**Chapter**

# A Hierarchy of Hilbert Spaces for Computing with Granular Semantics by Deep Learning Enhanced

*Vassilis Kaburlasos, George Siavalas and Lola Bris*

## Abstract

The all-popular Artificial Intelligence (AI) is pursued by deep learning (neural computing) models implemented in Data Centers (DCs) that require considerable resources. The aforementioned models are typically developed in the Euclidean (Hilbert) space $R^N$, where R is the set of real numbers for an integer N. This work introduces a Hibert space, namely, space of Generalized Intervals' Numbers (GINs) G as a hierarchy of Hilbert spaces stemming from R. Data processing is carried out in the convex cone $F_1$ of G, namely, space of Intervals' Numbers (INs), which includes partially ordered, Lebesque space $L_2$ distribution functions that may represent information granules, for example, probability/possibility distributions. It is detailed how enhanced deep learning models can be developed in $F_1^N$. As $F_1$ is a strict superset of R, all the conventional deep learning algorithms in $R^N$ are included. Additional advantages of deep learning in $F_1^N$ include (a) accommodation of spoken language semantics, represented by partial order, (b) potential engagement of axiomatic logic all along during data processing, and (c) tuning the number of tunable parameters toward reducing the demands for DC resources, including energy consumption, by engaging fewer models of greater flexibility without decreasing performance.

**Keywords:** deep learning, fuzzy lattice reasoning (FLR), granular semantics computing, hierarchy, lattice computing (LC), Hilbert space

## 1. Introduction

The influential review "Deep Learning" [1] is widely acknowledged for solidifying deep learning (neural computing) models as a foundational paradigm in Artificial Intelligence (AI). Deep learning is the core technology that has enabled the development of such popular technologies as Convolutional Neural Networks (CNNs) [2] for image recognition and Large Language Models (LLMs) [3] for human-like language generation. Deep learning models are currently implemented exclusively in the Euclidean (Hilbert) space $R^N$ by processing vectors of numbers.



IntechOpen

Conventional modeling has been largely initiated by Isaac Newton [4], regarding the physical world, based on measurements following, in a sense, the ancient Pythagorean doctrine that "(rational) numbers is the ultimate reality" – note that numbers emerge from physical world measurements. The aforementioned modeling has worked well for centuries in the physical world, for example, Maxwell's equations and/or Einstein's equations, and it has been extended successfully to alternative application domains such as the economy, physiology, psychology, and others. Lately, the aforementioned modeling has been extended to AI deep learning (statistical) models [1], whose remarkable performance can be attributed to the capacity of digital computer hardware [5] to process vast data fast rather than to compute differently. In all, conventional modeling deals exclusively with "flat data," that is, arrays of numbers. Nevertheless, when humans are involved, then non-numerical *percepts* emerge such as the degree of truth of propositions, data structures and symbols – note that percepts become data as soon as they are recorded.

Starting with "Industry 3.0," there has been an increasing demand for models that involve non-numerical human percepts and, historically, the truth values of propositions were among the first ones studied resulting in Boolean algebra/logic. It is noteworthy that Boolean algebra is the "par excellence" instrument for design in the digital computer industry today. In turn, the study of Boolean algebra has resulted in the introduction of mathematical Lattice Theory (LT) or, Order Theory, whose preeminent feature is a unifying capacity [6].

It is remarkable that non-numerical data have emerged in Physics as well. For instance, the 1954 Nobel prize laureate Max Born has instrumentally employed probability theory in quantum mechanics – note that Andrey Kolmogorov had already formalized probability theory in 1933 by introducing the notion of Probability Space. The latter was shown to be a mathematical lattice [6].

The proliferation of computers has triggered a sustained interest in applications of mathematical lattices [7, 8]. In conclusion, the lattice computing (LC) paradigm has been proposed [9] as a modeling paradigm shift to a lattice data domain, including $R^N$, where partial order may represent semantics.

The interest of this work is in a specific lattice, namely, the lattice of (type 1) Intervals' Numbers (INs), symbolically $(F_1,\preccurlyeq)$, where an IN may be interpreted either as a real number or as an information granule; the latter may be either a probability distribution or a fuzzy number [10].

The novelty of this work is the introduction of a hierarchy of Hilbert spaces stemming from R, culminating in the introduction of the space of Generalized Intervals' Numbers (GINs) $G_1$, where $G_1 \supset F_1$, within which $F_1$ is a convex cone. Hence, potentially useful mathematical tools are introduced in $F_1$ including a *similarity cosine* function between vectors of INs. Given the mathematical prerequisites for neurocomputing, namely (p1) a Hilbert space $H$, (p2) non-linear operations in $H$, and (p3) a derivative in $H$, this work further proposes a promising extension of deep learning to $F_1^N$ toward computing with (granular) semantics; axiomatic logic can also be involved by Fuzzy Lattice Reasoning (FLR) as explained below. Furthermore, a reduction of energy consumption is proposed (as a conjecture here to be confirmed in future work) by decreasing the total number of neurons in a deep learning architecture without decreasing the total number of parameters toward retaining performance.

The layout of this paper is as follows. Section 2 introduces the principal contribution of this work, that is, a novel mathematical background for enhancing conventional deep learning models. Section 3 describes an enhanced neural architecture. Section 4 delineates potential applications including CNNs and LLMs. Section 5 concludes by discussing both the proposed techniques and potential future work extensions. Appendix A includes mathematical proofs; Appendix B includes code (software) that implements selected functions.

## 2. The convex cone of intervals' numbers (INs)

The interest of this work is in *complete* mathematical lattices (L, $\sqsubseteq$) with minimum and maximum elements $o$ and $i$, respectively. A lattice (L, $\sqsubseteq$) is equipped with a *positive valuation* (real) function $v$: L → R, which, by definition, satisfies both a) $v(x) + v(y) = v(x \sqcap y) + v(x \sqcup y)$ and b) $x \sqsubset y \Rightarrow v(x) < v(y)$. A positive valuation results in a metric distance function $d$: L × L→$R_0^+$ given by $d(x,y) = v(x \sqcup y) - v(x \sqcap y)$.

### 2.1 Fuzzy lattice reasoning (FLR)

An *order measure* function σ: L × L → [0,1] in a lattice (L, $\sqsubseteq$) is defined by the two axioms: (A1) $u \sqsubseteq w \Leftrightarrow \sigma(u,w) = 1$ and (A2) $u \sqsubseteq w \Leftrightarrow \sigma(x,u) \leq \sigma(x,w)$. The following "reasonable axiom" may also be considered: (A0) $\sigma(x,o) = 0$, $\forall x \sqsupset o$ and $d(x,i) < +\infty$, $\forall x \in$L. Any employment of order measure function σ(.,.) is called *fuzzy lattice reasoning*, or FLR for short [11–16]. The FLR enables four types of axiomatic reasoning, namely, 1) inductive reasoning, 2) deductive reasoning, 3) reasoning by analogy, and 4) abductive reasoning.

An order measure function has been defined by function "sigma join" $\sigma_\sqcup(x,u) = v(u)/v(x \sqcup u)$ as well as by function "sigma meet" $\sigma_\sqcap(x,u) = v(x \sqcap u)/v(x)$, where $v(.)$ is a (parametric) positive valuation function. In particular, the order measure $\sigma_\sqcap(.,.)$ is the well-known "Rule of Bayes"; moreover, both order measures $\sigma_\sqcap(.,.)$ and $\sigma_\sqcup(.,.)$ are widely (though implicitly) used by Fuzzy Inference Systems (FISs) [12]. Note that function σ(.,.) has emerged from a unified generalization of Adaptive Resonance Theory's (ART's) *vigilance parameter* and *choice* (*Weber*) *function* [17]. Order measures are useful toward introducing axiomatic logic/reasoning in deep learning.

