EXPLAINABLE PARTS-BASED CONCEPT MODELING AND REASONING

A Thesis presented to
the Faculty of the Graduate School
at the University of Missouri

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

---

by
BLAKE RUPRECHT
Dr. Derek T. Anderson, Thesis Supervisor
DECEMBER 2022

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

EXPLAINABLE PARTS-BASED CONCEPT MODELING AND REASONING

presented by Blake Ruprecht,

a candidate for the degree of Master of Science in Computer Science,

and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Derek T. Anderson

Professor James M. Keller

Professor Mihail Popescu

Professor Grant J. Scott

DEDICATION

I dedicate this thesis to my family and friends. All of the work I've done the past few years has externally been an attempt to improve the existential future for humanity, but really, I just wanted to do something to bring me closer to you all. I'm forever grateful to my Mom and Dad, who have been my biggest supporters since before I can remember. I know I'll always have you two in my corner, pushing me to go farther than I could ever do on my own. Thank you to my sisters and my barnyard buddies for being the best friends I could ever ask for. Without you, I wouldn't have a future to fight for. I hope that we can eventually teach robots what it means to be loved so that they don't destroy us all, but if not it's been a hell of a ride. Cheers to a peaceful future!

ACKNOWLEDGEMENTS

I would like to give heaps of thanks to my advisor, Dr. Anderson. Without his patience and support, my graduate career would not have been possible, and I'd still be sitting in an armchair philosophizing, doing nothing! I came from the lands of thermofluids and dynamics into the world of bits and linear algebra, and couldn't have done it without the support of the Mindful lab. I've learned from everyone here, especially in my early days of computing. Charlie, I got Arch up and running, I'll never reach your level but you've inspired me to learn more about computers than any class has ever taught me. Bryce, you helped me realize that research is just as much about how much fun you have doing it as the end result. Drew, I couldn't have done half the things I did without your help and coding expertise, you are an absolute wizard behind the keys. I'll forever be grateful to all of my peers who helped me learn to code, think algorithmically, and appreciate the wonders of computation. Special thanks to Drs. Keller, Petry, Michael, Scott, Islam, Cannaday, and Davis for collaborating with me on research papers. Without your support, I would have never finished a single paper. Special thanks to Dr. Keller and Dr. Buck for laying the foundation of this thesis, nothing here would be possible without you guys. Thank you for the collaboration, passionate discussions, wild ideas, and for grounding my work the past few years. I had a difficult time using the word "I" in this thesis because I did none of this alone – without the support of the lab, my collaborators, and especially Dr. Anderson, I would still be all mechanical with no computer for guidance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

EXPLAINABLE PARTS-BASED CONCEPT MODELING AND REASONING

Blake Ruprecht

Dr. Derek T. Anderson, Thesis Supervisor

ABSTRACT

State-of-the-art artificial intelligence (AI) learning algorithms heavily rely on deep learning methods that exploit correlation between inputs and outputs. While effective, these methods typically provide little insight to the reasoning process used by the machine, which makes it difficult for human users to understand the process, trust the decisions made by the system, and control emergent behaviors in the system. One method to fix this is eXplainable AI (XAI), which aims to create algorithms that perform well while also providing explanations to users about the reasoning process to mitigate the problems outlined above. In this thesis, I focus on advancing the research around XAI techniques by introducing systems that provide explanations through the use of parts-based concept modeling and reasoning. Instead of correlating input to output, I correlate input to sub-parts or features of the overall concept being learned by the system. These features are used to model and reason about a concept using an explicitly defined structure. These structures provide explanations to the user by nature of how they are defined. Specifically, I introduce a shallow and deep Adaptive Neuro-Fuzzy Inference System (ANFIS) that can reason in noisy and uncertain contexts. ANFIS provides explanations in the form of learned rules that combine features to determine the overall output concept. I apply this system to real geospatial parts-based reasoning problems and evaluate the performance and explainability of the algorithm. I discover some drawbacks to the ANFIS system as traditionally defined due to dead and diminishing gradients. This leads me to focus on how to model parts-based concepts and their inherent uncertainty in other ways, namely through Spatially Attributed Relation Graphs (SARGs). I incorporate human feedback to refine the machine learning of concepts using SARGs. Finally, I present future directions for research to build on the progress presented in this thesis.

Chapter 1

INTRODUCTION

State-of-the-art *Artificial Intelligence* (AI), e.g. *Deep Learning* (DL), is deeply rooted in the exploitation of correlation. However, a major drawback to current DL approaches is we do not explicitly know what they have learned; leading to the term "black box". While we often assign meaning to the inputs and outputs of the system, we are not usually able to interpret and/or explain what happens "in-between". What chain of evidence went into a decision? What biases are in our data and solution? To what degree is our system confident in its decision? Many domains such as healthcare and public policy require explanations to justify a particular decision or course of action. Without these explanations, humans don't understand the decisions made by AI as well, they trust the AI system less, and there is more potential for unintended/emergent behavior [1] [2]. The competency of AI systems will continue to increase as it is deployed into the real world more and more. Long-term, the hope is for AI systems to continue behaving in ways that are in alignment with human values [3]

Creating safe AI is challenging, and it has led to the rise of a sub field of so-called *eXplainable AI* (XAI) [4]. XAI aims to create machine learning algorithms that maintain a high level of performance while producing better explanations that allow a human to understand, reasonably trust, and manage the emergent development of the AI system. One specific approach to XAI involves explicitly defining the meaning of mathematical terms one step before the output of an AI system. Instead of black box systems that learn output "concepts" (the meaning applied to the output as defined by humans) through correlation to inputs, this approach breaks up the output concept into explicitly defined sub-parts or features, which enables the system to correlate input to parts, and parts to output. In this way, the system models the output concept using parts, and reasons over these parts to determine the output concept. While this approach doesn't completely eliminate the ambiguity of black boxes, it adds an extra layer of explainability to the system.

While explainable parts-based concept modeling and reasoning can be used to help address pressing XAI concerns, it is not a solved problem. In this thesis, I focus on advancing this topic in three steps. Step 1 (outlined in Chapter 2) involves a new shallow and deep neuro-fuzzy logic system that can reason in noisy and uncertain contexts. Step 2 (Chapter 3) is the application of this approach to a real geospatial parts-based reasoning problem. Last, step 3 (Chapter 4) is a focus on how to model parts-based concepts and their inherent uncertainty. The next three sub-sections summarize these three challenges.

1.1 Robust shallow and deep neuro-fuzzy logic

DL approaches that exploit correlation perform well on many tasks [5]. A challenge of building XAI systems is utilizing DL techniques to exploit the performance advantages while incorporating explicitly defined techniques to improve explainability. One technique to do this is neuro-fuzzy logic [6], which combines the performance of neural networks with the explainability of fuzzy logic. Specifically, I investigated the type-1 *Takagi-Sugeno-Kang* (TSK) *adaptive neuro-fuzzy inference system* (ANFIS) [7] [8]. It is well accepted that many neural networks and fuzzy logic systems are universal function approximators [9]. A hope is that a neuro-fuzzy logic system can take advantage of neural learning algorithms and the explanation potential of fuzzy logic.

Specifically, in this thesis I address neuro-fuzzy logic using ANFIS in shallow and deep contexts with respect to incomplete and noisy contexts. An open source PyTorch library (https://github.com/Blake-Ruprecht/Fuzzy-Fusion [8]) was produced and shared for gradient descent (GD) based optimization of ANFIS. In addition, we discovered that GD-based ANFIS is not naturally robust. To address this problem, I utilized [10] the sequential possibilistic one mean (SPIM)[11]. My proposed solution leads to rules that are more compact and associated with trends in the data versus noise in the input and/or concepts. Furthermore, in [10] I showed vulnerabilities in GD-based ANFIS learning. Specifically, I showed that, as is, GD-based neuro-fuzzy logic optimization leads to XAI problems with duplicate rules, dead rules, bloated rules, and beyond. See Chapter 2 for further details.

1.2 ANFIS applied to geospatial parts-based task

Black-box style architectures like Convolution Neural Networks (CNNs) currently dominate performance on geospatial reasoning tasks [12]. The challenge of this work was to see if we could use a neuro-fuzzy logic system to recognize a complex scene with respect to its parts. The hope was

to show that not only could a solution be learned, and hopefully better than what a CNN could learn, but could we learn "logic" in the process and be able to shed light on what's going on. Ideally, the neuro-fuzzy logic system will be explainable enough to provide insight about the steps used to determine overall scene output.

In this task, I built on top of the research of Cannaday and Davis [13]. Specifically, Cannaday and Davis used Deep Neural Networks (DNNs) to detect component objects part of a larger, more complex scene. They then used different techniques to fuse the components together to determine overall output. In this work, I used DNNs to provide object-level confidences of spatial components. I fed these feature confidences into a ANFIS node in the hopes of using ANFIS rules to explain the logical decision making of the fusion system. A great feature of ANFIS is the explicit "IF-THEN" inference structure used to determine overall scene output. An example rule would be "IF Object A presence is HIGH, and Object B presence is HIGH, THEN scene X presence is HIGH." Each ANFIS node can contain multiple of these "IF-THEN" rules, constituting the entire rule base of that node. Each rule can contain as many conjuctions as needed to model the logical information required to determine overall ANFIS node firing.

A major benefit of ANFIS is the ability to encode logical rules as discussed previously. In this thesis, I found that expert knowledge in the form of "IF-THEN" rules can be embedded into an ANFIS node, effectively encoding the expert knowledge. Next, I found that ANFIS utilizing SP1M clustering could learn rule-bases from scratch, though the interpretability suffered from a dead/diminishing gradient problem that I discuss further in Chapter 3. Finally, I found that the best performing and most interpretable solution was using ANFIS to augment the encoded human rules, which helped avoid the dead/diminishing gradient while improving upon the human rules using linear regression learning. See Chapter 3 for further details.

1.3 Concept learning incorporating human feedback

After exploring the mathematical underpinnings of ANFIS and exploring it for a geospatial case study, it became apparent that such an approach is primarily focused on parts-based reasoning. But what was the concept that the model was learning? In this third part of my thesis [14] (see Chapter 4), I focused on parts-based modeling. Specifically, explicit focus was given to parts and modeling their uncertainty with respect to an Attributed Relation Graph (ARG) [15]. Concept learning is very similar to parts-based reasoning as discussed previously, but concept learning expands upon the features within a scene to also include relationships between features within a scene. We move away

from ANFIS and towards more general graph representation of features and relationships in the form of *Attributed Relation Graphs* (ARG). We maintain the parts-based complex scenes that motivated [16], but incorporate spatial relationships in the form of *Spatial* ARGs (SARG).

For example, consider the simple but effective example of learning a human face (a concept). A face consists of eyes, a nose, and mouth, or parts. These parts have attributes, e.g., size, shape, etc. These parts also have spatial relations relative to one another. A concept, e.g., human face, can take on a range of possible value assignments therein. In this part of my thesis, a spatial ARG (SARG) [17] was the center of attention. This was chosen because it relates to our earlier application of geospatial parts-based recognition.

Specifically, in this third challenge I focused on a new representation that enables incorporating human feedback into the refinement of machine learning of concepts. Histograms of forces [18], a form of type-1 fuzzy set, were extracted to model relative spatial relations. I show how features and relationships are used to model a concept, how concepts are learned from data, and how to compare concepts across SARGs. The SARG acts as a shared language that facilitates human interaction, enabling online concept learning of a SARG.

In the remainder of this thesis, Chapter 2 presents my implementation of robust shallow and deep neuro-fuzzy logic utilizing possibilistic clustering. Chapter 3 details this implementation applied to geospatial parts-based tasks. Chapter 4 explores a new form of concept learning utilizing SARGs that incorporate human feedback, and potential future routes for XAI research.

Chapter 2

POSSIBILISTIC CLUSTERING ENABLED NEURO FUZZY LOGIC

Blake Ruprecht[a], Wenlong Wu[a], Muhammad Aminul Islam[a], Derek T. Anderson[a], James Keller[a], Grant Scott[a,b], Curt Davis[a,b], Fred Petry[c], Paul Elmore[d], Kristen Nock[e], Elizabeth Gilmore[e],

[a] Dept. of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO

[b] Center for Geospatial Intelligence, University of Missouri, Columbia, MO

[c] U.S. Naval Research Laboratory, Stennis Space Center, MS

[d] Johns Hopkins Applied Physics Laboratory, Laurel, MD

[e] U.S. Naval Research Laboratory, Washington, DC

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.

