge with current FLNs, and ANFIS in particular, is they are not robust. Specifically, when noise is present it is likely that ANFIS will obtain variables with over inflated uncertainty: sets that are wider than they need to be. Also likely is poor placement: misalignment of the learned set relative to the underlying truth. Due to degenerate optimization conditions, if ANFIS is initialized with more uncertainty than what is required, then it is likely that uncertainty will not sufficiently decrease. Furthermore, it is possible to obtain rules that decrease the error function, but make little-to-no high level sense. Also, once a rule is dead (i.e., it never fires), it stays dead. Meaning, it does not get updated during training due to no data points contributing to the parameter update for that particular rule. Last, determining how many sets and rules are required for ANFIS has proven difficult, and not robust. As the reader can see, many challenges relate to ANFIS and FLN learning. The points mentioned above are explained and addressed in more detail later in the article.

Our contributions are as follows. First, possibilistic clustering is used to produce `(data point, typicality)`, where u_{ji} is the degree to which data point $\mathbf{x}_i \in [0, 1]$ belongs to cluster j. Note, if u_{ji} is *low* for all j, then a data point is generally considered an outlier. Second, we extend ANFIS to use the data point and its typicality degree in learning. To this end, we explore two ways to exploit this knowledge during optimization. As we show, the combination of possibilistic clustering and an extended ANFIS allows us to address many of the challenges discussed herein.

Blake Ruprecht is with the Electrical Engineering and Computer Science (EECS) department, University of Missouri (MU), MO, USA, e-mail: bcrf53@mail.missouri.edu. Wenlong Wu, Muhammad Aminul Islam, Derek Anderson, James Keller, Grant Scott, and Curt Davis are with EECS, MU. Fred Petry, Kristen Nock and Elizabeth Gilmour are with the U.S. Naval Research Laboratory. Paul Elmore is with Johns Hopkins Applied Physics Laboratory, USA.

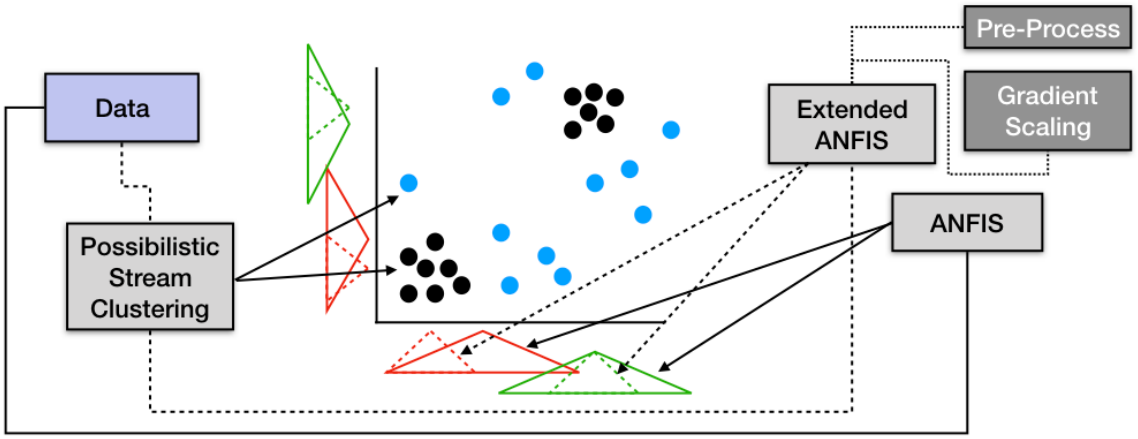


Fig. 1: High-level overview of the proposed work. Traditionally, ANFIS is applied to the raw full data. Here, possibilistic clustering is used to acquire data typicality degrees, which are fed to ANFIS during learning. Blue dots denote noise points (outliers) and black dots belong to a cluster. Red (and green, respectively) triangles are learned membership functions, where red indicates rule one and green indicates rule two. Solid lines are ANFIS learned solutions and dotted is extended ANFIS.

Before we delve into ANFIS and our extensions, it is important to note that ANFIS is not the only neuro-fuzzy architecture. In [3], Keller and Yager proposed a *multi layer perceptron* (MLP) to learn fuzzy logic. In [4], Keller and Tahani discussed the implementation of a conjunctive and disjunctive fuzzy logic rules with neural networks. In [5], Pal and Mitra explored an array of topics on neuro-fuzzy related to pattern recognition. In [6], Rajurkar and Verma put forth a deep fuzzy network with Takagi Sugeno fuzzy inference system. In [7], Blake et al. discussed deep ANFIS for remote sensing and open source PyTorch codes were made available at https://github.com/Blake-Ruprecht/Fuzzy-Fusion. Beyond fuzzy logic, there are numerous other recent fuzzy neuro investigations; e.g., *eXplainable AI* (XAI) based fuzzy integral neural network [8], fuzzy integrals for fusing heterogeneous architecture deep learners in remote sensing [9], fuzzy layers in deep learning [10], ordered weighted average networks [11], to name a few. The point is, past and present works exist connecting fuzzy set theory and neural networks at many levels.