Given an order measure $\sigma_i$, in a "constituent" lattice ($L_i$, $\sqsubseteq_i$), i∈{1, … , N}, an order measure σ is defined in the Cartesian product lattice ($L_1$, $\sqsubseteq_1$) × … × ($L_N$, $\sqsubseteq_N$) by the following convex combination function

$$\sigma_c(\bar{F}, \bar{E}) = k_1\sigma_1(F_1, E_1) + \ldots + k_N\sigma_N(F_N, E_N), \quad (1)$$

where $k_1, \ldots, k_N > 0$ such that $k_1 + \ldots + k_N = 1$, resulting in a capacity for rigorously fusing, hierarchically, disparate types of data.

It has been shown how an order measure can be extended to the lattice of intervals in a lattice (L, $\sqsubseteq$) based, in addition, on a *dual isomorphic* function $\theta$: L → L resulting in the positive valuation function $V([a,b]) = v(\theta(a)) + v(b)$ [13]. As much of the work below involves intervals, an order measure function may also be denoted as $\sigma(.,.; v,\theta)$ to indicate explicitly the underlying (parametric) functions $v(.)$ and $\theta(.)$.

### 2.2 The Hilbert space $G_1$ of generalized intervals' numbers (GINs)

The interest here is in complete sublattices (L, $\sqsubseteq$) stemming from the complete, totally ordered lattice ($\bar{R}$,≤), where $\bar{R}$ = R∪{−∞,+∞} and R is the set of real numbers with minimum and maximum elements $o = -\infty$ and $i = +\infty$, respectively. A number of mathematical results are presented next.

First, recall the definition of a (real) vector space [18].

**Definition 1**: Consider a set $V$ and the field R of real numbers satisfying the following two requirements: (i) given an arbitrary pair $(a, b)$ of elements in $V$, there exists a unique element $a + b$ (called *sum* of $a, b$) in $V$; (ii) given an arbitrary element $\kappa$ in R and an arbitrary element $a$ in $V$, there exists a unique element $\kappa a$ (called *scalar multiple* of $a$ by $\kappa$) in $V$. The set $V$ is called a *linear space over* R (or, *vector space over* R) if the following eight conditions are satisfied: (i) $(a + b) + c = a + (b + c)$; (ii) there exists an element $0 \in V$, called the *zero element* of $V$, such that $a + 0 = 0 + a = a$ for all $a \in V$; (iii) For any $a \in V$, there exists an element $x = -a \in V$ satisfying $a + x = x + a = 0$; (iv) $a + b = b + a$; (v) $\kappa(a + b) = \kappa a + \kappa b$; (vi) For $\lambda \in$R, $(\kappa\lambda)a = \kappa(\lambda a)$; (vii) For $\lambda \in$R, $(\kappa + \lambda)a = \kappa a + \lambda a$; (viii) $1a = a$ (where 1 is the unity element of R).

An element of $V$ is called *vector*; furthermore, an element of R is called *scalar*. R is called the *field of scalars*.

Second, recall the definition of a (real) Hilbert space [18].

**Definition 2**: Let R be the field of real numbers the elements of which are denoted by $a, b, \ldots$; furthermore, let $H$ be a linear space over R. To any pair of vectors $x,y \in H$, let us correspond a number $<x, y> \in$R satisfying the following five conditions: (i) $<x_1 + x_2, y> = <x_1, y> + <x_2, y>$; (ii) $<ax, y> = a<x, y>$; (iii) $<x, y> = <y, x>$; (iv) $<x, x> \geq 0$; and (v) $<x, x> = 0 \Leftrightarrow x = 0$. Then, $H$ is called a *pre-Hilbert space* and $<x,y>$ is called *inner product* of $x$ and $y$. With the norm $\|x\| = \sqrt{<x,x>}$, $H$ is a normed linear space. If $H$ is complete with respect to the metric distance $\|x - y\|$, that is, $\|x_n - y_m\| \to 0$ $(m, n \to \infty)$ implies the existence of $\lim_{n\to\infty} x_n = x \in H$, then $H$ is called a (real) Hilbert space.

A well-known Hilbert space example is the Euclidean space $R^N$, where R is the set of real numbers for an integer N.

An inner product may extend to a Cartesian product as follows.

**Lemma 3**: Assume the Cartesian product $H^N$, where N is integer, of a pre-Hilbert (inner product) space $H$. Let the sum $\bar{x}+\bar{z}$, where $\bar{x} = (x_1, \ldots, x_N)$ and $\bar{z} = (z_1, \ldots, z_N)$, be defined as $\bar{x}+\bar{z} = (x_1 + z_1, \ldots, x_N + z_N)$; moreover, let the scalar multiple $\kappa\bar{x}$ of $\bar{x} \in H^N$ by $\kappa \in$R be defined as $\kappa\bar{x} = (\kappa x_1, \ldots, \kappa x_N)$. Then, the (real) function $<.,.>: H^N \times H^N \to R$ defined as $<\bar{x}, \bar{y}> = <(x_1, \ldots, x_N), (y_1, \ldots, y_N)> = \sum_1^N \langle x_i, y_i \rangle$ is an inner product in $H^N$.

The proof of Lemma 3 is shown in Appendix A.

A number of lemmas are considered next.

**Lemma 4**: Let $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$ be the lattice of *generalized intervals*, where R is the field of real numbers. Let the *addition* of two generalized intervals $[a_1, b_1]$ and $[a_2, b_2]$ be defined as $[a_1, b_1] + [a_2, b_2] = [a_1 + a_2, b_1 + b_2]$; moreover, let the *scalar multiple* of vector $[a,b] \in \bar{R} \times \bar{R}$ by the scalar $\lambda \in$R be defined as $\lambda[a,b] = [\lambda a, \lambda b]$. Then, $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$ is a *vector space* over R.

The proof of Lemma 4 is shown in Appendix A.

**Lemma 5**: In vector space $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$ of generalized intervals, the mapping $<.,.>: V_I \times V_I \to R$ given by $<[a_1, b_1], [a_2, b_2]> = \frac{1}{2}(a_1a_2 + b_1b_2)$ is an *inner product*.

The proof of Lemma 5 is shown in Appendix A.

The coefficient $\frac{1}{2}$ in Lemma 5 is not critical, and it was inserted to align the result, when both $[a_1, b_1]$ and $[a_2, b_2]$ are trivial representing single real numbers, with the corresponding result between the real numbers. For example, let $a,b \in$R be represented by the trivial generalized intervals $[a, a]$ and $[b, b]$, respectively; then, the (inner) product of the real numbers $a$ and $b$ equals $ab$; moreover, the (inner) product of the trivial vectors $[a, a]$ and $[b, b]$ equals $\frac{1}{2}<[a, a], [b, b]> = \frac{1}{2}(ab + ab) = ab$.

$V_I$ is a *pre-Hilbert* (normed linear) space with norm $\|[a, b]\| = \sqrt{\frac{1}{2}\left(a^2 + b^2\right)}$.

**Lemma 6**: The inner product space $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$ of generalized intervals is *complete* with respect to the metric distance $\|[a_n, b_n] - [c_m, d_m]\|$.

The proof of Lemma 6 is shown in Appendix A.