2.1 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 causation [19]. 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) [7]. 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 [20], 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 [20], 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.

Table 2.1: ANFIS 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

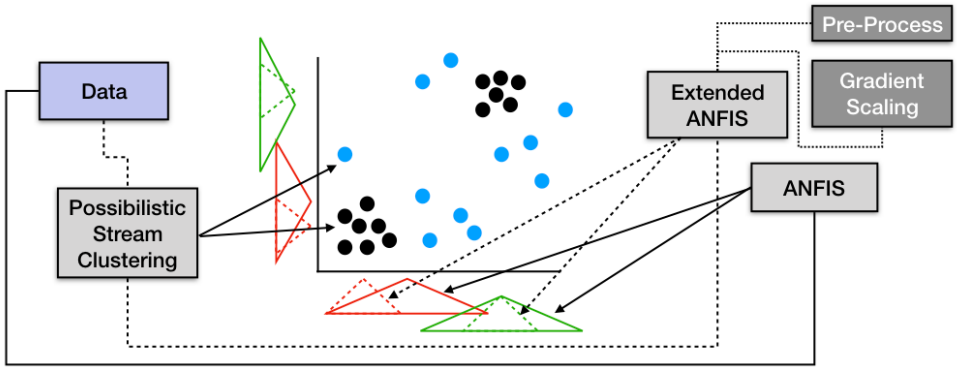


Figure 2.1: High-level overview of the proposed work. Traditionally, ANFIS is applied to the raw full data. Here, possibilistic clustering is used to acquire data typicality degrees, which are fed to ANFIS during learning. Blue dots denote noise points (outliers) and black dots belong to a cluster. Red (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.

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.

Before we delve into ANFIS and our extensions, it is important to note that ANFIS is not the only neuro-fuzzy architecture. In [20], Keller and Yager proposed a *multi layer perceptron* (MLP) to learn fuzzy logic. In [21], Keller and Tahani discussed the implementation of a conjunctive and disjunctive fuzzy logic rules with neural networks. In [22], Pal and Mitra explored an array of topics on neuro-fuzzy related to pattern recognition. In [23], Rajurkar and Verma put forth a deep fuzzy network with Takagi Sugeno fuzzy inference system. In [8], 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 [24], fuzzy integrals for fusing heterogeneous architecture deep learners in remote sensing [25], fuzzy layers in deep learning [26], ordered weighted average networks [27], 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 2.2 we review ANFIS, Section 2.3 is a possibilistic clustering approach, Section 2.4 discusses typicality extended ANFIS, and Section 2.5 is experiments and results. Table 2.1 is a summary of acronyms and notation and Figure 2.1 illustrates the structure of our paper.

2.2 ANFIS

In [7], Jang introduced ANFIS, which was initially based on TSK type fuzzy inference [28]. An illustrative overview and input-output depiction of ANFIS can be seen in Figure 2.2. 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,

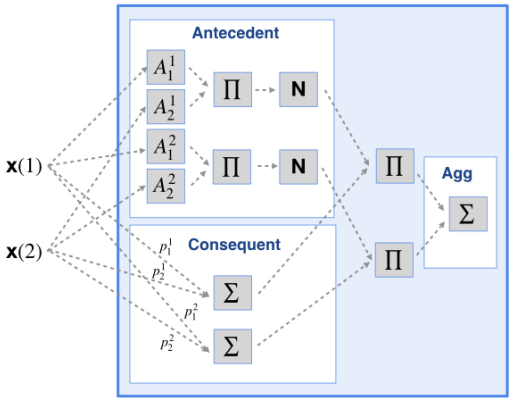


Figure 2.2: This figure illustrates the flow of data in a first order TSK ANFIS for the case of two inputs and two rules.

$A_k^r(\cdot)$, of each input feature, i.e.,

$$w_n^r = \prod_{k=1}^{K} A_k^r(\mathbf{x}_n(k)). \tag{2.1}$$

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.,

$$z_n^r = \left(\sum_{k=1}^{K} \mathbf{x}_n(k) p_k^r \right) + p_{bias}^r. \tag{2.2}$$

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.,

$$y_n = \frac{\sum_{r=1}^{R} z_n^r w_n^r}{\sum_{r=1}^{R} w_n^r}. \tag{2.3}$$

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.

2.2.1 Semantic Considerations

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

$$N(\mathbf{x}_n(k);\mu_k,\sigma_k) = e^{\left(-\frac{1}{2}\frac{(\mathbf{x}_n(k)-\mu_k)^2}{\sigma_k^2}\right)}. \tag{2.4}$$

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,

$$T(\mathbf{x}_n(k);\boldsymbol{\Theta}) = \begin{cases} 0 & (\mathbf{x}_n(k) < \Theta_1^k) \text{ or } (\mathbf{x}_n(k) > \Theta_4^k) \\ 1 & \Theta_2^k \leq \mathbf{x}_n(k) \leq \Theta_3^k \\ \frac{\mathbf{x}_n(k)-\Theta_1^k}{\Theta_2^k-\Theta_1^k} & \Theta_1^k \leq \mathbf{x}_n(k) < \Theta_2^k \\ \frac{\Theta_4^k-\mathbf{x}_n(k)}{\Theta_4^k-\Theta_3^k} & \Theta_3^k < \mathbf{x}_n(k) \leq \Theta_4^k. \end{cases} \tag{2.5}$$

2.2.2 ANFIS Optimization

For sake of article completeness, Table 2.2 are the partial derivatives for ANFIS, based on a first order TSK model. The reader can refer to [7], [8], and [16] for full mathematical explanation and derivations.

2.2.3 Open Source Codes

For reproducible research, in [8] 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.

2.2.4 Limitation of Traditional ANFIS

ANFIS [7] 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.

Table 2.2: ANFIS Derivatives for Gradient Descent

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

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.

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.

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.

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.

2.3 Sequential Possibilistic One Means (SP1M)

The *sequential possibilistic one means* (SP1M) algorithm [11] is based on the *possibilistic c-means* (PCM) algorithm [29]. The PCM abandons the membership sum-to-one constraint in the *fuzzy c-means* (FCM) [30] 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 [11].

In SP1M, the cluster centers are not initialized purely randomly. They are initialized from

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. 2.6
    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

probabilities based on the typicalities of the previously found clusters. The initial cluster centers are picked from dataset X with probabilities

$$p(\mathbf{x}_i) = \begin{cases} \frac{1}{n} & \text{if j = 1} \\ 0 & \text{if } \max\limits_{k=1,\ldots,j} u_{ki} > 0.5 \\ \frac{1 - \max_{k=1,\ldots,j} u_{ki}}{N - \sum_{s=1}^{N} \max_{k=1,\ldots,j} u_{ks}} & \text{otherwise} \end{cases} \tag{2.6}$$

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.

2.4 Possibilistic Clustering Informed ANFIS

2.4.1 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 [7], 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.

2.4.2 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.

2.4.3 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 2.2. 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.,

$$\left[\frac{\partial y_n}{\partial z_n^r}\right]_{\text{scaled}} = t_i \frac{\partial y_n}{\partial z_n^r}. \tag{2.7}$$

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.

2.5 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*.

2.5.1 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 2.3 shows Experiment 1.

Figure 2.3 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

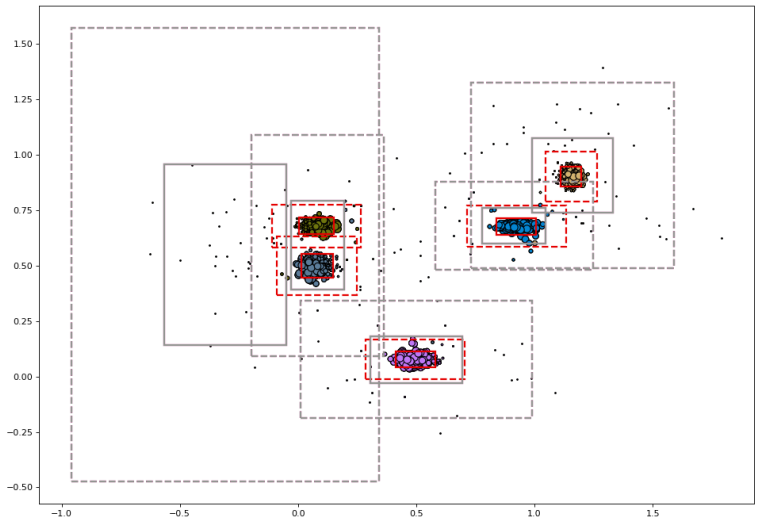


Figure 2.3: Experiment 1. Each class is color coded. The traditional and gradient scaled ANFIS trapezoidal membership functions are shown in grey. Solid is the core and dashed is support. Red solid and dashed lines are clustering-based pre-processed ANFIS. The data points are scaled based on typicality (note that the size has a lower bound to prevent points from disappearing).

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.

2.5.2 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 2.4 shows the dataset.

Figure 2.4 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.

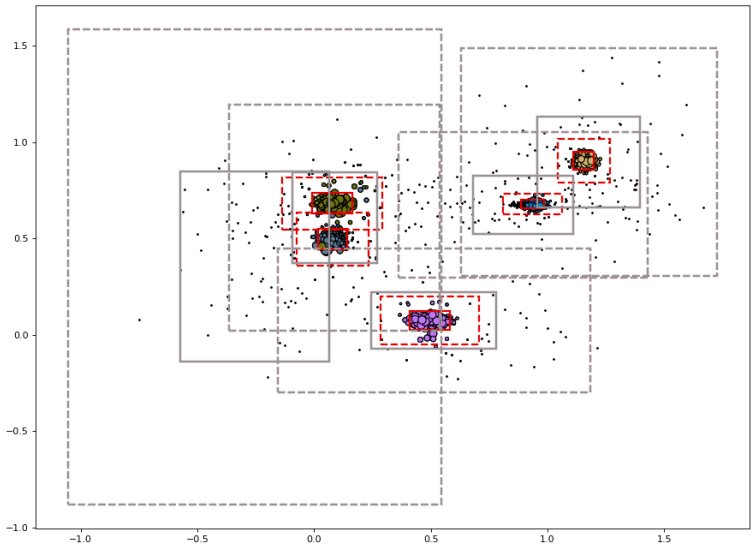


Figure 2.4: Experiment 2. Each class is color coded. The traditional and gradient scaled ANFIS trapezoidal membership functions are shown in grey. Solid is the core and dashed is support. Red solid and dashed lines are clustering-based pre-processed ANFIS. The data points are scaled based on typicality (note that the size has a lower bound to prevent points from disappearing).

2.5.3 Additional Experiments

Figure 2.5 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.

2.6 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 top

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

$$\frac{\partial y_n}{\partial \Theta_m^{r,k}} = \left(\frac{\sum_{j=1, j\neq r}^{R} w_n^j (z_n^r - z_n^j)}{(\sum_{i=1}^{R} w_n^i)^2} \right) \times \tag{2.8a}$$

$$\left(\prod_{j=1, j\neq k}^{K} A_j^r(\mathbf{x}_n(j)) \right) \times \tag{2.8b}$$

$$\left(\frac{\partial A_k^r(\cdot)}{\partial \Theta_m^{r,k}} \right). \tag{2.8c}$$

Consider the following two scenarios.

Proposition 1. *Term 1 (Equation 2.8a) results in a zero magnitude gradient (Equation 2.8) 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 2.8a 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 2.8 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 2.8b) results in a gradient (Equation 2.8) 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 2.8 is zero. □

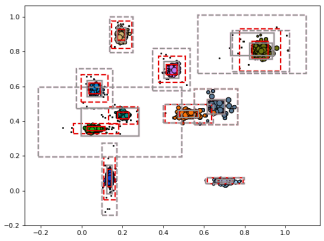
Proposition 2 has a number of ramifications. For example, if a data point does not result in a rule firing strength greater than zero, then that rule is not updated. While by itself, this does not seem overly alarming, consider the case of a rule initialized to a region corresponding to no data points. 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.