The remainder of the article is organized as such. In Section II we review ANFIS, Section III is a possibilistic clustering approach, Section IV discusses typicality extended ANFIS, and Section V is experiments and results. Table I is a summary of acronyms and notation and Figure 1 illustrates the structure of our paper.

II. ANFIS

In [2], Jang introduced ANFIS, which was initially based on TSK type fuzzy inference [12]. An illustrative overview and input-output depiction of ANFIS can be seen in Figure 2. Let a training dataset have dimensionality, $N \times K$. While it is possible to support batch and mini-batch processing in contexts like gradient descent-based optimization, herein we focus on sample-by-sample processing for sake of readability. Let $\mathbf{x}_n(k)$ denote the k-th feature of sample n, let ANFIS consist of R rules, and let $y_n \in \Re$ be the output of ANFIS. ANFIS performs three steps on $\mathbf{x}_n$ to determine y_n. The first two steps, the Antecedent Firing and the Consequent Component Building, can be run in parallel, and they produce an antecedent vector, $\mathbf{w}_n$, and consequent weighting vector, $\mathbf{z}_n$, respectively. The third step, Aggregation, takes $\mathbf{w}_n$ and $\mathbf{z}_n$, performs a weighted sum, and it produces the final output, y_n. The lengths of $\mathbf{w}_n$ and $\mathbf{z}_n$ are R. The rules in ANFIS follow the familiar *IF-THEN* format, each of which has an antecedent and consequent clause.

Antecedent Firing: consists of calculating the firing strength of all the antecedent clauses of each rule, w_n^r, which uses a t-norm (herein, we use the product operator) of the membership values, $A_k^r(\cdot)$, of each input feature, i.e.,

$$w_n^r = \prod_{k=1}^{K} A_k^r(\mathbf{x}_n(k)). \tag{1}$$

The membership functions $A_k^r(\cdot)$ are unique to each rule and feature, and can be learned. This step can be thought of as how well the input vector matches each individual rule.

Consequent Component Building: consists of calculating the components of the consequent clause of each rule, z_n^r, which is determined based on the summation of each input feature and weight p_k^r, plus a bias term, p_{bias}^r, i.e.,

$$z_n^r = \left(\sum_{k=1}^{K} \mathbf{x}_n(k) p_k^r \right) + p_{bias}^r. \tag{2}$$

The input feature weight p_k^r is a unique scalar to each rule and feature, which can be learned. This step can be thought of as the output each rule would make for the input vector that perfectly fires the antecedent clauses.

Aggregation Step: consists of calculating the weighted aggregation of the consequent clauses, $\mathbf{z}_n$, using the antecedent clauses, $\mathbf{w}_n$, as weights, to produce the output scalar y_n, i.e.,

$$y_n = \frac{\sum_{r=1}^{R} z_n^r w_n^r}{\sum_{r=1}^{R} w_n^r}. \tag{3}$$

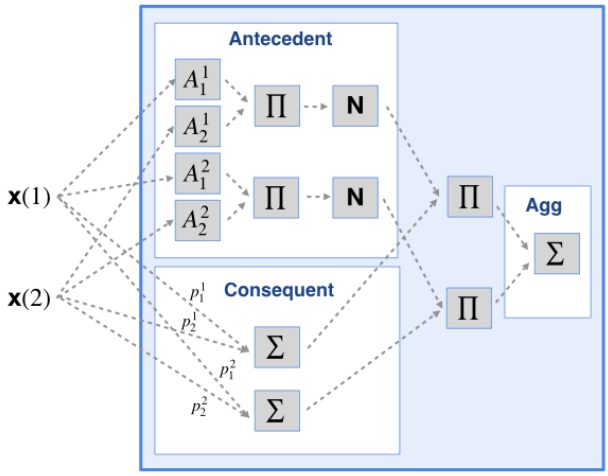


Fig. 2: This figure illustrates the flow of data in a first order TSK ANFIS for the case of two inputs and two rules.

This step can be thought of as the combination of the logical decisions from each rule based on how well the input vector matched each rule.

A. Semantic Considerations

By far, the most commonly utilized membership function for ANFIS is the Gaussian,

$$N(\mathbf{x}_n(k);\mu_k,\sigma_k) = e^{\left(-\frac{1}{2}\frac{(\mathbf{x}_n(k)-\mu_k)^2}{\sigma_k^2}\right)}. \tag{4}$$

While desirable, e.g., for differentiation, simplicity to work with, nonlinear, and infinite support (theoretically), a semantic limitation is that the standard Gaussian can only have a single element with membership value one. From a modeling standpoint, where one actually cares about the membership values and their qualitative interpretations, this can prove to be overly restrictive. While there are numerous functions to remedy this shortcoming, herein we focus, without loss of generality, on the trapezoidal membership function,

$$T(\mathbf{x}_n(k);\boldsymbol{\Theta}) = \begin{cases} 0 & (\mathbf{x}_n(k) < \Theta_1^k) \text{ or } (\mathbf{x}_n(k) > \Theta_4^k) \\ 1 & \Theta_2^k \leq \mathbf{x}_n(k) \leq \Theta_3^k \\ \frac{\mathbf{x}_n(k)-\Theta_1^k}{\Theta_2^k-\Theta_1^k} & \Theta_1^k \leq \mathbf{x}_n(k) < \Theta_2^k \\ \frac{\Theta_4^k-\mathbf{x}_n(k)}{\Theta_4^k-\Theta_3^k} & \Theta_3^k < \mathbf{x}_n(k) \leq \Theta_4^k. \end{cases} \tag{5}$$

B. ANFIS Optimization

For sake of article completeness, Table II are the partial derivatives for ANFIS, based on a first order TSK model. The reader can refer to [2], [7], and [13] for full mathematical explanation and derivations.

