Revisit fuzzy neural network: bridging the gap between fuzzy logic and deep learning

Lixin Fan
lixin.fan@nokia.com

Nokia Technologies
Tampere, Finland

Abstract. This article aims to establish a concrete and fundamental connection between two important fields in artificial intelligence i.e. *deep learning* and *fuzzy logic*. On the one hand, we hope this article will pave the way for fuzzy logic researchers to develop convincing applications and tackle challenging problems which are of interest to machine learning community too. On the other hand, deep learning could benefit from the comparative research by re-examining many trail-and-error heuristics in the lens of fuzzy logic, and consequently, distilling the essential ingredients with rigorous foundations. Based on the new findings reported in [38] and this article, we believe the time is ripe to revisit fuzzy neural network as a crucial bridge between two schools of AI research i.e. symbolic versus connectionist [93] and eventually open the black-box of artificial neural networks.

Table of Contents

Revisit fuzzy neural network: bridging the gap between fuzzy logic and deep learning 1

Introduction 3
1 A historical perspective on fuzzy logic 6
1.1 Overview of landscape 6
Principle of bivalence 6
The status quo 7
Fuzzy logic references 8
1.2 Prehistory of fuzzy logic 9
1.3 Many-valued logics 9
Łukasiwicz logics 9
Gödel logics 10
Product logic 11
1.4 Fuzzy sets and fuzzy logic 11
Fuzzy set 11
Level cuts 12
Standard operations on fuzzy sets 12
Ordered weighted averaging 13
Fuzzy XOR connective 13
Fuzzy number and fuzzy interval 15
Fuzzy arithmetic and the extension principle 15
Fuzzy relation 15
Approximate reasoning 15
Methods for constructing fuzzy sets 16
Criticisms 19
1.5 Foundations and extensions 19
Logic of inexact concept 20
Meta-mathematics of fuzzy logic 20
Probabilistic logic 20
Rough sets 21
1.6 Applications of fuzzy logic 22
Fuzzy clustering and pattern recognition 22
Fuzzy control 22
Fuzzy decision making 22
Fuzzy automata 22
Databases and information retrieval 23
Fuzzy Neural Network (FNN) 23
FNN with deep learning 23
2 Revisit fuzzy neural network with generalized hamming network 26
2.1 Introduction 26

2.2 Related work 27
2.3 Generalized Hamming Distance 28
2.4 Generalized Hamming Network 30
New perspective on neural computing 30
Ganeralized hamming network with induced fuzzy XOR 32
2.5 Performance evaluation 33
A case study with MNIST image classification 33
CIFAR10/100 image classification 34
Generative modelling with Variational Autoencoder 35
Sentence classification 35
2.6 Conclusion 35
3 Visualization of generalized hamming networks with deep epitomes 37
3.1 Introduction 37
3.2 Related work 38
3.3 Deep Epitome 39
Review of GHN 39
Generalized hamming distance and epitome 40
Deep epitome 43
3.4 Deep epitome for network visualization 45
Data independent visualization of deep epitomes 45
Data dependent feature extraction 46
3.5 Conclusion 47
4 Discussion and future work 48

Introduction

Since 2006 *deep learning* [85] has witnessed a striking development and record-breaking achievements on a variety of artificial intelligence problems such as image classification [82] , speech recognition [55] and Go game playing [121, 122]. The outstanding performances of deep learning is partially ascribed to the great generalization capability of deep neural networks in extracting relevant knowledge from large datasets [45, 73]. On the other hand, the acquired classification competence is vulnerable to adversarial attacks [46] and the flexibility in learning can be mis-used to fit meaningless random patterns [146]. To make things worse, the acquired knowledge are stored in the forms of *millions to billions of weights and bias parameters* which are surely out of the reach of human comprehension. This well-recognized black-box nature calls for an comprehensive understanding of the underlying mechanisms of the "paradoxical success" of deep learning. Unfortunately attempts to open the black-box of deep neural network thus far are unable to furnish satisfactory answers. On the contrary, the complex and dynamic learning process appears so mysterious to some people that they start to *worship the AI God* [AIg].

Fuzzy logic research, which aims to study and model *vagueness* in human reasoning with rigorous mathematical tools, has much to offer in formalizing and elucidating the deep learning underlying mechanism appropriately [62, 138]. For instance, the *data augmentation* technique commonly adopted in machine learning can be modelled as "fuzzified" data [61] or the structure of data spaces can be characterized in terms of fuzzy order relations [20, 114]. Despite of these promising advances which were reported in the fuzzy logic community only, it is extremely difficult to find a fuzzy logic related paper in a core machine learning conference or journal except for our recent NIPS publication [38]. Therefore it is the ultimate goal of this article to establish a concrete and fundamental connection between two important fields in artificial intelligence i.e. *deep learning* and *fuzzy logic*.

Looking retrospectively, it is not the first attempt to integrate strengths of the learning capability of neural networks and the interpretability provided by fuzzy logic theory — *fuzzy neural network* was proposed in 1980s exactly for this purpose [51]. The effort to interpret neural network in terms of propositional logic calculus even dated back to McCulloch & Pitts' seminial paper [91]. Based on the new findings reported in [38] and this article, we believe the time is ripe to revisit fuzzy neural network as a crucial bridge between two schools of AI research i.e. symbolic versus connectionist [93]. Our belief are mainly based on two reasons. First, the rapid development in (deep) neural networks has given rise to many new techniques that are not available in the last century when fuzzy neural networks were originated. These new techniques e.g. batch normalization, albeit extremely useful from an empirical point of view, are subject to critical examination in the lens of fuzzy logics. By doing so, one is able to grasp the most essential ingredient of deep learning. It is our hope that this kind of comparative study will shed light on future deep learning research and eventually open the "black box" of artificial neural networks [16]. Second, on the other hand,

the revisit provides powerful computational tools to perform fuzzy inferencing by exploiting the rapidly developed neural computing techniques such as deep convolutional neural networks (DCNN). Consequently, it is demonstrated for the first time that fuzzy neural networks, in particular, the *generalized hamming network* (GHN) can achieve the state of the art performance on par with its non-fuzzy counterparts for a variety of machine learning problems [38].

This article is suitable for researchers, practitioners, engineers and educators in the field of artificial intelligence in general, and in particular, those young generation who are familiar with the rapidly developed machine learning tools such as SVM [29], random forest [21] or recent deep learning frameworks (Caffe, CNTK, Torch or Tensorflow etc.), but are probably unfamiliar with the symbolic point of view of artificial intelligence and the joint force research called "soft computing" [93, 150]. We assume readers have general knowledge about machine learning, deep learning and pattern recognition. Familiarity with specific topics such as image categorization or sentence classification might be useful but not essential. We also assume readers have basic understanding about classical logic concepts like truth tables and logic operations, but not necessarily about their many-valued or fuzzy logic counterparts which are reviewed in this article.

In order to provide an appropriate context and background introduction to the in-depth discussion in Sections 2 and 3, the first part (Section 1) of the article therefore is devoted to a brief historical review of research topics in fuzzy logic. References for further readings are provided in case related topics are unable to elaborate due to the limited space of this article. The second part then reports our recent findings about furnishing deep learning with new insights and interpretations in terms of fuzzy logic. Specially, Sections 2 showcases motivations and advantages of measuring *generalized hamming distances* between neuron inputs and weights, followed by Section 3 illustrating how to convert a deep *generalized hamming network* into an equivalent shallow yet wide network using *deep epitomes*.

1 A historical perspective on fuzzy logic

1.1 Overview of landscape

Figure 1 illustrates an overview of related topics that are reviewed in this section. Topics are roughly aligned along two axes i.e. the *period of time* and the *applicability*, but these two notions should be understood as fuzzy sets. It must be also noted many important topics such as predicate fuzzy logic, applications in chemistry, physics, social sciences etc. are not covered in this article and we refer readers to [14] for a thorough historical review.

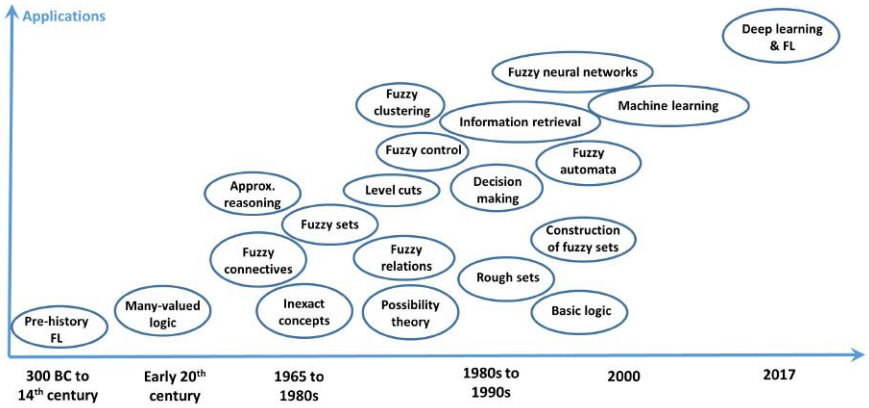


Fig. 1: Overview of related topics reviewed in this article. The x-axis *roughly* indicates the *period of time* during which the topic in question was proposed or actively studied. The y-axis *roughly* indicate topics' *applicability* to real world use cases (FL stands for fuzzy logic).

Principle of bivalence : a fundamental principle of classic logic states that any declarative sentence expressing a proposition has only two possible truth values, *true* and *false*. This is thus called *the principle of bivalence*. In spite of its simplicity and the foundational role in classical logic, most propositions in our real world communication are nevertheless not bivalent, instead, their *truth is a matter of degree*. As an example, consider the proposition "John is tall". In classical logic, one is forced to choose a threshold height $\mathfrak{h}$ such that the proposition is considered true if John's height is greater than $\mathfrak{h}$ otherwise false. According to our common sense, the notion of "tallness" can hardly be associated to any *sharp and specific* threshold $\mathfrak{h}$. Instead, it is *rough ranges of height* to which people often refer with phrases like "pretty tall", "somehow tall" or "absolutely not tall" to communicate about various degrees of truth in the proposition.

Fuzzy logic is therefore motivated to model *vague* notions as such used in our everyday life, and, *reject the principle of bivalence* by extending classical truth values with additional ones. These extended truth values are often interpreted as *degree of truth*, and may be represented by numbers in the unit interval $[0, 1]$. In this particular case, 0 and 1 represent the two extreme degrees of truth, while the numbers in between represent intermediate degrees of truth. Intermediate number in $[0, 1]$ can be either *finitely* or *infinitely* defined which lead to different forms of *many-valued logic*. Whether an object belongs to a class in question is now represented as a *graded membership* in $[0, 1]$ which essentially defines a *fuzzy set* for all objects in consideration. *Fuzzy connectives* and *fuzzy relations* are defined on fuzzy sets correspondingly, in the similar vein as their classical logical counterparts. Fuzzy deduction rules like *graded modus ponens* are thus used to carry out *approximate reasoning* from partially true assumptions or *inexact concepts* to partially true conclusions. In order to apply *fuzzy compositional rules of inference*, one has to first *construct fuzzy sets* for the given application context.

In response to criticisms questioning about the mathematical foundations of fuzzy logic, researchers put forward different mathematical forms of fuzzy logic e.g. *Basic Logic* to lay down theoretical foundations rigorously and provided *possibility theory* as one of its interpretations. Application-wise, fuzzy logic has been widely used for *clustering and pattern recognition*, *control*, *decision making*, *database and information retrieval* and *machine learning, deep learning*. New tools like *fuzzy neural network* and *fuzzy automata* have been proposed for all kinds of applications. Recently, the fuzzy logic interpretation has inspired the *generalized hamming network*, which is the first fuzzy neural network with deep structures that demonstrates state of the art performances on a variety of machine learning tasks.

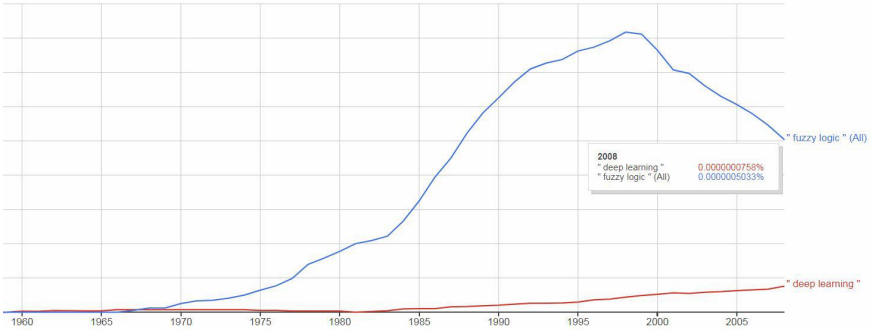


Fig. 2: Frequency chart of "fuzzy logic" and "deep learning" in 50 years (from 1958 to 2008) plotted by Google Ngram Viewer [47].

The status quo : Figure 2 illustrates frequencies of "fuzzy logic" as well as "deep learning" appeared in all books published between 1958 to 2008 according

to Google Ngram Viewer [47]. Clearly the popularity of "fuzzy logic" was ignited by Zadeh's seminal paper published in 1965 [138], nevertheless, the trend started to decline since 1998. On the other hand, "deep learning" is getting more popular since 2000 and, hypothetically, the frequency since 2008 could be much higher although the data is not available. Interestingly, correlation analysis shows that frequencies of these two terms are positively correlated from 1958 to 1998 (with Pearson correlation coefficient=0.916), but negatively correlated from 1998 to 2008 (with Pearson correlation coefficient=-0.973).

This declination of fuzzy logic research can be partially ascribed to the lack of convincing fuzzy logic applications on challenging (machine learning) problems. It was indeed well recognized by the fuzzy (logic) community [62] that research in this field has been restricted to out-of-date problems & approaches while many "hot topics" in machine learning are simply ignored by fuzzy logic scholars. Therefore the connection between the fuzzy logic and the core machine learning community is not well established at all, with very little (if any) interaction in the form of joint meetings, research initiatives or mutual conference attendance.

Fuzzy logic references : The evolution of fuzzy logic societies as well as dedicated conferences, specialized journals, educational programs etc. started with the first volume of the international journal on *Fuzzy Sets and Systems* published in 1978, followed by three important events — the publication of the book by Dubois and Prade [33], and the international Seminar on Fuzzy Set Theory held in Linz, Austria in Sept. 1979, and the publication of a quarterly bulletin known as BUSEFAL which served as a medium for quick communication of new ideas and relevant information [14]. The first IEEE International Conferences on *Fuzzy Systems* (FUZZ-IEEE) was held in 1992 and the first issue of the *IEEE Transaction on Fuzzy Systems* in 1993. Since late 1980, many new journals devoted ot various aspects of fuzzy logic have been founded too. Encyclopaedic resources and handbooks in fuzzy logic include *Handbook of Fuzzy Computation* [115] and recent *Handbook of Mathematical Fuzzy Logic* published in 2011 [28]. Many fuzzy logic definitions reviewed in this article are based on the book of Zimmermann [149] and a recent review book [14].

1.2 Prehistory of fuzzy logic

Although Aristotle (384–322 BC) is usually considered the founder of classical logic, he in fact questioned the applicability of the *principle of bivalence* to propositions concerning future contingencies. He also recognized that certain human categories apply to objects to various degrees and do not have sharp boundaries, a phenomenon which is directly related to the idea of fuzzy logic. The principle was also questioned "even more emphatically by one of his contemporaries, Epicurus (341 – 270 BC) and his followers — Epicureans. These philosophers basically rejected the principle of bivalence on the basis of their strong belief in free will" [14]. In spite of these challenges of the principle of bivalence, Aristotle still treated it as the cornerstone notion of classical logics.

The English philosopher William of Ockham (1287-1347) again questioned the principle of bivalence and introduced three-valued truth table (with neuter N included) in his analysis of chapter 9 of Aristotle's *De interpretatione* and *Topics* [14]. Vagueness of concepts and unsharp boundaries nicely fitted John Locke's empiricism and became recognized in his works. The principle of bivalence was challenged and the ideas of many-valued logic emerged in 19th century with worth-mentioning forerunners like Hugh MacColl, Charles Sanders Peirce and Nikolai A. Vasil'ev [14].

1.3 Many-valued logics

The late 19th and early 20th centuries witnessed various alternative logics which abandoned the principle of bivalence and introduced more than two truth values. The first influential and formal many-valued logics system was developed by Jan Łukasiewicz (1878-1936), who introduced a three-valued logic in 1920 and later generalized it to n-valued logics. He also described the *infinitely-valued logics* when the set of truth values are all real numbers in the unit interval $[0, 1]$. Paul Bernays (1888-1977) and Emil Post (1897-1954) independently invented similar many-valued logics systems at about the same time [14].

Łukasiwicz logics

Łukasiwicz first introduced the three-valued logic in June of 1920, denoted $Ł_3$, by defining a number of truth functions of connectives on a three-element set $\{0, \frac{1}{2}, 1\}$ of truth values. Two basic truth functions of the connectives of *implication*, $\rightarrow$, and *equivalence*, $\longleftrightarrow$, are defined according to tables

$\rightarrow$	0	$\frac{1}{2}$	1
0	1	1	1
$\frac{1}{2}$	$\frac{1}{2}$	1	1
1	0	$\frac{1}{2}$	1

and

$\leftrightarrow$	0	$\frac{1}{2}$	1
0	1	$\frac{1}{2}$	0
$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$
1	0	$\frac{1}{2}$	1

,

followed by three derived truth functions of negation $\neg$, disjunction $\vee$ and conjunction $\wedge$, respectively, according to following tables

φ	$\neg\varphi$
0	1
$\frac{1}{2}$	$\frac{1}{2}$
1	0

,

$\vee$	0	$\frac{1}{2}$	1
0	0	$\frac{1}{2}$	1
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1
1	1	1	1

and

$\wedge$	0	$\frac{1}{2}$	1
0	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$
1	0	$\frac{1}{2}$	1

.

Of particular interests, Łukasiwicz and his students thoroughly investigated the *degrees of tautology* for all 192 classical tautologies defined in Russell's Principia Mathematica [130]. It turned out 3 tautologies have degree values 0, some $1/2$ such as the law of excluded middle, $\varphi \vee \neg\varphi$, or the law of contractions, $\neg(\varphi \wedge \neg\varphi)$, while others remain the same as their classical logic counterparts.

Later in the 1920s, Łukasiwicz generalized his Ł$_3$ logic to n-valued logics by a set Ł$_n$ of equidistant rational numbers in $[0, 1]$:

$$Ł_n = \{0 = \frac{0}{n-1}, \frac{1}{n-1}, \cdots, \frac{n-1}{n-1} = 1\}.$$

Correspondingly, logic operations $\rightarrow, \neg, \vee, \wedge$ can be succinctly generalized by

$$a \rightarrow b = \min(1, 1-a+b), \quad \neg a = 1-a, \quad a \vee b = \min(1, a+b), \quad a \wedge b = \max(0, a+b-1).$$

Łukasiwicz also considered the case when the set of truth values are all real numbers in the unit interval $[0, 1]$ and this infinitely-valued logic, denoted Ł$_\infty$, was axiomatized and proved to be complete later on by [112].

Gödel logics

Kurt Gödel, arguably the greatest logician of the 20th century, introduced a family of many-valued connectives and later Arend Heyting proposed axioms for the so called *intuitionistic propositional logic* with three-valued structures defined as follows [14]:

φ	$\neg\varphi$
0	1
$\frac{1}{2}$	0
1	0

$\rightarrow$	0	$\frac{1}{2}$	1
0	1	1	1
$\frac{1}{2}$	0	1	1
1	0	$\frac{1}{2}$	1

$\vee$	0	$\frac{1}{2}$	1
0	0	$\frac{1}{2}$	1
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1
1	1	1	1

and

$\wedge$	0	$\frac{1}{2}$	1
0	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$
1	0	$\frac{1}{2}$	1

.

Correspondingly, logic operations $\rightarrow, \neg, \vee, \wedge$ can be succinctly generalized by

$$a \rightarrow b = \begin{cases} 1, \text{ if } a \leq b \\ \text{b, if } a > b \end{cases}, \quad \neg a = \begin{cases} 1, \text{ if } a = 0 \\ 0, \text{ if } a > 0 \end{cases}, \quad a \vee b = \max(a, b), \quad a \wedge b = \min(a, b).$$

Gödel then proved that there is no finite structure of truth values such that the tautologies are just the provable formulas of intuitionistic propositional logic. Furthermore, there is an infinite chain of systems between the intuitionistic and classical propositional logics, ordered by inclusion of the sets of their tautologies [14].

Product logic : operations $\rightarrow, \neg, \vee, \wedge$ for this logic are defined by

$$a \rightarrow b = \begin{cases} 1, \text{ if } a \leq b \\ \text{b/a, if } a > b \end{cases}, \quad \neg a = a \rightarrow 0, \quad a \vee b = ab, \quad a \wedge b = a \vee (a \rightarrow b).$$

It must be noted that each t-norm that is continuous as a real function can be obtained by *ordinal sum construction* from three basic t-norms — the Łukasiwicz, the Gödel and the product t-norm, as proved by [95].

1.4 Fuzzy sets and fuzzy logic

Many-valued logic was developed as a branch of abstract logic, nevertheless, there were no apparent applications of the well developed theory since its inception. It is a general agreement that the turning point was due to Zadeh's influential paper [138] and a sequence of follow up publications, which made significant contributions in the development of fuzzy logic. We briefly review below basic concepts of fuzzy logic and refer readers to [14, 138, 149] for thorough treatments of related subjects.

Fuzzy set : In classical logic, the *extension* A of each predicate P is uniquely defined, in other words, object x is a member within P's extension if the proposition "x is P" is true, and vice versa. Formally, the membership can be represented by an *indicator* (or *characteristic*) function $\mu_A : X \rightarrow \{0, 1\}$ defined as

$$\mu_A(x) := \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases},$$

where $A \subseteq X$ is the extension of the predicate P, and X is the *universal set* consisting of all objects that are of interest in a given context.

In fuzzy logic, the degree of truth in the proposition naturally leads to a *degree of membership* of object x in P's extension A. The correspondingly membership function is thus defined as

$$\mu_A : X \rightarrow [0, 1], \tag{1}$$

Now the non-classical set A, whose membership is a matter of degree, is referred to as a *fuzzy set*. Equivalently, the fuzzy set A may refer to the collection of pairs of each element and its membership i.e. $A := \{(x, \mu_A(x)) | x \in X, \mu_A(x) \in [0, 1]\}$[1].

For any fuzzy set A, the set of all objects of X for which $A(x) > 0$ is called a *support* of A, and the set of all objects of X for which $A(x) = 1$ is called a *core* of A. The *height* of fuzzy set A is given by $\mathfrak{h}(A) := \sup_{x \in X}(A(x))$. When $\mathfrak{h}(A) = 1$, the fuzzy set A is called *normal*; otherwise, *subnormal*.

[1] This definition is introduced in [149] and used in Section 2 of this article.

Level cuts : For a given fuzzy set A and a scalar $\alpha \in (0,1]$, the classical set ${}^{\alpha}A := \{x \in X | \mu_A(x) \geq \alpha\}$ is called a level-cut (or an α-cut) of A. Then all distinctive level-cuts form a family $\mathscr{A} = \{{}^{\alpha}A | \alpha \in [0,1]\}$ of nested classical sets. A level-cuts based representation of A is thus given by

$$\mu_A(x) = \sup_{\alpha \in [0,1]} \{\alpha \cdot {}^{\alpha}A(x)\}. \tag{2}$$

This representation is important since it connects fuzzy sets with underlying level-cuts which are nested classical sets. Moreover, the level-cut definition ${}^{\alpha}A := \{x \in X | \mu_A(x) \geq \alpha\}$ is indeed a *set-valued* mapping $F : (0,1] \to X; \alpha \mapsto {}^{\alpha}A$.