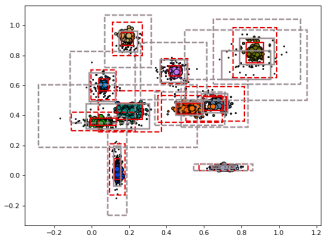
2.7 Conclusion and Future Work

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.

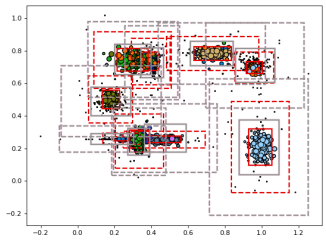
In future work, we will take this preliminary investigation and explore real-world applications of neuro fuzzy logic. We will also find a way to remedy the diminishing/dead gradient problem in ANFIS, which should make gradient scaling the top performer. Other future work will include taking definitions 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.



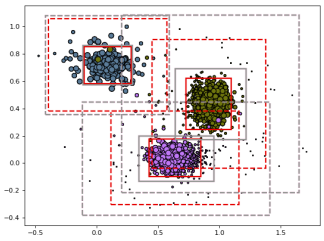
(a) 10% Noise, C=10, R=10. All rules discovered well by extended ANFIS.



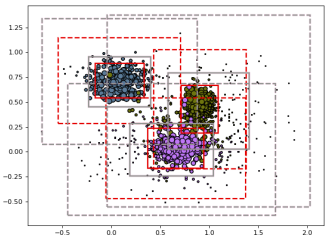
(b) 40% Noise, C=10, R=10. All rules discovered well by extended ANFIS.



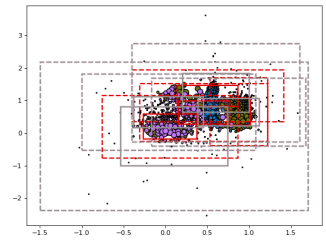
(c) 10% Noise, C=10, R=9. Less rules than clusters and high degree of overlap.



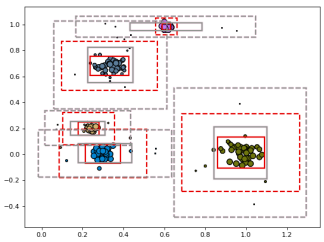
(d) 10% Noise, C=3, R=3. All rules discovered well by extended ANFIS.



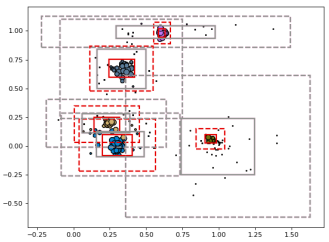
(e) 30% Noise, C=3, R=3. All rules discovered well by extended ANFIS.



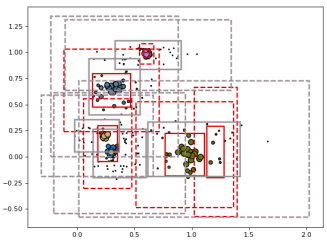
(f) 5% Noise, C=3, R=4. More rules than clusters and high degree of overlap.



(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.

Figure 2.5: 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.

Chapter 3

NEURO-FUZZY LOGIC FOR PARTS-BASED REASONING ABOUT COMPLEX SCENES IN REMOTELY SENSED DATA

Blake Ruprecht[a], Charlie Veal[a], Al Cannaday[a,b], Derek T. Anderson[a], Fred Petry[c], James Keller[a], Grant Scott[a], Curt Davis[a,b], Charles Norsworthy[c], Paul Elmore[d], Kristen Nock[e], Elizabeth Gilmore[e],

[a] Dept. of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO

[b] Center for Geospatial Intelligence, University of Missouri, Columbia, MO

[c] U.S. Naval Research Laboratory, Stennis Space Center, MS

[d] Johns Hopkins Applied Physics Laboratory, Laurel, MD

[e] U.S. Naval Research Laboratory, Washington, DC

Abstract

In this article, we explore the role and usefulness of neuro-fuzzy logic in the context of automatically reasoning under uncertainty about complex scenes in remotely sensed data. Specifically, we consider a first order Takagi-Sugeno-Kang (TSK) adaptive neuro-fuzzy inference system (ANFIS). First, we explore the idea of embedding an experts knowledge into ANFIS. Second, we explore the augmentation of this knowledge via optimization relative to training data. The aim is to explore the possibility of transferring then improving domain performance on tedious but important and challenging tasks. This route was selected, versus the popular modern thinking of learning a neural solution from scratch in an attempt to maintain interpretability and explainability of the resultant solution. An additional objective is to observe if the machine learns anything that can be returned to the human to improve their individual performance. To this end, we explore the task of detecting construction sites, an abstract concept that has a large amount of inner class variation. Our experiments show the usefulness of the proposed methodology and it sheds light onto future directions for neuro-fuzzy computing, both with respect to performance, but also with respect to glass box solutions.

3.1 Introduction

In the last decade, we have witnessed exponential growth of research and disparate applications of machine learning (ML) and artificial intelligence (AI). However, this advancement, primarily in performance according to standard measures like F1, has arose off the back of black box AI-ML, to varying degrees. While a success in and of itself, this arguably generates more problems than solutions. One conundrum is trust and accountability in these machines derived from data and the decisions they make, which falls under the umbrella of explainable AI (XAI). Another topic is domains where expert knowledge exists, perhaps exacerbated by low data volume and/or variety. In such a scenario, it might not be possible to learn a quality data-driven alone solution. In this article, we explore the concept of encoding expert human knowledge into a neural logic system. This allows for knowledge transfer, it maintains explainability and interpretability, and it provides a starting point for augmenting the solution via optimizing with respect to data. The question then pivots to do we see improvement and what logic was responsible for that gain? As a final payoff, any knowledge discovery can be returned to the expert to improve their knowledge and performance on a domain.

The contributions of this article are as follows. First, we discuss high-level benefits and drawbacks, with respect to XAI, of an adaptive neuro-fuzzy inference system (ANFIS) to machine reason about remotely sensed data. Second, we discuss how to import human knowledge into ANFIS. Third, we outline its optimization for augmented reasoning, which we call Human Augmentation. Fourth, we explore and learn from these ideas on a specific application, classification of complex and vague constructions sites. While a contribution in and of itself, this work sets us up to seek better neuro-fuzzy solutions and human centered ML-AI.

The remainder of this article is organized as follows. In Section 3.2, we discuss traditional ANFIS. Section 3.3 discusses initialization of ANFIS, Section 3.4 is different ANFIS membership functions, Section 3.5 is "types" of rules that can be learned, Section 3.6 is the construction site application and experiments, followed by a summary and future work. Figure 3.1 illustrates the main concepts of our article and their connectivity.

3.2 Adaptive Neuro-Fuzzy Inference System (ANFIS)

Jang introduced ANFIS in[31], which was initially based on TSK type fuzzy inference [32]. Figure 3.2 illustrates the flow of data of ANFIS. 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

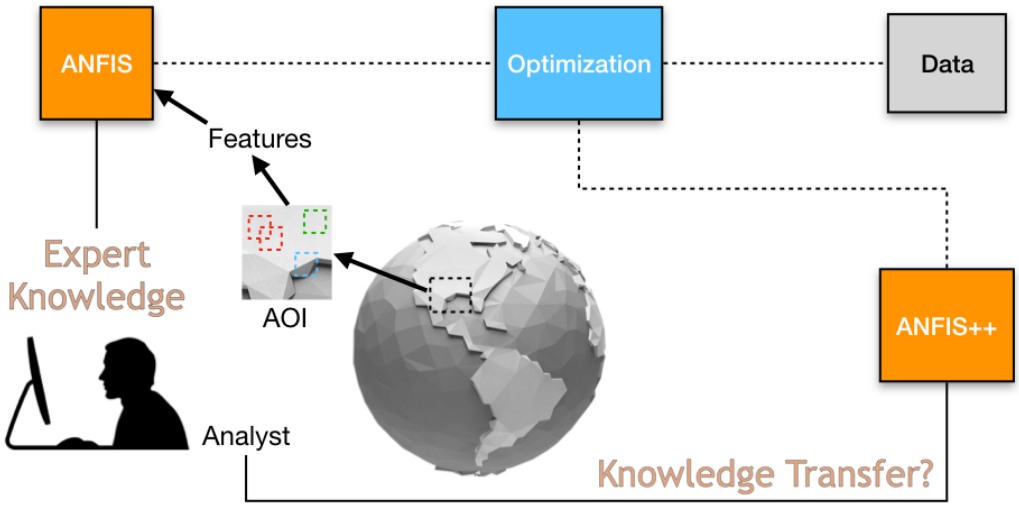


Figure 3.1: High-level illustration of this article. First, expert knowledge is transferred into an adaptive neuro-fuzzy inference system (ANFIS) for sake of automating some process, e.g., object detection or land classification in remote sensing. Next, data is used and the solution is optimized to produce an augmented ANFIS, "ANFIS++". The ANFIS++ is used in place of the expert and it is analyzed to determine differences for the sake of discovering new domain specific logic that might be of interest to the expert and/or analyzed for validation of the machine learned model.

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.

3.2.1 ANFIS Equations

The following are the equations that delineate the forward ANFIS system.

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.,

$$w_n^r = \prod_{k=1}^{K} A_k^r(\mathbf{x}_n(k)). \tag{3.1}$$

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.

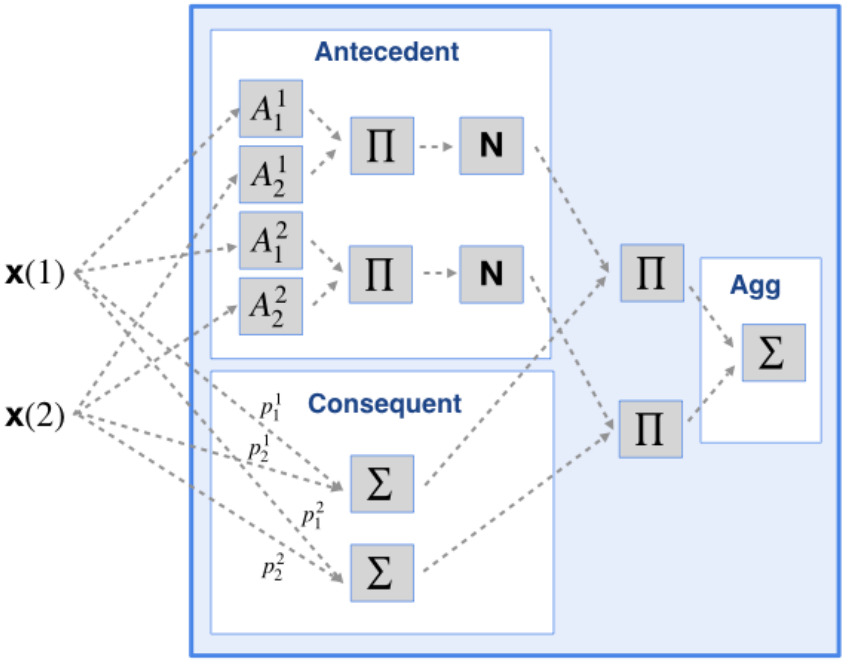


Figure 3.2: This figure illustrates the flow of data in a first order TSK ANFIS for the case of two inputs and two rules.

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

$$z_n^r = \left(\sum_{k=1}^{K} \mathbf{x}_n(k) p_k^r \right) + p_{bias}^r. \tag{3.2}$$

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.,

$$y_n = \frac{\sum_{r=1}^{R} z_n^r w_n^r}{\sum_{r=1}^{R} w_n^r}. \tag{3.3}$$

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.

Table 3.1: ANFIS Derivatives for Gradient Descent

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

3.2.2 Optimization

Table 3.1 are the partial derivatives for ANFIS, based on a first order TSK model. Backpropagation based on the partial derivatives can be performed to update or learn the ANFIS parameters. Different methods can be used to perform learning updates; herein we use PyTorch's automatic differentiation. We have open source PyTorch codes made available at https://github.com/Blake-Ruprecht/Fuzzy-Fusion.