C. Open Source Codes

For reproducible research, in [7] we provided free PyTorch codes for shallow and deep ANFIS; https://github.com/Blake-Ruprecht/Fuzzy-Fusion. For the current article, we have placed our possibilistic clustering extended ANFIS at https://github.com/Blake-Ruprecht/ANFIS-SP1M.

TABLE II: ANFIS Derivatives for Gradient Descent

$\frac{\partial y_n}{\partial z_n^r}$ =	$\frac{w_n^r}{\sum_{i=1}^R w_n^i}$
$\frac{\partial y_n}{\partial w_n^r}$ =	$\frac{\sum_{j=1,j\neq r}^R w_n^j(z_n^r - z_n^j)}{(\sum_{i=1}^R w_n^i)^2}$
$\frac{\partial y_n}{\partial p_k^r}$ =	$\frac{\partial y_n}{\partial z_n^r}\cdot \mathbf{x}_n(k) = \frac{w_n^r}{\sum_{i=1}^R w_n^i}\mathbf{x}_n(k)$
$\frac{\partial y_n}{\partial A_k^r(\cdot)}$ =	$\left(\frac{\partial y_n}{\partial w_n^r}\right)\left(\prod_{j=1,j\neq k}^K A_j^r(\mathbf{x}_n(j))\right)$
$\frac{\partial y_n}{\partial \Theta_m^{r,k}}$ =	$\left(\frac{\partial y_n}{\partial A_k^r(\cdot)}\right)\left(\frac{\partial A_k^r(\cdot)}{\partial \Theta_m^{r,k}}\right)$
$\frac{\partial N(\cdot)}{\partial \mu}$ =	$exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\left(\frac{x-\mu}{\sigma^2}\right)$
$\frac{\partial N(\cdot)}{\partial \sigma}$ =	$exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\left(\frac{(x-\mu)^2}{\sigma^3}\right)$

D. Limitation of Traditional ANFIS

ANFIS [2] is a powerful tool, but not without flaw. While a number of shortcomings have been identified, and addressed to varying degrees, we highlight a few relevant limitations.

1) Noise and Over Inflated Uncertainty: Noise *pulls* the estimated membership functions away from their true state of nature. For example, consider a trapezoidal membership function. Noise in the data will increase the core region (membership value one) and the support (membership value greater than zero) will widen. From a semantic standpoint, this means noise has the impact of "inflating" or creating greater uncertainty than what actually exists.

2) Initialization: A number of works have been proposed to estimate the number and parameters of rules from data. ANFIS, by default, requires the number of rules, and number of antecedents per rule, to be specified before optimization can begin. ANFIS is not a *structure* learning algorithm, it is a parameter estimation algorithm. Furthermore, what values should the membership functions (and TSK coefficients) be set to initially, i.e., how should ANFIS be initialized? This selection can have an impact on the result in the hands of an algorithm like stochastic gradient descent.

3) Non-Decreasing Uncertainty: An underappreciated aspect of ANFIS is its inability to reduce uncertainty during learning. Consider two inputs and two rules. Let each rule correspond, antecedent-wise, to an underlying Gaussian distribution and let these clusters have no overlap and sufficient separation. Suppose that the standard deviations are initialized to three times the true standard deviation. This is a simple example. The point is, if no data points exist in the space between the two underlying clusters, then there is no part of ANFIS learning that naturally tries to reduce uncertainty. There are no errors driving the sets to be as specific as possible (see Proposition 2). From a control standpoint this might be desirable if the goal is to partition the underlying space. However, if the goal is to listen to the data and to have the membership functions fit the underlying samples, then this is problematic.

4) Once Dead, Always Dead: Consider an ANFIS with one or more rules initialized such that their membership functions do not fire, i.e., return a rule firing strength greater than approximately zero for any sample in the training data. Here, we refer to such rules as dead. In ANFIS, once a rule is dead, there is little-to-no *force* (sufficiently scaled gradient) that drives that rule towards a location in space to remedy the problem (see Proposition 1). This is a massive limitation with respect to learning quality ANFIS solutions. It places a large burden on the initial position of the antecedent fuzzy sets and/or it requires us to initialize with wide uncertainties which conflicts with the above non-decreasing uncertainty challenge.

In summary, this subsection exists to raise awareness of ANFIS limitations that need addressing. The combination of possibilistic clustering and modified ANFIS learning leads to more robust neuro fuzzy logic learning.

III. Sequential Possibilistic One Means (SP1M)

The *sequential possibilistic one means* (SP1M) algorithm [14] is based on the *possibilistic c-means* (PCM) algorithm [15]. The PCM abandons the membership sum-to-one constraint in the *fuzzy c-means* (FCM) [16] algorithm and it has been shown to be robust against outliers. Each cluster in PCM is independent of the others and the user and/or algorithm may have to address coincident clusters. The SP1M was created to combat the coincident clusters problem of PCM by generating one cluster at a time until all the "dense" regions are found.