Standard operations on fuzzy sets : Zadeh put forward definitions of *union* $A \cup B$, *intersection* $A \cap B$ and *complement* $c(A)$ operations as follows [138]:

$$\begin{aligned}
&\mu_{\cup}(x) = \max\big(\mu_A(x), \mu_B(x)\big), x \in X, \\
&\mu_{\cap}(x) = \min\big(\mu_A(x), \mu_B(x)\big), x \in X, \\
&\mu_c(x) = 1 - \mu_A(x), x \in X.
\end{aligned}$$

While the above definitions are intuitive, they are not unique for different forms of fuzzy logics. It was suggested in the late 1970s to axiomatize intersections and unions operations on fuzzy sets by using triangular norms (t-norms) and triangular conorms (t-conorms or s-norms) [117]. This abstraction gave rise to a set of t-norms and s-norms operators which all satisfy the required axioms listed below. Fuzzy negation is also axiomatized in the same vein. Table 2 below summarizes some example pairs of t-norms and s-norms.

t-norm	$t(x,y) : [0,1] \times [0,1] \to [0,1]$
1. commutative	t(a,b) = t(b,a)
2. associative	t(t(a,b),c) = t(a, t(b,c))
3. monotonicity	$t(a,b) \leq t(a,d)$ if $b \leq d$
4. boundary	t(a,1) = a

,

s-norm	$s(x,y) : [0,1] \times [0,1] \to [0,1]$
1. commutative	s(a,b) = s(b,a)
2. associative	s(s(a,b),c) = s(a, s(b,c))
3. monotonicity	$s(a,b) \leq s(a,d)$ if $b \leq d$
4. boundary	s(a,0) = a

,

fuzzy-neg	$n(x) : [0,1] \to [0,1]$
1. boundary	n(0) = 1, and n(1)=0
2. ordered	$n(a) \leq n(b)$ if $b \geq d$
3. involutivity	n(n(a)) = a

.

Table 1: Axiomatization of fuzzy intersection, union and negation connectives.

t-norm	s-norm
$\min(a,b)$	$\max(a,b)$
$BD(a,b) = \max(0, a+b-1)$	$BS(a,b) = \min(1, a+b)$
$AP(a,b) = ab$	$AS(a,b) = a+b-ab$
$HP(a,b) = \frac{ab}{a+b-ab}$	$HS(a,b) = \frac{a+b-2ab}{1-ab}$
$EP(a,b) = \frac{ab}{2-[a+b-(ab)]}$	$ES(a,b) = \frac{a+b}{1+ab}$

Table 2: Pairs of t-norms and s-norms. BD – bounded difference, BS – bounded sum, AP – Algebraic product, AS – Algebraic sum, HP – Hamacher product, AS – Hamacher sum, EP – Einstein product, ES – Einstein sum.

Ordered weighted averaging : In addition to intersections and unions, fuzzy sets can be aggregated in others ways [48]. In particular, a parametrized class of *ordered weighted averaging* (OWA) operations has been widely used for many applications [132]. An OWA operation, $\mathfrak{h}_{\mathbf{w}}$, is defined with a weighting vector $\mathbf{w} = \{w_1, w_2, \cdots, w_N\}$ for each $w_i \in [0,1], i = 1, 2, \cdots, N$:

$$\mathfrak{h}_{\mathbf{w}}(u_1, u_2, \cdots, u_N) = \sum_{i=1}^{N} w_i \cdot v_i,$$

where v_i denotes the i-th largest value of $u_1, u_2, \cdots, u_N$. Since v_i is the *sorted* u_i, the lower and upper bounds of $\mathfrak{h}_{\mathbf{w}}$ are obtained, respectively, for the weighting vector $\underline{\mathbf{w}} = \{0, 0, \cdots, 1\}$ and $\bar{\mathbf{w}} = \{1, 0, \cdots, 0\}$, or $\min(u_1, u_2, \cdots, u_N)$ and $\max(u_1, u_2, \cdots, u_N)$ correspondingly.

Contrary to intersections and unions, averaging operations defined as such are dedicated to fuzzy sets and not applicable to classical sets — an average of characteristic functions of classical sets is not a characteristic function in general. OWA satisfy four conditions required for meaningful averaging functions i.e. 1) monotonicity; 2) idempotency; 3) continuity; and 4) symmetry [14].

Fuzzy XOR connective : Specific definitions of fuzzy exclusive-OR (XOR) connective were considered in [78, 92], and a generalized XOR operations was introduced as a composition of the fuzzy negation, t-norms and s-norms [106]. An *autonomous*[2] definition for the *fuzzy XOR connective* was finally provided in [10], from which the definition is excerpted as follows.

Definition 1. *A function $E : U^2 \to U$ is a fuzzy XOR if it satisfies the properties:*

1) $E(x,y) = E(y,x)$ *(symmetry)*;
2) $E(x, E(y,z)) = E(E(x,y), z)$ *(associativity)*;
3) $E(0,x) = x$ *(0-Identity)*;
4) $E(1,1) = 0$ *(boundary condition)*.

[2] By autonomous, it means the definition is independent of the other connectives.

[9] examined 12 specific definitions of fuzzy XOR connective and discussed additional properties required for these definitions (see Figure 3 for a summary). Among all these definitions, $x \oplus y = x + y - 2xy$ is of particular interests to our work. This definition is studied extensively and used in the generalized hamming networks (GHN) (see Sections 2 and 3 of this article). Note that fuzzy neural network with fuzzy neurons employing generalized multivalued exclusive-OR (XOR) operations has been proposed in [106], nevertheless, the domain and range of the XOR connective function F are restricted to unit hypercubes i.e. $F : [0,1]^n \to [0,1]^m$ in [106].

Follow on research [144] studied the robustness of fuzzy XOR operator based on the sensitive to small changes in the inputs, and [101] extended the operator to the lattice-valued version.

f-Xor	Properties
1. $E_{KK_\perp}(x,y) = \lvert x-y\rvert$	**E3**, **E4**, **E6**, **E8-E11**, **E14**, **E16**
2. $E_{ML}(x,y) = x + y - 2xy$	**E4**, **E5**, **E8-E12**, **E14-E17**
3. $E_K(x,y) = \max(\min(x, 1-y), \min(1-x, y))$	**E4**, **E5**, **E8-E12**, **E14-E17**
4. $E_{KK_\top}(x,y) = \min(x+y, 2-(x+y))$	**E4-E5**, **E8-E12**, **E14-E17**
5. $E_5(x,y) = \min(\max(x,y), \max(1-x, 1-y))$	**E4**, **E4-E12**, **E17**
6. $E_6(x,y) = \max(x,y) - xy$	**E3-E5**, **E9-E12**, **E14-E17**
7. $E_7(x,y) = \min(x+y, \max(1-x, 1-y))$	**E3**, **E4**, **E8-E12**, **E16**, **E17**
8. $E_8(x,y) = \begin{cases} 0 & \text{if } x = y = 0 \\ \dfrac{\lvert x-y\rvert}{\max(x,y)} & \text{otherwise} \end{cases}$	**E3**, **E6**, **E11**
9. $E_9(x,y) = \begin{cases} 0 & \text{if } x = y \\ \max(1-x, 1-y) & \text{otherwise} \end{cases}$	**E3**, **E9-E11**, **E14**
10. $E_\perp(x,y) = \begin{cases} 1 & \text{if } \lvert x-y\rvert = 1 \\ 0 & \text{otherwise} \end{cases}$	**E3**, **E6**, **E12**, **E14**, **E16**
11. $E_{11}(x,y) = \begin{cases} \min(x, 1-x) & \text{if } x = y \\ 1 & \text{otherwise} \end{cases}$	**E11**, **E13**, **E14**
12. $E_C(x,y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{otherwise} \end{cases}$	**E3**, **E6**, **E11**, **E13**

Fig. 3: Examples of fuzzy XOR connectives [9].

Fuzzy number and fuzzy interval : A *fuzzy interval* is a fuzzy set $A : \mathcal{R} \to [0, 1]$ that satisfies following requirements:

1) level sets of A,${}^{\alpha}A$, are closed intervals of real numbers for all $\alpha \in (0, 1]$,
2) the support of A is bounded,
3) A is a normal fuzzy set,
4) A is a piecewise continuous funcion.

A *fuzzy number* $\tilde{M}$ is thus a normal fuzzy interval which has exactly one member x_0 such that $\mu_{\tilde{M}}(x_0) = 1$. Such x_0 is called the *mean value* of fuzzy number $\tilde{M}$.

Fuzzy arithmetic and the extension principle : Given two fuzzy intervals, A and B, the four fuzzy arithmetic operations are defined by

$$(A * B)(c) = \sup_{c=a*b} \min\{A(a), B(b)\}, \tag{3}$$

where $a, b, c \in \mathcal{R}$ and $*$ denotes any of the four arithmetic operations. Eq. (3), which can be trivially extended to the case with $n > 2$ fuzzy intervals, is referred to as the *extension principle* for fuzzy sets.

The basic ingredients of fuzzy numbers was already introduced in [138]. The concept developed in follow up research was summarized as *fuzzy arithmetic* in a book published in 1985 [71].

Fuzzy relation : A *binary fuzzy relation* is a fuzzy set $R : X \to [0, 1]$, that is defined on the Cartesian products of universes $X = X_1 \times X_2$ by

$$R = \Big\{ \big((x_1, x_2), \mu_R(x_1, x_2)\big) \Big| (x_1, x_2) \in X, \mu_R \in [0, 1] \Big\}, \tag{4}$$

where X_1, X_2 are nonempty classical sets. Note that this definition can be trivially extended to the *N-ary fuzzy relation* where $N > 2$.

Approximate reasoning : The foundations of approximate reasoning based on fuzzy logic was published in a series of connected papers by Zadeh [140], in which the cornerstone notion of *linguistic variable* is formally defined as a tuple of three interrelated components — *base variable* $\mathbf{V}$, a set of *linguistic terms* $L = \{L_i | i = 1, 2, \cdots, N\}$, and a set of *fuzzy sets* $F = \{F_i | i = 1, 2, \cdots, N\}$, which are all defined on $\mathbf{V}$ such that fuzzy sets in F are paired with linguistic terms in L via the common index i. Linguistic variables may involve different types of linguistic terms as well as fuzzy propositional forms, of which we refer readers to [14] for a detailed account.

Take as an example the linguistic variable introduced in Zadeh's paper [140], the base variable "age" in this case has a set of numerical states in the range e.g. $A = [0, 100]$. the set of linguistic terms are *very young*, *young* and *old*, which all refer to the base variable "age". Corresponding fuzzy sets F_i are defined on

the base variable and supposed to represent the meaning of respective linguistic terms (see Figure 4 for Zadeh's original example)[3].

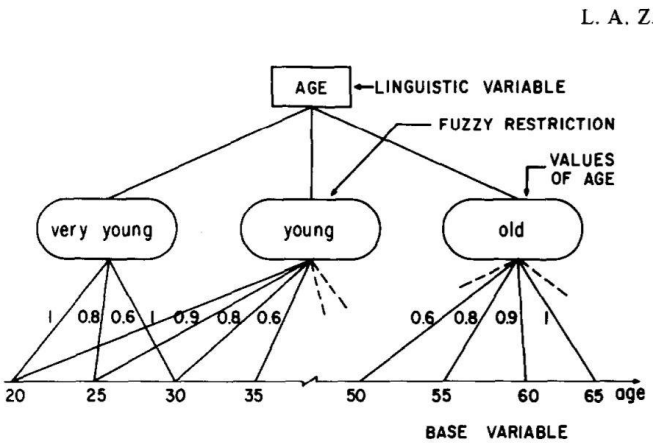


Fig. 4: Hierarchical structure of a linguistic variable.

Based on the defined linguistic variables, the so called *compositional rule of inference* refers to "the process of solving a simultaneous system of so-called relational assignment equations in which linguistic values are assigned to fuzzy restriction" [140]. The compositional rule of inference defined as such actually leads to a *generalized modus ponens* which can be reformulated as:

$$\frac{\begin{array}{l}(\mathbf{V},\mathbf{W}) \text{ is } R,\\ \mathbf{V} \text{ is } A\end{array}}{\mathbf{W} \text{ is } B} \tag{5}$$

where "$\mathbf{V}$ is $A, \mathbf{W}$ is B" are two fuzzy propositions based on the fuzzy sets A, B, and R denotes a fuzzy relation on $V \times W$ with degree of truth in $[0, 1]$. This formulation of inference rule can be implemented by the sup-min composition as $B(w) = \sup_{v \in V} \min(A(v), R(v, w))$ for each $w \in W$.

Methods for constructing fuzzy sets : In order to apply *fuzzy compositional rules of inference* defined above to practical applications, one has to first construct fuzzy sets e.g. A, R in the generalized modus ponens (5) for the given application context. Therefore the need for constructing methods of fuzzy sets, albeit not a problem of fuzzy set theory per se, is of crucial importance for any

[3] Note that in this example fuzzy sets F_i are pre-defined and the issue of how to construct them was not discussed in Zahel's original paper. We will review various methods for constructing (estimating) fuzzy sets below.

real world applications. A great variety of methods of many diverse types have been proposed for this aim, and we review them below following an extensive survey [19] made in 2000.

[12] first proposed a principle approach to address the issue of constructing fuzzy sets based on two assumption:

1. the fuzzy set is exemplified by a finite set of samples;
2. there exists a family of candidate functions, each of which maps from some universe of discourse X to $[0, 1]$.

Under these assumptions, the constructing of fuzzy sets or membership functions is formulated as an optimization problem, i.e. to find the optimal function that best fits the given samples. This *data fitting* style of approach is not unfamiliar to machine learning practitioners, nevertheless, the approach was not chased up further by authors of [12].

Bilgic et. al. provided in [19] a thorough and systematic review to classify various methods into five categories according to different views/interpretations of membership functions. The example fuzzy proposition "John is tall", that has been discussed from the beginning of this article, is again used below for the illustration.

1. *likelihood view*: [57] illustrated a representative method of this category, which considered the likelihood that "John is tall" as the fuzzy membership e.g. $\mu_{\text{tall}}(180cm) = P(\text{tall} \mid x = 180cm) = 0.7$. Empirically, Hisdal proposed to estimate the likelihood by employing a group of subjects to provide their decisions in response to the question "Is John tall?". Then the estimated likelihood is based on the fact that 70% of subjects declared that John is tall by considering his measured height (180cm). Noticeably, Hisdal argued that for a singe subject equipped with perfect information about John's height there cannot be any fuzziness.

2. *random set view*: As illustrated in (2) fuzzy membership functions can be viewed as a nested family of "level-cuts". This so called "horizontal" view therefore allows one to interpret the membership function $\mu_{\text{tall}}(x) = 0.7$ as 70% of the subjects describe "tall" as random sets on X that contain John's measured height. Empirically the membership can be estimated based on the statistics of subjects' responses as in Hisdal's experiments mentioned above.

3. *utility view*: [41] offers a decision-theoretic interpretation in which each fuzzy proposition is associated with a pay-off function. In other word, the fuzzy proposition "John is tall" with membership 0.7 will cost credibility of the speaker if it is far from the truth. By minimizing the overall pay-off, the optimal membership functions is estimated accordingly. It must be noted that the connectives for this view are no longer truth functional, and therefore, none of existing triangular norms and conorms are candidates for disjunction and conjunction.

4. *similarity view*: the similarity view treats the membership as a "degree of similarity" with respect to a perfect (or representative) sample $\bar{x}$ of the set in question [84, 111]. Oftentimes, this similarity view assumes that the *distance* $d(x, \bar{x})$ can be strictly measured such that the membership function is defined accordingly e.g. by $\mu(x) = \frac{1}{1+f(d(x,\bar{x}))}$ (see Section 2.4 of this article for a concrete examples of f). Then the fuzzy proposition $\mu_{\text{tall}}(180cm) = 0.7$ can be interpreted as having a measured distance between the given height i.e. 180cm and the perfect sample of "tallness" such that the membership $\mu(x) = 0.7$.
   While this similarity view is in accordance with many techniques employed in machine learning and pattern recognition applications, it implicitly assumes that there exist 1) a perfect sample; and 2) a continuous *metric space* in which distances can be strictly measured. These two requirements need critical examination before one can commit to the similarity view[4].

5. *measurement view*: Bilgic et. al. argued that the measurement theory introduced in [79, 80, 81, 97] allows one to model fuzzy notions with abstract algebraic structure, and subsequently, map them into numerical structures that can be assigned as memberships. In particular, they argued that one should combine both the membership measurements and property ranking to arrive at *ordered* algebraic structures which can be taken to model fuzzy sets. In essence, their studies showed that the assumption of metric space in the similarity view is not necessary [19].

Noticeably, fuzzy neural network techniques were also adopted in 1990s to come up with estimations of membership functions for a number of tasks such as handwriting recognition [40, 134]. We review below specific neural networks that have been utilized to come up with membership functions from a given set of data. More work related to fuzzy neural networks are reviewed in Section 1.6.

- Jang proposed a hybrid *generalized neural network* (GNN) to incorporate prior knowledge about the original fuzzy system and fine-tune the membership functions of the fuzzy rules, via the gradient-descent-based supervised learning [65]. All experiments are based on 1D *function simulations* with various number of network parameters.

- Takagi & Hayashi proposed to combine an artificial neural network (NN) and fuzzy reasoning so that the learning function and nonlinearity of NN can be exploited to design and adapt membership functions. Different networks with various parameters and layers were tested with two 1D signal simulation problems [125].

- Furukawa & Yamakawa described two algorithms that yield membership functions for a fuzzy neuron and their application to recognition of hand

[4] In Section 3 of this article, we illustrate how to construct such a perfect sample, called *deep epitomes*, from a trained *generalized hamming network*.

writing [40, 134]. This rudimentary research with hand-crafted feature extractions is by no means comparable to the state of the art e.g. GHN based MNIST hand written recognition though [38].

Criticisms : Despite the recognized success and significant developments of fuzzy logic made in 70s and 80s, the topic as a whole has been constantly criticized with critical arguments focused on three aspects:

1. **mathematical foundation**: "the large number of publications under the name of *fuzzy logic* or *fuzzy set theory* is not based on sound mathematical foundations". And in a paper entitled "The paradoxical success of fuzzy logic", Elkan claimed that fuzzy logic actually collapses to classical logic under certain conditions [35]. This paper gave rise to a documented debate which was published in IEEE Expert, with Zadeh and others supporters share their views in response to Eklan's questioning.
2. **applications**: "there is no demonstration of innovative and *non-trivial* applications, for which classical logic is inadequate and fuzzy logic is essential". Applications of fuzzy logic were developed significantly in 1980s to 1990s (see Section 1.6). This criticism, in its original form, was no longer valid. However, the rapid development of deep learning in the new millennium has dwarfed those of fuzzy logic. In this sense, further developments of fuzzy logic applications are urgently needed.
3. **philosophical interpretation**: "probability theory is the only sensible description of uncertainty, thus fuzzy logic is unnecessary". In particular, Lindley held an extreme position by stating "the only satisfactory description of uncertainty is probability. ... alternative descriptions of uncertainty are unnecessary. ... anything that can be done with fuzzy logic, ... can better be done with probability"[14]. Zadeh responded with a number of publications including [139, 142] to defend the importance of fuzzy logic[5].

The following subsections illustrate follow up research from proponents of fuzzy logic, in response to each of these criticisms.

1.5 Foundations and extensions

Through the work with significant contributions towards *mathematics of fuzzy logic* [43, 44], the first above-mentioned critical argument is no longer valid. Some of these contributions e.g. of Goguen and many-valued logics were actually available but not well-known before 1990s. We review below some ground-breaking contributions as well as extensions of fuzzy logic.

[5] In our view, although the interpretation of fuzzy sets (membership function) or degree of truths is intuitive, it should be re-examined from the machine learning point of view. It is one of our motivations for this article to set off exploration along this direction.

Logic of inexact concept : Shortly after the publication of Zadeh's seminal paper [138], his student Goguen provided a solid mathematical and logical treatment to one crucial characteristic of human reasoning i.e. vagueness in the paper entitled "The logic of inexact concepts" [43]. Goguen first cautiously differentiated the calculus of "inexact predicates" (or vagueness) from the calculus of probability: "we are not concerned with the likelihood that a man is short/tall, after many trials; we are concerned with the shortness/tallness of one observation". Goguen then derived the *deduction rule from partially true assumptions* e.g. with the *graded modus ponens* as

$$\frac{\varphi \text{ with degree } a, \quad \varphi \to \psi \text{ with degree } c}{\psi \text{ with degree } a \otimes c}. \tag{6}$$

where $a \otimes b$ is the truth function of a many-valued conjunction and can be implemented as the usual product of $a \cdot b$ as proposed by Goguen. The graded modus ponens in (6) has the desired property that the results of "a long chain of only slightly unreliable deductions can be very unreliable", which helps to resolve the well-known *sorites paradox* (see [14, 43]).

In a related paper [44], Goguen also generalized the product conjunction on unit interval $[0, 1]$ to a complete lattice $< L, \otimes, \wedge, \vee, 0, 1 >$, with an associative operation $\otimes$, infima $\wedge$, suprema $\vee$ and boundary points $0, 1$ defined on the set L of truth degrees. Goguen illustrated required conditions on the lattice, which guarantees the graded modus ponens is both *sound* and *strong* [44].

Meta-mathematics of fuzzy logic : The development of foundations of fuzzy logic was significantly influenced by Hájek's remarkable book *Metamathematics of Fuzzy Logic* [53], in which he nicely put it: "Fuzzy logic is not a poor man's logic nor poor man's probability. Fuzzy logic (in the narrow sense) is a reasonably deep theory" and "Fuzzy logic in the narrow sense is a beautiful logic, but is also important for applications: it offers foundations". Indeed he developed the *basic logic*, denoted BL, which generalizes the three previously developed logics, namely the Łukasiwicz logics Ł, Gödel logics G, and product logic $\sqcap$, based on the three basic continuous t-norms. By adding further axioms to BL, in the same vein, BL also generalizes classical logic [14].

Probabilistic logic : While the classical logic is concerned with inferences from *certainly true* statements to *certainly true* conclusions, there is a long-standing efforts to extend the classical logic to account for *probably true* inference rules. In a broad sense, all such efforts lead to the so called *probabilistic semantics* of propositional logic or *probabilistic logic*. For instance, Łukasiwicz assigned to a proposition φ the ratio between the number of occurrences for which the proposition is true and the number of all occurrences, and called the ratio the *truth value* of the proposition denoted as $\|\varphi\|_p = \frac{\#(true)}{\#(all)}$ [14]. Łukasiwicz then postulated three axioms and derived a number of theorems based on the defined truth value. In this view, a classical tautology is a propositional formula φ with the truth value

$\|\varphi\|_p$=1. By this definition, however, the probabilistic logic is not *truth functional* in the sense that the truth value of composite propositions e.g. $\|\varphi \vee \psi\|_p$ cannot be determined by individual truth values $\|\varphi\|_p$ and $\|\psi\|_p$[6]. This issue or criticism is well-known to researchers of many-valued (fuzzy) logic and the key to resolve the apparent confusion lies in the distinction between *degrees of truth* and *degrees of belief*.

In essence, degrees of belief model "vagueness" while degrees of truth deal with "uncertainty due to lack of evidence". For example, "John is tall" is vague but not uncertain, thus, this proposition is "believed" to be true with certain degree of confidence by the speaker. "John will come tomorrow", on the other hand, is uncertain but not vague. The degree of truth of this proposition is between $[0, 1]$ for now, but will be either 1 or 0 by tomorrow. Many research thus has focused on studying how uncertainties in belief flow from premises to conclusions in deductive inferences [5, 52, 54, 103].