Based on Definition 2, as well as on Lemmas 4, 5 and 6, it follows that $V_I$ is a (real) Hilbert space.

A *Generalized Intervals' Number* (GIN) is defined as a function $E$: $[0,1] \rightarrow (\bar{R}\times\bar{R}, \geq \times \leq) = V_I$. The value $E(h)$ may, alternatively, be denoted as $E_h$. Let G denote the set of GINs. It turns out that (G, $\preccurlyeq$) is a complete lattice, as the Cartesian product (G, $\preccurlyeq$)= $\times(\bar{R}\times\bar{R}, \geq \times \leq)_h$, where $h\in[0,1]$, whose dimension is uncountably infinite.

The previous results extend to G, next.

**Theorem 7**: Consider the space G of GINs and let R be the field of real numbers. Let (a) the *addition* of two GINs $E_1$ and $E_2$ be defined as $E_s(h) = E_1(h) + E_2(h)$, $h\in[0,1]$ and (b) the *scalar multiplication* of a scalar $\lambda\in$R times a GIN $E$ be defined as $E_p(h) = \lambda E(h)$, $h\in[0,1]$. Then, G is a *vector space* over R.

The proof of Theorem 7 is shown in Appendix A.

**Remark 8**: As the interest of this work is in (deep learning) applications rather than in abstract mathematics, the Lebesgue space $L_2$ of square-integrable functions will be considered here exclusively. Recall that $L_2$ is the only Hilbert space among integrable $L_p$ spaces of functions, where $p > 0$. In particular, the *inner product* of two functions $f,g\in L_2$, whose domain is the interval [0,1], is defined by $<f(h), g(h)> = \int_0^1 f(h)g(h)dh$ [18] (168B, 197D).

The following Corollary extends the inner product of Lemma 5 to the vector space G, in a straightforward manner, based on (a) Remark 8 and (b) Lemma 3.

**Corollary 9**: Let $G_1 = \{[a(h), b(h)]: a(.), b(.)\in L_2\}$, $F_h = [a_1(h), b_1(h)]$ and $E_h = [a_2(h), b_2(h)]$, $h\in[0,1]$. Then, the mapping $<.,.>$: $G_1 \times G_1 \rightarrow$ R given by $<F,E> = \int_0^1 \langle F_h, E_h\rangle dh = =\frac{1}{2}\int_0^1 [a_1(h)a_2(h) + b_1(h)b_2(h)]dh$ is *inner product*.

It is pointed out that $G_1 \subset G$ is a vector space because it satisfies all the eight conditions of Theorem 7. The completeness of $G_1$ is shown next.

**Theorem 10**: The inner product space $G_1$ of GINs is complete with respect to the metric distance $\|E_n - E_m\|$, where $E_i$, $i = 1,2, \ldots$ is a Cauchy sequence.

The proof of Theorem 10 is shown in Appendix A. It is pointed out that any violation on null sets can be modified without changing the $L_2$ class.

Based on Definition 2, as well as on both Corollary 9 and Theorem 10, it follows that $G_1$ is a (real) Hilbert space.

It has been acknowledged that the closest relatives of Euclidean spaces are Hilbert spaces [19]. Since the utility of Euclidean spaces is well established during millennia of practice, the potential of the proposed Hilbert space needs to be scrutinized. In particular, this work considers a result from the theory of *inner product spaces*, namely, the Cauchy-Schwarz inequality $|<x, y>| \leq \|x\| \, \|y\|$ that computes the *similarity cosine* function ρ: $V \times V \rightarrow [-1,1]$ by

$$\rho(x, y) = \frac{<x, y>}{\|x\| \, \|y\|}, , \tag{2}$$

toward calculating the angle $\measuredangle$(x,y) between any two vectors $x$,$y$ in an inner product space $V$ as $\measuredangle(x,y) = \arccos\frac{<x,y>}{\|x\| \, \|y\|}$.

## 2.3 The convex cone $F_1$ of intervals' numbers (INs)

The "resolution identity theorem," of fuzzy set theory, has shown that a fuzzy set can be equivalently represented by either its membership function or its α-cuts [12]. Note that the α-cuts of fuzzy numbers are intervals. Recall that the universe of discourse $[o, i]$ here is a totally ordered, complete lattice ($\mathsf{L} = [o,i]$, $\leq$) with minimum element $o \in \bar{\mathsf{R}}$ and maximum element $i \in \bar{\mathsf{R}}$, where $\bar{\mathsf{R}} = \mathsf{R} \cup \{-\infty, +\infty\}$ is the totally ordered lattice of real numbers with minimum element "$-\infty$" and maximum element "$+\infty$". The complete lattice of intervals in ($\mathsf{L} = [o,i]$, $\leq$), denoted by ($\mathsf{I}_1$, $\subseteq$), is defined as $\mathsf{I}_1 = \{[a,b] \in (\mathsf{L} \times \mathsf{L}, \geq \times \leq): a \leq b\} \cup \{\emptyset\}$, where $\mathsf{L} = [o, i] \subseteq \bar{\mathsf{R}}$ with $o < i$ and ($\mathsf{L} \times \mathsf{L}$, $\geq \times \leq$) is the lattice of *generalized intervals* in L. The empty interval in $\mathsf{I}_1$ is denoted by $[i, o]$, and it corresponds to all generalized intervals $[a, b]$ with $a > b$.

An IN is defined by (C0) a function $E$: $[0,1] \to \mathsf{I}_1$ such that (C1) $h_1 \leq h_2 \Rightarrow E_{h_1} \supseteq E_{h_2}$ and (C2) $\forall S \subseteq [0,1]$: $\bigcap_{h \in S} E_h = E_{\vee S}$. Note that, typically, in applications, the *height* of an IN $E$ equals 1 meaning that the range of IN $E$: $[0,1] \to \mathsf{I}_1$ does not include the empty interval [i, o] on a domain of non-zero measure. An IN is a mathematical object which may represent either a possibility distribution or a probability distribution [10] as explained next with reference to **Figure 1**.

**Figure 1** displays INs with interval support [1.80, 3.15]. Each IN was induced from 70 samples. In particular, IN $F$ was induced by Algorithm "distrIN" [20], whereas IN $G$ by Algorithm CALFIN [21]. Note that the interval representation of an IN requires two numbers per level, that is, the two interval ends. In particular, regarding the interval representation of IN $F$ in **Figure 1(c)** only one number is required per level