Pseudocode for the latest version of SP1M is shown in Algorithm 1, where X is the input samples, c is the estimated input for cluster number, ϵ is the threshold, m is the fuzzifier, K is defined to be the number of points whose maximum typicality is smaller than 0.5. Note that the (*) detail of dynamic η computation in Algorithm 1 is discussed in [14].

Algorithm 1 SP1M Pseudocode

INPUT: X, c, ϵ
OUTPUT: U: final typicality partition
OUTPUT: V: final cluster prototypes
Initialize U, V as empty
while $j{+}{+} < c$ and $\#(P1M) < K$ **do**
  repeat ▷ loop to find a suitable cluster
    Pick $\mathbf{v} \in X$ with probabilities Eq. 6
    repeat ▷ loop to execute P1M
      Compute η_j dynamically (*)
      Compute typicality $u_{ji} = \frac{1}{1+\left(\frac{d_{ji}^2}{\eta_j}\right)^{\frac{1}{m-1}}}$
      Compute cluster center: $\mathbf{v}_j = \frac{\sum_{i=1}^{N} u_{ji}^m \mathbf{x}_i}{\sum_{i=1}^{N} u_{ji}^m}$
    until $\Delta v_j < \epsilon$
  until $\min_{\mathbf{w} \in V} ||\mathbf{v}_j - \mathbf{w}|| \geq 2\eta$
  Append $\mathbf{u}_j$ to U
  Append $\mathbf{v}_j$ to V
end while

In SP1M, the cluster centers are not initialized purely randomly. They are initialized from probabilities based on the typicalities of the previously found clusters. The initial cluster centers are picked from dataset X with probabilities

$$p(\mathbf{x}_i) = \begin{cases} \frac{1}{n} & \text{if j = 1} \\ 0 & \text{if } \max\limits_{k=1,\ldots,j} u_{ki} > 0.5 \\ \frac{1 - \max_{k=1,\ldots,j} u_{ki}}{N - \sum_{s=1}^{N} \max_{k=1,\ldots,j} u_{ks}} & \text{otherwise} \end{cases} \tag{6}$$

The returned matrix U from the SP1M is the typicality matrix that measures how "typical" a particular point is to each cluster, that is, how close the point is to each cluster prototype. Outliers have a large distance to all the existing clusters so that they naturally have low typicality to all clusters. SP1M open source MATLAB code: https://github.com/waylongo/sp1m-de.

IV. Possibilistic Clustering Informed ANFIS

A. Initialization

Herein, we do not rely on random ANFIS parameter initialization. ANFIS is a supervised learning algorithm. First, the number of underlying clusters must be determined from the data (we will create one rule for each underlying cluster). Next, user-specified membership functions are fit to each cluster. In the case of a Gaussian, we calculate the respective mean and standard deviation. In the case of a trapezoidal membership function, we set the core to three standard deviations and the support to five standard deviations. The point is, the user has flexibility over what function to use and how to best fit the antecedent clause membership functions to the data. Next, once the membership functions are estimated for each cluster, we use the *least means squared* (LMS), like Jang [2], to estimate the ANFIS consequent weights (p_k^r). Note, in traditional ANFIS the user selects the number of rules and antecedents per rule. In our possibilistic clustering-based ANFIS, SP1M informs us about the number of underlying clusters, which is used to pick the number of rules.

B. Method 1: Pre-Processing or Data Filtering

Our first proposal is to use the possibilistic clustering algorithm results to pre-process the training data. Training data typicalities are generated using the SP1M algorithm. These typicalities are then analyzed, and any typicality below a user defined threshold, Γ, are removed from the training data. Specifically, we take the max typicality of a data point to all clusters, $t_i = \max_{j=1,\ldots,c} u_{ji}$, i.e., the *strongest* degree that it belongs to any cluster. This process has numerous benefits. First, there are fewer data points, which lets an algorithm train faster. Second, there are fewer outliers, which improves initialization and helps us combat challenges like non-decreasing uncertainty. However, this procedure is crisp in the fact that it partitions data into use/not use, versus utilizing the degree to which a data point is an outlier. Furthermore, results will likely vary based on parameter Γ.

C. Method 2: Gradient Scaling

Our second approach is to use SP1M typicalities to modify the ANFIS algorithm itself, not the data presented to it. Specifically, we modify the learning algorithm. Herein, gradient descent is used to optimize ANFIS. The criteria function used is the *sum of squared error* (SSE); which leads to the derivatives in Table II. We first summarize our data point typicalities according to their max typicality, $t_i = \max_{j=1,\ldots,c} u_{ji} \in [0, 1]$. Next, we scale the partials, i.e.,

$$\left[\frac{\partial y_n}{\partial z_n^r}\right]_{\text{scaled}} = t_i \frac{\partial y_n}{\partial z_n^r}. \tag{7}$$

Thus, if a sample has a *high degree* of belonging to a cluster then ANFIS operates "as is". However, outliers have a *low* typicality, which dampens their impact on learning. Note, as this is uniform scaling, i.e., all gradients are scaled the same way, it does not change the direction of the gradient, just its magnitude. In summary, Method 2 is different from Method 1 as it utilizes the typicality information to scale gradients during learning, rather than partition data during pre-processing.

as it instead listens to the data with respect to their individual degrees of importance.