Rough sets : Pawlak proposed to augment a fuzzy set by taking into consideration th indiscernibility between objects [104, 105]. For example, if a group of patients are described by using several symptoms, many patients would share the same symptoms, and hence are indistinguishable in terms of symptoms in question. The indiscernibility is typically characterized by an equivalence relation and rough sets are the results of approximating crisp sets using equivalence classes. Formally indiscernibility may be described by a *reflexive*, *symmetric* and *transitive* equivalence relation $\mathfrak{R} \subseteq X \times X$ on a finite and non-empty universe X [135]. The relation $\mathfrak{R}$ partitions X into a family of disjoint subsets $X/\mathfrak{R}$. For any given relation, if two elements $x\mathfrak{R}y, x, y \in X$ then they are indistinguishable. Probabilistic approaches have been applied to rough sets in the form of decision-theoretic analysis, variable precision analysis and information-theoretic analysis [136]. More recent advances in the field were reviewed in [147].

[6] Considering the case where $\|\varphi\|_p$=0.5 and $\psi = \varphi$, then $\|\varphi \vee \psi\|_p$=0.5; and for the case where $\|\varphi\|_p$=0.5 and $\psi = \neg\varphi$, instead $\|\varphi \vee \psi\|_p = \|\varphi \vee \neg\varphi\|_p$=1. Reichenbach proposed to resolve the non-truth functional by including an additional parameter u to represent the correlation between φ and ψ such that $\|\varphi \vee \psi\|_p = \|\varphi\|_p + \|\psi\|_p - \|\varphi\|_p \cdot u$ [14].

1.6 Applications of fuzzy logic

Fuzzy clustering and pattern recognition : *clustering* is a traditional data analysis technique, which refers to the task of *classifying* a given set of objects into a number of *clusters* according to a set of distinctive *features*. A typical optimization goal of clustering is to put similar objects into the same cluster, while dissimilar ones into different clusters as much as possible. In this sense, clustering may also refer to classification, recognition, categorization or grouping etc. Classical clustering techniques often assign each object into a designated cluster, in other words, each cluster is a classical crisp set with element membership in $\{0, 1\}$. This requirement is abandoned for fuzzy clustering, and therefore, each cluster is a fuzzy set.

Fuzzy clustering or pattern classification was definitely one of the principal motivation for Zadeh to put forward the conception of fuzzy sets in his seminal paper [138] followed by another early paper devoted fully to fuzzy cluster analysis [141]. The early history of fuzzy clustering can also be traced in the book which contains 20 papers covering the principal ideas and different aspects of the topic [18]. The classical k-means clustering method was extended to a fuzzy Iterative Self-Organizing Data Analysis method (ISODATA) and fuzzy c-means clustering method in these early studies [17, 34]. Often a fixed number of clusters and some proximity measure (e.g. Euclidean/Minkowski/Mahalanobis) were adopted for these methods. We refer readers to a comprehensive book on fuzzy clustering analysis for further developments in the field [58].

Fuzzy control : As an important application area of fuzzy logic, a successful fuzzy controller for steam engine based on fuzzy if-then rules was first demonstrated by Mamdani and his student in 1975 [90]. The first fuzzy controller of an *inverted pendulum* was implemented and demonstrated by Yamakawa by using 49 fuzzy if-then rules applied on two input variables (with each having 7 linguistic terms) [133]. This system was a great success when it was demonstrated at the Second IFSA Congress in 1987, as it showed the remarkable robustness in an uncontrolled environment.

Fuzzy decision making : The idea of employing fuzzy logic in decision making (thus coined as fuzzy decision making or FDM) was first introduced in the paper [13], followed by exploded research along this direction [75, 148]. A general fuzzy linear programming model for decision making can be formulated as follows: Maximize $\sum_{j=1}^{n} C_j X_j$ subject to $\sum_{j=1}^{n} A_{ij} X_j \leq B_i (i \in \mathbb{N}_m)$ and $X_j \geq 0$, where A_{ij}, B_i, C_j are fuzzy intervals, X_j are fuzzy intervals representing states of n linguistic variables, and symbols $\leq$ or $\geq$ denote ordering of fuzzy intervals [14]. Depending on the types of decision criteria employed involved in the decision-making, the whole area of FDM are often categorized into a number sub-areas including multi-attribute, multi-objective, and multi-criteria FDM [25, 131]

Fuzzy automata : Finite automata theory is a branch of computer science that deals with abstract machines that follow a predetermined sequence of operations

automatically. A finite automaton has a finite number of states, inputs, outputs, internal and transition functions. Correspondingly, a finite *fuzzy automata* is defined as a quintuple $< I, V, Q, f, g >$ in which I, V, Q are, respectively, ordinary non-empty finite sets of objects in input, output and internal states while f, g, on the other hand, are membership functions of fuzzy sets [129]. A fuzzy automaton behaves in a deterministic fashion. However, it has many properties similar to that of stochastic automata. Fuzzy auotomata may also handle continuous spaces thus are able to model uncertainty in many applications [32]. Mathematically, an isomorphism between a category of fuzzy automata and a category of chains of non-deterministic automata was proved in [94]. Also, deterministic fuzzy automata and nondeterministic fuzzy automata are all equivalent in the sense that they recognize the same class of fuzzy languages [24].

Databases and information retrieval : an important notion in application of fuzzy logic is the *fuzz relational database* [22], which extends entries of the classical relation $R \subset D_1 \times \cdots \times D_n$ from a single value to a whole set of values indicating the graded relationships i.e. $R \subset 2^{D_1} \times \cdots \times 2^{D_n}$. Correspondingly, each domain is equipped with a *fuzzy similarity relation*, which allows vague and natural similarity queries like "look for houses *near* New York and cost *approximately* \$200,000". [109] later considered a rational database table as a fuzzy relation $R : D_1 \times D_n \to [0, 1]$, and [37] introduced the groundbreaking and powerful threshold algorithm which is widely used in numerous applications and systems [14].

Fuzzy Neural Network (FNN) : Since 1980s the integration of fuzzy logic and computational *neural networks* has given birth to the *fuzzy neural networks* (FNN) [51] which, on the one hand, attempted to furnish neural networks with the interpretability of fuzzy logic [14, 138, 149]. On the other hand, neural networks have been used as a computational tool to come up with both *membership functions* and fuzzy inference rules [40, 134]. A fuzzy inference system implemented in the framework of adaptive networks by using a hybrid learning procedure was proposed in [66] and follow up research between 2002 to 2012 were reviewed in [69]. In essence, a *fuzzy neuron* of FNNs is designed to take *fuzzy sets as inputs* and aggregate inputs by *fuzzy aggregation operations* (e.g. fuzzy union, fuzzy intersected and/or OWA) [39]. In terms of system architecture, [124] reviewed thoroughly different integration models of neural network and fuzzy systems (NN+FS), and classified them into 9 categories which are illustrated in Figure 5 below.

FNN with deep learning : The so called "soft computing" joint force endeavour remains active in the new millennium e.g. in [60, 70, 87, 98, 107]. Nevertheless, FNNs have been largely overlooked nowadays by scholars and engineers in machine learning (ML) community, partially due to the lack of convincing demonstrations on ML problems with large datasets. The exceptional case is

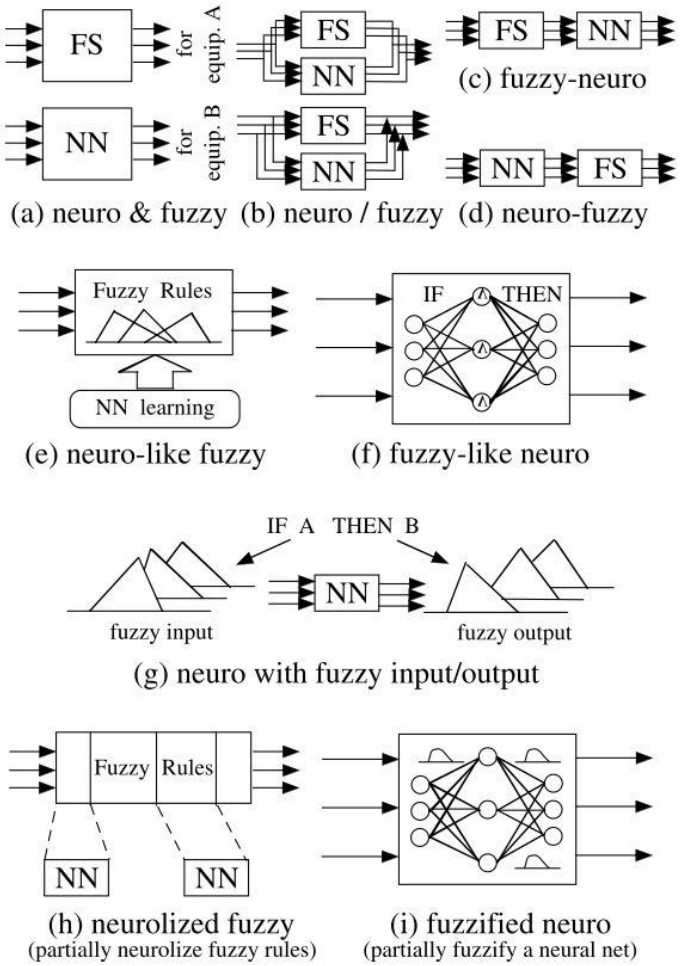


Fig. 5: Hayashi & Umono's categorization of different integration models of neural network and fuzzy system (NN+FS) [124].

the recent NIPS publication [38], which re-interpreted ReLU and batch normalization within a novel Generalized Hamming Network (GHN) framework and demonstrated the state-of-the-art performances on a variety of machine learning tasks.

The proposed *generalized hamming network* (GHN) forms its foundation on the cornerstone notion of *generalized hamming distance* (GHD), which is essentially defined as $h(x, w) := x + w - 2xw$ for any $x, w \in \mathbb{R}$. Its connection with the inferencing rule in neural computing is obvious: the last term $(-2xw)$ corresponds to element-wise multiplications of neuron inputs and weights, and since we aim to measure the GHD between inputs x and weights w, the bias term then should take the value $x + w$. Since the underlying GHD induces a fuzzy XOR connective, GHN lends itself to rigorous analysis within the fuzzy logics theory. GHN also benefits from the rapid developments of neural computing techniques, in particular, those employing parallel computing on GPUs. Due to this efficient implementation of GHNs, it is the first fuzzy neural network that has demonstrated state-of-the-art performances on learning tasks with large scale datasets.

2 Revisit fuzzy neural network with generalized hamming network

In this section, we revisit fuzzy neural network with a cornerstone notion of *generalized hamming distance* which provides a novel and theoretically justified framework to re-interpret many useful neural network techniques in terms of fuzzy logic. In particular, we conjecture and empirically illustrate that, the celebrated batch normalization (BN) technique actually adapts the "normalized" bias such that it approximates the rightful bias induced by the generalized hamming distance. Once the due bias is enforced analytically, neither the optimization of bias terms nor the sophisticated batch normalization is needed. Also in the light of generalized hamming distance, the popular rectified linear units (ReLU) can be treated as setting a minimal hamming distance threshold between network inputs and weights. This thresholding scheme, on the one hand, can be improved by introducing double-thresholding on both positive and negative extremes of neuron outputs. On the other hand, ReLUs turn out to be *non-essential* and can be removed from networks trained for simple tasks like MNIST classification. The proposed *generalized hamming network* (GHN) as such not only lends itself to rigorous analysis and interpretation within the fuzzy logic theory but also demonstrates fast learning speed, well-controlled behaviour and state-of-the-art performances on a variety of learning tasks.

2.1 Introduction

Since early 1990s the integration of fuzzy logic and computational neural networks has given birth to the *fuzzy neural networks* (FNN) [51]. While the formal fuzzy set theory provides a strict mathematical framework in which vague conceptual phenomena can be precisely and rigorously studied [7, 11, 126, 138], application-oriented fuzzy technologies lag far behind theoretical studies. In particular, fuzzy neural networks have only demonstrated limited successes on some toy examples such as [107, 127]. In order to catch up with the rapid advances in recent neural network developments, especially those with deep layered structures, it is the goal of this paper to demonstrate the relevance of FNN, and moreover, to provide a novel view on its non-fuzzy counterparts.

Our revisiting of FNN is not merely for the fond remembrances of the golden age of "soft computing" [150]. Instead it provides a novel and theoretically justified perspective of neural computing, in which we are able to re-examine and demystify some useful techniques that were proposed to improve either effectiveness or efficiency of neural networks training processes. Among many others, *batch normalization* (BN) [63] is probably the most influential yet mysterious trick, that significantly improved the training efficiency by adapting to the change in the distribution of layers' inputs (coined as *internal covariate shift*). Such kind of adaptations, when viewed within the fuzzy neural network framework, can be interpreted as rectifications to the deficiencies of neuron outputs with respect to the rightful *generalized hamming distance* (see definition 2) between inputs and neuron weights. Once the appropriate rectification is applied , the ill effects

of internal covariate shift are automatically eradicated, and consequently, one is able to enjoy the fast training process without resorting to a sophisticated learning method used by BN.

Another crucial component in neural computing, Rectified linear unit (ReLU), has been widely used due to its strong biological motivations and mathematical justifications [4, 42, 108]. We show that within the *generalized hamming group* endowed with generalized hamming distance, ReLU can be regarded as setting a minimal hamming distance threshold between network input and neuron weights. This novel view immediately leads us to an effective double-thresholding scheme to suppress fuzzy elements in the generalized hamming group.

The proposed *generalized hamming network* (GHN) forms its foundation on the cornerstone notion of *generalized hamming distance* (GHD), which is essentially defined as $h(x, w) := x + w - 2xw$ for any $x, w \in \mathbb{R}$ (see definition 2). Its connection with the inferencing rule in neural computing is obvious: the last term $(-2xw)$ corresponds to element-wise multiplications of neuron inputs and weights, and since we aim to measure the GHD between inputs x and weights w, the bias term then should take the value $x + w$. In this article we define any network that has its neuron outputs fulfilling this requirement (13) as a *generalized hamming network*. Since the underlying GHD induces a fuzzy XOR logic, GHN lends itself to rigorous analysis within the fuzzy logics theory (see definition 5). Apart from its theoretical appeals, GHN also demonstrates appealing features in terms of fast learning speed, well-controlled behaviour and simple parameter settings (see Section 2.5).

2.2 Related work

Fuzzy logic and fuzzy neural network: the notion of fuzzy logic is based on the rejection of the fundamental *principle of bivalence* of classical logic i.e. any declarative sentence has only two possible truth values, *true* and *false*. Although the earliest connotation of fuzzy logic was attributed to Aristotle, the founder of classical logic [14], it was Zadeh's publication in 1965 that ignited the enthusiasm about the theory of fuzzy sets [138]. Since then mathematical developments have advanced to a very high standard and are still forthcoming to day [7, 11, 126]. *Fuzzy neural networks* were proposed to take advantages of the flexaible knowledge acquiring capability of neural networks [51, 88]. In theory it was proved that fuzzy systems and certain classes of neural networks are equivalent and convertible with each other [16, 64]. In practice, however, successful applications of FNNs are limited to some toy examples only [107, 127].

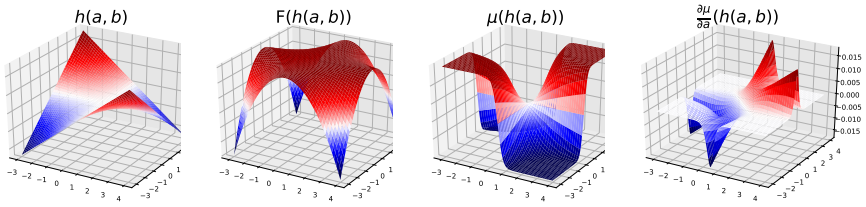
Demystifying neural networks: efforts of interpreting neural networks by means of propositional logic dated back to McCulloch & Pitts' seminial paper [91]. Recent research along this line include [60] and the references therein, in which First Order Logic (FOL) rules are encoded using soft logic on continuous truth values from the interval $[0, 1]$. These interpretations, albeit interesting, seldom explain effective neural network techniques such as batch normalization or ReLU. Recently [116] provided an improvement (and explanation) to batch

normalization by removing dependencies in weight normalization between the examples in a minibatch.

Binary-valued neural network: Restricted Boltzmann Machine (RBM) was used to model an "ensemble of binary vectors" and rose to prominence in the mid-2000s after fast learning algorithms were demonstrated by Hinton et. al. [56, 96]. Recent binarized neural network [30, 110] approximated standard CNNs by binarizing filter weights and/or inputs, with the aim to reduce computational complexity and memory consumption. The XNOR operation employed in [110] is limited to binary hamming distance and not readily applicable to non-binary neuron weights and inputs.

Ensemble of binary patterns: the distributive property of GHD described in (7) provides an intriguing view on neural computing – even though real-valued pattens are involved in the computation, the computed GHD is strictly equivalent to the mean of binary hamming distances across two *ensembles of binary patterns*! This novel view illuminates the connection between generalized hamming networks and efficient binary features, that have long been used in various computer vision tasks, for instance, the celebrated Adaboost face detection[128], numerous binary features for key-point matching [23, 113] and binary codes for large database hashing [83, 86, 99, 100].

2.3 Generalized Hamming Distance



(a) (b) (c) (d)

Fig. 6: (a) $h(a,b)$ has one *fuzzy region* near the identity element 0.5 (in white), two *positively confident* (in red) and two *negatively confident* (in blue) regions from above and below, respectively. (b) Fuzziness $F(h(a,b)) = h(a,b) \oplus h(a,b)$ has its maxima along $a = 0.5$ or $b = 0.5$. (c) $\mu(h(a,b)) : U \rightarrow I$ where $\mu(h) = 1/(1 + exp(0.5 - h))$ is the logistic function to assign membership to fuzzy set elements (see definition 5). (d) partial derivative of $\mu(h(a,b))$. Note that magnitudes of gradient in the fuzzy region is non-negligible.

Definition 2. *Let $a, b, c \in U \subseteq \mathbb{R}$, and a generalized hamming distance (GHD), denoted by $\oplus$, be a binary operator $h : U \times U \rightarrow U$;* $\boxed{h(a,b) := a \oplus b = a + b - 2 \cdot a \cdot b}$. *Then*

(i) for $U = \{0, 1\}$ GHD de-generalizes to binary hamming distance with $0 \oplus 0 = 0; 0 \oplus 1 = 1; 1 \oplus 0 = 1; 1 \oplus 1 = 0$;

(ii) for $U = [0.0, 1.0]$ the unitary interval I, $a \oplus b \in I$ (closure);
Remark: this case is referred to as the "restricted" hamming distance, in the sense that inverse of any elements in I are not necessarily contained in I (see below for definition of inverse).

(iii) for $U = \mathbb{R}$, $\mathcal{H} := (\mathbb{R}, \oplus)$ is a group satisfying five *abelian group* axioms, thus is referred to as the *generalized hamming group* or *hamming group*:
- $a \oplus b = (a + b - 2 \cdot a \cdot b) \in \mathbb{R}$ (closure);
- $a \oplus b = (a + b - 2 \cdot a \cdot b) = b \oplus a$ (commutativity);
- $(a \oplus b) \oplus c = (a + b - 2 \cdot a \cdot b) + c - 2(a + b - 2 \cdot a \cdot b)c$
  $= a + (b + c - 2 \cdot b \cdot c) - 2 \cdot a \cdot (b + c - 2 \cdot b \cdot c) = a \oplus (b \oplus c)$ (associativity);
- $\exists e = 0 \in \mathbb{R}$ such that $e \oplus a = a \oplus e = (0 + a - 2 \cdot 0 \cdot a) = a$ (identity element);
- for each $a \in \mathbb{R} \setminus \{0.5\}$, $\exists a^{-1} := a/(2 \cdot a - 1)$ s.t. $a \oplus a^{-1} = (a + \frac{a}{2 \cdot a - 1} - 2a \cdot \frac{a}{2 \cdot a - 1})$
  $= 0 = e$; and we define $\infty := (0.5)^{-1}$ (inverse element).

Remark: note that $1 \oplus a = 1 - a$ which *complements* a. "0.5" is a fixed point since $\forall a \in \mathbb{R}, 0.5 \oplus a = 0.5$, and $0.5 \oplus \infty = 0$ according to definition[7].

(iv) GHD naturally leads to a measurement of *fuzziness*: $F(a) := a \oplus a, \mathbb{R} \to (-\infty, 0.5] : F(a) \geq 0, \forall a \in [0, 1]; F(a) < 0$ otherwise. Therefore $[0, 1]$ is referred to as the *fuzzy region* in which $F(0.5) = 0.5$ has the maximal fuzziness and $F(0) = F(1) = 0$ are two boundary points. Outer regions $(-\infty, 0]$ and $[1, \infty)$ are negative and positive *confident regions* respectively. See Figure 6 (a) for the surface of $h(a, b)$ which has one central *fuzzy region*, two *positive confident* and two *negative confident* regions.

(v) The *direct sum* of hamming group is still a hamming group $\mathcal{H}^L := \oplus_{l \in L} \mathcal{H}_l$: let $\mathbf{x} = \{x_1, \ldots, x_L\}, \mathbf{y} = \{y_1, \ldots, y_L\} \in \mathcal{H}^L$ be two group members, then the *generalized hamming distance* is defined as the arithmetic mean of element-wise GHD: $\mathcal{G}^L(\mathbf{x} \oplus^L \mathbf{y}) := \frac{1}{L}(x_1 \oplus y_1 + \ldots + x_L \oplus y_L)$.
And let $\tilde{x} = (x_1 + \ldots x_L)/L, \tilde{y} = (y_1 + \ldots y_L)/L$ be arithmetic means of respective elements, then $\boxed{\mathcal{G}^L(\mathbf{x} \oplus^L \mathbf{y}) = \tilde{x} + \tilde{y} - \frac{2}{L}(\mathbf{x} \cdot \mathbf{y})}$, where $\mathbf{x} \cdot \mathbf{y} = \sum_{l=1}^{L} x_l \cdot y_l$ is the dot product.

(vi) *Distributive property*: let $\bar{\mathbf{X}}^M = (\mathbf{x}^1 + \ldots \mathbf{x}^M)/M \in \mathcal{H}^L$ be *element-wise arithmetic mean* of a set of members $\mathbf{x}^m \in \mathcal{H}^L$, and $\bar{\mathbf{Y}}^N$ be defined in the same vein. Then GHD is *distributive*:

$$\begin{aligned} \mathcal{G}^L(\bar{\mathbf{X}}^M \oplus^L \bar{\mathbf{Y}}^N) = \frac{1}{L}\sum_{l=1}^{L} \bar{x}_l \oplus \bar{y}_l = \frac{1}{M}\frac{1}{N}\frac{1}{L}\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{l=1}^{L} x_l^m \oplus y_l^n \\ = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N} \mathcal{G}^L(\mathbf{x}^m \oplus^L \mathbf{y}^n). \end{aligned} \tag{7}$$

[7] By this extension, it is $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ instead of $\mathbb{R}$ on which we have all group members.

Remark: in case that $x_l^m, y_l^n \in \{0,1\}$ i.e. for two sets of binary patterns, the *mean of binary hamming distance* between two sets can be efficiently computed as the GHD between two real-valued patterns $\bar{\mathbf{X}}^M, \bar{\mathbf{Y}}^N$. Conversely, a real-valued pattern can be viewed as the element-wise average of an ensemble of binary patterns.