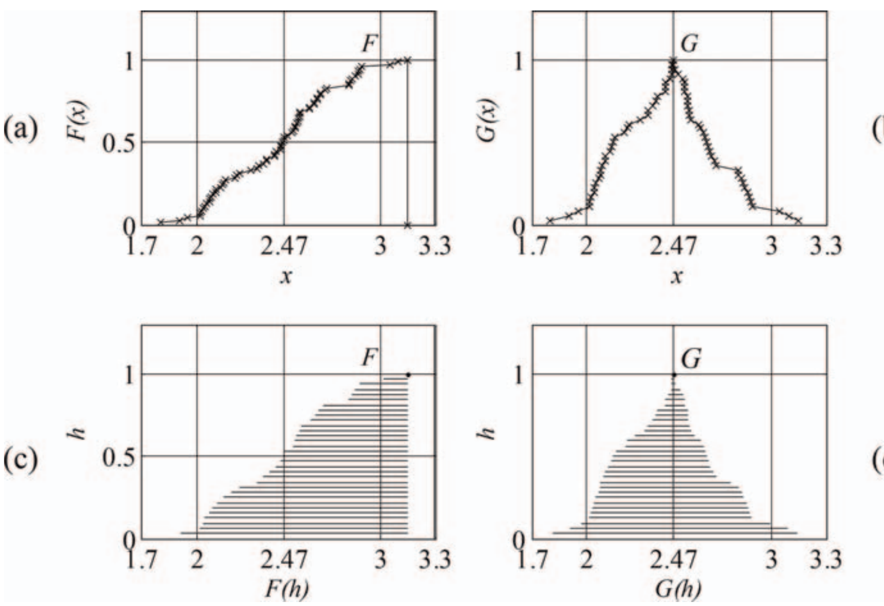


**Figure 1.**
*(a) A probability distribution IN* F *membership-function-representation with horizontal* x*-axis domain R; samples are indicated by an "x" mark. (b) A fuzzy number (possibility distribution) IN* G *membership-function-representation with horizontal* x*-axis domain R; samples are indicated by an "x" mark. (c) The corresponding probability distribution* F *interval-representation with* L = *32 levels and vertical axis domain [0,1]. (d) The corresponding possibility distribution (fuzzy number)* G *interval-representation with* L = *32 levels and vertical axis domain [0,1].*

because all the right interval ends coincide. In conclusion, the IN $F$ in **Figure 1(c)** is represented by $L = 32$ numbers ordered increasingly.

The IN $F$ in **Figure 1(a)** corresponds to a Cumulative Distribution Function (CDF) induced from a population of numerical data samples indicated with an "x" mark; it is $F(2.47) = 0.5$ [20]. Actually, the IN $F$ in **Figure 1(a)** represents only part of a CDF since a CDF in non-decreasing on its domain, whereas function $F(x)$ drops to 0 as soon as $F(x)$ reaches its largest value of 1. Nevertheless, apparently, there is a bijective mapping between a proper subset of INs and CDFs. In the aforementioned sense, an IN represents a CDF. The corresponding interval-representation is shown in **Figure 1 (c)** with $L = 32$ levels.

The IN $G$ in **Figure 1(b)** represents a fuzzy number induced from the aforementioned population of numerical data samples indicated with an "x" mark. The membership function $G(x)$ equals $G(x) = 2F(x)$ for $1.80 \leq x \leq 2.47$ and $G(x) = 2(1\text{-}F(x))$ for $2.47 \leq x \leq 3.15$ [21]. The corresponding interval-representation is shown in **Figure 1(d)**.

The similarity cosine function $\rho(x)$ between two INs $E$ and $F$ is computed as

$$\rho(x) = \rho(E(\tau - x), F(\tau)) = \frac{< E(\tau - x), F(\tau) >}{\|E(\tau - x)\| \, \|F(\tau)\|} = \frac{< E(\tau - x), F(\tau) >}{\sqrt{< E(\tau - x), E(\tau - x) >} \sqrt{< F(\tau), F(\tau) >}}$$
$$= \frac{\int_0^1 [(a_1(h) - x)a_2(h) + (b_1(h) - x)b_2(h)]dh}{\sqrt{\int_0^1 \left[(a_1(h) - x)^2 + (b_1(h) - x)^2\right] dh} \sqrt{\int_0^1 \left[a_2^2(h) + b_2^2(h)\right] dh}}, \quad (3)$$

and it is demonstrated in **Figure 2**. In particular, function $\rho(x)$ takes on values in the range $[-1, 1]$. Note that, for $\tau = 0.381$, INs $G(x\text{-}\tau)$ and $F(x)$ are orthogonal to one another because $< T(x\text{-}0.381), F(x) > = 0$, that is, their inner product equals zero.

In practice, an IN may (approximately) be represented by an $L$-dimensional vector of real numbers ordered increasingly. An advantage of an IN is its potential to represent *all-order data statistics* to any degree of accuracy depending on the number $L$ of levels [12]; moreover, an IN represents an *information granule*. An interval $F_h \in \mathrm{I}_1$, $h \in [0,1]$ will be denoted either by $[a_h, b_h]$ or, equivalently, by $[a(h), b(h)]$, $h \in [0,1]$. The set of INs is lattice ordered according to $F \preccurlyeq G \Leftrightarrow F_h \subseteq G_h$, $h \in [0,1] \Leftrightarrow F(x) \leq G(x)$, $x \in \mathrm{R}$. The corresponding lattice of INs is denoted by $(\mathrm{F}_1, \preccurlyeq)$.

The interest here is in the lattice $(\mathrm{F}_1, \preccurlyeq)$ of INs, which is a sublattice of $(\mathrm{G}_1, \preccurlyeq)$. In particular, the set of *trivial* INs corresponds to the vector space R of real numbers. However, $\mathrm{F}_1$ is not a vector space because if $E \in \mathrm{F}_1$ then $(-E) \notin \mathrm{F}_1$. It turns out that $\mathrm{F}_1$ is a *convex cone* in $\mathrm{G}_1$ in the sense that if $x,y \in \mathrm{F}_1$, then $(\lambda x + \mu y) \in \mathrm{F}_1$, $\forall \lambda,\mu \geq 0$.

**Figure 3** illustrates the notion of the convex cone $\mathrm{I}_1$ of intervals in the vector space $V_\mathrm{I}$ of generalized intervals. In particular, the closed half-plane "$a \leq b$" is a convex cone. Points on the line "$a = b$" correspond to trivial intervals. An IN $F \in \mathrm{F}_1$ representation includes a set of points $F_h$, $h \in [0,1]$, and it is not demonstrated here.

The inner product in $\mathrm{G}_1$, given by Corollary 9, also applies in the convex cone $\mathrm{F}_1$ of INs. Furthermore, any Cauchy sequence in $\mathrm{F}_1$ converges in $\mathrm{F}_1$ as shown next.

**Theorem 11**: The convex cone $\mathrm{F}_1$ of Intervals' Numbers (INs) is complete.

The proof of Theorem 11 is shown in Appendix A.

In the lattice $(\mathrm{I}_1, \subseteq)$ of intervals, given a strictly increasing function $v$: $\mathrm{L} \rightarrow \bar{\mathrm{R}}$ as well as a strictly decreasing function $\theta$: $\mathrm{L} \rightarrow \mathrm{L}$, there follow a *metric distance* function $d_1$: $\mathrm{I}_1 \times \mathrm{I}_1 \rightarrow \mathrm{R}_0^+$ as well as two order measure functions σ: $\mathrm{L} \times \mathrm{L} \rightarrow [0,1]$, respectively, as

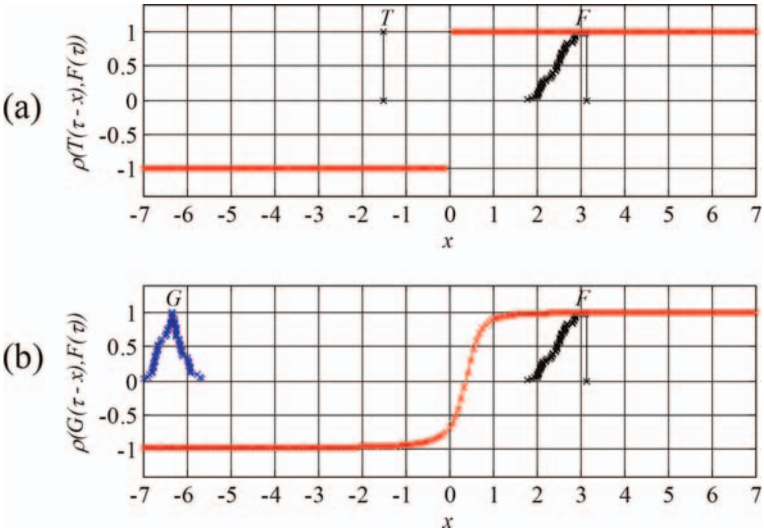


**Figure 2.**
*Similarity cosine function* ρ(x) *between the constant IN* F *and another IN moving along the* x*-axis. (a)* ρ(x) = ρ(T(τ-x), F(τ)), *where the "moving" IN* T(τ-x) *is trivial. (b)* ρ(x) = ρ(G(τ-x), F(τ)). *For* x = 0.381, *the "moving" IN* G(τ-x) *and* F(τ) *are orthogonal to one another because* <G(τ-0.381), F(τ)> = 0, *that is, their inner product equals zero.*