3.3 Initialization

The parameters of the Antecedent Firing step, namely the membership functions $A_k^r(\cdot)$ must be determined during initialization. The parameters for the Consequent Component Building, namely p_k^r, must be initialized.

3.3.1 Antecedent Parameters

Due to the dead gradient problem with ANFIS, as discussed in [10], antecedent parameter initialization is crucial to the performance of the algorithm. Different methods may be used, including random initialization, expert-determined, and clustering to inform where potential membership functions should instantiate. We have used K-means and SP1M clustering to estimate the initial parameters for

the membership functions. As shown in [10], SP1M clustering creates the most accurate membership function estimations. Expert knowledge can also be used to determine the initial position of the membership functions, as discussed in section 3.6.

3.3.2 Consequent Parameters

Different methods may be used to initialize the consequent parameters, including random initialization, and Least-Means Squared (LMS) estimation. Herein, we use LMS, since the TSK system is linear, and it provides a good starting point for learning.

3.4 Membership Functions

The choice of membership functions shapes not only the learning behavior of ANFIS, but the semantic interpretation too. Factors such as differentiability, simplicity, nonlinearity, fuzzy support and core, and more must be considered when choosing membership functions in ANFIS.

3.4.1 Gaussian Membership Function

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

$$N(\mathbf{x}_n(k); \mu_k, \sigma_k) = e^{\left(-\frac{1}{2}\frac{(\mathbf{x}_n(k)-\mu_k)^2}{\sigma_k^2}\right)}. \tag{3.4}$$

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. Furthermore, Gaussian functions are symmetric about the mean, meaning any type of skew cannot be learned. This restricts the membership function further.

3.4.2 Trapezoidal Membership Function

While there are numerous functions to remedy this shortcoming, herein we focus on the trapezoidal function,

$$T(\mathbf{x}_n(k);\boldsymbol{\Theta}) = \begin{cases} 0 & (\mathbf{x}_n(k) < \Theta_1^k) \text{ or } (\mathbf{x}_n(k) > \Theta_4^k) \\ 1 & \Theta_2^k \leq \mathbf{x}_n(k) \leq \Theta_3^k \\ \frac{\mathbf{x}_n(k)-\Theta_1^k}{\Theta_2^k-\Theta_1^k} & \Theta_1^k \leq \mathbf{x}_n(k) < \Theta_2^k \\ \frac{\Theta_4^k-\mathbf{x}_n(k)}{\Theta_4^k-\Theta_3^k} & \Theta_3^k < \mathbf{x}_n(k) \leq \Theta_4^k. \end{cases} \tag{3.5}$$

The trapezoidal function remedies the semantic limitations of the Gaussian through the inclusion of many elements that may have membership value one, and not necessarily infinite support. The trapezoidal function also allows asymmetric behavior to be learned, increasing the quality of interpretations.

3.5 Different "Types" of Rules

In the pursuit of explainable and interpretable solutions, it is not enough to say that ANFIS is automatically understandable simply because it consists of rules. This story might hold if we just encode human knowledge into ANFIS. But, if we learn an ANFIS from scratch and/or if we augment—start with the human and optimized from there—is the resultant solution understandable? To this end, we contemplate some of the scenarios that could arise in the context of a (potentially partially) machine learned ANFIS solution. Note, we discuss different types of rules that ANFIS can discover, but we do not consider below any post processing procedures to identify and/or possibility remediate their existence. We mention this because we do not want to discredit such a possibility, but we do not want to consider one as well since we are unaware of how it would be conducted.

Exception Rules: These rules are rare or they are not present in the data. In the case of the latter, they can be added after the fact by a human based on expert knowledge. In the case of the prior, there is a low probability that ANFIS will naturally discover them. Nevertheless, they are important as they present real logic that the system needs to learn.

Noise Rules: These rules correspond to noise in the underlying data and they may arise from factors like initialization and/or rule over specification. The problem is these rules are not real and they can lead to over fitting. From a learning standpoint, they are not desirable but they may be otherwise indistinguishable by the machine from exception rules.

Counter Rule: Counter rules are cancel one another. In theory, these rules should have similar antecedents, they just have opposite effects (consequences). These rules are problematic because they lead to wasted computation and from an XAI standpoint they complicate matters.

Duplicate Rules: Similar to counter rules, duplicate rules consist of multiple rules covering more-or-less the same antecedent space. If the consequent components are similar then this leads to increased weighting of an underlying rule during aggregation. In a sense, this is ANFIS gaming the system, as its a way to high jack the rule aggregation process and improve the importance of a rule.

Split Rules: These types of rules occur when multiple rules subdivide a rule and work together, versus a more compact and succinct single rule solution. In terms of XAI, this equates to an "overly wordy" machine.

Dead Rule: These are rules that are not associated with any underlying data samples (clusters or noise). Instead, they never fire and they pose a real risk, e.g., ANFIS not generalizing to new data. The reader can refer to our recent work[10], which proves their existence due to initialization and a dead gradient problem.

Good Rules: We could not end this section without discussing the most positive outcome of ANFIS. A good rule is a real relationship that exists and we have learned. Technically speaking, a good rule would be a rule whose antecedents accurately capture the underlying uncertainty in the antecedents and well approximate the consequence. These are the underlying generative structure that we desire ANFIS to learn and wish to learn and explain to a user.

3.6 Case Study: Parts-Based Construction Site Reasoning

Construction site detection from satellite imagery is difficult from a machine learning perspective. There is much variation in the different features and amounts of those features that make up construction sites. Due to this variation, parts-based detectors can be used to classify the features of a construction site to simplify the problem. These simpler detectors can be used as the input to a neuro-fuzzy system, which will then build fuzzy rules and TSK linear relations to fuse the parts into a decision.

The dataset used is from the "DIUx xView 2018 Detection Challenge" [33]. The dataset consists of a series of large image scenes with sets of feature/object bounding boxes. xView is one of the largest and most diverse publicly available satellite imagery object-detection datasets. The initial results used in this paper, i.e. the human expert, were provided by Cannaday et al. [34].

The construction site problem consists of identifying construction sites from satellite imagery

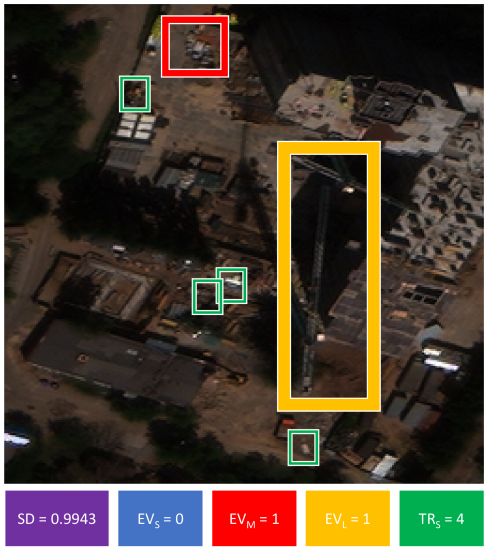


Figure 3.3: This figure illustrates an example of our construction site application. We show some of the features based on their parts-detector categories and their resultant outputs. Note, the entire image (region of interest in the context of broad area scanning) is analyzed for SD, the construction site detector confidence.

based on parts-detection. Our usage of the xView dataset involves four different parts-detectors: small, medium, and large Engineering Vehicles (EV_S, EV_M, and EV_L, respectively), small Trucks (TR_S), and the inference output from the ProxylessNAS DNN trained model (SD), detailed in [4] [34]. An example of a construction site image with some example parts-detectors is shown in Figure 3.3.

3.6.1 Experiment 1: Human Replication

Parts-based detection can be done by ANFIS with the goal of replicating a human expert's knowledge. Since the rules used by a human in this task typically look like threshold based IF-THEN rules (see Figure 3.4. These rules can be manually instantiated in the antecedent firing step of ANFIS, allowing the system to replicate the logical rules used by an expert. The trapezoidal membership function can be made to perform like a step-function when the Θ_1^k and Θ_2^k parameters are equal (creating a vertical line), and the Θ_3^k parameter is suitably large enough to create a core encompassing all possible input values greater than the vertical threshold. The consequent component building step must be instantiated to reflect a step-function, which we accomplished using a linear equation solver. Note, for the sake of this experiment, we turned off the learning feature of ANFIS to ensure the rules did not improve, and only replicated expert knowledge. Under these restrictions and instantiations, ANFIS

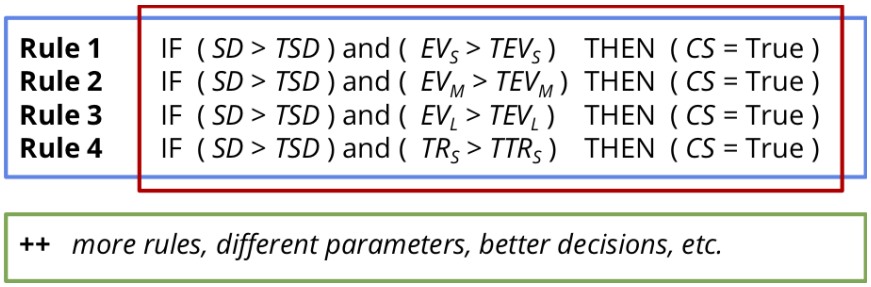


Figure 3.4: An example of the four rules used by our expert. The variables with a T in front are thresholds. These four rules were instantiated in Experiment 1 to replicate human knowledge (the blue box). In Experiment 2, ANFIS from scratch learned new rules and parameters in an attempt to make better decisions. Experiment 3 (the red box), shows what happens when we combine the two different strategies to utilize the benefits of both.

was able to perfectly replicate the expert-informed rules, leading to the same exact performance and interpretability.

3.6.2 Experiment 2: ANFIS from Scratch

In our second experiment, we instantiated ANFIS using SP1M clustering and least-means squared for the antecedent and the consequent parameters, respectively. The purpose of this experiment was to determine if smart instantiation of ANFIS from scratch could approach the performance of ANFIS with expert knowledge, while maintaining interpretability. In these experiments, learning was turned back on to allow ANFIS to refine the instantiation methods over time. ANFIS mostly learned the consequent parameters, and the initialized antecedent parameters tended to not move due to the diminishing/dead gradient problem. This led to an ANFIS solution from scratch the performed about as well as the human expert did on the task, but with worse interpretability. The rules ANFIS learned are able to be analyzed, and look similar to the human's IF-THEN structure, albeit with more fuzziness than the crisp thresholds used by the human. The rules are more difficult to analyze with respect to expected results are, losing some interpretability compared to the human rules. Learning ANFIS from scratch proves to be an okay method in terms of both performance and interpretability, but it doesn't transfer expert knowledge in any way, which could improve performance on this domain while maintaining interpretability.

3.6.3 Experiment 3: Human Augmentation

After the expert rules were replicated, we turned learning on to allow ANFIS to optimize the parameters of the antecedent firing and the consequent component building. Ideally, this method can

utilize the learning ability of ANFIS along with the transference of expert knowledge to create solution that performs better and maintains interpretability. Based on the expert rules, ANFIS was able to optimize the parameters of the consequent step, resulting in better F1-scores, while maintaining the interpretability of the human rules in the antecedent step. This improved performance was accomplished using backpropgation of error, creating a better TSK linear regressor. The antecedent step continued to use interpretable thresholds, and the increase in performance can be interpreted as better consequent component parameters being learned to combine the antecedent firings in the aggregation step. Basically, the linear equation can learn how important the different number of vehicles are in comparison. Since the expert knowledge was transferred to this system, ANFIS did not have to learn from scratch, which prevented some of the known issues of ANFIS (see Section 3.5) from hampering learning.