2.4 Generalized Hamming Network

Despite the recent progresses in deep learning, artificial neural networks has long been criticized for its "black box" nature: "they capture *hidden* relations between inputs and outputs with a highly accurate approximation, but no definitive answer is offered for the question of how they work" [16]. In this section we provide an interpretation on neural computing by showing that, if the condition specified in (13) is fulfilled, outputs of each neuron can be strictly defined as the generalized hamming distance between inputs and weights. Moreover, the computations of GHD induces fuzzy implication of XOR connective, and therefore, the inferencing of entire network can be regarded as a logical calculus in the same vein as described in McCulloch & Pitts' seminial paper [91].

New perspective on neural computing The bearing of *generalized hamming distance* on neural computing is elucidated by looking at the negative of generalized hamming distance, (GHD, see definition 2), between inputs $\mathbf{x} \in \mathcal{H}^L$ and weights $\mathbf{w} \in \mathcal{H}^L$ in which L denotes the length of neuron weights e.g. in convolution kernels:

$$-\mathcal{G}^L(\mathbf{w} \oplus^L \mathbf{x}) = \frac{2}{L}\mathbf{w} \cdot \mathbf{x} - \frac{1}{L}\sum_{l=1}^{L} w_l - \frac{1}{L}\sum_{l=1}^{L} x_l \tag{8}$$

Divide (12) by the constant $\frac{2}{L}$ and let

$$\boxed{b = -\frac{1}{2}\Big(\sum_{l=1}^{L} w_l + \sum_{l=1}^{L} x_l\Big)} \tag{9}$$

then it becomes the familiar form ($\mathbf{w}\cdot\mathbf{x}+b$) of neuron outputs save the non-linear activation function. By enforcing the bias term to take the given value in (13), standard neuron outputs measure negatives of GHD between inputs and weights. Note that, for each layer, the bias term $\sum_{l=1}^{L} x_l$ is averaged over neighbouring neurons in individual input image. The bias term $\sum_{l=1}^{L} w_l$ is computed separately for each filter in fully connected or convolution layers. When weights are updated during the optimization, $\sum_{l=1}^{L} w_l$ changes accordingly to keep up with weights and maintain stable neuron outputs. We discuss below (re-)interpretations of neural computing in terms of GHD.

Fuzzy inference: As illustrated in definition 5 GHD induces a fuzzy XOR connective. Therefore the negative of GHD quantifies the *degree of equivalence*

between inputs $\mathbf{x}$ and weights $\mathbf{w}$ (see definition 5 of fuzzy XOR), i.e. the fuzzy truth value of the statement "$\mathbf{x} \leftrightarrow \mathbf{w}$" where $\leftrightarrow$ denotes a fuzzy equivalence relation. For GHD with multiple layers stacked together, neighbouring neuron outputs from the previous layer are integrated to form composite statements e.g. "$(\mathbf{x}_1^1 \leftrightarrow \mathbf{w}_1^1, \ldots, \mathbf{x}_i^1 \leftrightarrow \mathbf{w}_i^1) \leftrightarrow \mathbf{w}_j^2$" where superscripts correspond to two layers. Thus stacked layers will form more complex, and hopefully more powerful, statements as the layer depth increases.

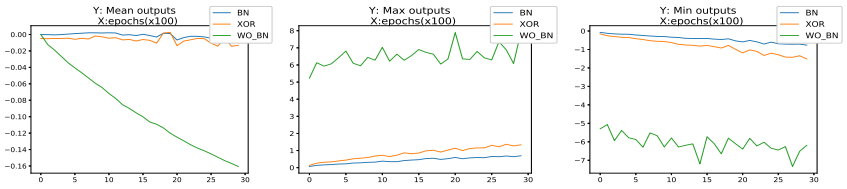


Fig. 7: Left to right: mean, max and min of neuron outputs, with/without batch normalized (BN, WO_BN) and generalized hamming distance (XOR). Outputs are averaged over all 64 filters in the first convolution layer and plotted for 30 epochs training of a MNIST network used in our experiment (see Section 2.5).

Batch normalization demystified: When a mini-batch of training samples $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^M\}$ is involved in the computation, due to the distributive property of GHD, the data-dependent bias term $\sum_{l=1}^{L} x_l$ equals the arithmetic mean of corresponding bias terms computed for each sample in the mini-batch i.e. $\frac{1}{M}\sum_{m=1}^{M}\sum_{l=1}^{L} x_l^m$. It is almost impossible to maintain a constant scalar b that fulfils this requirement when mini-batch changes, especially at deep layers of the network whose inputs are influenced by weights of incoming layers. The celebrated batch normalization (BN) technique therefore proposed a learning method to compensate for the input vector change, with additional parameters γ, β to be learnt during the training [63]. It is our conjecture that batch normalization is approximating these rightful bias through optimization, and this connection is empirically revealed in Figure 7 with very similar neuron outputs obtained by BN and GHD. Indeed they are highly correlated during the course of training (with Pearson correlation coefficient=0.97), confirming our view that BN is attempting to influence the bias term according to (13).

Once b is enforced to follow (13), *neither the optimization of bias terms nor the sophisticated learning method of BN is needed.* In the following section we will illustrate a rectified neural network designed as such.

Rectified linear units (ReLU) redesigned: Due to its strong biological motivations [4] and mathematical justifications [108], rectified linear unit (ReLu) is the most popular activation function used for deep neural network [85]. If neuron outputs are rectified as the generalized hamming distances, the

activation function $max(0, 0.5 - h(\mathbf{x}, \mathbf{w}))$ then simply sets a minimal hamming distance threshold of 0.5 (see Figure 6). Astute readers may immediately spot two limitations of this activation function: a) it only takes into account the negative confidence region while disregards positive confidence regions; b) it allows elements in the fuzzy regime near 0.5 to misguide the optimization with their non-negligible gradients.

A straightforward remedy to ReLU is to suppress elements within the fuzzy region by setting outputs between $[0.5 - r, 0.5 + r]$ to 0.5, where r is a parameter to control acceptable fuzziness in neuron outputs. In particular, we may set thresholds adaptively e.g. $[0.5 - r \cdot O, 0.5 + r \cdot O]$ where O is the maximal magnitude of neuron outputs and the threshold ratio r is adjusted by the optimizer. This *double-thresholding* strategy effectively prevents noisy gradients of fuzzy elements, since 0.5 is a fixed point and $x \oplus 0.5 = 0.5$ for any x. Empirically we found this scheme, in tandem with the rectification (13), dramatically boosts the training efficiency for challenging tasks such as CIFAR10/100 image classification. It must be noted that, however, the use of non-linear activation as such is *not essential* for GHD-based neural computing. When the double-thresholding is switched-off (by fixing $r = 0$), the learning is prolonged for challenging CIFAR10/100 image classification but its influence on the simple MNIST classification is almost negligible (see Section 2.5 for experimental results).

Ganeralized hamming network with induced fuzzy XOR

Definition 3. *A generalized hamming network (GHN) is any networks consisting of neurons, whose outputs* $\mathbf{h} \in \mathcal{H}^L$ *are related to neuron inputs* $\mathbf{x} \in \mathcal{H}^L$ *and weights* $\mathbf{w} \in \mathcal{H}^L$ *by* $\boxed{\mathbf{h} = \mathbf{x} \oplus^L \mathbf{w}}$.

Remark: In case that the bias term is computed directly from (13) such that $\mathbf{h} = \mathbf{x} \oplus^L \mathbf{w}$ *is fulfilled strictly, the network is called a rectified GHN or simply a GHN. In other cases where bias terms are approximating the rightful offsets (e.g. by batch normalization [63]), the trained network is called an approximated GHN.*

Compared with traditional neural networks, the optimization of bias terms is no longer needed in GHN. Empirically, it is shown that the proposed GHN benefits from a fast and robust learning process that is on par with that of the batch-normalization approach, yet without resorting to sophisticated learning process of additional parameters (see Section 2.5 for experimental results). On the other hand, GHN also benefits from the rapid developments of neural computing techniques, in particular, those employing parallel computing on GPUs. Due to this efficient implementation of GHNs, it is the first time that fuzzy neural networks have demonstrated state-of-the-art performances on learning tasks with large scale datasets.

Often neuron outputs are clamped by a logistic activation function to within the range $[0, 1]$, so that outputs can be compared with the target labels in supervised learning. As shown below, GHD followed by such a non-linear activation actually induces a fuzzy XOR connective. We briefly review basic notion of fuzzy

set used in our work and refer readers to [14, 138, 149] for thorough treatments and review of the topic.

Definition 4. ***Fuzzy Set:*** *Let X be an universal set of elements $x \in X$, then a fuzzy set A is a set of pairs: $A := \{\big(x, \mu_A(x)\big) | x \in X, \mu_A(x) \in I\}$, in which $\mu_A : X \to I$ is called the membership function (or grade membership).*

Remark: In this work we let X be a Cartesian product of two sets $X = P \times U$ where P are (2D or 3D) collection of neural nodes and U are real numbers in $\subseteq I$ or $\subseteq R$. We define the membership function $\mu_X(x) := \mu_U(x_p), \forall x = (p, x_p) \in X$ such that it is dependent on x_p only. For the sake of brevity we abuse the notation and use $\mu(x)$, $\mu_X(x)$ and $\mu_U(x_p)$ interchangeably.

Definition 5. Induced fuzzy XOR*: let two fuzzy set elements $a, b \in U$ be assigned with respective grade or membership by a membership function $\mu : U \to I : \mu(a) = i, \mu(b) = j$, then the generalized hamming distance $h(a, b) : U \times U \to U$ induces a fuzzy XOR connective $E : I \times I \to I$ whose membership function is given by*

$$\mu_R(i, j) = \mu(h(\mu^{-1}(i), \mu^{-1}(j))). \tag{10}$$

Remark: For the restricted case $U = I$ the membership function can be trivially defined as the identity function $\mu = \mathrm{id}_I$ as proved in [11].

Remark: For the generalized case where $U = \mathbb{R}$, the fuzzy membership μ can be defined by a sigmoid function such as logistic, *tanh* or any function $: U \to I$. In this work we adopt the logistic function $\mu(a) = \frac{1}{1+\exp(0.5-a)}$ and the resulting fuzzy XOR connective is given by following membership function:

$$\mu_R(i, j) = \frac{1}{1 + \exp\big(0.5 - \mu^{-1}(i) \oplus \mu^{-1}(j)\big)}, \tag{11}$$

where $\mu^{-1}(a) = -\ln(\frac{1}{a} - 1) + \frac{1}{2}$ is the inverse of $\mu(a)$. Following this analysis, it is possible to rigorously formulate neuron computing of the entire network according to inference rules of fuzzy logic theory (in the same vein as illustrated in [91]). Nevertheless, research along this line is out of the scope of the present article and will be reported elsewhere.

2.5 Performance evaluation

Generalized Hamming Networks were tested with four learning tasks, namely MNIST image classification, CIFAR10/100 image classification, Variational autoencoding, and sentence classification.

A case study with MNIST image classification :

Overall performance: we tested a simple four-layered GHN (cv[1,5,5,16]-pool-cv[16,5,5,64]-pool-fc[1024]-fc[1024,10]) on the MNIST dataset with 99.0% test accuracy obtained. For this relatively simple dataset, GHN is able to reach test

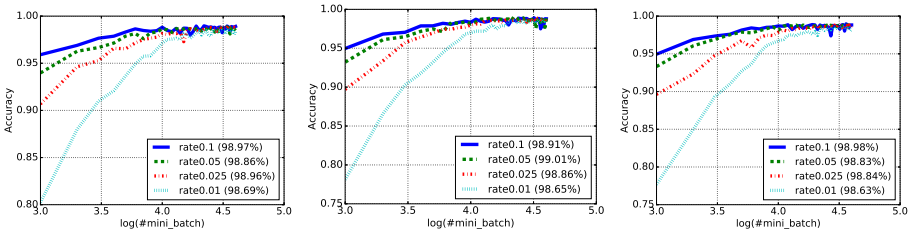


Fig. 8: Test accuracies of MNIST classification with Generalized Hamming Network (GHN). Left: test accuracies without using non-linear activation (by setting $r = 0$). Middle: with r optimized for each layer. Right: with r optimized for each filter. Four learning rates i.e. $\{0.1, 0.05, 0.025, 0.01\}$ are used for each case with the final accuracy reported in brackets. Note that the number of mini-batch are in logarithmic scale along x-axis.

accuracies above 0.95 with 1000 mini-batches and a learning rate 0.1. This learning speed is on par with that of the batch normalization (BN), but without resorting to the learning of additional parameters in BN. It was also observed a wide range of large learning rates (from 0.01 to 0.1) all resulted in similar final accuracies (see below). We ascribe this well-controlled robust learning behaviour to rectified bias terms enforced in GHNs.

Influence of learning rate: This experiment compares performances with different learning rates and Figure 8 (middle,right) show that a very large learning rate (0.1) leads to much faster learning without the risk of divergences. A small learning rate (0.01) suffice to guarantee the comparable final test accuracy. Therefore we set the learning rate to a constant 0.1 for all experiments unless stated otherwise.

Influence of non-linear double-thresholding: The non-linear double-thresholding can be turned off by setting the threshold ratio $r = 0$ (see texts in Section 2.4). Optionally the parameter r is automatically optimized together with the optimization of neuron weights. Figure 8 (left) shows that the GHN without non-linear activation (by setting $r = 0$) performs equally well as compared with the case where r is optimized (in Figure 8 left, right). There are no significant differences between two settings for this relative simple task.

CIFAR10/100 image classification :

In this experiment, we tested a six-layered GHN (cv[3,3,3,64]-cv[64,5,5,256]-pool-cv[256,5,5,256]-pool-fc[1024]-fc[1024,512]-fc[1024,nclass]) on both CIFAR10 (nclass=10) and CIFAR100 (nclass=100) datasets. Figure 9 shows that the double-thresholding scheme improves the learning efficiency dramatically for these challenging image classification tasks: when the parameter r is optimized for each feature filter the numbers of iterations required to reach the same level of test accuracy are reduced by 1 to 2 orders of magnitudes. It must be noted

that performances of such a simple generalized hamming network (89.3% for CIFAR10 and 60.1% for CIFAR100) are on par with many sophisticated networks reported in [Benenson]. In our view, the rectified bias enforced by (13) can be readily applied to these sophisticated networks, although resulting improvements may vary and remain to be tested.

Generative modelling with Variational Autoencoder :

In this experiment, we tested the effect of rectification in GHN applied to a generative modelling setting. One crucial difference is that the objective is now to minimize reconstruction error instead of classification error. It turns out the double-thresholding scheme is no longer relevant for this setting and thus not used in the experiment.

The baseline network (784-400-400-20) used in this experiment is an improved implementation [3] of the influential paper [77], trained on the MNIST dataset of images of handwritten digits. We have rectified the outputs following (13) and, instead of optimizing the lower bound of the log marginal likelihood as in [77], we directly minimize the reconstruction error. Also we did not include weights regularization terms for the optimization as it is unnecessary for GHN. Figure 10 (left) illustrates the reconstruction error with respect to number of training steps (mini-batches). It is shown that the rectified generalized hamming network converges to a lower minimal reconstruction error as compared to the baseline network, with about 28% reduction. The rectification also leads to a faster convergence, which is in accordance with our observations in other experiments.

Sentence classification :

A simple CNN has been used for sentence-level classification tasks and excellent results were demonstrated on multiple benchmarks [76]. The baseline network used in this experiment is a re-implementation of [76] made available from [2]. Figure 10 (right) plots accuracy curves from both networks. It was observed that the rectified GHN did improve the *learning speed*, but did not improve the final accuracy as compared with the baseline network: both networks yielded the final evaluation accuracy around 74% despite that the training accuracy were almost 100%. The over-fitting in this experiment is probably due to the relatively small Movie Review dataset size with 10,662 example review sentences, half positive and half negative.

2.6 Conclusion

In summary, we proposed a rectified *generalized hamming network* (GHN) architecture which materializes a re-emerging principle of fuzzy logic inferencing. This principle has been extensively studied from a theoretic fuzzy logic point of view, but has been largely overlooked in the practical research of ANN. The rectified

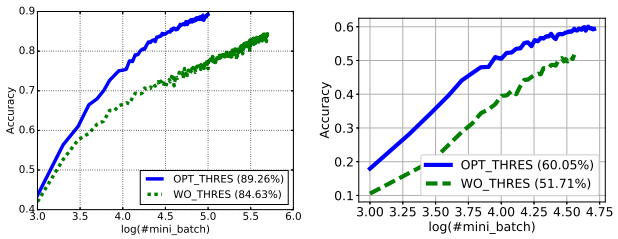


Fig. 9: Left: GHN test accuracies of CIFAR10 classification (OPT THRES: parameter r optimized; WO THRES: without nonlinear activation). Right: GHN test accuracies of CIFAR100 classification(OPT THRES: parameter r optimized; WO THRES: without non-linear activation).

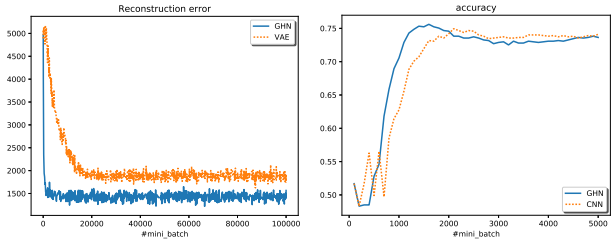


Fig. 10: Left: Reconstruction errors of convolution VAE with and w/o rectification. Right: Evaluation accuracies of Sentence classification with GHN rectification and w/o rectification).

neural network derives fuzzy logic implications with underlying *generalized hamming distances* computed in neuron outputs. Bearing this rectified view in mind, we proposed to compute bias terms analytically without resorting to sophisticated learning methods such as batch normalization. Moreover, we have shown that, the rectified linear units (ReLU) was theoretically non-essential and could be skipped for some easy tasks. While for challenging classification problems, the double-thresholding scheme did improve the learning efficiency significantly.

The simple architecture of GHN, on the one hand, lends itself to being analysed rigorously and this follow up research will be reported elsewhere. On the other hand, GHN is the first fuzzy neural network of its kind that has demonstrated fast learning speed, well-controlled behaviour and state-of-the-art performances on a variety of learning tasks. By cross-checking existing networks against GHN, one is able to grasp the most essential ingredient of deep learning. It is our hope that this kind of comparative study will shed light on future deep learning research and eventually open the "black box" of artificial neural networks [16].

3 Visualization of generalized hamming networks with deep epitomes

This section gives a rigorous analysis of trained Generalized Hamming Networks (GHN) proposed by [38] and discloses an interesting finding about GHNs, i.e., *stacked convolution layers in a GHN is equivalent to a single yet wide convolution layer*. The revealed equivalence, on the theoretical side, can be regarded as a constructive manifestation of the *universal approximation theorem* [31, 59]. In practice, it has profound and multi-fold implications. For network visualization, the constructed *deep epitomes* at each layer provide a visualization of network internal representation that does not rely on the input data. Moreover, deep epitomes allows the direct extraction of features in just one step, without resorting to regularized optimizations used in existing visualization tools.

3.1 Introduction

Despite the great success in recent years, neural networks have long been criticized for their black-box natures and the lack of comprehensive understanding of underlying mechanisms e.g. in [16, 50, 119, 120]. The earliest effort to interpret neural computing in terms of logic inferencing indeed dated back to the seminal paper of [91], followed by recent attempts to provide explanations from a multitude of perspectives (reviewed in Section 3.2).

As an alternative approach to deciphering the mysterious neural networks, various network visualization techniques have been actively developed in recent years (e.g. [49, 118] and references therein). Such visualizations not only provide general understanding about the learning process of networks, but also disclose operational instructions on how to adjust network architecture for performance improvements. Majority of visualization approaches probe the relations between input *data* and neuron *activations*, by showing either how neurons *react* to some sample inputs or, reversely, how desired activations are attained or *maximized* with regularized reconstruction of inputs [6, 36, 72, 89, 123, 137, 145]. Input data are invariably used in visualization to probe how the information flow is transformed through the different layers of neural networks. Although insightful, visualization approaches as such have to face a critical open question: to what extend the conclusions drawn from the analysis of sample inputs can be safely applied to new data?

In order to furnish confirmatory answer to the above-mentioned question, ideally, one would have to employ a visualization tool that is *independent* of input data. This ambitious mission appears impossible at a first glance — the final neuron outputs cannot be readily decomposed as the product of *inputs* and neuron *weights* because the thresholding in ReLU activations is input data dependent. By following the principle of fuzzy logic, [38] recently demonstrated that ReLUs are not essential and can be removed from the so called generalized hamming network (GHN). This simplified network architecture, as reviewed in section 3.3, facilitates the analysis of neuron interplay based on connection weights only. Consequently, stacked convolution layers can be merged into a

single hidden layer without taking into account of inputs from previous layers. Equivalent weights of the merged GHN, which is called *deep epitome*, are computed analytically without resorting to any learning or optimization processes. Moreover, deep epitomes constructed at different layers can be readily applied to new data to extract *hierarchical features* in just *one step* (section 3.4).

3.2 Related work

Despite the great success in recent years, neural networks have long been criticized for their black-box natures e.g. in [16]: "they capture *hidden* relations between inputs and outputs with a highly accurate approximation, but no definitive answer is offered for the question of how they work". The spearhead [91] attempted to interpret neural computing in terms of logic inferencing, followed by more "recent" interpretations e.g. in terms of the *universal approximation* framework [31, 59], *restricted Boltzmann machine* [56], *information bottleneck* theory [120], Nevertheless the mission is far from complete and the training of neural networks (especially deep ones) is still a trail-and-error based practice.

The early 1990s witnessed the birth of *fuzzy neural networks* (FNN) [51, 74] which attempted to furnish neural networks with the interpretability of fuzzy logic [14, 138, 149]. On the other hand, neural networks have been used as a computational tool to come up with both *membership functions* and fuzzy inference rules [40, 124]. This joint force endeavour remains active in the new millennium e.g. [60, 70, 87, 98, 107]. Nevertheless, FNNs have been largely overlooked nowadays by scholars and engineers in machine learning (ML) community, partially due to the lack of convincing demonstrations on ML problems with large datasets. The exception case is the recent [38], which re-interpreted celebrated ReLU and batch normalization with a novel Generalized Hamming Network (GHN) and demonstrated the state-of-the-art performances on a variety of machine learning tasks. While GHNs adopted deep networks with multiple convolution layers, in this paper, we will show how to merge multiple stacked convolution layers into a single yet wide convolution layer.

There are abundant empirical evidences backing the belief that deep network structures is preferred to shallow ones [45], on the other hand, it was theoretically proved by the *universal approximation theorem* that, a single hidden layer network with non-linear activation can well approximate any arbitrary decision functions [31, 59]. Also, empirically, it was shown that one may reduce depth and increase width of network architecture while still attaining or outperforming the accuracies of deep CNN [8] and residual network [143]. Nevertheless, it was unclear how to convert a trained deep network into a shallow equivalent network. To this end, the equivalence revealed in Section 3.3 can be treated as a constructive manifestation of the *universal approximation theorem*.

Various network visualization techniques have been actively developed in recent years, with [36] interpreting high level features via maximizing activation and sampling; [72, 145] learning hierarchical convolutional features via energy or cost minimization; [123] computing class saliency maps for given images; [89] reconstructing images from CNN features with an natural image prior applied;

[137] visualizing live activations as well as deep features via regularized optimization; [6] monitoring prediction errors of individual linear classifiers at multiple iterations. Since all these visualization methods are based on the analysis of examples, the applicability of visualization methods to new data is questionable and no confirmatory answers are provided in a principled manner.

The name "deep epitome" is reminiscent of [26, 27, 67, 68], in which miniature, condensed "epitomes" consisting of the most essential elements were extracted to *model* and *reconstruct* a set of given images. During the learning process, the self-similarity of image(s), either in terms of pixel-to-pixel comparison or spatial configuration, was exploited and a "smooth" mapping between epitome and input image pixels was estimated.