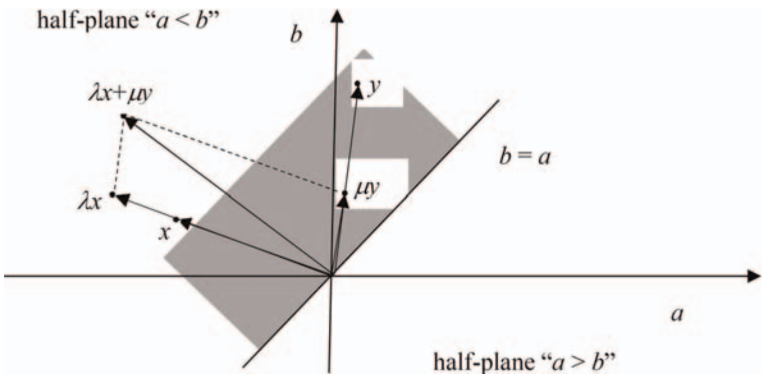


**Figure 3.**
*A point on the plane corresponds bijectively to a generalized interval in the set (lattice)* $V_I = (\overline{R} \times \overline{R}, \geq \times \leq)$. *Half-plane "*a ≤ b*" corresponds bijectively to the set I of intervals, whereas the whole half-plane "*a > b*" corresponds to the empty interval* ∅ = [i, o] *in* $I_1$ = I∪{∅}; *the latter is a convex cone in* $V_I$, *that is, if* x,y∈$I_1$, *then* (λx + μy)∈$I_1$, ∀λ,μ ≥ 0.

$$d_1([a_1,b_1],[a_2,b_2]) = [v(\theta(a_1 \wedge a_2)) - v(\theta(a_1 \vee a_2))] + [v(b_1 \vee b_2) - v(b_1 \wedge b_2)] \quad (4)$$

$$\sigma_{\sqcap}([a_1,b_1],[a_2,b_2]) = \begin{cases} 1, & \text{if } [a_1,b_1]=\varnothing \\ \dfrac{v(\theta(a_1 \vee a_2)) + v(b_1 \wedge b_2)}{v(\theta(a_1)) + v(b_1)}, & \text{otherwise} \end{cases} \quad (5)$$

$$\sigma_{\sqcup}([a_1,b_1],[a_2,b_2]) = \begin{cases} 1, & \text{if } [a_1,b_1]=\varnothing=[a_2,b_2] \\ \dfrac{v(\theta(a_2)) + v(b_2)}{v(\theta(a_1 \wedge a_2)) + v(b_1 \vee b_2)}, & \text{otherwise} \end{cases} \quad (6)$$

Given INs $E$ and $F$, there follows a metric distance function $D_1$: $F_1 \times F_1 \rightarrow R_0^+$ as well as two order measure functions σ: F × F → [0,1], respectively, as

$$D_1(E,F) = \int_0^1 d_1(E_h, F_h; v, \theta) dh \quad (7)$$

and

$$\sigma_\wedge(E,F) = \int_0^1 \sigma_\sqcap(E_h, F_h; v, \theta) dh, \quad (8)$$

$$\sigma_\curlyvee(E,F) = \int_0^1 \sigma_\sqcup(E_h, F_h; v, \theta) dh. \quad (9)$$

### 2.4 Non-linearities in $F_1$

Non-linearities can be introduced in $F_1$ by parametric functions $v(.)$ and $\theta(.)$ via metric functions as explained in the following.

A metric is introduced in a Hilbert space by the corresponding inner product. The previous section has detailed that the inner product of Corollary 9 is available in $F_1$. Moreover, $F_1$ is equipped with additional metric distance functions by Eq. (7). More metrics can be introduced in $F_1^N$ based on the following result [18]. Given the metric spaces $(X_1, d_1), \ldots, (X_N, d_N)$, a family of metric distance functions can be computed in the Cartesian product $X = X_1 \times \ldots \times X_N$ as follows:

$$\bar{d}_p(X,Y) = \left\{ \left(d_1(x_1,y_1)\right)^p + \ldots + \left(d_N(x_N,y_N)\right)^p \right\}^{1/P}, \quad (10)$$

between N-tuples $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_N)$, where $p \geq 1$; in particular, $p = 1$ corresponds to the *Manhattan distance*, $p = 2$ corresponds to the *Euclidean distance*, furthermore $\bar{d}_\infty(x,y) = \max\{d_1(x_1,y_1), \ldots, d_N(x_N,y_N)\}$.

Non-linear transformations can be introduced in the lattice $I_1$ of intervals by extending a strictly increasing (positive valuation) real function $v$: L → R to $v$: $I_1 \to I_1$ by defining $v([a_1,b_1]) = [v(a_1), v(b_1)]$. Since $a < b \Rightarrow v(a) < v(b)$, it follows that if $[a, b] \in I_1$ then $v([a,b]) \in I_1$. An extension $v$: $F_1 \to F_1$ follows by defining IN $G = v(E)$ such that $G_h = v(E_h)$, $\forall h \in [0,1]$ – note that, for $E \in F_1$, it follows $0 \leq h_1 \leq h_2 \leq 1 \Rightarrow E_{h_1} \supseteq E_{h_2} \Rightarrow v(E_{h_1}) \supseteq v(E_{h_2})$; hence, $v(E) \in F_1$ [22]. Lattice $(v(F_1), \preccurlyeq)$ is the *filtered image* of $(F_1, \preccurlyeq)$. A number of potentially useful mathematical results are presented next.

**Lemma 12**: Let $v(.)$ be a strictly increasing real function $v$: R → R. Then, lattices $(I_1, \subseteq)$ and $(v(I_1), \subseteq)$ are *order-isomorphic*, symbolically $(I_1, \subseteq) \approx (v(I_1), \subseteq)$, in the sense that $\forall A,B \in I_1$, 1) $v(.)$ is bijective (that is, one-to-one and onto), 2) $v(A \wedge B) = v(A) \wedge v(B)$ and 3) $v(A \vee B) = v(A) \vee v(B)$.

The proof of Lemma 12 is shown in Appendix A.

Corollary 13 follows.

**Corollary 13**: Let $v(.)$ be a strictly increasing real function $v$: R → R. Then, lattices $(F_1, \preccurlyeq)$ and $(v(F_1), \preccurlyeq)$ are order-isomorphic, symbolically $(F_1, \preccurlyeq) \approx (v(F_1), \preccurlyeq)$, in the sense of Lemma 12.