All three methods offer the same level of explainability that is inherent in ANFIS. In each case, the main way to analyze the decisions made by the algorithm consist in looking at the core and support of the antecedent firing functions. In Experiment 1, it is easy to see that the core and support form a step-function antecedent for each rule. This is by design, since we implemented the antecedent initialization to function like this, and didn't turn on learning to change it. Experiment 3 hardly changes the antecedent parameters (due to previously mentioned problems), but in this case, explainability is maintained because the decision thresholds are still clear. These step-function antecedents are interpretable as such: if the input value is greater than the rule threshold, the antecedent firing strength for that rule is high (normally 1). If the antecedent firing strength is high, the consequent component is "listened to" during aggregation. Similarly, Experiment 3 has interpretable antecedent parameters. Since the trapezoidal function is defined by four parameters, we can view the parameters for each rule to determine the core and support for each rule. In our experimentation, we discovered that ANFIS from scratch did not create step threshold functions, which is not surprising due to the initialization strategy, but did create explainable rules. It is easy to see that the core and support of the ANFIS from scratch rules define what types of values each rule fires for. In ANFIS, the consequent component step creates a linear logical decision; the explainability of the system resides within the antecedent firing parameters.

3.7 Summary and Future Work

Herein, we explored the role of neuro-fuzzy logic on a real-world problem, parts-based reasoning of complex scenes for remote sensing. First, we showed a way to encode expert knowledge into a

Table 3.2: Experimental Results

EXPERIMENT	TP	FP	F1-SCORE
Expert Rules	341	37	90.93%
1) Human Replication	341	37	90.93%
2) ANFIS from Scratch	344	44	90.53%
3) Human Augmentation	335	16	92.67%

machine. Next, we showed that ANFIS could optimize this starting logic and find a better solution, which makes sense because it is hard for a user to optimize their logic. The final machine logic could be understood and explained back to the expert. However, the weakest part of the explanations are the rule consequences, which are likely due to a type-one TSK inference formulation. Last, we also showed that it is possible to learn the fuzzy logic system from scratch—even in light of the fact that ANFIS is hindered by diminishing to dead gradients that lessen its ability to learn—and we found a similar performing solution but with different logic. In summary, we demonstrated a way to take a good explanation, from the human, to encode it into the machine and to improve on that logic. This process is a good example of human-machine teaming and the answer was better than the machine learned only solution likely due to the domain knowledge of the human.

In light of the above case study, a number of weaknesses were identified. First, another neuro-fuzzy inference system beyond ANFIS needs to be used or ANFIS needs to be improved. There are severe hindrances that restrict ANFIS from achieving desired logic. Next, we spent a lot of time trying to get "good explanations". However, what is a good explanation? That needs to be defined and likely folded into the learning process if the goal is to find high quality answers that are also high quality explanations to the expert. Last, we took the output from a set of parts-based detectors. In future work it would be more beneficial if the machine could extract its own features, perhaps based on aspects like spatial relationships between evidence in the scene. However, our concern is that the quality degree of an explanation largely depends on the interpretability of the input; e.g., black box in, black box out.

Chapter 4

CONCEPT LEARNING BASED ON HUMAN INTERACTION AND EXPLAINABLE AI

Blake Ruprecht[a,b], Derek T. Anderson[a], Fred Petry[c], James Keller[a,b], Christopher Michael[c], Andrew Buck[a,b], Grant Scott[a,b], Curt Davis[a,b]

[a] Dept. of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO

[b] Center for Geospatial Intelligence, University of Missouri, Columbia, MO

[c] U.S. Naval Research Laboratory, Stennis Space Center, MS

Abstract

In this article, we explore the role and usefulness of parts-based spatial concept learning about complex scenes. Specifically, we consider the process of teaching a spatially attributed graph how to utilize parts-detectors and relative positions as attributes in order to learn concepts and to produce human oriented explanations. First, we endow the graph with parts detectors and relative positions to determine the possible range of attributes that will limit the types of concepts that are learned. Next, we enable the graph to learn concepts in the context of recognizing structured objects in imagery and the spatial relations between these objects. As the graph is learning concepts, we allow human operators to give feedback on attribute knowledge, creating a system that can augment expert knowledge for any similar task. Effectively, we show how to perform online concept learning of a spatially attributed graph. This route was chosen due to the vast representational capabilities of attributed graphs, and the low-data requirement of online learning. Finally, we explore how well this method lends itself to human augmentation, leveraging human expertise to perform otherwise difficult tasks for a machine. Our experiments shed light on the usefulness of spatially attributed graphs utilizing online concept learning, and shows the way forward for more explainable image reasoning machines.

4.1 Introduction

In the last decade, we have seen an exponential rise in research and funding of machines that can perform visual object detection exceptionally well. However, this advancement, primarily in regards to standard measures of performance like F1, has originated from techniques that function as correlation machines, rather than concept learners. While a success no doubt, these so-called black box machines arguably generate more problems than solutions. One major problem is the explainability and accountability of these machines, which generate no natural explanations and don't promise similar outcomes under similar circumstances. Explainable AI (XAI) attempts to rectify this in one way by developing novel techniques to perform concept learning rather than correlation learning. In essence, concept learning is the search for attributes that are used to distinguish exemplars from non-exemplars of a certain category. This is inherently explainable due to the ability to point to the attributes that did or did not lead to a decision made by the machine. Furthermore, the inclusion of a human in the loop provides oversight and faster machine learning from the interaction of the human and machine. While broad in scope, there are many application areas that can benefit from concept learning based on human interaction and XAI, including computer vision, geospatial sensing, pattern recognition, object detection, natural language processing, and more. In the context of this paper, we are less focused on specific application, and more focused on the dialogue that is created between human and machine when using explainable AI techniques. An explainable AI machine can naturally generate explanations for decisions, a human in the loop can evaluate these explanations and decisions and provide feedback, and the machine accepts the feedback to improve concept definitions, leading to better explanations and performance.

The contributions of this article are as follows. First, we discuss the high-level benefits and drawbacks of concept learning based on human interaction and explainable AI, along with discussing some potential application areas. Second, we describe the low-level methods and techniques of our machine, and the spatially attributed graph that combines these techniques into one learner. Third, we illustrate the explanation-feedback loop of concept learning with the human in the loop, highlighting the knowledge that the machine is using, the generated explanations, how humans can provide feedback, and the improvements for both human and machine that this technique provides. Finally, we provide an example of what concept learning of a spatially attributed graph with human interaction would look like on a synthetic concept example, and show how this method would benefit a scenario where a difficult false positive is encountered.

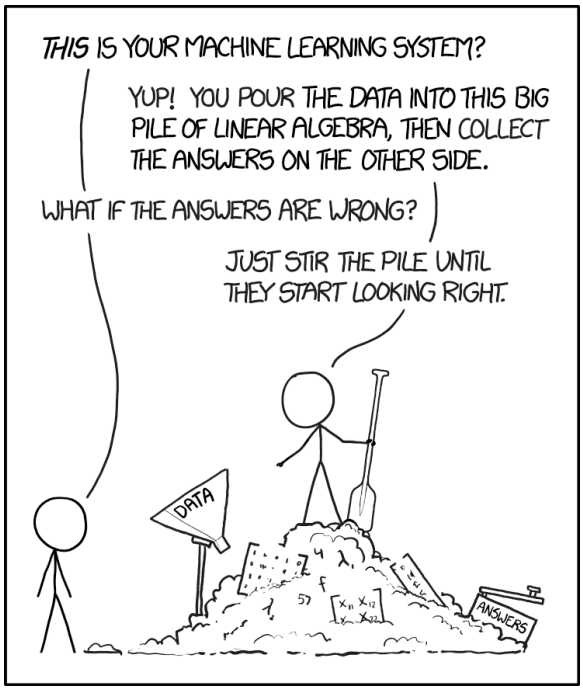


Figure 4.1: A humorous XKCD comic showing some of the problems with correlation machine learning techniques and their inherent lack of explainability. From xkcd.com/1838/.

The remainder of this article is organized as follows. In Section 4.2, we discuss the high level overview of the method, and some specific applications. Section 4.3 discusses concept learning, spatially attributed graphs, human interaction, parts detectors, and spatial relations. Section 4.4 discusses the overall method and how the specific methods are combined. Section 4.5 provides an example of the method, followed by a summary and future work.

4.2 Big Picture

Computational solutions to problems do not always make intuitive sense to humans, especially when the pile of equations used to solve the problem gets larger and larger (see Figure 4.1). Machines solve tasks using the mathematical techniques we endow them with. When the techniques aren't interpretable, there is little hope for creating an explanation. Fuzzy uncertainty is an effective way of dealing with many of the problems that come up in machine learning. Unfortunately, many methods cannot adequately handle processes with fuzzy uncertainty, which leads to worse decisions and explanations. However, when interpretable math techniques are used, explanations naturally exist, and because of this a human can interpret what the machine is doing, and provide feedback[35]. Interpretable explanations require that a human can view the computation the machine is performing

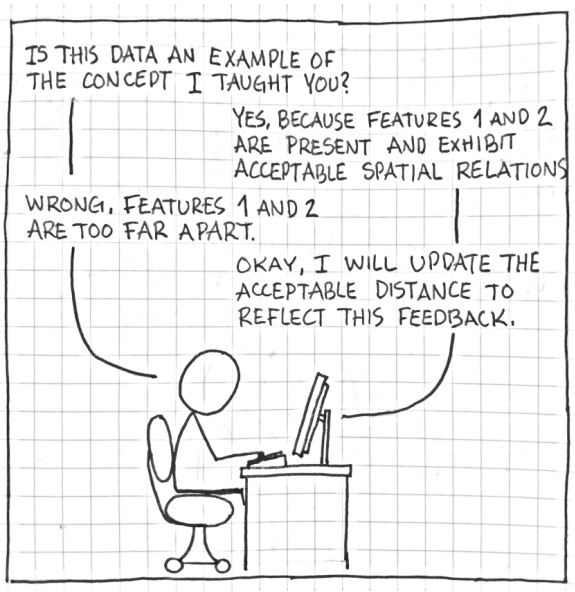


Figure 4.2: The big picture of where we want to go with machine learning – explainable human-machine interaction. The human gives the machine a problem to solve, the machine provides an explainable solution, it's wrong, but since the explanation is interpretable, the human is able to provide feedback, and the machine corrects its concept model. Admittedly, not as humorous.

and understand exactly what is going on. This type of explanation allows the human to both trust what the machine is doing and easily understand where and why the machine makes mistakes. Furthermore, actionable explanations allow the human to provide feedback to the machine by correcting the mistakes made by the machine. This is useful because there is no conversion step needed between feedback and update, maintaining the full explainability of the feedback.

Features and the relationships between features are used to distinguish exemplars of a concept from non-exemplars [36]. A machine can represent features and relationships between features as a model, and this model can be learned from data and human interaction. In Figure 4.2, an example dialogue between human and machine is shown. The human has pre-trained the machine to learn a concept model based on features and the relationships between features. Input data is then given to the machine, and it utilizes the learned model to determine if the data is representative of the concept or not. This determination is made based on the presence of certain features, and how well the relationships between these features match with the learned concept model. The machine's answer explains exactly what features and relationships contributed to the decision as well as the degree to which each feature and relationship contributed. Because the explanation can represent the entire chain from input to computation to output, the human can easily interpret why the machine made a certain decision[37]. Of course, this explanation could be too much information for the user; the scope of the explanation can be intelligently limited. The explanations provide the perfect interface

for the human to provide actionable feedback to adjust the concept model. This feedback can adjust which features are important, how present they must be, which relationships are important, and the nature of those relationships. It is then straightforward for the machine to update its concept model to reflect the feedback provided by the human.

The process of updating the concept model lends itself well to tackling important problems related to concept drift and concept evolution [38]. There isn't always curated data to train on, leading to systems that may not generalize well, and overfit the data. Streaming data is a good way to address this problem, but leads to problems of its own. The concepts that were initially designed may need to be updated or changed to reflect the changing meaning of those concepts, and our method can allow for these changes thanks to the human feedback and update abilities.

This method is particularly well-suited to complex scenes and objects that have regular spatial attributes. Figure 4.3a shows how airplane identification follows spatial patterns with changing objects. The figure details the shapes and spatial relations involved with recognizing friendly aircraft. These are very distinct spatial relationships that lend themselves to classification [39]. Figure 4.3b shows geospatial reasoning over complex scenes, which can be simplified to reasoning over individual parts and their spatial relationships to one another. The individual parts of a construction site can include things like work trucks, engineering vehicles, cranes, etc. These parts can then be reasoned about spatially to determine if the scene involves construction. Figure 4.3c shows an example of sentiment understanding. The process can be improved by tracking changing spatial relationships of individual facial features to one another. In this case, the shape of the mouth changes relative to the other features of the face, leading to a likely change in sentiment. Other applications involving visual reasoning [40] could be good use cases for our method.