3.3 Deep Epitome

We briefly review generalized hamming networks (GHN) introduced in [38] and present in great detail a method to derive the deep epitome of a trained GHN. Note that we follow notations in [38] with minor modifications for the sake of clarity and brevity.

Review of GHN According to [38], the cornerstone notion of generalized hamming distance (GHD) is defined as $g(a, b) := a \oplus b = a + b - 2 \cdot a \cdot b$ for any $a, b \in \mathcal{R}$. Then the negative GHD is used to quantify the similarity between neuron inputs $\mathbf{x}$ and weights $\mathbf{w}$:

$$-g(\mathbf{w}, \mathbf{x}) = \frac{2}{L}\mathbf{w} \cdot \mathbf{x} - \frac{1}{L}\sum_{l=1}^{L} w_l - \frac{1}{L}\sum_{l=1}^{L} x_l, \tag{12}$$

in which L denotes the length of neuron weights e.g. in convolution kernels, and $g(\mathbf{w}, \mathbf{x})$ is the arithmetic mean of generalized hamming distance between elements of $\mathbf{w}$ and $\mathbf{x}$. By dividing the constant $\frac{2}{L}$, (12) becomes the common representation of neuron computing $(\mathbf{w} \cdot \mathbf{x} + b)$ provided that:

$$b = -\frac{1}{2}\Big(\sum_{l=1}^{L} w_l + \sum_{l=1}^{L} x_l\Big). \tag{13}$$

It was proposed by [38] that neuron bias terms should follow the condition (13) analytically without resorting to an optimization approach. Any networks that fulfil this requirement are thus called generalized hamming networks (GHN). In the light of fuzzy logic, the negative of GHD quantifies the *degree of equivalence* between inputs $\mathbf{x}$ and weights $\mathbf{w}$, i.e. the fuzzy truth value of the statement "$\mathbf{x} \leftrightarrow \mathbf{w}$" where $\leftrightarrow$ denotes a fuzzy equivalence relation. Moreover, $g(\mathbf{x}, \mathbf{x})$ leads to a measurement of *fuzziness* in $\mathbf{x}$, which reaches the maximal fuzziness when $\mathbf{x} = \mathbf{0.5}$ and monotonically decreases when $\mathbf{x}$ deviates from $\mathbf{0.5}$. Also it can be shown that GHD followed by a non-linear activation induces a fuzzy XOR connective [38].

When viewed in this GHN framework, the ReLU activation function $max(0, 0.5 - g(\mathbf{x}, \mathbf{w}))$ actually sets a minimal hamming distance threshold of 0.5 on neuron outputs. [38] then argued that the use of ReLU activation is not essential because bias terms are analytically set in GHNs. [38] reported only negligible influences when ReLU was completely skipped for the easy MNIST classification problem. For more challenging CIFAR10/100 classifications, removing ReLUs merely prolonged the learning process but the final classification accuracies remained almost the same. To this end, we restrict our investigation in this paper to those GHNs which have no ReLUs. As illustrated below, this simplification allows for strict derivation of deep epitome from individual convolution layers in GHNs.

Generalized hamming distance and epitome [38] postulated that one may analyse the entire GHN in terms of fuzzy logic inference rules, yet no elaboration on the analysis was given. Inspired by the universal approximation framework, we show below how to unravel a deep GHN by merging multiple convolution layers into a single hidden layer.

We first reformulate the convolution operation in terms of *generalized hamming distance* (GHD) for each layer, then illustrate how to combine multiple convolution operations across different layers. As said, this combination is only made possible with GHNs in which bias terms strictly follow condition (13). Without loss of generality, we illustrate derivations and proofs for 1D neuron inputs and weights (with complete proofs elaborated in appendix A). Nevertheless, it is straightforward to extend the derivation to 2D or high dimensions. And appendices B to D illustrate deep epitomes of GHNs trained for 2D MNIST and CIFAR10/100 image classifications.

Definition 6. *For two given tuples* $\mathbf{x}^K = \{x_1, \ldots, x_K\}, \mathbf{y}^L = \{y_1, \ldots, y_L\}$, *the hamming outer product, denoted* $\bigoplus$, *is a set of corresponding elements* $\mathbf{x}^K \bigoplus \mathbf{y}^L = \{x_k \oplus y_l | k = 1 \ldots K; l = 1 \ldots L\}$, *where* $\oplus$ *denotes the generalized hamming distance operator. Then the product has following properties,*

1. non-commutative: in general $\mathbf{x}^K \bigoplus \mathbf{y}^L \neq \mathbf{y}^L \bigoplus \mathbf{x}^K$ *but they are permutation equivalent, in the sense that there exist permutation matrices* $\mathbf{P}$ *and* $\mathbf{Q}$ *such that* $\mathbf{x}^K \bigoplus \mathbf{y}^L = \mathbf{P}(\mathbf{y}^L \bigoplus \mathbf{x}^K)\mathbf{Q}$.

2. non-linear: in contrast to the standard outer product which is bilinear in each of its entry, the hamming outer product is non-linear since in general $\mathbf{x}^K \bigoplus (\mathbf{y}^L + \mathbf{z}^L) \neq (\mathbf{x}^K \bigoplus \mathbf{y}^L) + (\mathbf{x}^K \bigoplus \mathbf{z}^L)$ *and* $(\mu \mathbf{x}^K) \bigoplus \mathbf{y}^L \neq \mathbf{x}^K \bigoplus (\mu \mathbf{y}^L) \neq \mu(\mathbf{x}^K \bigoplus \mathbf{y}^L)$ *where* $\mu \in \mathcal{R}$ *is a scalar. Therefore, the hamming outer product defined as such is a pseudo outer product.*

3. associative: $(\mathbf{x}^K \bigoplus \mathbf{y}^L) \bigoplus \mathbf{z}^M = \mathbf{x}^K \bigoplus (\mathbf{y}^L \bigoplus \mathbf{z}^M) = \mathbf{x}^K \bigoplus \mathbf{y}^L \bigoplus \mathbf{z}^M$ *because of the associativity of GHD. This property holds for arbitrary number of tuples.*

4. iterated operation: the definition can be trivially extended to multiple tuples $\mathbf{x}^K \bigoplus \mathbf{y}^L \bigoplus \ldots \mathbf{z}^M = \{x_k \oplus y_l \oplus \ldots, z_m | k = 1 \ldots K; l = 1 \ldots L; \ldots; m = 1, \ldots M\}$.

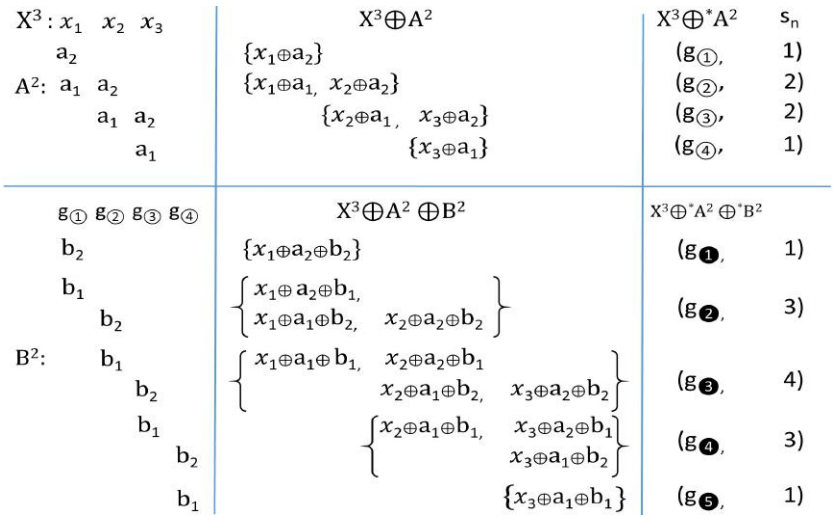


Fig. 11: **Left panel**: example tuples X^3, A^2, B^2; **Middle**: *Hamming outer products* $X^3 \bigoplus A^2$, $X^3 \bigoplus A^2 \bigoplus B^2$; **Right**: *Hamming convolutions* $X^3 \bigoplus^* A^2$, $X^3 \bigoplus^* A^2 \bigoplus^* B^2$ and corresponding *epitomes*. Circled indices denote subsets $S(1), S(2) \dots S(n)$ in which element indices satisfying $k + (L - l) = n$ and $k + (L - l) + (M - m) = n$.

Definition 7. *The convolution of hamming outer product or hamming convolution, denoted* $\bigoplus^*$*, of two tuples is a binary operation that sums up corresponding hamming outer product entries:*

$$\mathbf{x}^K \overset{*}{\bigoplus} \mathbf{y}^L := \Big\{ \sum_{(k,l)\in S(n)} x_k \oplus y_l \Big| \text{ for } n = 1, \dots, K + L - 1 \Big\} \tag{14}$$

where the subsets $S(n) := \{(k, l) \mid k+(L-l) = n\}$ *for* $n = 1, \dots, K+L-1$*, and the union of all subsets constitute a partition of all indices* $\bigcup_{n=1,\dots,K+L-1} S(n) = \big\{(k, l) \big| k = 1 \dots K; l = 1 \dots L\big\}$*. The hamming convolution has following properties,*

1. commutative: $\mathbf{x}^K \bigoplus^* \mathbf{y}^L = \mathbf{y}^L \bigoplus^* \mathbf{x}^K$ *since the partition subsets* $S(n)$ *remains the same.*

2. non-linear: this property is inherited from the non-linearity of the hamming outer product.

3. non-associative: in general $\big(\mathbf{x}^K \bigoplus^* \mathbf{y}^L\big) \bigoplus^* \mathbf{z}^M \neq \mathbf{x}^K \bigoplus^* \big(\mathbf{y}^L \bigoplus^* \mathbf{z}^M\big)$ *since the summation of GHDs is non-associative. Note this is in contrast to the associativity of the hamming outer product.*

4. iterated operation: likewise, the definition can be extended to multiple tuples $\mathbf{x}^K \bigoplus^* \mathbf{y}^L \dots \mathbf{z}^M = \big\{ \sum_{(k,l,\dots,m)\in S(n)} x_k \oplus y_l \dots \oplus z_m \big| \text{ for } n = 1, \dots, K + (L-1) + \dots + (M-1) \big\}$.

Figure 11 illustrates an example in which GHDs are accumulated through two consecutive convolutions. Note that the conversion from the hamming outer products to its convolution is *non-invertible*, in the sense that, it is impossible to recover individual summands $x_k \oplus y_l$ from the summation $\sum_{(k,l)\in S(n)} x_k \oplus y_l$. As proved in proposition 4, it is possible to compute the convolution of tuples in two (or more) stacked layers without explicitly recovering individual outer product entries of each layer. Due to the non-linearity of the hamming convolutions, computing the composite of two hamming convolutions is non-trivial as elaborated in Section 3.3. In order to illustrate how to carry out this operation, let us first introduce the *epitome* of a hamming convolution as follows.

Definition 8. *An epitome consists of a set of* N *pairs* $\mathbb{E} = \big\{(g_n, s_n),\ \big|n = 1, \ldots, N\big\}$ *where* g_n *denotes the summation of GHD entries from some hamming convolutions,* s_n *the number of summands or the cardinality of the subset* $S(n)$ *defined above, and* N *is called the length of the epitome.*

A normalized epitome is an epitome with $s_n = 1$ *for all* $n = 1, \ldots N$*. Any epitome can then be normalized by setting* $(g_n/s_n, 1)$ *for all elements. A normalized epitome may also refer to input data* $\mathbf{x}$ *or neuron weights* $\mathbf{w}$ *that are not yet involved in any convolution operations. In the latter case,* g_n *is simply the input data* $\mathbf{x}$ *or neuron weights* $\mathbf{w}$*.*

Remark: the summation of GHD entries g_n is defined abstractly, and depending on different scenarios, the underlying outer product may operate on arbitrary number of tuples $g_n = \left(\mathbf{x}^K \bigoplus{}^* \mathbf{y}^L \ldots \mathbf{z}^M\right)(n) = \sum_{(k,l,\ldots,m)\in S(n)} x_k \oplus y_l \ldots \oplus z_m$.

Fuzzy logic interpretation: in contrast to the traditional signal processing point of view, in which neuron weights $\mathbf{w}$ are treated as parameters of linear transformation and bias terms b are appropriate thresholds for non-linear activations, the generalized hamming distance approach *treats* $\mathbf{w}$ *as fuzzy templates and sets bias terms analytically* according to (13). In this view, the normalization g_n/s_n is nothing but the *mean GHD* of entries in the subset $S(n)$, which indicates a *grade of fitness* (or a fuzzy set) between templates $\mathbf{w}$ and inputs $\mathbf{x}$ at location n. This kind of arithmetic mean operator has been used for aggregating evidences in fuzzy sets and empirically performed quite well in decision making environments (e.g. see [149]).

Still in the light of signal processing, the generalized hamming distance naturally induces an *information enhancement and suppression* mechanism. Since the gradient of $g(\mathbf{x}, \mathbf{w})$ with respect to $\mathbf{x}$ is $1 - 2\mathbf{w}$, the information in $\mathbf{x}$ is then either enhanced or suppressed according to $\mathbf{w}$: a) the output $g(\mathbf{x}, \mathbf{w})$ is always $\mathbf{x}$ for $\mathbf{w} = 0$ (conversely $1 - \mathbf{x}$ for $\mathbf{w} = 1$) with no information loss in $\mathbf{x}$; b) for $\mathbf{w} = 0.5$, the output $g(\mathbf{x}, \mathbf{w})$ is always 0.5 regardless of $\mathbf{x}$, thus input information in $\mathbf{x}$ is completely suppressed; c) for $\mathbf{w} < 0.0$ or $\mathbf{w} > 1.0$ information in $\mathbf{x}$ is proportionally enhanced. It was indeed observed, during the learning process in our experiments, a small faction of prominent feature pixels in weights $\mathbf{w}$ gradually attain large positive or negative values, so that corresponding input pixels play decisive roles in classification. On the other hand, large majority of obscure pixels remain in the fuzzy regime near 0.5, and correspondingly, input pixels have

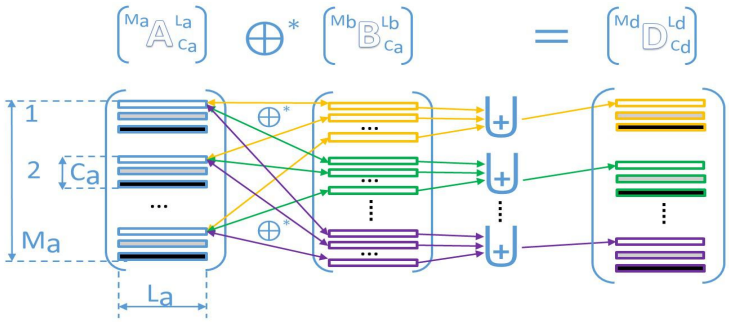


Fig. 12: The hamming convolution of two banks of epitomes. Remarks: **a**) for the inputs $\mathbb{A}, \mathbb{B}$ the number of epitomes M_a must be the same as the number of channels C_b; and for the output bank $M_d = M_b, C_d = C_a, L_d = (L_a + L_b - 1)$. **b)** the notation $\bigoplus^*$ refers to the hamming convolution between two banks of epitomes (see Definition 9 for details). The convolution of two single-layered epitomes is treated as a special case with all $M_a, C_a, M_b, C_b = 1$. **c)** the notation $\biguplus$ refers to the summation of multiple epitomes of the same length, which is defined in Definition 10. **d)** multiple (coloured) epitomes in $\mathbb{D}$ correspond to different (coloured) epitomes in $\mathbb{B}$; and different (shaded) channels in $\mathbb{D}$ correspond to different (shaded) channels of inputs in $\mathbb{A}$.

virtually no influence on the final decision (see experimental results in Section 3.4). This observation is also in accordance with the *information compression* interpretation advocated by [120], and the connection indicates an interesting research direction for future work.

Deep epitome This subsection only illustrates main results concerning how to merge multiple hamming convolution operations in stacked layers into a single-layer of epitomes i.e. *deep epitome*. Detailed proofs are given in appendix A.

Notation: for the sake of brevity, let $[{}^{M_a}\mathbb{A}^{L_a}_{C_a}]$ denote a *bank of epitomes*: $\left\{{}^{m}\mathbb{A}^{L_a}_{C_a} \;\middle|\; m = 1, \dots M_a\right\}$, where $\mathbb{A}^{L_a}_{C_a} = \left\{\mathbb{A}^{L_a}_{c} \;\middle|\; c = 1, \dots C_a\right\}$ are C_a-channels of length-L_a epitomes, and M_a is the number of epitomes as such in the bank or set $[\mathbb{A}]$. Figure 12 illustrates example banks of epitomes and two operations defined on them (also see Appendix A for detailed definition of $\biguplus$).

Theorem 2. *A generalized hamming network consisting of multiple convolution layers, is equivalent to a bank of epitome, called deep epitome* $[{}^{\star}\mathbb{D}^{\diamond}_{\triangledown}]$*, which can be computed by iteratively applying the composite hamming convolution in equation (19) to individual layer of epitomes:*

$$[{}^{\star}\mathbb{D}^{\diamond}_{\triangledown}] := [{}^{M_a}\mathbb{A}^{L_a}_{C_a}] \overset{*}{\bigoplus} [{}^{M_b}\mathbb{B}^{L_b}_{C_b}] \overset{*}{\bigoplus} \dots \overset{*}{\bigoplus} [{}^{M_z}\mathbb{Z}^{L_z}_{C_z}], \tag{21}$$

in which $\triangledown = C_a$ *is the number of channels in the first bank* $\mathbb{A}$, $\star = M_z$ *is the number of epitomes in the last bank* $\mathbb{Z}$, *and* $\diamond = L_a + (L_b - 1) + \ldots + (L_z - 1)$ *is the length of composite deep epitome. Note that for the hamming convolution to be a valid operation, the number of epitomes in the previous layer and the number channels in the current layer must be the same e.g.* $C_b = M_a$.

Proof. For given inputs represented as a bank of normalized epitomes $[{}^{M_x}\mathbb{X}^{L_x}_{C_x}]$ the final network output $[{}^{M_z}Y^{L_y}_{C_x}]$ is obtained by recursively applying equation (19) to outputs from the previous layers, and factoring out the input due to the associativity proved in proposition 4:

$$\begin{aligned}[{}^{M_z}Y^{L_y}_{C_x}] &= \Big(\big(([{}^{M_x}\mathbb{X}^{L_x}_{C_x}] \overset{*}{\bigoplus} [{}^{M_a}\mathbb{A}^{L_a}_{C_a}]) \overset{*}{\bigoplus} [{}^{M_b}\mathbb{B}^{L_b}_{C_b}]\big) \overset{*}{\bigoplus} \ldots \overset{*}{\bigoplus} [{}^{M_z}\mathbb{Z}^{L_c}_{C_z}]\Big) \\ &= [{}^{M_x}\mathbb{X}^{L_x}_{C_x}] \overset{*}{\bigoplus} \underbrace{\Big([{}^{M_a}\mathbb{A}^{L_a}_{C_a}] \overset{*}{\bigoplus} [{}^{M_b}\mathbb{B}^{L_b}_{C_b}] \overset{*}{\bigoplus} \ldots \overset{*}{\bigoplus} [{}^{M_z}\mathbb{Z}^{L_c}_{C_z}]\Big)}_{[{}^{\star}\mathbb{D}^{\diamond}_{\triangledown}]}. \end{aligned} \tag{22}$$

□

Remark: due to the non-linearity of underlying hamming outer products, to prove the associativity of the *convolution of epitomes* is by no means trivial (see proposition 4). In essence, we have to use proposition 4 to compute the convolution of two epitomes even though individual entries of the underlying hamming outer product are not directly accessible. Consequently, the updating rule outlined in equations (15) and (16) play the crucial role in setting due bias terms analytically for generalized hamming networks (GHN), as opposed to the optimization approach often adopted by many non-GHN deep convolution networks.

Fuzzy logic inferencing with deep epitomes: Eq. (22) can be treated as a fuzzy logic inferencing rule, with which elements of input $\mathbf{x}$ are compared with respect to corresponding elements of deep epitomes $\mathbf{d}$. More specifically, the negative of GHD quantifies the *degree of equivalence* between inputs $\mathbf{x}$ and epitome weights $\mathbf{d}$, i.e. the fuzzy truth value of the assertion "$\mathbf{x} \leftrightarrow \mathbf{d}$" where $\leftrightarrow$ denotes a fuzzy *logical biconditional*. Therefore, output scores in $\mathbf{y}$ indicate the grade of fuzzy equivalences truth values between $\mathbf{x}$ and the shifted $\mathbf{d}$ at different spatial locations. This inferencing rule, in the same vein of [38], is applicable to either a single layer neuron weights or the composite deep epitomes as proved by (22).

Constructive manifestation of the universal approximation theorem: it was proved that a single hidden layer network with non-linear activation can well approximate any arbitrary decision functions [31, 59], yet it was also argued by [45] that such a single layer may be infeasibly large and may fail to learn and generalize correctly. Theorem 2 proves that such a simplified single hidden layer network can actually be constructed from a trained GNH. In this sense Theorem 2 illustrates a concrete solution which materializes the universal approximation theorem.

3.4 Deep epitome for network visualization

We illustrate below deep epitomes extracted from three generalized hamming networks trained with MNIST, CIFAR10/100 classification respectively. Detailed descriptions about the network architectures (number of layers, channels etc.) are included in the appendix.

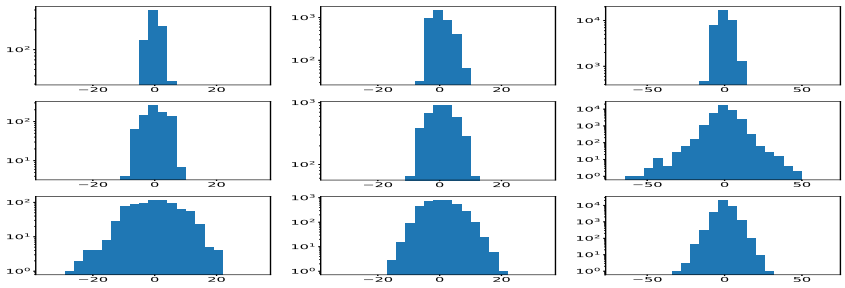


Fig. 13: Histograms of *normalized* deep epitomes at different layers/iterations for GHN trained with MNIST classification. Left to right: layers 1,2,3. Top, middle and bottom rows: iteration 200, 1000 and 10000 respectively.

Data independent visualization of deep epitomes Deep epitomes derived in the previous section allows one to build up and visualize hierarchical features in an on-line manner during the learning process. This approach is in contrast to many existing approaches, which often apply additional optimization or learning processes with various type of regularizations e.g. in [36, 89, 123, 137, 145]. Figures 15, 18 and 21, 22 in appendices illustrate deep epitomes learnt by three generalized hamming networks for the MNIST and CIFAR10/100 image classification tasks. It was observed that geometrical structures of hierarchical features were formed at different layers, rather early during the learning process (e.g. 1000 out of 10000 iterations). Substantial follow up efforts were invested on refining features for improved details. The scrutinization of normalized epitome histograms in Figure 13 showed that a majority of pixel values remain relatively small during the learning process, while a small fraction of epitome weights gradually accumulate large values over thousands of iterations to form prominent features.

The observation of sparse features has been reported and interpreted in terms of sparse coding e.g. [102] or the information compression mechanism as advocated by [120]. Following [38] we adopt the notion of *fuzziness* (also reviewed in Section 3.3) to provide a fuzzy logic interpretation: *prominent features correspond to neuron weights with low fuzziness*. It was indeed observed in Figure 14 that *fuzziness* of deep epitomes in general decrease during the learning process despite of fluctuations at some layers. The inclination towards *reduced fuzzi-*

ness seems in accord with the minimization of classification errors, although the fuzziness is not explicitly minimized.