V. Preliminary Experiments and Results

In this section, we demonstrate a set of controlled experiments to show the before and after effects of clustering informed versus traditional ANFIS. Synthetic data is used because we can control the conditions and range and we know the answer. We use two inputs/dimensions because the results can be visualized. Furthermore, we focus on the quality of the results. ANFIS, as is, can be used to achieve minimum error relative to a user specified criteria function. By quality, rather than accuracy or an index like F1 score, we specifically mean generating fuzzy sets and rules that fit the data well. For our synthetic experiments, this quality will be evident in how well the rules fit the underlying clusters and ignore noise, since we can use this information to present to a decision maker or use for a purpose such as *eXplainable artificial intelligence*.

A. Experiment 1: Low Amount of Noise

In Experiment 1, we focus on a dataset that should have five rules, which is intended to model a *few* close rules whose noise can influence each other. Furthermore, five rules were selected because it can be visualized; i.e., there are not too many points and overlaid resulting clustering information to inhibit the readers viewing and understanding. Each cluster has 300 samples and 10% of the data is noise. Specifically, noise samples are generated beyond four standard deviations of the underlying generative Gaussian clouds. The five class (rule) centers are $[0.080, 0.501]$, $[0.074, 0.680]$, $[0.496, 0.077]$, $[0.918, 0.679]$, $[1.151, 0.903]$, with corresponding x and y dimension standard deviations of $[0.033, 0.0256]$, $[0.0353, 0.0193]$, $[0.0397, 0.0148]$, $[0.0384, 0.0091]$, $[0.0251, 0.0272]$. It is worth noting that we selected close, but not overlapped clusters because in a real scenario if two clusters/rules overlap, it makes less sense. Meaning, two rules with similar IF but different THEN counterparts. A single data point could potentially belong to multiple different rules. Such a scenario is semantically confusing. The number of features may be inadequate to properly separate rules, or something else may be causing issues. Figure 3 shows Experiment 1.

Figure 3 shows the result of traditional ANFIS, clustering-based pre-processing, and gradient scaled ANFIS. Note, in traditional ANFIS one has to either select or engage in some external method to pick R. Herein, we use SP1M to select R, which provides ANFIS more benefit than what the core algorithm affords. Clearly, clustering-based pre-processing produces better rule structures. Namely, the core (membership value one) region is a tight fit to the underlying data and the support (membership value greater than zero) has a reasonable footprint; specifically, it does not include all of the noise. On the other hand, traditional and gradient scaled ANFIS have over inflated cores and support regions. Furthermore, both have a problem with rule placement for the two close clusters (green and bluegray). Overall, typicality-based pre-processing yields a good fit of the underlying data, making the resulting fuzzy sets more faithful and interpretable descriptions of the data.

B. Experiment 2: Moderate Amount of Noise

In Experiment 2, we keep the same dataset, allowing Experiment 2 to be compared to Experiment 1. The only thing that has changed is that 25% of each cluster's data is now noise, compared to 10% in Experiment 1. The reason for Experiment 2 is to observe the impact and behavior of the proposed approaches in light of a greater amount of noise. Figure 4 shows the dataset.

Figure 4 also displays the result of the three methods. The take away is as follows. As expected, an increased level of noise impacts ANFIS, but the power of possibilistic clustering is able to address and overcome shortcomings, namely typicality-based pre-processing.

C. Additional Experiments

Figure 5 shows nine additional arbitrary ANFIS experiments with varying numbers of R, C, and cluster overlap. The reason for these mini experiments is to give the reader a feel for performance in different contexts. As can be seen, typicality weighted ANFIS is more resilient and its largest shortcoming is if SP1M can estimate the true underlying generative cluster structure. While we have only demonstrated results for two dimensions and a few clusters ($2 \leq C \leq 10$), the reader can download and experiment with our open source codes, https://github.com/Blake-Ruprecht/Fuzzy-Fusion and https://github.com/Blake-Ruprecht/ANFIS-SP1M.

VI. Insights and Summary

Experiments 1 and 2 highlight the benefit of using possibilistic typicality degrees in neuro fuzzy logic. This allows us to combat a number of underlying qualitative concerns about a learned neuro fuzzy logic solution. However, it was our expectation that Method 2, gradient scaling, would be the