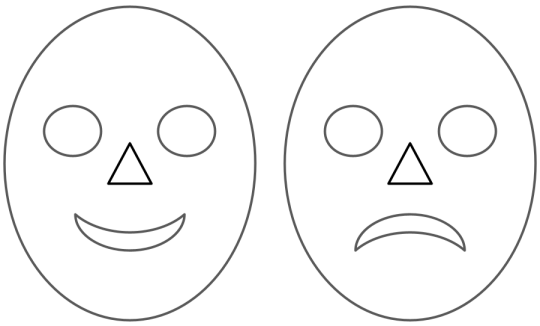
4.3 Specific Methods

4.3.1 Concept Learning

The driving force behind choosing concept learning as the paradigm to view our general problem through is the explainability benefit. Our goal here is not to completely replace pattern recognition as a technique, but rather to rely less on it for complex scenes and objects. Similarly to how complex scenes can be broken up into their constituent objects, complex objects can be broken up into their constituent features. Building a reasoning machine from simpler pattern recognition tasks allows for greater explainability. Complex pattern recognition systems, e.g. CNNs, have no specific, defined

(a) Airplane identification from 1942

(b) Geospatial reasoning



(c) Sentiment analysis

Figure 4.3: Some application areas for concept learning based on human-interaction and explainable AI

internal method for dealing with sub-features, causing these techniques to appear as a black box. A system informed by simpler parts detectors that then reasons on top of the simple parts is by-design more explainable since all of the simpler features are built into the system and easily interpretable. We call this technique concept learning because it serves as a good reference for reasoning over features to identify larger concepts/objects/scenes [37]. Simply put, we are moving the pattern recognition layer one level down from object detection to parts detection, and then performing concept learning on the parts to determine the presence of an object. Since this method is more explainable, humans can easily interpret the reasoning behind decisions made by the machine, which also provides a common language for humans to give feedback to the machine[35].

4.3.2 Spatially Attributed Graph

Our main goal in building the framework of this technique is to learn concepts from features. The learning process utilizes the features themselves, and also the relationships between features [36][41]. The features and relationships are analogous to the nodes and edges of a graph, as shown in Figure 4.4. The features are parts detectors, and the relationships are spatial relations between parts. This natural graph structure and the usage of spatial attributes is combined into the term spatially attributed graph [17]. Here, the nodes of the graph represent the parts detectors used in the system. The edges represent every relationship and function defined between parts in the system. Formally, within the system there exists F, the set of all features. Specific features are denoted as F_i. The set of all possible relationships is denoted as R, with each unique type of relationship denoted R^t, where t is the type of relationship (distance, histogram of forces, etc.). Finally, each relationship applies to multiple features, denoted with a subscript as R^t_{ij}, where ij are all nodes that the relationship relates together, in the direction i to j if needed. In the next subsections, we discuss some of the specific parts and relationships used.

4.3.3 Human Interaction

A driving force behind the choice of concept learning a spatially attributed graph is the natural interaction humans can have with those specific techniques. Concept learning allows humans to give feedback to machines in order to teach them how to better recognize a concept. It is a paradigm that allows for human-in-the-loop interaction[42]. In a spatially attributed graph, the behavior of each node and edge is interpretable, allowing the human to see exactly how each node and edge contributes to the decision. Using this setup, when a machine makes a wrong decision, we can see

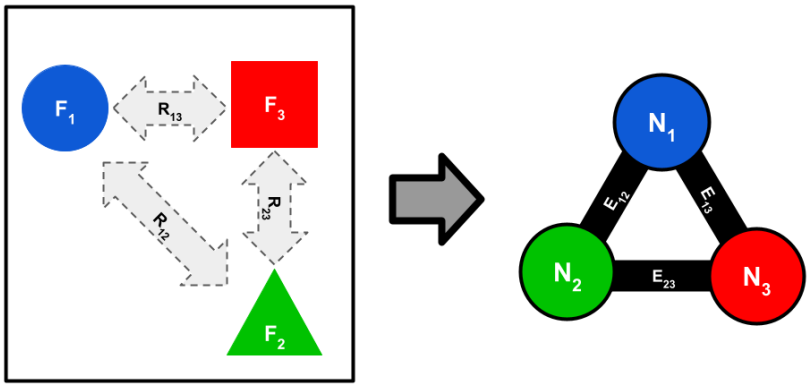


Figure 4.4: On the left, an image containing three features (the circle, square, and triangle). The gray arrows between the features represent the different relationships between features (distances, spatial relations, etc.).The features and relationships between features directly translate to the nodes and the edges between nodes of a graph, shown on the right.

precisely which nodes and edges don't fit with the concept definition. The language of the spatially attributed graph allows us to communicate which nodes or edges are wrong, and how to fix them to better represent the concept.

Finally, the human in the loop has direct supervision over the learning process, since the human can provide precise feedback to directly change the feature or relationship that was incorrect. This can help prevent large errors from dominating a system, and allows for faster machine learning. This is particularly helpful for problems that require a high-degree of accuracy, yet are limited in the amount of training data they have.

4.3.4 Parts Detectors

Any type of parts detector can be used in our system. We don't do anything to deal with the uncertainty of the parts detectors, we treat them as a given. Currently, black-box architectures perform well at simple pattern recognition tasks[43]. We seek to utilize state-of-the-art parts detectors at the low-level, while focusing on the explainability and human interaction benefits of spatially attributed graph concept learning at the high-level. This combination allows us to use the advantages produced by state of the art parts detectors, but also not fully trust their outputs, letting us deal with the uncertainty at a higher level.

4.3.5 Spatial Relations

Concepts are learned from independent features, but the relationships between features provides important contextual details that allow for richer concepts to be developed by a machine. Spatial

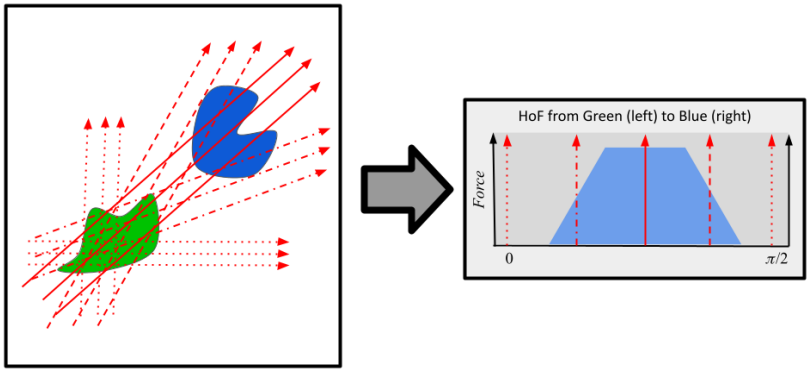


Figure 4.5: Two objects exist in the image on the left; the blue blob in the top right is "above and to the right" of the green blob in the bottom left of the image. The Histogram of Forces for these two objects is shown on the right.

relationships are implicitly explainable, since they are interpreted by humans easily. For example, an object that exists at 90 degrees to another object could easily be interpreted as "to the right" (depending on reference frame), since spatial relationships are concepts that humans are very familiar with. Many different ways of relating objects spatially have been developed; here we focus on binary relationships. Simple relationships such as relative distances are trivial to define and can change based on context.

We use histograms of forces as one of the key tools to relate objects spatially. This type of relation calculates the force exerted by some object B on some object A, at all angles $-\pi$ to π, and creates a histogram representing the force at every angle[18]. The force at an angle relates the number of particles of object B encountered by vectors emanating from A, and the distance those particles are relative to A. As shown in Figure 4.5, the histogram is built from scenes where one object exerts force on another object; this force represents the direction and intensity of that force at all angles emanating outward from the argument object. This creates a histogram that is interpreted as the relative spatial positioning of the referent object to the argument object. In the case of Figure 4.5, it is clear that the blue blob in the top right is "above and to the right" of the green blob in the bottom left. The histogram to the right in the figure shows this as the force being most intense at and angle of $\pi/4$. The histogram is the description of the spatial relationship, and can be translated into linguistic descriptions like "above and to the right". This histogram can then inform linguistic descriptions of scenes[18]. Histograms are compared to one another using similarity functions, as defined in [44], which provides the basis for how they are used in spatially attributed graph concept learning.

4.4 Overall Method

4.4.1 Combining Features and Relationships into a Graph

For the purposes of this paper, we assume that concepts are partially constructed, and we focus on the human machine interaction. The spatially attributed graph model is composed of nodes and edges. The nodes are chosen from the set of features, or parts, that currently have parts detectors in the system. These parts detectors independently analyze the image data for their respective parts, acting as modular components that are added and removed as necessary. The edges are comprised of all relationships between two parts that are currently defined in the system. These spatial relationships are also modular, since different relations can be defined based on the problem domain. Simply, the nodes and edges come from the features found in the input data, and the relationships between features, respectively. The graph is limited by the total number of parts detectors in the system. Features that are not detectable by the system cannot be included in the graph, which is a limitation for all machine learning systems. Similarly, relationships that have not been defined are impossible to detect and add to the graph. This is to say that the possible combinations of nodes and edges are limited to what is detectable by the system (nodes), and what we have defined to be relationships between nodes (edges). This limited set of nodes and edges forms the shared "language" that machine and human can use to create and refine concepts. In the context of images in space, we refer to this as a spatially attributed graph, since the nodes and edges have spatial attributes. An example of this limited language is shown in Figure 4.6, where three different objects are currently detected, and all other objects in the scene are not part of the shared language of that system.

4.4.2 From Input to Concept Graph

This graph model can represent a concept present in an image by including features of the concept (nodes) and specific relationships between those features (edges) from the total set of features and relationships available in the system. The presence of each unique feature is determined from the set of parts detectors in the system. Using parts detectors introduces some uncertainty into the system, so the presence of each part can have a relative match strength, e.g. a number between 0 and 1 of how present that part is in the image. This presence value is used to determine which features from the total set of detectable features are present in any given input image. Once the present features are determined, each spatial relationship between features is calculated. These relationships can be of any arity, but for our purposes as defined in the previous section, we use binary relationships.

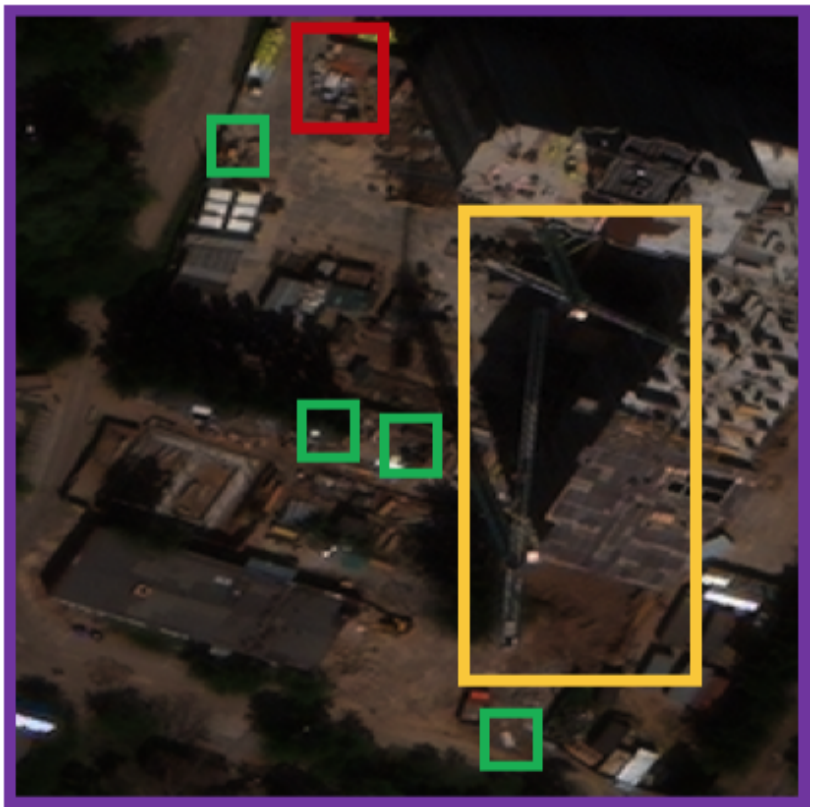


Figure 4.6: In this scene, three different types of objects are currently being detected by their respective parts detectors, the red box is medium trucks, the green boxes are small trucks, and the yellow box is a crane. Currently, concepts can only be built using these parts, since all other parts of the image are not being detected at this time – they are unknown to the system.