Finally we re-iterate that the internal representation of deep epitomes is *input data independent*. For instance in MNIST handwritten images, it is certain constellations of strokes instead of digits that are learnt at layer 3 (see Figure 15). The matching of arbitrary input data with such "fuzzy templates" is then quantified by the generalized hamming distance, and can be treated as generic fuzzy logic inferencing rules learnt by GHNs. The matching score measured by GHDs can also be treated as salient features that are subsequently fed to the next layer (see Section 3.4 with Figures 16 and more results in appendices B and C)[8].

Data dependent feature extraction Feature extraction of given inputs is straightforward with deep epitomes applied according to eq. (22). Figures 16 (and more results in appendices B and C) show example features extracted at different layers of GHN trained on MNIST, CIFAR10/100 image datasets. Clearly extracted features represent different types of salient features e.g. oriented strokes in hand written images, oriented edgelets, textons with associated colours or even rough segmentations in CIFAR images. These features all become gradually more discriminative during the learning process.

It must be noted that the extraction of these hierarchical salient features is not entirely new and has been reported e.g. in [36, 72]. Nevertheless, the equivalence of deep epitomes disclosed in Theorem 2 leads to an unique characteristic of GHNs — deep layer features do not necessarily rely on features extracted from previous layers, instead, they can be extracted in *one step* using deep epitomes at desired layers. For extremely deep convolution networks e.g. those with over 100 layers, this simplification may bring about substantial reduction of computational and algorithmic complexities. This potential advantage is worth follow up exploration in future research.

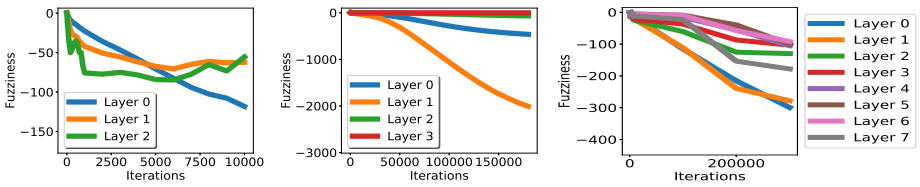


Fig. 14: Fuzziness in *normalized* deep epitomes at different layers and learning iterations. **Left**: a GHN trained with MNIST classification. **Middle**: CIFAR10. **Right**: CIFAR100. See section 3.3 for definition of *fuzziness*.

[8] This learnt "fuzzy template" is reminiscent of epitomes in [67] and gives rise to the name *deep epitome*.

3.5 Conclusion

We have proposed in this paper a novel network representation, called *deep epitome*, which is proved to be equivalent to stacked convolution layers in generalized hamming networks (GHN). Theoretically this representation provides a constructive manifestation for the *universal approximation theorem* [31, 59], which states that a single layered network, in principle, is able to approximate any arbitrary decision functions up to any desired accuracy. On the other hand, it is a dominant belief [45], which is supported by abundant empirical evidences, that deep structures play an indispensable role in decomposing the combinatorial optimization problem into layer-wise manageable sub-problems. We concur with the view and supplement with our demonstration that, a trained deep GHN can be converted into a simplified networks for the sake of high interpretability, reduced algorithmic and computational complexities.

The success of our endeavours lies in the rigorous derivation of convolving epitomes across different layers in eq. (15) and (16), which set due bias terms analytically without resorting to optimization-based approaches. Consequently, deep epitomes at all convolution layers can be computed without using any input data. Moreover, deep epitomes can be used to extract hierarchical features in just *one step* at any desired layers. In the light of fuzzy logic, the normalized epitome (definition 8) encodes a *grade of fitness* between the learnt templates and given inputs at certain spatial locations. This fuzzy logic interpretation furnishes a refreshing perspective that, in our view, will open the black box of deep learning eventually.

4 Discussion and future work

By illustrating our recent findings with the generalized hamming network, this article explores and establishes a concrete and fundamental connection between deep learning and fuzzy logic. The significance of this work probably may be best appreciated in terms of its point of view instead of any particular results. While basic concepts (like hamming outer product, deep epitomes) have been formed, no doubt many others have yet to be formulated. Therefore, one motivation of this article is to visualize the outline of future work from both fuzzy logic and deep learning points of view:

- *Fuzzy logic* research has reached a culmination in 1998 and declined since then (as shown in Figure 2). The declination can be partially ascribed to the lack of convincing applications on challenging (machine learning) problems. It is our hope that this article will pave the way for fuzzy logic researchers to develop convincing applications and tackle challenging problems which are of interest to machine learning community too. In particular, we believe expertise and knowledge in fuzzy logic are well suited to *model ambiguities in data*, *model uncertainty in knowledge representation* and *furnish transfer learning with non-inductive inference* etc. as suggested in [62].

- On the other hand, *deep learning* and *machine learning* could benefit from the comparative research by re-examining many trail-and-error heuristics in the lens of fuzzy logic, and consequently, distilling the essential ingredients with rigorous foundations. In particular, we feel the exploration of *combined binary features* may not only reformulate neural computing but also lead to highly memory and computational efficient algorithms in practice.

Last but not least, it is my humble vision that researchers of young generation will be inspired by this article, and set off for the rejuvenated endeavour of "soft computing" with their valuable wisdoms and enthusiasms.

Bibliography

[AIg] An AI god will emerge by 2042 and write its own bible. Will you worship it? https://venturebeat.com/2017/10/02/an-ai-god-will-emerge-by-2042-and-write-its-own-bible-will-you-worship-it/. Accessed: 2017-11-08.

[2] A baseline cnn network for sentence classification implemented with tensorflow. https://github.com/dennybritz/cnn-text-classification-tf. Accessed: 2017-05-19.

[3] A baseline variational auto-encoder based on "auto-encoding variational bayes". https://github.com/y0ast/VAE-TensorFlow. Accessed: 2017-05-19.

[4] R. Hahnloser, R. Sarpeshkar, M. Mahowald, R.J. Douglas, H.S.Seung (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. 405.

[5] Adams, E. (1998). *A Primer of Probability Logic*. Center for the Study of Language and Information Publication Lecture Notes. Cambridge University Press.

[6] Alain, G. et Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. *CoRR*, abs/1610.01644.

[7] Atanassov, K. (2011). On Zadeh's intuitionistic fuzzy disjunction and conjunction. *NIFS*, 17(1):1–4.

[8] Ba, L. J. et Caurana, R. (2013). Do deep nets really need to be deep? *CoRR*, abs/1312.6184.

[9] BEDREGAL, B., REISER, R. H. S. et DIMURO, G. P. (2013). Revisiting xor-implications: Classes of fuzzy (co)implications based on f-xor (f-xnor) connectives. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 21(06):899–925.

[10] Bedregal, B. C., Reiser, R. H. S. et Dimuro, G. P. (2009a). Xor-implications and e-implications: Classes of fuzzy implications based on fuzzy xor. *Electron. Notes Theor. Comput. Sci.*, 247:5–18.

[11] Bedregal, B. C., Reiser, R. H. S. et Dimuro, G. P. (2009b). Xor-Implications and E-Implications: Classes of Fuzzy Implications Based on Fuzzy Xor. *Electronic Notes in Theoretical Computer Science*, 247:5–18.

[12] Bellman, R., Kalaba, R. et Zadeh, L. (1966). Abstraction and pattern classification. *Journal of Mathematical Analysis and Applications*, 13(1):1 – 7.

[13] Bellman, R. et Zadeh, L. A. (1970). Decision making in a fuzzy environment. *Management Sciences*, 17:141–164.

[14] Belohlavek, R., Dauben, J. et Klir, G. (2017). *Fuzzy Logic and Mathematics: A Historical Perspective*. Oxford University Press.

[Benenson] Benenson, R. What is the class of this image? Discover the current state of the art in objects classification. http://rodrigob.github.io/are_

we_there_yet/build/classification_datasets_results.html. Accessed: 2017-07-19.

[16] Benítez, J. M., Castro, J. L. et Requena, I. (1997). Are artificial neural networks black boxes? *IEEE Trans. Neural Networks*, 8(5):1156–1164.

[17] Bezdek, J. C., Ehrlich, R. et Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers and Geosciences*, 10(2):191 – 203.

[18] Bezdek, J. C. et Pal, S. K. (1992). *Fuzzy Models for Pattern Recognition: Methods That Search for Structures in Data*. IEEE Press.

[19] Bilgiç, T. et Türkşen, I. B. (2000). *Measurement of Membership Functions: Theoretical and Empirical Work*, pages 195–227. Springer US, Boston, MA.

[20] Bodenhofer, U. et Klawonn, F. (2008). Robust rank correlation coefficients on the basis of fuzzy orderings: initial steps. *Mathware Soft Comput.*, 15(1):5–20.

[21] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[22] Buckles, B. P. et Petry, F. E. (1982). A fuzzy representation of data for relational databases. *Fuzzy Sets and Systems*, 7(3):213 – 226.

[23] Calonder, M., Lepetit, V., Strecha, C. et Fua, P. (2010). Brief: Binary robust independent elementary features. *In Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 778–792.

[24] Cao, Y. et Ezawa, Y. (2010). Nondeterministic fuzzy automata. *CoRR*, abs/1012.2162.

[25] Chen, S.-J. J. et Hwang, C. L. (1992). *Fuzzy Multiple Attribute Decision Making: Methods and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[26] Cheung, V., Frey, B. et Jojic, N. (2005). Video epitomes. *In Proceedings of the Computer Vision and Pattern Recognition*, CVPR '05.

[27] Chu, X., Yan, S., Li, L., Chan, K. L. et Huang, T. S. (2010). Spatialized epitome and its applications. *In The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 311–318.

[28] Cintula, P., Hájek, P. et Noguera, C. (2011). *Handbook of Mathematical Fuzzy Logic - volume 2*. Numéro 38 de Studies in Logic, Mathematical Logic and Foundations. College Publications, London, petr cintula, petr hájek, carles noguera édition.

[29] Cortes, C. et Vapnik, V. (1995). Support-vector networks. *Mach. Learn.*, 20(3):273–297.

[30] Courbariaux, M. et Bengio, Y. (2016). Binarized neural network: Training deep neural networks with weights and activations constrained to +1 or -1. *CoRR*, abs/1602.02830.

[31] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314.

[32] Doostfatemeh, M. et Kremer, S. C. (2005). New directions in fuzzy automata. *International Journal of Approximate Reasoning*, 38(2):175 – 214.

[33] Dubois, D. et Prade, H. (1980). *Fuzzy Sets and Systems: Theory and Applications*. Numéro v. 144 de Fuzzy Sets and Systems: Theory and Applications. Academic Press.

[34] Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

[35] Elkan, C., Berenji, H. R., Chandrasekaran, B., de Silva, C. J. S., Attikiouzel, Y., Dubois, D., Prade, H., Smets, P., Freksa, C., Garcia, O. N., Klir, G. J., Yuan, B., Mamdani, E. H., Pelletier, F. J., Ruspini, E. H., Turksen, B., Vadiee, N., Jamshidi, M., Wang, P.-Z., Tan, S.-K., Tan, S., Yager, R. R. et Zadeh, L. A. (1994). The paradoxical success of fuzzy logic. *IEEE Expert*, 9(4):3–49.

[36] Erhan, D., Bengio, Y., Courville, A. et Vincent, P. (2009). Visualizing higher-layer features of a deep network. Rapport technique 1341, University of Montreal. Also presented at the ICML 2009 Workshop on Learning Feature Hierarchies, Montréal, Canada.

[37] Fagin, R. (1996). Combining fuzzy information from multiple systems (extended abstract). *In Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '96, pages 216–226, New York, NY, USA. ACM.

[38] Fan, L. (2017). Revisit fuzzy neural network: Demystifying batch normalization and ReLU with generalized hamming network. *In Advances in Neural Information Processing Systems*.

[39] Fullér, R. (1995). Neural fuzzy systems.

[40] Furukawa, M. et Yamakawa, T. (1995). The design algorithms of membership functions for a fuzzy neuron. *Fuzzy Sets and Systems*, 71(3):329 – 343. Fuzzy Neural Control.

[41] Giles, R. (1988). The concept of grade of membership. *Fuzzy Sets Syst.*, 25(3):297–323.

[42] Glorot, X., Bordes, A. et Bengio, Y. (2011). Deep sparse rectifier neural networks. *In* Gordon, G., Dunson, D. et Dudík, M., éditeurs : *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 de *Proceedings of Machine Learning Research*, pages 315–323.

[43] Goguen, J. A. (1969). The logic of inexact concepts. *Synthese*, 19(3/4):325–373.

[44] Goguen, J. A. et Zadeh, L. (1967). L-Fuzzy Sets. *JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS*, 18:145–174.

[45] Goodfellow, I., Bengio, Y. et Courville, A. (2016). *Deep Learning*.

[46] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. et Bengio, Y. (2014). Generative adversarial nets. *In* Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D. et Weinberger, K. Q., éditeurs : *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

[47] Google (2012). Google ngram viewer. http://books.google.com/ngrams/datasets.

[48] Grabisch, M., Marichal, J., Mesiar, R. et Pap, E. (2009). *Aggregation Functions*. Encyclopedia of Mathematics and its Applications. Cambridge University Press.

[49] Grün, F., Rupprecht, C., Navab, N. et Tombari, F. (2016). A Taxonomy and Library for Visualizing Learned Features in Convolutional Neural Networks. *Proceedings of the Workshop on Visualization for Deep Learning at International Conference on Machine Learning (ICML)*, 48.

[50] Gülçehre, Ç. et Bengio, Y. (2013). Knowledge matters: Importance of prior information for optimization. *CoRR*, abs/1301.4083.

[51] Gupta, M. M. et Rao, D. H. (1994). Invited Review on the principles of fuzzy neural networks. *Fuzzy Sets and Systems*, 61:1–18.

[52] Hailperin, T. (1996). Sentential probability logic.

[53] Hajek, P. (1998). *The Metamathematics of Fuzzy Logic*. Kluwer.

[54] Halpern, J. Y. (2003). *Reasoning About Uncertainty*. MIT Press, Cambridge, MA, USA.

[55] Hinton, G., Deng, L., Yu, D., Dahl, G. E., r. Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. et Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

[56] Hinton, G. et Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507.

[57] Hisdal, E. (1988). Are grades of membership probabilities? *Fuzzy Sets and Systems*, 25:325–348.

[58] Hopner, F., Hoppner, F. et Klawonn, F. (1999). *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. Wiley and Sons, 1., auflage édition.

[59] Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257.

[60] Hu, Z., Ma, X., Liu, Z., Hovy, E. H. et Xing, E. P. (2016). Harnessing deep neural networks with logic rules. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

[61] Hüllermeier, E. (2013). Learning from Imprecise and Fuzzy Observations: Data Disambiguation through Generalized Loss Minimization. *ArXiv e-prints*.

[62] Hüllermeier, E. (2015). Does machine learning need fuzzy logic? *Fuzzy Sets Syst.*, 281(C):292–299.

[63] Ioffe, S. et Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In* Bach, F. R. et Blei, D. M., éditeurs : *ICML*, volume 37, pages 448–456.

[64] Jang, J. R. et Sun, C. (1993). Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Trans. Neural Networks*, 4(1):156–159.

[65] Jang, J.-S. R. (1991). Fuzzy modeling using generalized neural networks and kalman filter algorithm. *In Proceedings of the Ninth National Conference on Artificial Intelligence - Volume 2*, AAAI'91, pages 762–767. AAAI Press.

[66] Jang, J. S. R. (1993). Anfis: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685.

[67] Jojic, N., Frey, B. J. et Kannan, A. (2003). Epitomic analysis of appearance and shape. *In Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 34–.

[68] Jojic, N., Perina, A. et Murino, V. (2010). Structural epitome: a way to summarize one's visual experience. *In Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 1027–1035.

[69] Kar, S., Das, S. et Ghosh, P. K. (2014a). Review article: Applications of neuro fuzzy systems: A brief review and future outline. *Appl. Soft Comput.*, 15:243–259.

[70] Kar, S., Das, S. et Ghosh, P. K. (2014b). Review article: Applications of neuro fuzzy systems: A brief review and future outline. *Appl. Soft Comput.*, 15:243–259.

[71] Kaufmann, A. et Gupta, M. (1991). *Introduction to fuzzy arithmetic: theory and applications*. Electrical-Computer Science and Engineering Series. Van Nostrand Reinhold Co.

[72] Kavukcuoglu, K., Sermanet, P., lan Boureau, Y., Gregor, K., Mathieu, M. et Cun, Y. L. (2010). Learning convolutional feature hierarchies for visual recognition. *In* Lafferty, J., Williams, C., Shawe-taylor, J., Zemel, R. et Culotta, A., éditeurs : *Advances in Neural Information Processing Systems 23*, pages 1090–1098.

[73] Kawaguchi, K., Pack Kaelbling, L. et Bengio, Y. (2017). Generalization in Deep Learning. *ArXiv e-prints*.

[74] Keller, J., Yager, R. et Tahani, H. (1992). Neural Network Implementation of Fuzzy Logic. *Fuzzy Sets and Systems*, 45(1).

[75] Kickert, W. (1979). *Fuzzy Theories on Decision Making: A Critical Review*. Frontiers in System Research. Springer US.

[76] Kim, Y. (2014). Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

[77] Kingma, D. P. et Welling, M. (2013). Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

[78] Kohout, L. J. et Kim, E. (2004). Characterization of interval fuzzy logic systems of connectives by group transformations. *Reliable Computing*, 10(4): 299–334.

[79] Krantz, D., Luce, R., Suppes, P. et Tversky, A. (1971). *Foundations of measurement*. Academic Press, New York.

[80] Krantz, D., Luce, R., Suppes, P. et Tversky, A. (1990). *Foundations of Measurement: Representation, axiomatization, and invariance*. Foundations of Measurement. Academic Press.

[81] Krantz, D., Suppes, P., Luce, R. et Tversky, A. (1989). *Foundations of measurement: Geometrical, threshold, and probabilistic representations*. Foundations of Measurement. Academic Press.

[82] Krizhevsky, A., Sutskever, I. et Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA. Curran Associates Inc.

[83] Kulis, B. et Darrell, T. (2009). Learning to hash with binary reconstructive embeddings. *In Proceedings of the 22Nd International Conference on Neural Information Processing Systems*, NIPS'09, pages 1042–1050.

[84] Lakoff, G. (1990). *Women, Fire, and Dangerous Things*. Cognitive science/linguistics/philosophy. University of Chicago Press.

[85] Lecun, Y., Bengio, Y. et Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[86] Lin, K., Yang, H.-F., Hsiao, J.-H. et Chen, C.-S. (2015). Deep learning of binary hash codes for fast image retrieval. *In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[87] Liu, P. et Li, H. (2004a). *Fuzzy Neural Network Theory and Application*. World Scientific Press.

[88] Liu, P. et Li, H. (2004b). *Fuzzy Neural Network Theory and Application*. Series in machine perception and artificial intelligence. World Scientific.

[89] Mahendran, A. et Vedaldi, A. (2015). Visualizing deep convolutional neural networks using natural pre-images. *CoRR*, abs/1512.02017.

[90] Mamdani, E. H. et Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Machine Studies*, 7 (1):1–13.

[91] Mcculloch, W. et Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:127–147.

[92] Mela, C. F. et Lehmann, D. R. (1995). Using fuzzy set theoretic techniques to identify preference rules from interactions in the linear model: an empirical study. *Fuzzy Sets and Systems*, 71(2):165 – 181.

[93] Minsky, M. (1991). Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI Magazine*, 12(2):34–51.

[94] Močkoř, J. (1999). Fuzzy and non-deterministic automata. *Soft Computing*, 3(4):221–226.

[95] Mostert, P. S. et Shields, A. L. (1957). On the structure of semigroups on a compact manifold with boundary. *Annals of Mathematics*, 65(1):117–143.

[96] Nair, V. et Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *In* Fürnkranz, J. et Joachims, T., éditeurs : *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814. Omnipress.

[97] Narens, L. (1985). *Abstract measurement theory*. MIT Press.

[98] Nauck, D. D. et Nürnberger, A. (2013). *Neuro-fuzzy Systems: A Short Historical Review*, pages 91–109. Springer Berlin Heidelberg, Berlin, Heidelberg.

[99] Norouzi, M. et Blei, D. M. (2011). Minimal loss hashing for compact binary codes. *In Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 353–360, New York, NY, USA.

[100] Norouzi, M., Fleet, D. J. et Salakhutdinov, R. R. (2012). Hamming distance metric learning. *In* Pereira, F., Burges, C. J. C., Bottou, L. et Weinberger, K. Q., éditeurs : *Advances in Neural Information Processing Systems 25*, pages 1061–1069.

[101] Palmeira, E. et Bedregal, B. (2016). *Some Results on Extension of Lattice-Valued XOR, XOR-Implications and E-Implications*, pages 809–820. Springer International Publishing, Cham.

[102] Papyan, V., Romano, Y. et Elad, M. (2016). Convolutional neural networks analyzed via convolutional sparse coding. *CoRR*, abs/1607.08194.

[103] Paris, J. (1994). *The Uncertain Reasoner's Companion: A Mathematical Perspective*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

[104] Pawlak, Z. (1982). Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356.

[105] Pawlak, Z. (1992). *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, Norwell, MA, USA.

[106] Pedrycz, W. et Succi, G. (2002a). fxor fuzzy logic networks. *Soft Computing*, 7(2):115–120.

[107] Pedrycz, W. et Succi, G. (2002b). fXOR fuzzy logic networks. *Soft Computing*, 7.

[108] R Hahnloser, H. S. (2001). Permitted and forbidden sets in symmetric threshold-linear networks. *In NIPS*.

[109] Raju, K. V. S. V. N. et Majumdar, A. K. (1988). Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems. *ACM Trans. Database Syst.*, 13(2):129–166.

[110] Rastegari, M., Ordonez, V., Redmon, J. et Farhadi, A. (2016). Xnor-net: Imagenet classification using binary convolutional neural networks. *CoRR*, abs/1603.05279.

[111] Rosch, E. et Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7(4):573 – 605.

[112] Rose, A. et Rosser, J. B. (1958). Fragments of many-valued statement calculi. *Transactions of the American Mathematical Society*, 87(1):1–53.

[113] Rublee, E., Rabaud, V., Konolige, K. et Bradski, G. (2011). Orb: An efficient alternative to sift or surf. *In Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA.

[114] Ruiz, M. D. et Hüllermeier, E. (2012). A formal and empirical analysis of the fuzzy gamma rank correlation coefficient. *Inf. Sci.*, 206:1–17.

[115] Ruspini, E., Bonissone, P. et Pedrycz, W. (1998). *Handbook of Fuzzy Computation*. Taylor & Francis.

[116] Salimans, T. et Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. page 901.

[117] Schweizer, B. (2005). Triangular norms, looking back—triangle functions, looking ahead. *In* Klement, E. P. et Mesiar, R., éditeurs : *Logical, Algebraic, Analytic and Probabilistic Aspects of Triangular Norms*, pages 3 – 15. Elsevier Science B.V., Amsterdam.

[118] Seifert, C., Aamir, A., Balagopalan, A., Jain, D., Sharma, A., Grottel, S. et Gumhold, S. (2017). *Visualizations of Deep Neural Networks in Computer Vision: A Survey*, pages 123–144. Springer International Publishing, Cham.

[119] Shrikumar, A., Greenside, P. et Kundaje, A. (2017). Learning important features through propagating activation differences. *CoRR*, abs/1704.02685.

[120] Shwartz-Ziv, R. et Tishby, N. (2017). Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810.

[121] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. et Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

[122] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. et Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.