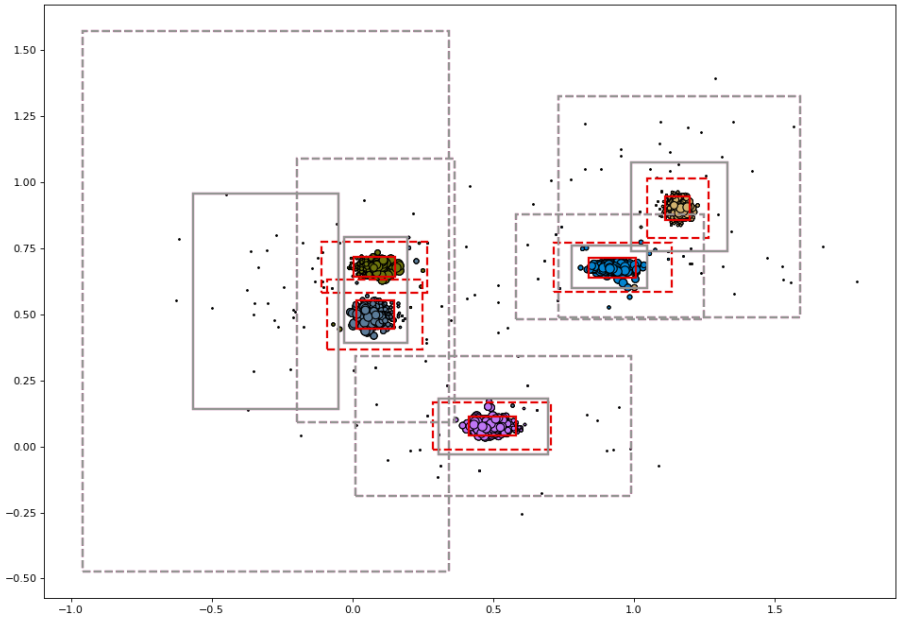


Fig. 3: Experiment 1. Each class is color coded. The traditional and gradient scaled ANFIS trapezoidal membership functions are shown in grey. Solid is the core and dashed is support. Red solid and dashed lines are clustering-based pre-processed ANFIS. The data points are scaled based on typicality (note that the size has a lower bound to prevent points from disappearing).

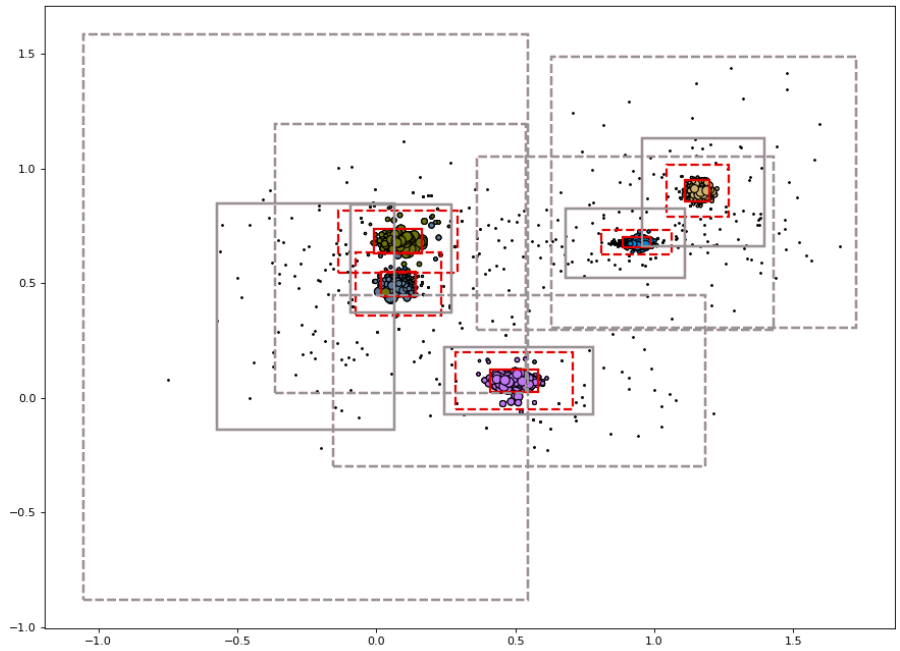


Fig. 4: Experiment 2. Each class is color coded. The traditional and gradient scaled ANFIS trapezoidal membership functions are shown in grey. Solid is the core and dashed is support. Red solid and dashed lines are clustering-based pre-processed ANFIS. The data points are scaled based on typicality (note that the size has a lower bound to prevent points from disappearing).

top performer. Because this is not the case, we dug deeper into ANFIS. Consider the update equation for a membership function parameter,

$$\frac{\partial y_n}{\partial \Theta_m^{r,k}} = \left(\frac{\sum_{j=1, j\neq r}^{R} w_n^j (z_n^r - z_n^j)}{(\sum_{i=1}^{R} w_n^i)^2} \right) \times \tag{8a}$$

$$\left(\prod_{j=1, j\neq k}^{K} A_j^r(\mathbf{x}_n(j)) \right) \times \tag{8b}$$

$$\left(\frac{\partial A_k^r(\cdot)}{\partial \Theta_m^{r,k}} \right). \tag{8c}$$

Consider the following two scenarios.

Proposition 1. Term 1 (Equation 8a) results in a zero magnitude gradient (Equation 8) for rule r when it is the only rule that fires, i.e., $w_n^r > 0$, and $\forall j \in \{1, ..., R\}, j \neq r, w_n^j = 0$.

Proof. The numerator in Equation 8a is all rules other than r. As each of these rules have $w_n^j = 0$, the numerator is zero and the denominator is not. Thus, Equation 8 is zero. □

Proposition 1 can be interpreted as diminishing the gradient when the rules are properly separated with no overlap. This means that the conditions for Proposition 1 are constantly being met each time a data point that belongs to rule r is being trained on. This prevents learning from occurring for the rule membership parameters that the training data matches with. This is a drastic result if the goal is to make the distribution fit the underlying data.