**Lemma 14**: Let $v(.)$ be a strictly increasing (positive valuation) function $v$: R → R such that $v(-x) = -v(x)$, let $\theta(x) = -x$ and let $A,B \in (I_1, \subseteq)$. Then, (a) $d_1(A,B; v,\theta) = d_1(v(A), v(B); x,\theta)$; (b) $\sigma_\sqcap(A,B; v,\theta) = \sigma_\sqcap(v(A),v(B); x,\theta)$; (c) $\sigma_\sqcup(A,B; v,\theta) = \sigma_\sqcup(v(A),v(B); x,\theta)$.

The proof of Lemma 14 is shown in Appendix A.

**Theorem 15**: Let $v(.)$ be a strictly increasing (positive valuation) $v$: R → R such that $v(-x) = -v(x)$, let $\theta(x) = -x$ and let $E,F \in (F_1, \preccurlyeq)$. Then, (a) $D_1(E,F; v,\theta) = D_1(v(E),v(F); x,\theta)$; (b) $\sigma_\wedge(E,F; v,\theta) = \sigma_\wedge(v(E),v(F); x,\theta)$; (c) $\sigma_\curlyvee(E,F; v,\theta) = \sigma_\curlyvee(v(E),v(F); x,\theta)$.

The proof of Theorem 15 is shown in Appendix A.

Theorem 15(a) indicates that the isomorphic lattices $(F_1, \preccurlyeq)$ and $(v(F_1), \preccurlyeq)$ are also *isometric* in the sense the distance between any two elements in one lattice equals the distance between their bijective images in the other lattice.

Previous work [14] has proposed two "reasonable constraints (RC)" regarding a positive valuation function $v(.)$ in a complete lattice $(L, \sqsubseteq)$ with minimum- and maximum-elements $o$ and $i$, respectively, namely, (RC1) $v(o) = 0$ and (RC2) $v(i) < +\infty$, which correspond to axiom (A0) in the definition of an order measure. In particular, (RC1) implies $\sigma_{\sqcap}(x, o) = 0 = \sigma_{\sqcup}(x, o)$, whereas (RC2) implies $d(x,i) < +\infty$, $\forall x \in L$. Theorem 15 implicitly introduces (RC3) $v(-x) = -v(x) \Rightarrow v(0) = 0$. In applications, constraints (RC1) and (RC2) are often useful as a neuron's activation function, whereas (RC3) is useful as a link's weight function.

Theorem 15 substantiates that lattice $(F_1, \preccurlyeq)$, equipped with $v(x) = -v(-x)$ and $\theta(x) = -x$, preserves the distances $D_1(.,.)$ as well as the order measures values $\sigma_{\curlywedge}(.,.)$ and $\sigma_{\curlyvee}(.,.)$ between bijective elements in lattices $(F_1, \preccurlyeq)$ and $(v(F_1), \preccurlyeq)$. Hence, Theorem 15 guarantees that the shapes of the distributions $F$ and $E$ may be kept intact, thus retaining the corresponding "probabilistic" and/or "possibilistic" interpretations, nevertheless the distance as well as the order measure between the distributions $E$ and $F$ may be tuned non-linearly toward optimizing decision-making.

**Theorem 16**: Let $v(.)$ be a strictly increasing (positive valuation) real function $v$: $R \to R$ such that $v(0) \neq 0$. Let $v_0(x) = v(x) - v(0)$. If (either $v(x)$ or $v_0(x)$) and $\theta(x) = -x$ is used, then $D_1(E,F; v,\theta) = D_1(E,F; v_0,\theta)$.

The proof of Theorem 16 is shown in Appendix A.

Theorem 16 implies that lattices 1) $(F_1, \preccurlyeq)$ with a strictly increasing positive valuation function $v(x)$ and $\theta(x) = -x$ and 2) $(F_1, \preccurlyeq)$ with a strictly increasing positive valuation function $v(x)$-$v(0)$ and $\theta(x) = -x$ are isometric in the sense that the metric distance between corresponding pairs of elements remains constant. Theorem 16 confirms the aforementioned constraint (RC3), that is, $v(-x) = -v(x) \Rightarrow v(0) = 0$.

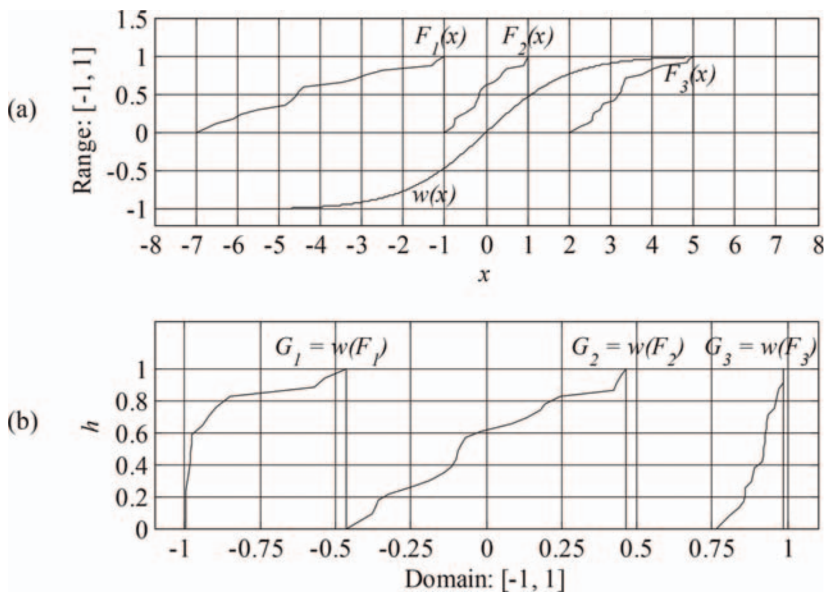


**Figure 4.**
*(a) The logistic function* $w(x) = (1 - e^{-x})/(1 + e^{-x})$ *with range* $[-1, 1]$*, and three INs* $F_1$, $F_2$ *and* $F_3$*. (b) Effect of the strictly increasing link function* $w(x)$*. The domain* $[-1, 1]$ *INs* $G_1 = w(F_1)$, $G_2 = w(F_2)$ *and* $G_3 = w(F_3)$ *equals the range of* $w(.)$*.*

**Figure 4** demonstrates nonlinear transformation of INs by a strictly increasing link function $w(.)$. In particular, **Figure 4(a)** displays the logistic function $w(x) = (1 - e^{-x})/(1 + e^{-x})$ as well as three INs $F_1$, $F_2$, and $F_3$. **Figure 4(b)** shows how the three INs $F_1$, $F_2$, and $F_3$ of **Figure 4(a)** are filtered by the logistic function $w(x)$ to result in the INs $G_1 = w(F_1)$, $G_2 = w(F_2)$, and $G_3 = w(F_3)$, respectively.

## 3. An enhanced deep learning architecture

### 3.1 Conventional deep learning in $R^N$

Feedforward deep learning (neural computing) architectures, during their training phase, operate successively both in a forward mode and in a backward mode; whereas, during their testing phase, they operate exclusively in the forward mode. The input to a neuron is computed by the "dot product" $\bar{w}\bullet\bar{x} = (w_1, \ldots, w_i, \ldots, w_N)\bullet(x_1, \ldots, x_i, \ldots, x_N) = w_1x_1 + \ldots + w_ix_i + \ldots + w_Nx_N$, where $\bar{x}$ is a neuron's input vector and $\bar{w}$ is the corresponding weight vector. In terms of a Hilbert space, $\bar{w}\bullet\bar{x}$ is an inner product. In particular, first, in the forward mode, deep learning (neural computing) models compute the "dot product" of vectors of real numbers, furthermore the numerical outcome of a dot (inner) product is transformed non-linearly by a neuron's activation function. Second, in the backward mode, the models optimize the values of weights between neurons by derivatives of the output error. Pooling is practiced between layers toward reducing the spatial dimensions (width/height) of feature maps while preserving important information.