[123] Simonyan, K., Vedaldi, A. et Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034.

[124] Takagi, H. (2000). Fusion Technology of Neural Networks and Fuzzy Systems : A Chronicled Progression from the Laboratory to Our Daily Lives. *Journal of Applied Mathematics*, 10(4):1–20.

[125] Takagi, H. et Hayashi, I. (1991). Nn-driven fuzzy reasoning. *International Journal of Approximate Reasoning*, 5(3):191 – 212.

[126] Tick, J., Fodor, J. et Von Neumann, J. (2005). Fuzzy Implications and Inference Process. *Computing and Informatics*, 24:591–602.

[127] Ulsari, A. B. (1992). Training Artificial Neural Networks for Fuzzy Logic. *Complex Systems*, 6:443–457.

[128] Viola, P. et Jones, M. J. (2004). Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154.

[129] Wee, W. G. et Fu, K. S. (1969). A formulation of fuzzy automata and its application as a model of learning systems. *IEEE Transactions on Systems Science and Cybernetics*, 5(3):215–223.

[130] Whitehead, A. et Russell, B. (1912). *Principia Mathematica*. Numéro v. 2 de Principia Mathematica. University Press.

[131] Xu, J. et Zhou, X. (2011). *Fuzzy-Like Multiple Objective Decision Making*. Springer Publishing Company, Incorporated, 1st édition.

[132] Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190.

[133] Yamakawa, T. (1989). Stablization of an inverted pendulum by a high-speed fuzzy logic controller hardware system. *Fuzzy Sets Syst.*, 32(2):161–180.

[134] Yamakawa, T. et Furukawa, M. (1992). A design algorithm of membership functions for a fuzzy neuron using example-based learning. *In [1992 Proceedings] IEEE International Conference on Fuzzy Systems*, pages 75–82.

[135] Yao, Y. (1998). A comparative study of fuzzy sets and rough sets. *Information Sciences*, 109(1):227 – 242.

[136] Yao, Y. (2008). Probabilistic rough set approximations. *International Journal of Approximate Reasoning*, 49(2):255 – 271. Special Section on Probabilistic Rough Sets and Special Section on PGM'06.

[137] Yosinski, J., Clune, J., Nguyen, A. M., Fuchs, T. J. et Lipson, H. (2015). Understanding neural networks through deep visualization. *CoRR*, abs/1506.06579.

[138] Zadeh, L. (1965). Fuzzy sets. *Information Control*, 8:338–353.

[139] Zadeh, L. (1986). *Is Probability Theory Sufficient for Dealing With Uncertainty in AI?* Uncertainty in Artificial Intelligence. North-Holland.

[140] Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning. *Journal of Information Science*, page 199.

[141] Zadeh, L. A. (1977). Classification and clustering. chapitre Fuzzy Sets and Their Application to Pattern Classification and Clustering Analysis, pages 251–299. Academic Press, NJ, USA.

[142] Zadeh, L. A. (1995). Probability theory and fuzzy logic are complementary rather than competitive. *Technometrics*, 37(3):271–276.

[143] Zagoruyko, S. et Komodakis, N. (2016). Wide residual networks. *CoRR*, abs/1605.07146.

[144] Zanotelli, R. M., Reiser, R. H. S., da Costa Cavalheiro, S. A., Foss, L. et Bedregal, B. R. C. (2014). Robustness on the fuzzy f-xor class: Implication, bi-implications and dual constructions. *In 2014 XL Latin American Computing Conference (CLEI)*, pages 1–8.

[145] Zeiler, M. D., Krishnan, D., Taylor, G. W. et Fergus, R. (2010). Deconvolutional networks. *In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535.

[146] Zhang, C., Bengio, S., Hardt, M., Recht, B. et Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *ArXiv e-prints*.

[147] Zhang, Q., Xie, Q. et Wang, G. (2016). A survey on rough set theory and its applications. *CAAI Transactions on Intelligence Technology*, 1(4):323 – 333.

[148] Zimmermann, H.-J. (1985). *Decision Making in Fuzzy Environment*, pages 213–260. Springer Netherlands, Dordrecht.

[149] Zimmermann, H.-J. (2001). *Fuzzy Set Theory — and Its Applications*. Kluwer Academic Publishers, Norwell, MA, USA.

[150] Zimmermann, H.-J. (2010). Fuzzy set theory review. *Advanced Review*.

Appendix A: theoretical analysis of deep epitome

Definition 6. *For two given tuples* $\mathbf{x}^K = \{x_1, \ldots, x_K\}, \mathbf{y}^L = \{y_1, \ldots, y_L\}$, *the hamming outer product, denoted* $\bigoplus$, *is a set of corresponding elements* $\mathbf{x}^K \bigoplus \mathbf{y}^L = \big\{x_k \oplus y_l \big| k = 1 \ldots K; l = 1 \ldots L\big\}$, *where* $\oplus$ *denotes the generalized hamming distance operator. Then the product has following properties,*

1. non-commutative: in general $\mathbf{x}^K \bigoplus \mathbf{y}^L \neq \mathbf{y}^L \bigoplus \mathbf{x}^K$ *but they are permutation equivalent, in the sense that there exist permutation matrices* $\mathbf{P}$ *and* $\mathbf{Q}$ *such that* $\mathbf{x}^K \bigoplus \mathbf{y}^L = \mathbf{P}(\mathbf{y}^L \bigoplus \mathbf{x}^K)\mathbf{Q}$.

2. non-linear: in contrast to the standard outer product which is bilinear in each of its entry, the hamming outer product is non-linear since in general $\mathbf{x}^K \bigoplus (\mathbf{y}^L + \mathbf{z}^L) \neq (\mathbf{x}^K \bigoplus \mathbf{y}^L) + (\mathbf{x}^K \bigoplus \mathbf{z}^L)$ *and* $(\mu \mathbf{x}^K) \bigoplus \mathbf{y}^L \neq \mathbf{x}^K \bigoplus (\mu \mathbf{y}^L) \neq \mu(\mathbf{x}^K \bigoplus \mathbf{y}^L)$ *where* $\mu \in \mathcal{R}$ *is a scalar. Therefore, the hamming outer product defined as such is a pseudo outer product.*

3. associative: $\big(\mathbf{x}^K \bigoplus \mathbf{y}^L\big) \bigoplus \mathbf{z}^M = \mathbf{x}^K \bigoplus \big(\mathbf{y}^L \bigoplus \mathbf{z}^M\big) = \mathbf{x}^K \bigoplus \mathbf{y}^L \bigoplus \mathbf{z}^M$ *because of the associativity of GHD. This property holds for arbitrary number of tuples.*

4. iterated operation: the definition can be trivially extended to multiple tuples $\mathbf{x}^K \bigoplus \mathbf{y}^L \bigoplus \ldots \mathbf{z}^M = \big\{x_k \oplus y_l \oplus \ldots, z_m \big| k = 1 \ldots K; l = 1 \ldots L; \ldots; m = 1, \ldots M\big\}$.

Proof. associativity: by definition it suffices to prove element-wise $(x_k \oplus y_l) \oplus z_m = x_k \oplus (y_l \oplus z_m)$ because of the associativity of the generalized hamming distance.

non-linearity: by definition $\mathbf{x}^K \bigoplus (\mathbf{y}^L + \mathbf{z}^L)$ has elements $x_k \oplus (y_l + z_l)$, then it suffices to prove non-linearity for each element i.e. $x_k \oplus (y_l + z_l) = x_k + (y_l + z_l) - 2x_k(y_l + z_l) \neq (x_k + y_l - 2x_k y_l) + (x_k + z_l - 2x_k z_l) = (x_k \oplus y_l) + (x_k \oplus z_l)$. Similarly, $(\mu x_k \oplus y_l) = \mu x_k + y_l - 2\mu x_k y_l \neq x_k + \mu y_l - 2\mu x_k y_l \neq \mu(x_k + y_l - 2x_k y_l)$ in general. □

Definition 7. *The convolution of hamming outer product or hamming convolution, denoted* $\overset{*}{\bigoplus}$, *of two tuples is a binary operation that sums up corresponding hamming outer product entries:*

$$\mathbf{x}^K \overset{*}{\bigoplus} \mathbf{y}^L := \Big\{ \sum_{(k,l) \in S(n)} x_k \oplus y_l \Big| \text{ for } n = 1, \ldots, K + L - 1 \Big\} \tag{14}$$

where the subsets $S(n) := \{(k, l) \mid k + (L - l) = n\}$ *for* $n = 1, \ldots, K + L - 1$, *and the union of all subsets constitute a partition of all indices* $\bigcup_{n=1,\ldots,K+L-1} S(n) = \big\{(k, l) \big| k = 1 \ldots K; l = 1 \ldots L\big\}$. *The hamming convolution has following properties,*

1. commutative: $\mathbf{x}^K \overset{*}{\bigoplus} \mathbf{y}^L = \mathbf{y}^L \overset{*}{\bigoplus} \mathbf{x}^K$ *since the partition subsets* $S(n)$ *remains the same.*

2. non-linear: this property is inherited from the non-linearity of the hamming outer product.

3. non-associative: in general $\left(\mathbf{x}^K \bigoplus^* \mathbf{y}^L\right) \bigoplus^* \mathbf{z}^M \neq \mathbf{x}^K \bigoplus^* \left(\mathbf{y}^L \bigoplus^* \mathbf{z}^M\right)$ *since the summation of GHDs is non-associative. Note this is in contrast to the associativity of the hamming outer product.*

4. iterated operation: likewise, the definition can be extended to multiple tuples $\mathbf{x}^K \bigoplus^* \mathbf{y}^L \ldots \mathbf{z}^M = \Big\{ \sum_{(k,l,\ldots,m)\in S(n)} x_k \oplus y_l \ldots \oplus z_m \Big| \text{ for } n = 1, \ldots, K + (L-1) + \ldots + (M-1) \Big\}$.

Proof. non-associativity: by definition it suffices to prove element-wise in general $\sum_{(n,m)\in S'(n')} \big(\sum_{(k,l)\in S(n)} x_k \oplus y_l \big) \oplus z_m \neq \sum_{(k,n)\in S'(n')} x_k \oplus \big(\sum_{(l,m)\in S(n)} y_l \oplus z_m \big)$. □

Definition 8. *An epitome consists of a set of* N *pairs* $\mathbb{E} = \big\{(g_n, s_n), \big| n = 1, \ldots, N\big\}$ *where* g_n *denotes the summation of GHD entries from some hamming convolutions,* s_n *the number of summands or the cardinality of the subset* $S(n)$ *defined above, and* N *is called the length of the epitome.*

A normalized epitome is an epitome with $s_n = 1$ *for all* $n = 1, \ldots N$. *Any epitome can then be normalized by setting* $(g_n/s_n, 1)$ *for all elements. A normalized epitome may also refer to input data* $\mathbf{x}$ *or neuron weights* $\mathbf{w}$ *that are not yet involved in any convolution operations. In the latter case,* g_n *is simply the input data* $\mathbf{x}$ *or neuron weights* $\mathbf{w}$.

Given two tuples $\mathbf{x} = \{x_k | k = 1 \ldots K\}$ and $\mathbf{y} = \{y_l | l = 1 \ldots L\}$, then

$$\sum_k^K \sum_l^L (x_k \oplus y_l) = \big(\sum_k^K x_k\big) \oplus \big(\sum_l^L y_l\big) + (L-1) \sum_k^K x_k + (K-1) \sum_l^L y_l. \quad (15)$$

Proof.

$$\begin{aligned} LHS &= \sum_k^K \sum_l^L (x_k + y_l - 2x_k y_l) = L \sum_k^K x_k + K \sum_l^L y_l - 2 \sum_k^K x_k \sum_l^L y_l \\ &= \Big(\sum_k^K x_k + \sum_l^L y_l - 2 \sum_k^K x_k \sum_l^L y_l \Big) + (L-1) \sum_k^K x_k + (K-1) \sum_l^L y_l \\ &= \big(\sum_k^K x_k\big) \oplus \big(\sum_l^L y_l\big) + (L-1) \sum_k^K x_k + (K-1) \sum_l^L y_l = RHS \end{aligned}$$

□

Remark: eq. (15) allows one to compute summation of all hamming outer product elements on the right hand side, even though individual elements x_k and y_l are unable to recover from the given summands $\sum_k x_k$ and $\sum_l y_l$. The definition below immediately follows and illustrates how to merge elements of two epitomes.

Definition 9. *Given two epitomes* $\mathbb{E}_a = \{(g_n, s_n)|n = 1, \dots N\}, \mathbb{E}_b = \{(g'_m, s'_m)|m = 1, \dots M\}$*, the convolution of two epitomes* $\mathbb{E}_c = \mathbb{E}_a \bigoplus^* \mathbb{E}_b$ *is given by:*

$$\mathbb{E}_c = \{(g''_c, s''_c)|c = 1, \dots, N + M - 1\}; \tag{16a}$$

$$\text{where } g''_c = \sum_{(n,m)\in S(c)} \Big(g_n \oplus g'_m + (s'_m - 1)g_n + (s_n - 1)g'_m \Big), \tag{16b}$$

$$s''_c = \sum_{(n,m)\in S(c)} s_n s'_m, \tag{16c}$$

$$S(c) := \{(n, m)|n + (M - m) = c\}. \tag{16d}$$

Proof. For each pair of epitome elements (g_n, s_n) and (g'_m, s'_m) since by definition 8 $g_n = \sum_{k=1}^{s_n} x_k$ is a summation of elements and g'_m in the same vein, then the summation of hamming outer product elements $\sum_{k=1}^{s_n} \sum_{l=1}^{s'_m} (x_k \oplus y_l)$ follows eq. (15). The number of elements s''_c is simply the convolution of s_n and s'_m of two given epitomes. □

Remark: this operation is applicable to the case when two epitomes are merged via *spatial convolution* (see Figure 12 for an example). Note that this merging operation is *associative* due to the following theorem.

Theorem 1. *The convolution of multiple epitomes, as defined in 9, is associative:*

$$\mathbb{E}_a \overset{*}{\bigoplus} \mathbb{E}_b \overset{*}{\bigoplus} \mathbb{E}_c = (\mathbb{E}_a \overset{*}{\bigoplus} \mathbb{E}_b) \overset{*}{\bigoplus} \mathbb{E}_c = \mathbb{E}_a \overset{*}{\bigoplus} (\mathbb{E}_b \overset{*}{\bigoplus} \mathbb{E}_c). \tag{17}$$

Proof. By definition 9, elements of $\mathbb{E}_a \bigoplus^* \mathbb{E}_b$ are the summations of hamming outer product elements denoted by $\sum_{k=1}^{s_n} \sum_{l=1}^{s'_m} (x_k \oplus y_l)$. Then elements of $\big(\mathbb{E}_a \bigoplus^* \mathbb{E}_b\big) \bigoplus^* \mathbb{E}_c$ are the summation of hamming outer product elements $\sum_{k=1}^{s_n} \sum_{l=1}^{s'_m} \sum_{i=1}^{s''_q} (x_k \oplus y_l \oplus z_i)$, which are equal to elements of $\mathbb{E}_a \bigoplus^* \big(\mathbb{E}_b \bigoplus^* \mathbb{E}_c\big)$, due to the associativity of hamming outer products by definition 6.

□

Remark: this associative property is of paramount importance for the derivation of deep epitomes, which factor out the inputs $\mathbf{x}$ from subsequent convolutions with neuron weights $\mathbf{w}$.

Definition 10. *Given two epitomes of the same size* $\mathbb{E}_a = \{(g_n, s_n)|n = 1, \dots N\}, \mathbb{E}_b = \{(g'_n, s'_n)|n = 1, \dots N\}$*, the summation of two epitomes* $\mathbb{E}_c = \mathbb{E}_a \biguplus \mathbb{E}_b$ *is trivially defined by element-wise summation:*

$$\begin{aligned} \mathbb{E}_c &= \{(g''_n, s''_n)|n = 1, \dots, N\}; \\ \text{where } g''_n &= g_n + g'_n, \\ s''_n &= s_n + s'_n. \end{aligned} \tag{18}$$

Remark: the summation operation is applicable to the case when epitomes are (iteratively) merged cross different channels (see Figure 12 for an example). Note that the size of two input epitomes must be the same, and the size of output epitome remain unchanged. Moreover, the operation is trivially extended to multiple epitomes $\biguplus_{\{1,2,\ldots,M\}} := \mathbb{E}_1 \uplus \mathbb{E}_2 \uplus \ldots \uplus \mathbb{E}_M$.

Notation: for the sake of brevity, let $[^{M_a}\mathbb{A}^{L_a}_{C_a}]$ denote a *bank of epitomes*: $\left\{ {}^{m}\mathbb{A}^{L_a}_{C_a} \;\middle|\; m = 1, \ldots M_a \right\}$, where $\mathbb{A}^{L_a}_{C_a} = \left\{ \mathbb{A}^{L_a}_{c} \;\middle|\; c = 1, \ldots C_a \right\}$ are C_a-channels of length-L_a epitomes, and M_a is the number of epitomes as such in the bank or set $[\mathbb{A}]$. Figure 12 illustrates example banks of epitomes and two operations defined on them.

Definition 11. *The composite convolution of two banks of epitomes* $[^{M_a}\mathbb{A}^{L_a}_{C_a}]$ *and* $[^{M_b}\mathbb{B}^{L_b}_{C_b}]$ *with* $M_a = C_b$*, is defined as*

$$[^{M_a}\mathbb{A}^{L_a}_{C_a}] \overset{*}{\bigoplus} [^{M_b}\mathbb{B}^{L_b}_{C_b}] := \Big\{ \biguplus_{m_a=1, c_b=1}^{M_a, C_b} \left({}^{m_a}\mathbb{A}^{L_a}_{c_a} \overset{*}{\bigoplus} {}^{m_b}\mathbb{B}^{L_b}_{c_b} \right) \Big| \; c_a = 1, \ldots C_a; m_b = 1, \ldots M_b \Big\}. \tag{19}$$

The output of this operation, in turn, is a bank with M_b of C_a-channel length-$(L_a + L_b - 1)$ epitomes denoted as $[^{M_d}\mathbb{D}^{L_d}_{C_d}]$ with $M_d = M_b, C_d = C_a, L_d = L_a + L_b - 1$. See Figure 12 for an example.

The composite convolutions of multiple epitome banks, as given in definition 11, is associative:

$$\left([^{M_a}\mathbb{A}^{L_a}_{C_a}] \overset{*}{\bigoplus} [^{M_b}\mathbb{B}^{L_b}_{C_b}] \right) \overset{*}{\bigoplus} [^{M_c}\mathbb{C}^{L_c}_{C_c}] = [^{M_a}\mathbb{A}^{L_a}_{C_a}] \overset{*}{\bigoplus} \left([^{M_b}\mathbb{B}^{L_b}_{C_b}] \overset{*}{\bigoplus} [^{M_c}\mathbb{C}^{L_c}_{C_c}] \right) \tag{20}$$

Proof. The associativity immediately follows the associativity of Theorem 1 and definition 10. □

Remark: this associative property, which is inherited from theorem 1, can be trivially extended to multiple banks and lead to the main theorem of the paper as follows.

Theorem 2. *A generalized hamming network consisting of multiple convolution layers, is equivalent to a bank of epitome, called deep epitome* $[^{\star}\mathbb{D}^{\diamond}_{\triangledown}]$*, which can be computed by iteratively applying the composite hamming convolution in equation (19) to individual layer of epitomes:*

$$[^{\star}\mathbb{D}^{\diamond}_{\triangledown}] := [^{M_a}\mathbb{A}^{L_a}_{C_a}] \overset{*}{\bigoplus} [^{M_b}\mathbb{B}^{L_b}_{C_b}] \overset{*}{\bigoplus} \ldots \overset{*}{\bigoplus} [^{M_z}\mathbb{Z}^{L_z}_{C_z}], \tag{21}$$

in which $\triangledown = C_a$ *is the number of channels in the first bank* $\mathbb{A}$*,* $\star = M_z$ *is the number of epitomes in the last bank* $\mathbb{Z}$*, and* $\diamond = L_a + (L_b - 1) + \ldots + (L_z - 1)$ *is the length of composite deep epitome. Note that for the hamming convolution to be a valid operation, the number of epitomes in the previous layer and the number channels in the current layer must be the same e.g.* $C_b = M_a$*.*

Proof. For given inputs represented as a bank of normalized epitomes $[^{M_x}\mathbb{X}^{L_x}_{C_x}]$ the final network output $[^{M_z}Y^{L_y}_{C_x}]$ is obtained by recursively applying equation (19) to outputs from the previous layers, and factoring out the input due to the associativity proved in proposition 4:

$$\begin{aligned}[^{M_z}Y^{L_y}_{C_x}] &= \left(\left(([^{M_x}\mathbb{X}^{L_x}_{C_x}] \overset{*}{\bigoplus} [^{M_a}\mathbb{A}^{L_a}_{C_a}]) \overset{*}{\bigoplus} [^{M_b}\mathbb{B}^{L_b}_{C_b}]\right) \overset{*}{\bigoplus} \cdots \overset{*}{\bigoplus} [^{M_z}\mathbb{Z}^{L_c}_{C_z}]\right) \\ &= [^{M_x}\mathbb{X}^{L_x}_{C_x}] \overset{*}{\bigoplus} \underbrace{\left([^{M_a}\mathbb{A}^{L_a}_{C_a}] \overset{*}{\bigoplus} [^{M_b}\mathbb{B}^{L_b}_{C_b}] \overset{*}{\bigoplus} \cdots \overset{*}{\bigoplus} [^{M_z}\mathbb{Z}^{L_c}_{C_z}]\right)}_{[^{\star}\mathbb{D}^{\diamond}_{\nabla}]}. \end{aligned} \tag{22}$$

□

Network architectures used in Appendices B,C,D:

We summarize in Tables below architectures of three generalized hamming networks trained with MNIST, CIFAR10/100 classification respectively. Note that for kernels with stride 2, we resize original kernels to their effective size ($\times 2$) when computing deep epitomes. Also we use average-pooling, instead of max-pooling, in all three networks. Subsequent fully connected layers are also reported although they are not involved in the computation of deep epitomes.