Proposition 2. Term 2 (Equation 8b) results in a gradient (Equation 8) whose magnitude is 0 when a data sample (n) is not in a rule (r), i.e., $w_n^r = 0$.

Proof. If data sample n is not covered by rule r, then $A_j^r(\mathbf{x}_n(j)) = 0$. As each of these terms are zero, their product (and other t-norms at that, e.g., the minimum) is zero. As a result, Equation 8 is zero. □

Proposition 2 has a number of ramifications. For example, if a data point does not result in a rule firing strength greater than zero, then that rule is not updated. While by itself, this does not seem overly alarming, consider the case of a rule initialized to a region corresponding to no data points. The rule will never get updated. This is the definition of "once dead, always dead."

These two propositions highlight two common cases. More scenarios that give rise to diminishing-to-dead gradients can be identified. The point is, it is clear why our typicality weighted and initialized procedure performs best. Until factors like these are remedied with ANFIS, a procedure like gradient scaling, while elegant in design, is rendered ineffective.

VII. Conclusion and Future Work

In this article we explored the role of possibilistic clustering to generate data point typicality degrees to improve challenges in ANFIS, a neuro fuzzy logic learning tool. Specifically, we used SP1M, which helps us combat coincident clusters in PCM, and it allows one to support, if desired, realtime, online learning and/or Big Data. We explored two possible methods, pre-processing data by removing SP1M identified outliers, and gradient scaling during ANFIS learning utilizing SP1M typicalities. Pre-processing had the best results, namely due to a diminishing/dead gradient shortcoming in ANFIS.

In future work, we will take this preliminary investigation and explore real-world applications of neuro fuzzy logic. We will also find a way to remedy the diminishing/dead gradient problem in ANFIS, which should make gradient scaling the top performer. Other future work will include taking definitions of what constitutes a "good logic explanation", and folding that into the learning algorithms to promote better explanations. Last, once our work related to understanding shallow neuro fuzzy logic networks is mature, we will open our processes up to deep inference nets.

References

[1] Judea Pearl and Dana Mackenzie, *The Book of Why: The New Science of Cause and Effect*, Basic Books, Inc., USA, 1st edition, 2018.

[2] J. . R. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE Transactions on SMC*, vol. 23, no. 3, pp. 665–685, May 1993.

[3] James M. Keller and Ronald R. Yager, "Fuzzy Logic Inference Neural Networks," in *Intelligent Robots and Computer Vision VIII: Algorithms and Techniques*, David P. Casasent, Ed. 1990, vol. 1192, pp. 582 – 591, SPIE.

[4] James M. Keller and Hossein Tahani, "Implementation of conjunctive and disjunctive fuzzy logic rules with neural networks," *International Journal of Approximate Reasoning*, vol. 6, no. 2, pp. 221 – 240, 1992.

[5] Sankar K. Pal and Sushmita Mitra, *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing*, John Wiley Sons, Inc., USA, 1st edition, 1999.

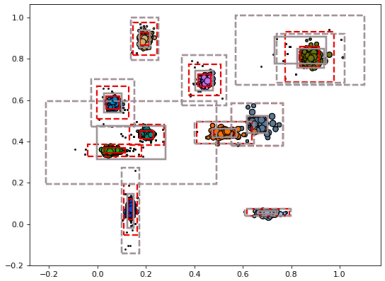
[6] S. Rajurkar and N. K. Verma, "Developing deep fuzzy network with takagi sugeno fuzzy inference system," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2017, pp. 1–6.

[7] Blake Ruprecht, Charlie Veal, Bryce Murray, Muhammad Aminul Islam, Derek Anderson, Fred Petry, James Keller, Grant Scott, and Curt Davis, "Fuzzy logic-based fusion of deep learners in remote sensing," New Orleans, USA, 2019, FUZZ-IEEE, Late Breaking Research and Poster Presentation.

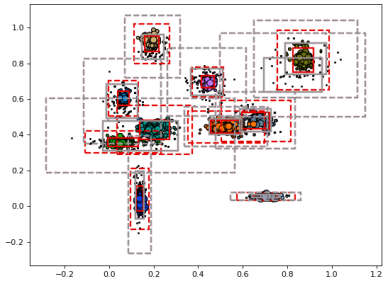
[8] M. A. Islam, D. T. Anderson, A. Pinar, T. C. Havens, G. Scott, and J. M. Keller, "Enabling explainable fusion in deep learning with fuzzy integral neural networks," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2019.

[9] G. J. Scott, K. C. Hagan, R. A. Marcum, J. A. Hurt, D. T. Anderson, and C. H. Davis, "Enhanced fusion of deep neural networks for classification of benchmark high-resolution image data sets," *IEEE GRSL*, vol. 15, no. 9, pp. 1451–1455, Sep. 2018.

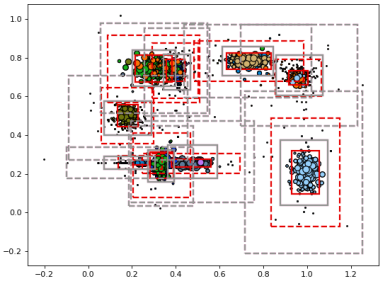
[10] S. Price, S. Price, and D. Anderson, "Introducing fuzzy layers for deep learning," in *FUZZ-IEEE*, June 2019.