Objects of interest in pattern recognition are represented by vectors of numbers such that similar objects are "forced," during the training phase, to be located nearby; furthermore, the similarity of objects of interest is often quantified by the "similarity cosine" between their corresponding vector representations. Recall that a similarity cosine is always available by the inner product in a Hilbert space.

A criterion for training convergence is to keep reducing successive output vector error to arbitrarily small values. The aforementioned criterion corresponds to a Cauchy sequence whose convergence is guaranteed in a Hilbert space.

All the above data processing capacities of neural computing exist in a Hilbert space. The basic idea of this work is to introduce an enhanced Hilbert space that includes the conventional Euclidean (Hilbert) space; then, develop likewise neural computing techniques as illustrated next.

### 3.2 Deep learning enhanced

An inner product exists in the Hilbert space $G_1$; hence, it is available in its convex cone $F_1$ of INs. The aforementioned "inner product" is a hint toward extending deep learning from $R^N$ to alternative Hilbert spaces such as $G_1^N$ including $F_1^N$. However, a straightforward extension is not possible as explained next.

The set R of real numbers is both a (mathematical) field and a real vector space. Conventional deep learning models treat the set R indiscriminately both as a (mathematical) field and as a real vector space. In this work, those roles are separated regarding the vector space $G_1$ of generalized intervals.

One way to enable the aforementioned separation of roles is to interpret a link weight $w_i \in R$, regarding a conventional deep learning neuron, as a real linear (link weight) function, namely $w_i(x) = w_ix$. Then, for an input $x_i \in R$, the corresponding (filtered) link output is computed as the product $w_ix_i$. According to Section 2.4, for $w_i \geq 0$ the input $x_i$

could be a non-trivial IN; furthermore, function $w_i(x)$ could be strictly increasing. In all aforementioned cases, the output $w_i(x)$ is an IN. However, a technical problem arises, for $w_i < 0$ as well as when $w_i(x)$ is a strictly decreasing function, for a non-trivial IN input $x_i$; then, $w_i(x)$ is not an IN. The following heuristic solution has been developed.

For $w_i < 0$, filtering a $L$-dimentional vector (which represents an IN, say IN $E$), results in a $L$-dimentional vector $w_i(E)$ whose entries are ordered decreasingly. Therefore, the output vector $r(w(E))$ was computed; where given a vector $\mathbf{x} = [x_1, \dots, x_N]$, the entries of vector $\mathbf{y} = [y_1, \dots, y_N] = r(\mathbf{x})$ are in reverse order i.e. $y_i = x_{N+1-i}$, $i \in \{1, \dots, N\}$ [15]. The abovementioned heuristic is compatible with the multiplication of real numbers in the sense that the result is the same. Furthermore, a linear weight function can be replaced by a strictly monotone (either increasing or decreasing) parametric function and handle INs likewise. In such a manner, link weight filtering can change not only the location of an IN but also its shape. **Figure 5** demonstrates the effect of filtering three non-trivial INs $F_1$, $F_2$, and $F_3$ by a strictly decreasing link weight function which satisfies constraint (RC3). The code (software) that produced the graphs in **Figure 5** is shown in Appendix B.

As $F_1$ is a convex cone in the Hilbert space $G_1$, an inner product is available for computing the similarity between N-dimensional vectors of INs. Furthermore, a neuron IN input (i.e., sum of INs) can be transformed non-linearly by a neuron's activation function.

Operations such as convolution and/or pooling can be extended to the proposed deep learning enhancement. In particular, pooling can be implemented by calculating probabilistic INs (**Figure 1**) representing all-order data statistics of big data to an arbitrary degree of accuracy using an appropriate number of $L$ levels to represent an IN in its interval representation. In particular, the induction of an IN from recorded data corresponds to a "first level" down-sampling, that is, pooling; further pooling on INs can be pursued for "higher level" down-sampling.

In the context of the work in [15], the IN Neural Network (INNN) architecture shown in **Figure 6** was used – note that the alternative term "Type-1 Neural Network

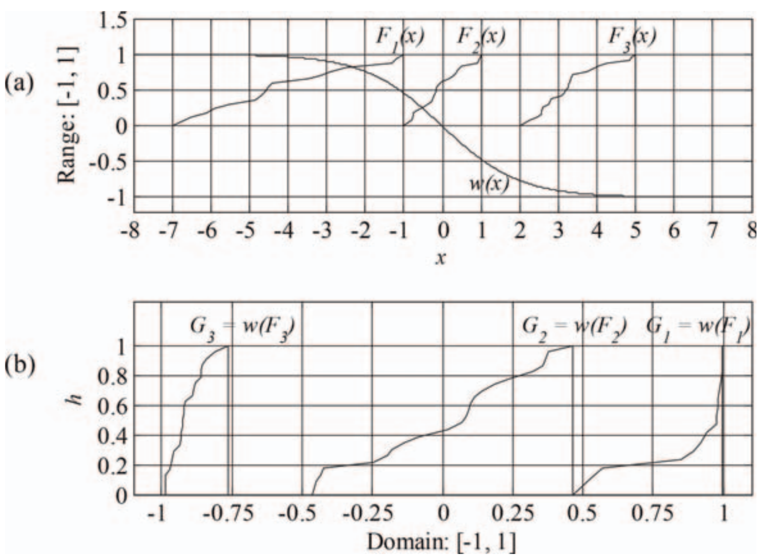


**Figure 5.**
*(a) The logistic function* $w(x) = (1 - e^x)/(1 + e^x)$ *with range* $[-1, 1]$*, and three INs* $F_1$, $F_2$ *and* $F_3$*. (b) Effect of the strictly decreasing link function* $w(x)$*. The domain* $[-1, 1]$ *INs* $G_1 = w(F_1)$, $G_2 = w(F_2)$ *and* $G_3 = w(F_3)$ *equals the range of* $w(.)$*.*

(T1NN)" may be used instead of INNN; however, a conventional deep learning neural network is called "Type-0 Neural Network (T0NN)" because it processes only vectors of numbers in $R^N$.

The *sigmoid* function $\varphi$: $R \rightarrow R_0^+$ given by

$$\varphi(x; A_\varphi, \lambda_\varphi, \mu_\varphi) = \frac{A_\varphi}{1 + e^{-\lambda_\varphi(x - \mu_\varphi)}} \tag{11}$$

is typically used as a neuron activation function; however, the sigmoid function offset by "-A/2," namely, the *logistic function*, is typically used as a link weight i.e.

$$w(x; A_w, \lambda_w, \mu_w) = \frac{A_w}{1 + e^{-\lambda_w(x - \mu_w)}} - \frac{A_w}{2} = \frac{A_w}{2} \frac{1 - e^{-\lambda_w(x - \mu_w)}}{1 + e^{-\lambda_w(x - \mu_w)}} \tag{12}$$

Parameter $\lambda_w$ could be either positive or negative resulting in a strictly increasing or a strictly decreasing function.