GHN for MNIST classification					
Layers	Kernel	Str.	resized	Ch I/O	**Epitome size**
conv1	5x5	1	5x5	3 / 32	5x5
conv2	5x5	2	10x10	32 / 32	14x14
conv3	5x5	2	10x10	32/ 128	23x23
fc1	-	-	-	* / 1024	-
fc2	-	-	-	1024 / 10	-

GHN for CIFAR10 classification					
Layers	Kernel	Str.	resized	Ch I/O	**Epitome size**
conv1	3x3	1	3x3	3 / 64	3x3
conv2	3x3	1	3x3	64 / 64	5x5
conv3	5x5	2	10x10	64 / 256	14x14
conv4	5x5	2	10x10	256 / 256	23x23
fc1	-	-	-	*/1024	-
fc2	-	-	-	1024/512	-
fc3	-	-	-	512/10	-

GHN for CIFAR100 classification					
Layers	Kernel	Str.	resized	Ch I/O	**Epitome size**
conv1	3x3	1	3x3	3 / 64	3x3
conv2	5x5	2	10x10	64 / 64	12x12
conv3	5x5	1	5x5	64 / 64	16x16
conv4	5x5	1	5x5	64 / 64	20x20
conv5	5x5	1	5x5	64 / 64	24x24
conv6	5x5	1	5x5	64 / 64	28x28
conv7	5x5	2	10x10	64 / 128	37x37
fc1	-	-	-	*/1024	-
fc2	-	-	-	1024/512	-
fc3	-	-	-	512/10	-

Appendix B: deep epitomes with MNIST handwritten recognition

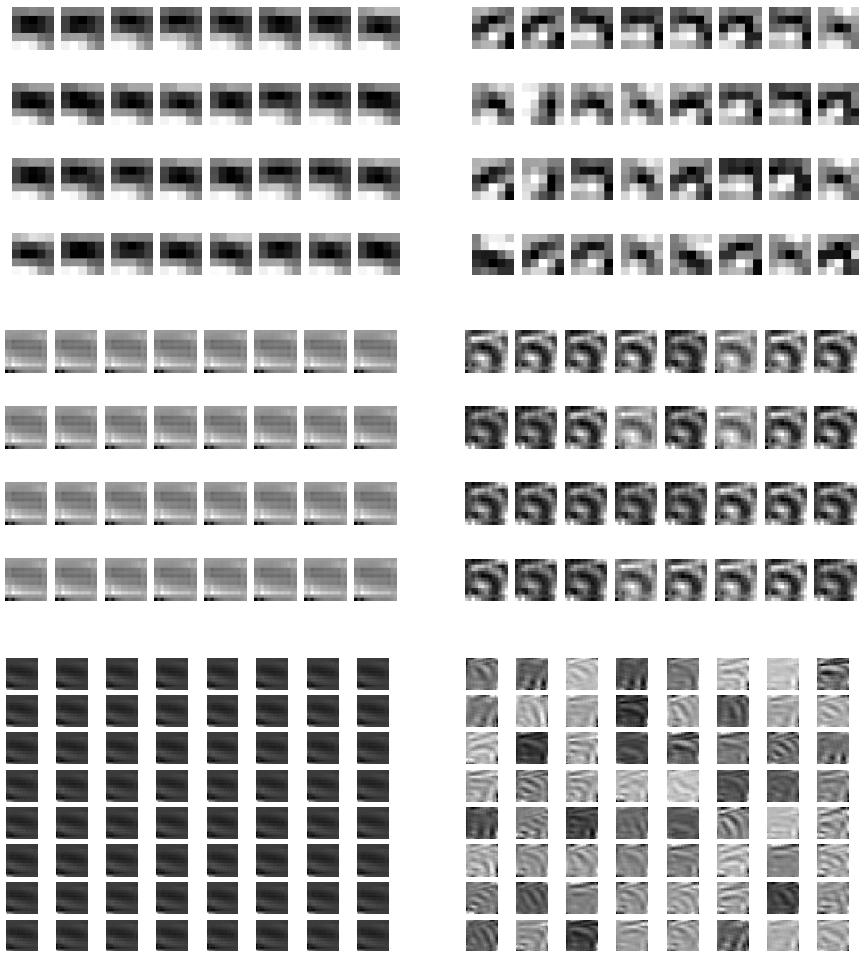


Fig. 15: Deep epitomes at layers 1,2 and 3 for a GHN trained with MNIST classification at iterations 100 and 10000 respectively.

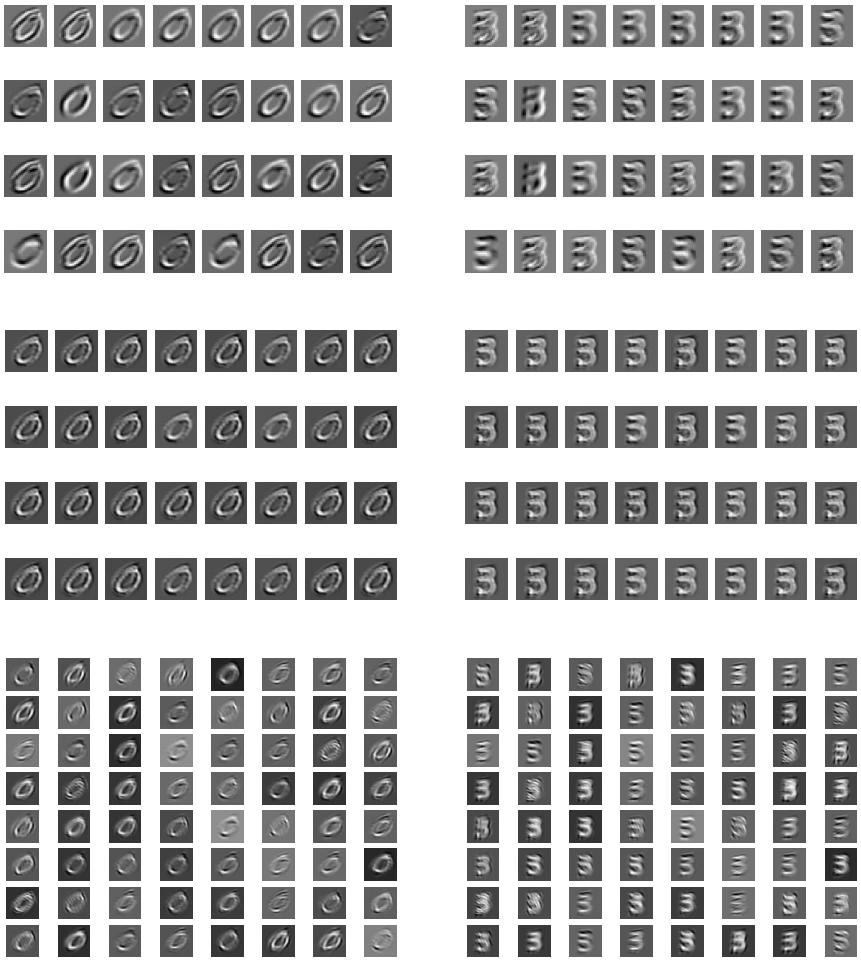


Fig. 16: Example hierarchical features extracted at layers 1,2, and 3 for a GHN trained with MNIST classification.

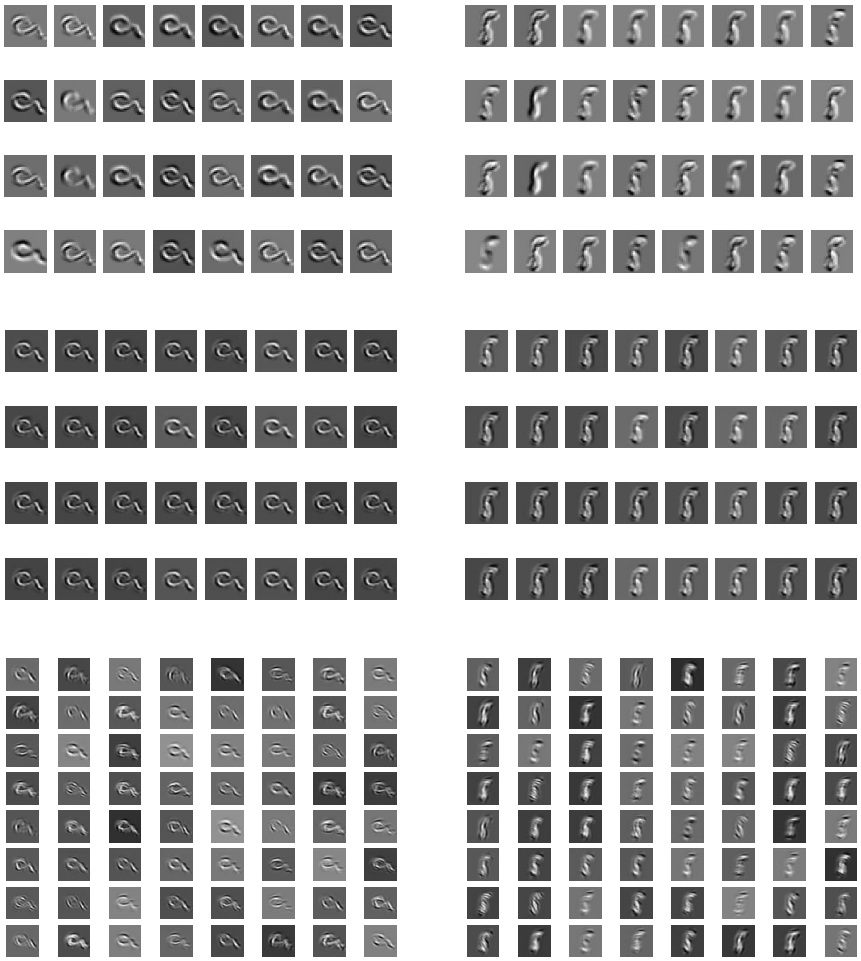


Fig. 17: Example hierarchical features extracted at layers 1,2, and 3 for a GHN trained with MNIST classification.

Appendix C: deep epitomes with CIFAR10 image classification

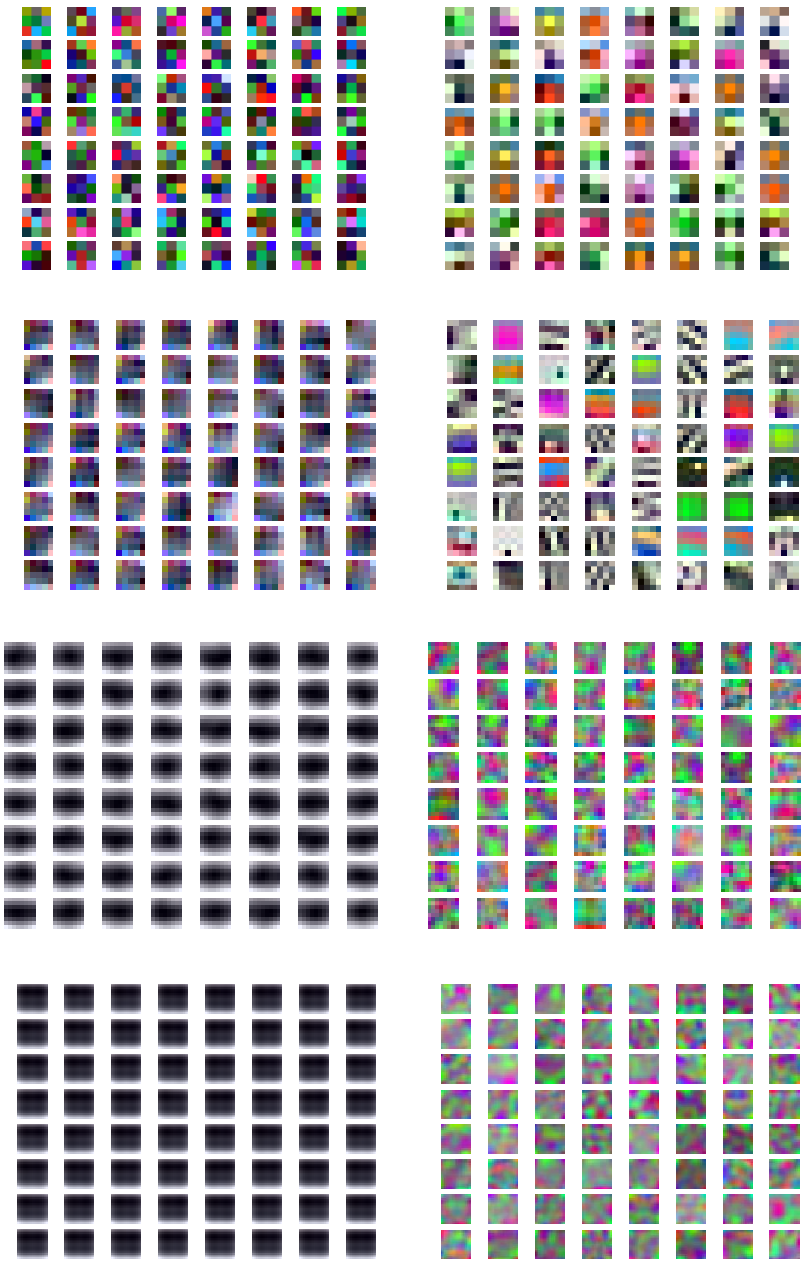


Fig. 18: Top to bottom: deep epitomes for first 64 filters at layers 1,2, 3 and 4 of a GHN trained with CIFAR10 classification. Pseudo colour images correspond to three channels of merged epitomes from the first layer filters. **Left** column: iteration 100; **Right** column: iteration 180000.

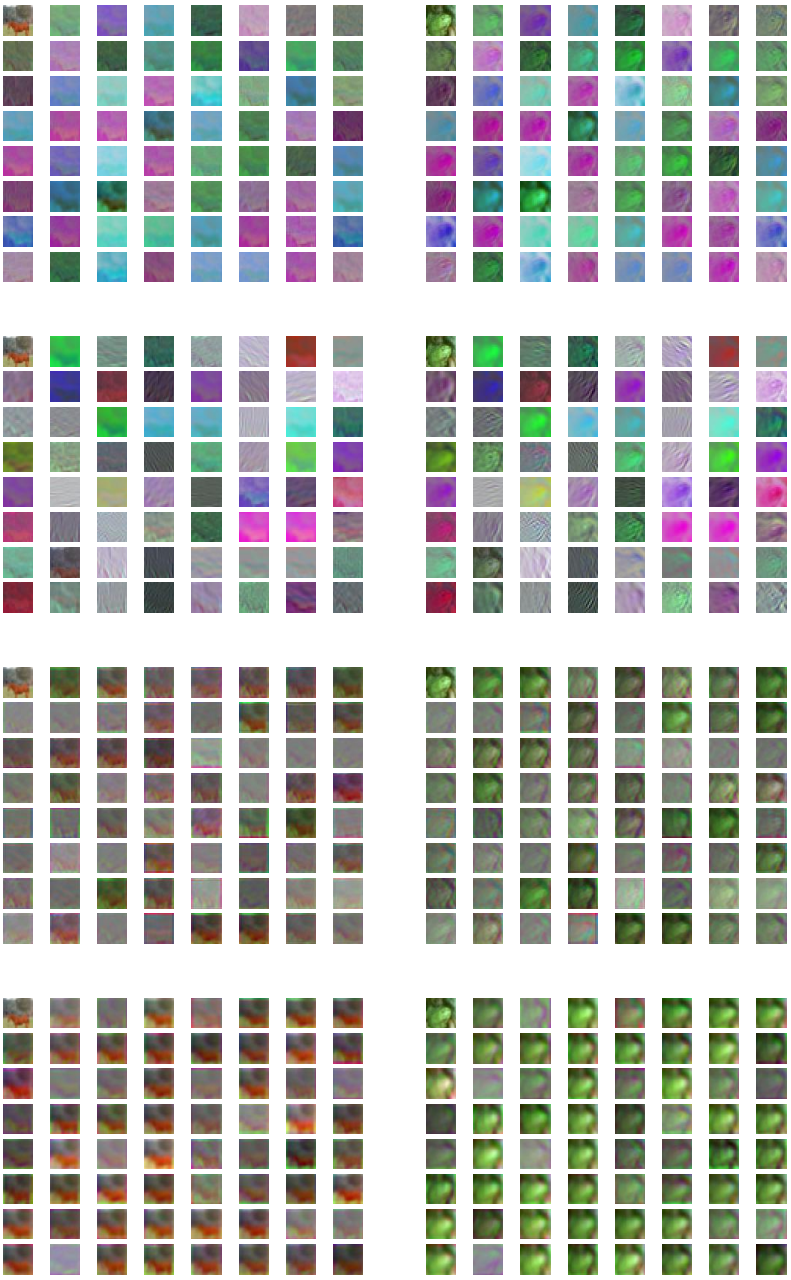


Fig. 19: Hierarchical features extracted at layers 1,2,3 and 4 for a GHN trained with CIFAR10 at 180000 iterations. The top-left most image in each panel is the input image, and the rest are features extracted with different epitomes (only first 63 features are shown for layer 4). Pseudo colour images correspond to three channels of features outputs for input RGB colour channels. Note that oriented edgelets (layer 1,2), textons with associated colours (layer 2,3) and rough segmentations (layer 4) are extracted from different layers.

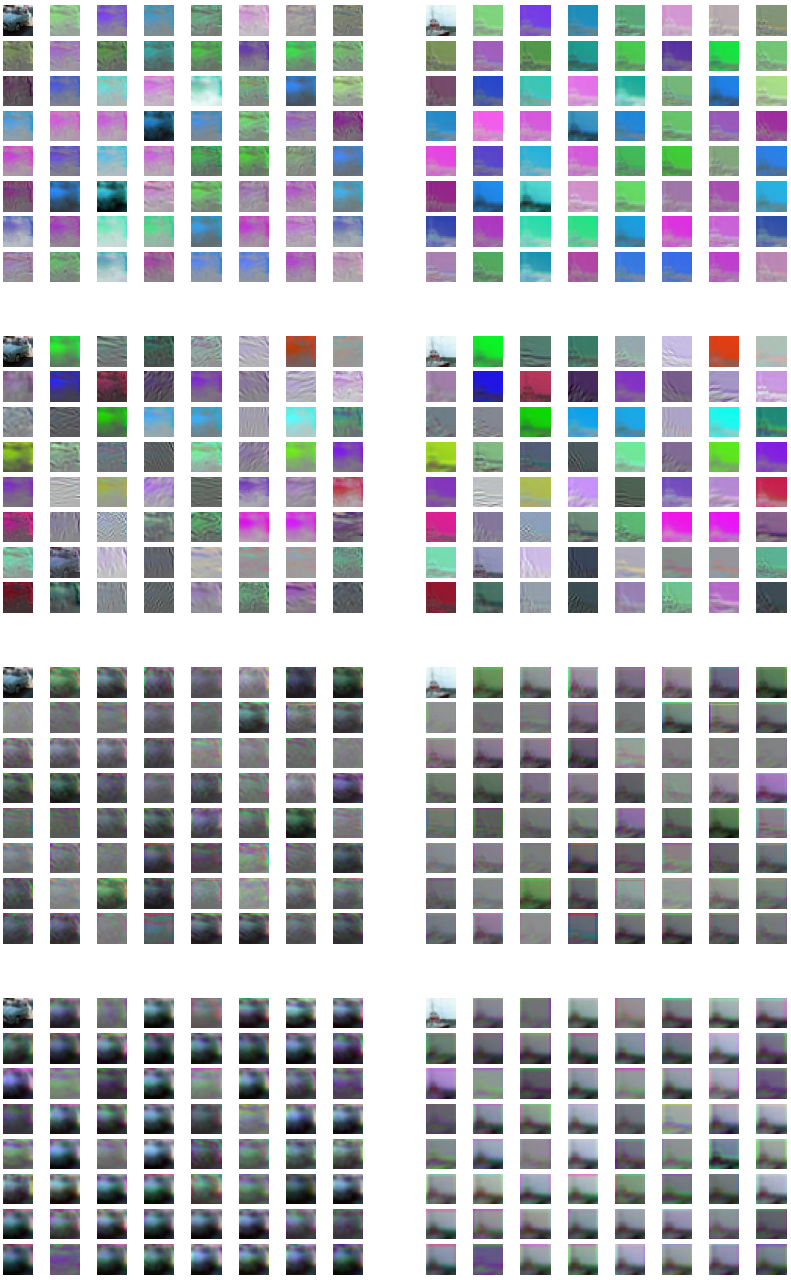


Fig. 20: Hierarchical features extracted at layers 1,2,3 and 4 for a GHN trained with CIFAR10 at 180000 iterations. The top-left most image in each panel is the input image, and the rest are features extracted with different epitomes (only first 63 features are shown for layer 4). Pseudo colour images correspond to three channels of features outputs for input RGB colour channels. Note that oriented edgelets (layer 1,2), textons with associated colours (layer 2,3) and rough segmentations (layer 4) are extracted from different layers.

Appendix D: deep epitomes with CIFAR100 image classification

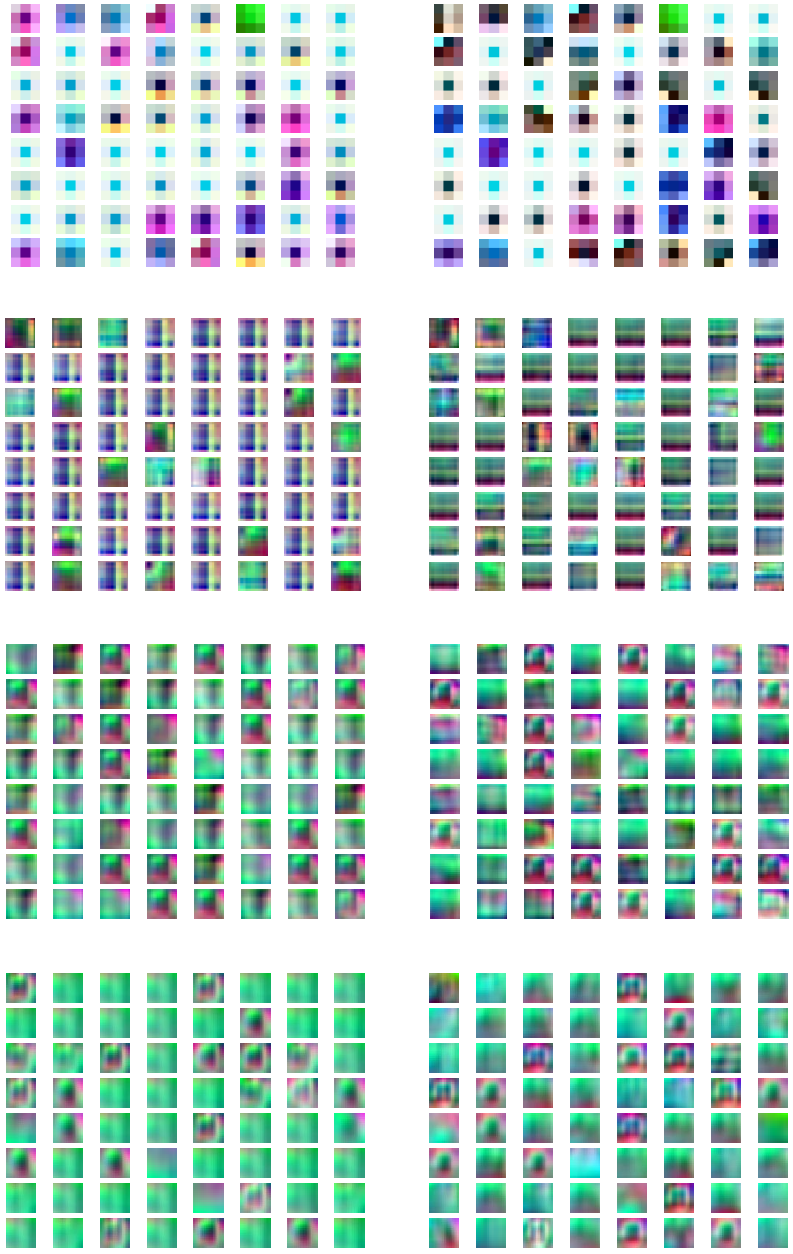


Fig. 21: Top to bottom: deep epitomes for first 64 filters at layers 1,2, 3 and 4 of a GHN trained with CIFAR100 classification. Pseudo colour images correspond to three channels of merged epitomes from the first layer filters. **Left** column: iteration 10000; **Right** column: iteration 30000.

Fig. 22: Top to bottom: deep epitomes for first 64 filters at layers 5, 6 and 7 of a GHN trained with CIFAR100 classification. Pseudo colour images correspond to three channels of merged epitomes from the first layer filters. **Left** column: iteration 10000; **Right** column: iteration 30000.

Fig. 23: Hierarchical features extracted at layers 1,2,3 and 4 for a GHN trained with CIFAR100 at 10000 iterations. The top-left most image in each panel is the input image, and the rest are features extracted with different epitomes (only first 63 features are shown for different layers). Pseudo colour images correspond to three channels of features outputs for input RGB colour channels.

Fig. 24: Hierarchical features extracted at layers 5,6 and 7 for a GHN trained with CIFAR100 at 10000 iterations. The top-left most image in each panel is the input image, and the rest are features extracted with different epitomes (only first 63 features are shown for different layers). Pseudo colour images correspond to three channels of features outputs for input RGB colour channels.