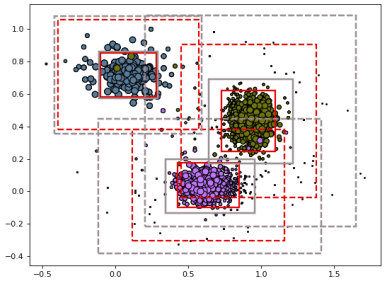
(a) 10% Noise, C=10, R=10.
All rules discovered well by extended ANFIS.



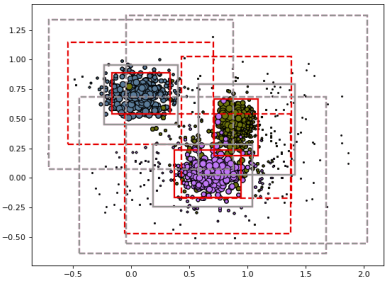
(b) 40% Noise, C=10, R=10.
All rules discovered well by extended ANFIS.



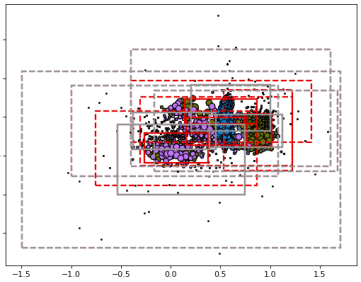
(c) 10% Noise, C=10, R=9.
Less rules than clusters and high degree of overlap.



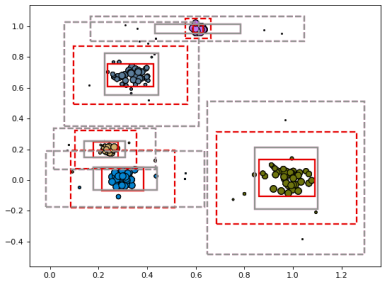
(d) 10% Noise, C=3, R=3.
All rules discovered well by extended ANFIS.



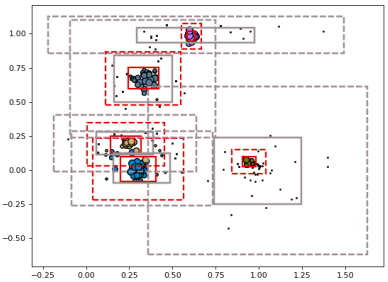
(e) 30% Noise, C=3, R=3.
All rules discovered well by extended ANFIS.



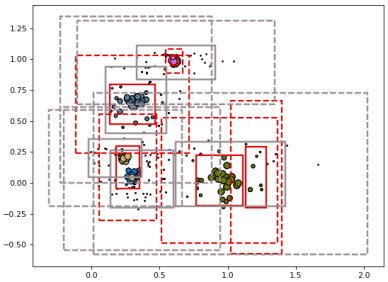
(f) 5% Noise, C=3, R=4.
More rules than clusters and high degree of overlap.



(g) 10% Noise, C=5, R=5.
All rules discovered well by extended ANFIS.



(h) 30% Noise, C=5, R=5.
All rules discovered well by extended ANFIS.



(i) 60% Noise, C=5, R=5.
Some rules in wrong place.

Fig. 5: Additional experiments that show variety and ANFIS behavior in a range of contexts. The goal is a tight fit of boxes to data. C is number of underlying clusters, R is number SP1M found. (a)-(c) have 1500 data points. (a) and (b) have the same means. (d)-(f) have 1500 data points. (d) and (e) have the same means. (g)-(i) have 250 data points and the same means.

[11] C. Veal, A. Yang, A. Hurt, M. Islam, D. Anderson, G. Scott, J. Keller, T. Havens, and B. Tang, "Linear order statistic neuron," in *FUZZ-IEEE*, June 2019.

[12] Michio Sugeno, *Industrial Applications of Fuzzy Control*, Elsevier Science Inc., New York, NY, USA, 1985.

[13] Blake Ruprecht, Charlie Veal, Al Cannaday, Derek T. Anderson, Fred Petry, James Keller, Grant Scott, Curt Davis, Charles Norsworthy, Kristen Nock, and Elizabeth Gilmour, "Neuro-fuzzy logic for parts-based reasoning about complex scenes in remotely sensed data," in *Signal Processing, Sensor/Information Fusion, and Target Recognition XXIX*, Ivan Kadar, Erik P. Blasch, and Lynne L. Grewe, Eds. International Society for Optics and Photonics, 2020, vol. 11423, pp. 45 – 52, SPIE.

[14] Wenlong Wu, James M. Keller, and Thomas A. Runkler, "Sequential possibilistic one-means clustering with dynamic eta," in *FUZZ-IEEE*, 2018, pp. 1–8.

[15] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *Trans. Fuz Sys.*, vol. 1, no. 2, pp. 98–110, May 1993.

[16] James Bezdek, Robert Ehrlich, and William Full, "Fcm—the fuzzy c-means clustering-algorithm," *Computers Geosciences*, vol. 10, pp. 191–203, 12 1984.