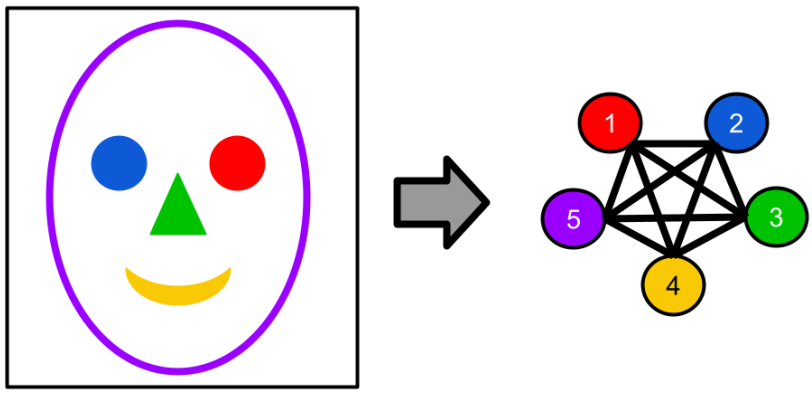


Figure 4.7: Each feature in the image on the left maps to a node in the graph on the right. Specifically, Node 1 = right circle, Node 2 = left circle, Node 3 = triangle, Node 4 = crescent, Node 5 = outer oval

These relationships, along with the features themselves, form the complete spatially attributed graph for the given input image. This graph model is easily interpretable, since for each input image, every feature that was detected and the relationships between features are easy to inspect. Specifically, to demonstrate what the process would look like, we will look at the example of a cartoon face shown in Figure 4.7. In this Figure, each of the five features of the cartoon face in the image on the left map to nodes on the graph. This example is simple to describe, and recognizable as an example that heavily depends on the relationships between features.

4.4.3 Learning a Concept from Scratch

For the purposes of human-machine interaction, and for the sake of speed, a human expert can define an approximate concept using the shared language of the spatially attributed graph by selecting important features and relationships. The process of defining a concept is as simple as selecting which features must be present, and then determining the approximate spatial relations between features. The problem of learning the concept from scratch is currently unsolved, and outside of the scope of this paper. We don't discuss exactly how to learn the concept, instead we focus on comparing concepts and the human feedback loop of concept learning.

4.4.4 Comparing Concept Graphs

The graph model representing the input image will be referred to as the input graph, and the graph model representing the existing concept model will be referred to at the reference graph. It is easy to determine which features and relationships are necessary for a given concept by analyzing the reference graph. Each specific required feature is present in the reference graph, and the relationships

between each feature are easily inspected to determine the spatial relationships between features. The input graph is compared to the reference graph to determine if the necessary concept features in the reference graph are all present in the input graph.

Using Figure 4.8 as an example, we can see that the reference graph on top contains 5 nodes, each specifying a specific feature. The input graph on bottom contains those same five nodes. If there are more or fewer features present in the input graph compared to the reference graph, then the concept is not present, by definition. However, extra or missing features do not necessarily mean that the input data does not demonstrate the concept or at least a partial concept, both of which can be useful. Due to the fuzzy nature of defining concepts, partial concepts may be present in images and will need to be dealt with, e.g. by forming a new concept, removing requirements from the current concept, etc. Humans can easily verify visually that the same features are present in both images. The spatial relationships between features are compared depending on the type of relationship. If the relationships are relative distances, the scalars are compared using allowable plus/minus thresholds, or a matter of degree represented by a fuzzy set. If the relationships are relative positions using histograms of forces, the histograms are compared using similarity functions. Of course, spatial relationships don't necessarily have to perfectly match. Thresholds of allowable deviation are set to allow for some variation in spatial positioning.

Depending on how each node and edge is compared, the overall graph comparison is determined using any combination function – minimum usually makes sense, because if any part of the input is wrong, the entire input doesn't fit the concept. Again, if this isn't the case, the human can provide feedback to the machine to learn a better concept. In Figure 4.8, the spatial relationship between Nodes 1 and 2 are shown on the right. Nodes 1 and 2 (the right eye and left eye of the cartoon face), are the same relative distance away from each other in both images, but the histogram of forces clearly shows that they are at different relative positions in the different images. The similarity between the two histograms is low, which shows that the input image does not match the reference image. This overall comparison leads to the decision of whether a concept is found within the input image or not.

4.4.5 Feedback Loop

For an input, the machine computes the spatially attributed graph and performs the comparison to the reference graph, deciding if the concept is present or not. Any of the set of features, relationships, and comparisons leading to the final decision can then be displayed to the human expert for evaluation,

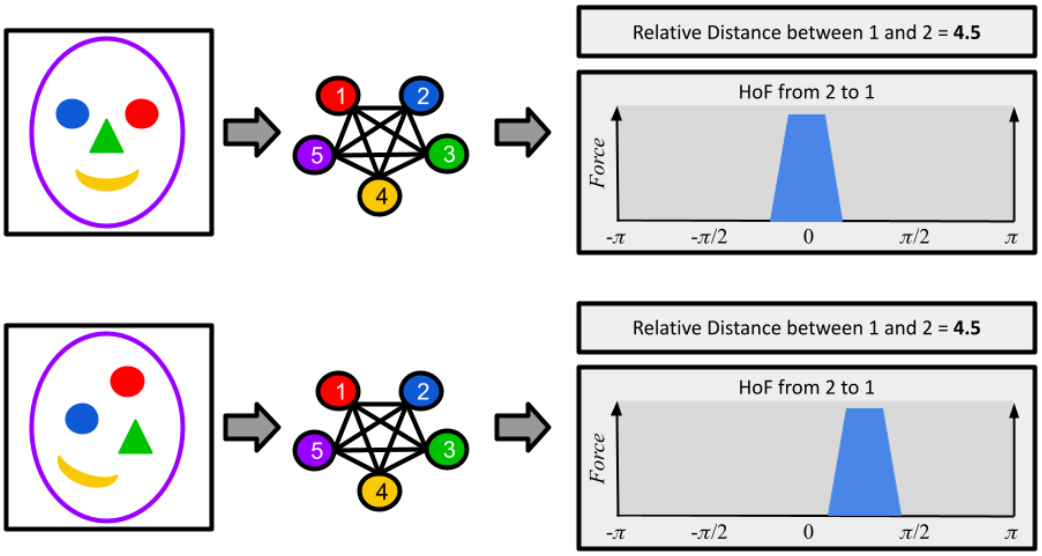


Figure 4.8: The comparison between input graph and reference graph. Note that while Features 1 and 2 are the same distance in both, and the histograms look similar, the histogram mean is at a different angle for the two graphs.

as shown in Figure 4.8. Of course, all of these are displayed easily, showing the full explainability of the system, but this could also be an overwhelming amount of information, given the problem domain. The human is able to analyze each step that led to the decision and determine if the decision is correct or incorrect, and either way, if any of the steps were correct or incorrect. The human can then provide feedback of varying degrees and specificity explaining what aspects are incorrect, and why, using the common language of the spatially attributed graph. An example of what this looks like is shown in Figure 4.9. In the example, the human gives linguistic feedback, specifically the term "farther". The machine uses this to shrink the allowable histogram of forces between the two objects to better match the intended relationship based on the term "farther". Using feedback, the machine can refine the features, relationships, thresholds, and comparisons to better represent the given concept. This process can then be repeated for more input images, and even the same input image to compare performance of concepts as they develop across time.

4.5 Example

4.5.1 Prototype

Here, we call the allowable ranges for each feature and relationship the *prototype* of the concept. This prototype represents the currently defined concept, including all relevant features, and the

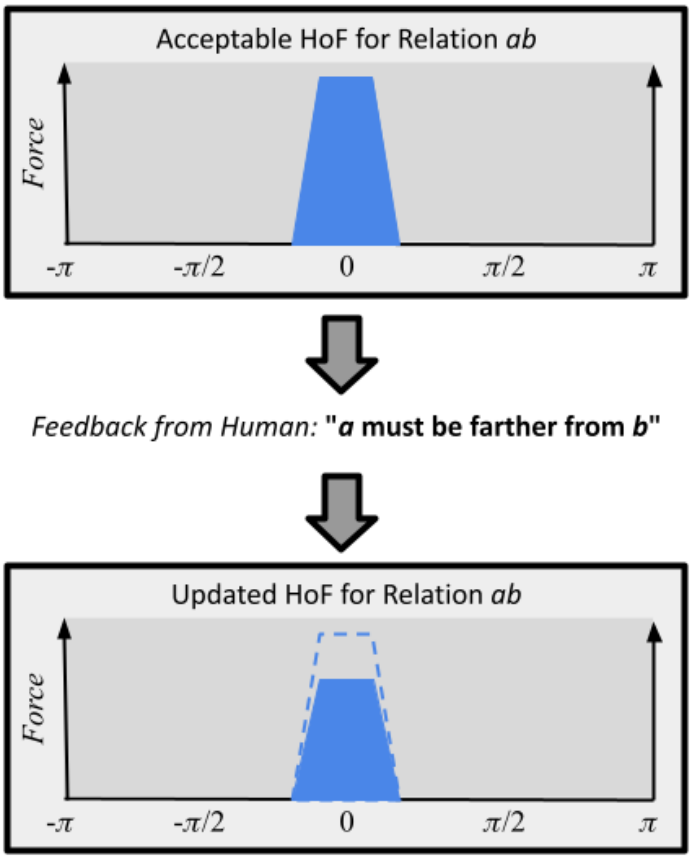


Figure 4.9: An example of the shared language the human can use to provide feedback to the machine to assist in concept learning. The machine has a method of translating the linguistic term "farther" into an operation on the histogram values to reduce the allowable force.

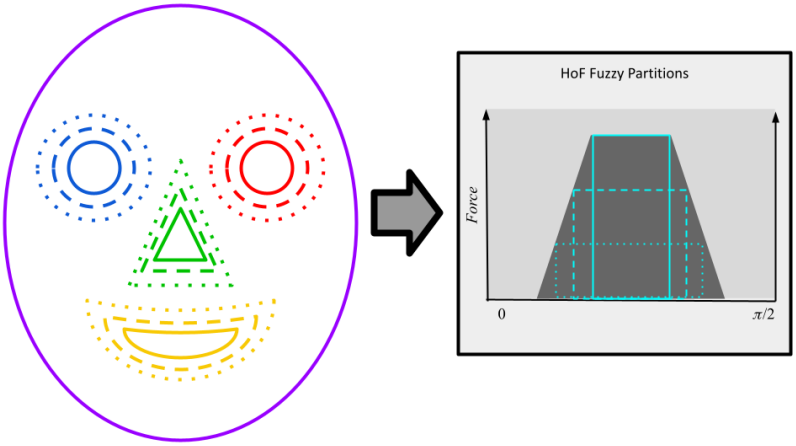


Figure 4.10: An example of the shared language the human can use to provide feedback to the machine to assist in concept learning.

specifics of the relationships allowed. We use the cartoon face example because it does a good job of demonstrating why this technique is useful: specifically, because each feature is distinct and detectable, and the relationships between features defines whether or not the features represent a cartoon face or not. The prototype can be represented using crisp relationships with margins of error, or fuzzy relationships that build uncertainty into the relationship directly. The visual representation of the prototype, shown in Figure 4.10, is meant to show roughly where the allowable spatial positions for each feature are, based on the allowable spatial relations between features The prototype on the left represents the core of each image allowable spatial position, along with a higher and lower confidence interval to show the range of positions into which each feature could fit. The spatial relations, in effect, constrain the features to certain positions relative to one another. Of course, the figure doesn't show the relative angles or distances between individual features – for example, the right eye should be 0 degrees (to the right) of the left eye, and can't be at -2 or +2 degrees. These constraints allow the machine to determine if the image represents a certain concept or not. The flexibility in the constraints allows the machine to deal with uncertainty, as shown in the more translucent areas surrounding each crisp feature on the prototype.

4.5.2 Difficult False Positive

Here, we show what would happen when a false-positive is shown to the system, i.e. an input image that presents the current features and relationships of the prototype, but shouldn't, so the human has an opportunity to demonstrate the process of fixing this. In Figure 4.11, we see an image that currently matches the prototype for all features. The prototypical relationship between the eyes,

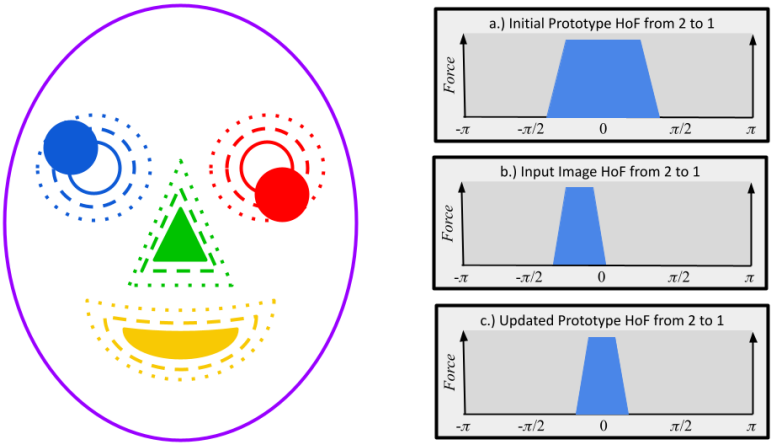


Figure 4.11: Input image on the left, overlaid on top of the prototype image. a.) is the prototype relationship of right eye to left eye. b.) is the input image HoF. c.) is the updated prototype after human feedback.

Nodes 1 and 2, is shown in every histogram. Histogram a.) shows the allowable spatial relation between the eyes for the protoype initially. Since histogram b.), which shows the relationship between the eyes for the input image, clearly falls within the allowable region defined in histogram a.), the prototype needs to be refined to specify the relationship better. The human expert has decided that this cartoon face does not satisfy their concept of a face, so provides feedback to the machine that the right eye is too far above the left eye. The human feedback triggers and update to the prototype, which leads to the refined prototype relationship shown in histogram c.), where the relationship between the eyes has been specified to allow fewer angles.

4.6 Conclusion

In this article, we explored concept learning based on human interaction and explainable AI. In the context of parts-based concept learning about complex scenes, we showed how features found using parts detectors act as nodes in a graph, and the spatial relations between features act as edges, creating a spatially attributed graph. We discussed the process of inputting an image into this system, and how the image data forms the spatially attributed graph. Next, we showed how concepts are represented in the spatially attributed graph, and the current state of learning a concept from data. Next, we showed how concepts are compared across different graphs due to the interpretable nature of the graphs, allowing humans to understand and take part in the process. This naturally leads to the opportunity for human interaction, where humans can improve the system using the shared language of the spatially attributed graph. Effectively, we showed how to perform online

concept learning of a spatially attributed graph. We emphasized the explainability of this method by demonstrating the process using a synthetic example, helping to show the way forward towards more explainable artificial intelligence.

Chapter 5

CONCLUSION

5.1 Summary

In summary, this thesis attempted to build an XAI system to address problems with AI safety, mainly the lack of a system combining explicit knowledge representation, concept learning, and human feedback. To this end, I implemented Type-1 TSK ANFIS in PyTorch, and attempted to improve performance using possibilistic clustering. After implementing and testing, I found some problems with the logic learned. Further experimentation performing parts-based reasoning further showed limitations to ANFIS. While it could replicate and improve on human rules, learning rules from scratch proved much more challenging. "Good logic" was not learned by the system, instead we showed that due to diminishing and dead gradients in ANFIS, many bad types of rules were learned like "dead" rules and "noisy" rules. This showed that more constraints or other mechanisms to enforce good logic are necessary for ANFIS to really shine. This led to our exploration into different explicit techniques to perform concept learning, where we tried to focus more on creating a feedback loop using a human expert. We just got started on that last area or research, which leads to the future work necessary to make this explainable logic system function.

5.2 Future Work

In the future, more work needs to be done to address the problems with ANFIS. The dead and diminishing gradients lead to poor logic, so the gradient descent part of the ANFIS algorithm needs updates or revisions to learn better logic rules. Furthermore, we discussed using a SARG to perform concept learning, and did not get far enough to evaluate whether this structure is an improvement over ANFIS or not. Concept learning using features and relationships is a layer above correlation

learning, but the features are still learned using correlation techniques. This may be the best solution in the end, but more work is needed to evaluate the safety of using correlation techniques at that level of learning. Further research must be performed into the human feedback loop. We did not define what type of information is best to pass between human and machine, nor how this should be done in practice. We did not evaluate any human factors during the course of this research, so that needs to be done as well. Finally, more rigorous definitions of safety and explainability need to be applied to AI research to ensure that future machines stay aligned to our human values.

Much work needs to be done in the future to improve the systems outlined in this thesis. Neuro fuzzy logic gradient descent learning needs a major overhaul to fix problems with dead and diminishing rules, especially if we want to stay committed to being able to incorporate neuro-fuzzy with existing neural algorithms. We need to find a way to incorporate metrics of "goodness" into the learning process, either as a cost function, constraints, etc., to encourage better rules. Further, to create better rules, membership functions need some form of "shrinkage" to be able to better fit the data. We saw problems with noise, so some technique needs to be developed to address that. Next, there isn't consensus on what constitutes an explainable deep fuzzy logic system, and more exploration needs to be done to make the algorithm more transparent. If the inputs/outputs aren't 100% explainable by themselves, what makes an explanation? Finally, how do we use the rules/logic learning by ANFIS to generate an explanation? I think the work done here is a good start, but many of these questions need to be addressed to create a more effective and explainable ANFIS.

Next, the application of ANFIS brought up some future questions. How do we best make use of human knowledge/rules prior to training? During training, how do we best make use of human-in-the-loop feedback? When doing so, how do we align this feedback and learn the humans true intentions, so that human bias doesn't negatively impact the learning process? We defined rules on parts, but can we further define rules that incorporate the parts to learn concepts better? We must analyze the trade offs between performance and explainability, and see if we can do both, or if explanation comes at the cost of worse performance. In our application, we used CNNs to perform parts detection, and I wonder if this is the best method to feed parts to ANFIS. How do we handle variable numbers of parts, and how do we learn ANFIS rules on the fly with different parts appearing?

Finally, I made good progress on the SARG route, but many questions are left unanswered. What if parts are missing? What if there are extra parts? How do we learn concepts on the fly using many different parts detectors? Ideally, I'm picturing a system that can update concepts based on human feedback using whatever parts that system can currently detect, without the human having to specify each part. How do we learn to do this? Furthermore, can we reason about what the machine learns

when doing this, and can the machine generate a satisfactory explanation? Human feedback is a promising research route, yet we need to determine how to "optimally" interact with a human to learn their preferences, and ideally, none of their biases, to truly learn the intention of the human. Much work needs to be done to create more explainable parts-based concept modeling and reasoning machines.

BIBLIOGRAPHY

[1] "Ai risk management framework: Second draft," Tech. Rep., National Institute of Standards and Technology, August 2022.

[2] Select Committee on Artificial Intelligence, "The national artificial intelligence research and development strategic plan: 2019 update," Tech. Rep., National Science & Technology Council, June 2019.

[3] "Research priorities for robust and beneficial artificial intelligence: An open letter," October 2015.

[4] David Gunning, "Explainable artificial intelligence (xai)," *Defense advanced research projects agency (DARPA), nd Web*, vol. 2, no. 2, pp. 1, 2017.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[6] J.-S.R. Jang and Chuen-Tsai Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378–406, 1995.

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

[8] 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.

[9] J Sun Jang, "R., sun, ct., and mizutani, e.(1997): Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence," *Prentice Hall, Inc., Simon & Schuster/A Viacom Company, Upper Saddle River, NJ*, vol. 7458, pp. 23, 1997.

[10] Blake Ruprecht, Wenlong Wu, Muhammad Aminul Islam, Derek Anderson, James Keller, Grant Scott, Curt Davis, Fred Petry, Paul Elmore, Kristen Nock, et al., "Possibilistic clustering enabled neuro fuzzy logic," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2020, pp. 1–8.

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

[12] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[13] Alan B. Cannaday II, Curt H. Davis, Grant J. Scott, Blake Ruprecht, and Derek T. Anderson, "Broad area search and detection of surface-to-air missile sites using spatial fusion of component object detections from deep neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4728–4737, 2020.

[14] Blake Ruprecht, Derek T Anderson, Fred Petry, James Keller, Christopher Michael, Andrew Buck, Grant Scott, and Curt Davis, "Concept learning based on human interaction and explainable ai," in *Pattern Recognition and Tracking XXXII*. SPIE, 2021, vol. 11735, pp. 106–119.

[15] Wen-Hsiang Tsai and King-Sun Fu, "Error-correcting isomorphisms of attributed relational graphs for pattern analysis," *IEEE Transactions on systems, man, and cybernetics*, vol. 9, no. 12, pp. 757–768, 1979.

[16] 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.

[17] M. A. Eshera and K. S. Fu, "An image understanding system using attributed symbolic representation and inexact graph-matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 5, pp. 604–618, 1986.

[18] P. Matsakis, J. M. Keller, L. Wendling, J. Marjamaa, and O. Sjahputera, "Linguistic description

of relative positions in images," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 4, pp. 573–588, 2001.

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

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

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

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

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

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

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

[33] Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord, "xview: Objects in context in overhead imagery," 2018.

[34] A. B. Cannaday II, R. L. Chastain, J. A. Hurt, C. H. Davis, G. J. Scott, and A. J. Maltenfort, "Decision-level fusion of dnn outputs for improving feature detection performance on large-scale remote sensing image datasets," in *2019 IEEE International Conference on Big Data (Big Data)*, Dec 2019, pp. 5428–5436.

[35] H. Hagras, "Toward human-understandable, explainable ai," *Computer*, vol. 51, no. 9, pp. 28–36, 2018.

[36] Jesus Gonzalez, Lawrence B. Holder, and Diane J. Cook, "Graph-based concept learning," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. AAAI Press*. 2000, p. 1072, The MIT Press.

[37] Kaj Sotala, "Concept learning for safe autonomous AI," in *Artificial Intelligence and Ethics, Papers from the 2015 AAAI Workshop, Austin, Texas, USA, January 25, 2015*, Toby Walsh, Ed. 2015, vol. WS-15-02 of *AAAI Workshops*, AAAI Press.

[38] M. M. Masud, Q. Chen, L. Khan, C. Aggarwal, J. Gao, J. Han, and B. Thuraisingham, "Addressing concept-evolution in concept-drifting data streams," in *2010 IEEE International Conference on Data Mining*, 2010, pp. 929–934.

[39] S. F. Ali, J. Jaafar, and A. S. Malik, "Proposed technique for aircraft recognition in intelligent video automatic target recognition system (ivatrs)," in *2010 International Conference on Computer Applications and Industrial Electronics*, 2010, pp. 174–179.

[40] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick, "Clevr: A diagnostic dataset for compositional language and elementary

visual reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[41] Weili Ding, Bo Hu, Han Liu, Xinming Wang, and Xiangsheng Huang, "Human posture recognition based on multiple features and rule learning," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 11, pp. 2529–2540, Nov 2020.

[42] Jesper E. van Engelen and Holger H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, Feb 2020.

[43] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, Feb 2020.

[44] Costas P. Pappis and Nikos I. Karacapilidis, "A comparative assessment of measures of similarity of fuzzy values," *Fuzzy Sets and Systems*, vol. 56, no. 2, pp. 171–174, 1993.