An input IN $s_j$ to a neuron $n_j$ activation function is computed (**Figure** 7) as

$$s_j = \left[ \sum_{i=1}^{N} w_{ji}(O_{pji}) \right] + b_j, \tag{13}$$

where $w_{ji}(.)$, $i \in \{1, \ldots, N\}$ is an incoming link's (from the previous layer of neurons) weight function, $O_{pji}$, $i \in \{1, \ldots, N\}$ is the corresponding output IN of a neuron from the previous layer regarding pattern $p$, and $b_j$ is the bias IN of neuron $n_j$. For input vectors in $R^N$ the INNN's (T1NN's) operation reduces to the operation of a conventional deep learning neural network T0NN.

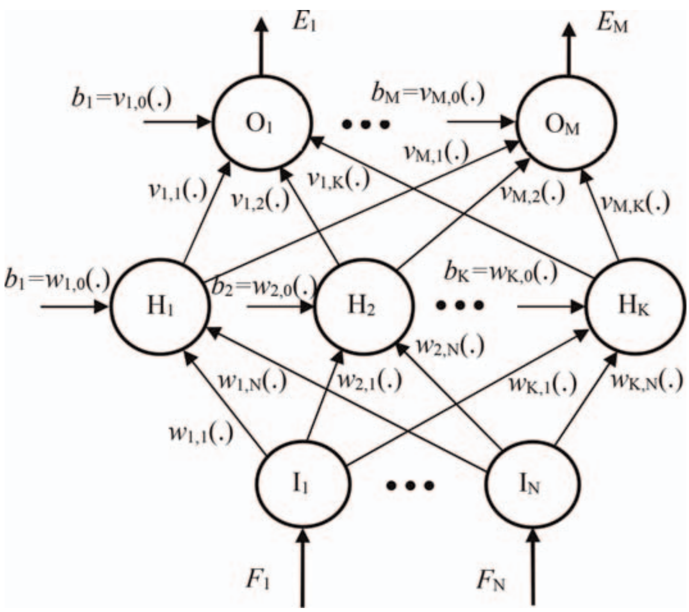


**Figure 6.**
*A three-layer IN neural network (INNN) or, alternatively, Type-1 neural network (T1NN), architecture whose inputs* $F_1, \ldots, F_N$ *are INs. For input vectors in* $R^N$ *the INNN's (T1NN's) operation reduces to the operation of a conventional deep learning neural network T0NN.*

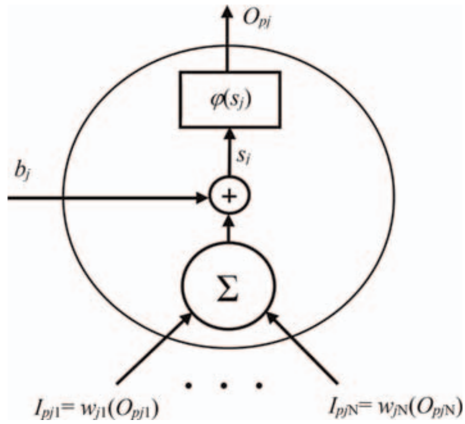


**Figure 7.**
*A single neuron model "j" of the INNN (T1NN) architecture, with N input INs* $I_{pj1}, \ldots, I_{pjN}$ *and 1 output IN* $O_{pj}$ *for an input pattern* p *– the latter is an N-dimensional IN. The activation function* $\varphi(.)$ *as well as all link weights,* $w_{ji}(.)$, i *{1, … ,N}, are strictly monotone functions, which filter their input INs* $s_j$ *and* $O_{pji}$, i *{1, … ,N}, respectively.*

## 4. Potential applications

This section presents two application domains where the proposed T1NN is expected to be especially useful.

### 4.1 Convolutional neural networks (CNNs)

Convolutional Neural Networks (CNNs) are inspired by biological processes. Specifically, certain neurons, consisting of simple and complex cells, respond differently in specific visual fields. Those insights from biology have paved the way for computational approaches to mimic aspects of visual perception based on these principles [2]. In particular, the LeNet-5 was designed to recognize handwritten postal codes and its architecture, including a backpropagation approach with conventional layers, pooling (subsampling) layers and dense (fully connected layers) has created a framework that is still applied by CNNs. A significant improvement was introduced by the AlexNet model [23], which outperforms all previous approaches by reducing the classification error by over 10%. AlexNet's success relies on several factors such as deeper multi-layered structure, the use of Rectified Linear Unit (ReLU) activation, the use of GPU to handle large-scale data, the use of max pooling layers, and the use of dropout layers.

Lately, hybrid models like the Interval Neural Networks have been introduced toward improving decision making under uncertainty in vision tasks [24, 25]. Interval Neural Networks are "conceptually" the nearest deep learning schemes to IN Neural Networks (INNNs) as demonstrated in [15], where an INNN was used in cascade with a YOLO deep learning architecture. Extensions of CNNs to INNNs/T1NNs are planned at our RIS-Lab especially regarding robot vision applications.

### 4.2 Large language models (LLMs)

A project is currently under development at IHU's RIS-Lab at the intersection of robotics and AI involving the social robot NAO (**Figure 8**) toward assisting children with special needs, e.g. children in the autism spectrum, as well as elderly with dementia. The team is supported by psychologists, sociologists and educators.

**Figure 8.**
*Social robot NAO integrated with AI at IHU's RIS-Lab.*

AI is integrated with the NAO robot to make it "intelligent" and capable of voice and image recognition in order to carry out conversations, to recognize faces, gestures, colors, even emotions and to adapt to a patient. The AI integration is pursued using a pre-trained Large Language Model (LLM) based on the Transformer architecture [3], which is accessed via an Application Programming Interface (API). The interaction begins when the NAO robot receives an audio/visual input from a human, through its microphone/camera. The raw data is, first, pre-processed by a local computer, which acts as an interface between NAO and the LLM, to extract meaningful content toward transforming audio/visual signals to text. Then, it sends a request to the AI via its API. Hence, it is routed to a remote server hosting the LLM, where the input is analyzed using deep learning models. In conclusion, it generates a structured output in a text format, returns it to the local computer and, finally, converts it via NAO's text-to-signal engine toward communicating with a human.

A number of improvements are planned for future work, including a tunable parameterization by T1NN also toward filtering the LLM received answers by a user-defined rule base also to comply with ethics.

## 5. Conclusion

High technology pioneers are successfully developing algorithms based on T0NN models. This work has introduced a straightforward enhancement of T0NN deep learning models to T1NN models. The critical difference is that T0NN models are developed in the Euclidean (Hilbert) space $R^N$, whereas T1NN models are developed in the Hilbert space $G_1^N$ of GINs including the convex cone $F_1^N$ of (Type-1) INs, where $F_1^N \supset R^N$. An IN is interpreted either as a real number or as an information granule, the latter may be either a probability distribution or a fuzzy number.

It is interesting to point out that other authors [26] have proposed a "fuzzy number algebra" in a Banach space – Recall that a Hilbert space is a Banach space but not always vice versa. The interest in Hilbert spaces here was motivated by the fact that a Hilbert space is a direct abstraction of the successful in modeling Euclidean space $R^N$. A likewise success is expected by a Hilbert space which may also include semantics.

The critical difference between classical interval arithmetic and the arithmetic of generalized intervals regards the definition of the product of an interval by a scalar. In particular, the classical interval arithmetic defines the product as $c[a, b] = [ca, cb]$ if $c \geq 0$, and $c[a, b] = [cb, ca]$ if $c < 0$. In contrast, the arithmetic of generalized intervals defines the product as: $c[a, b] = [ca, cb]$ $\forall c \in$R. The important consequence is the introduction of an inner product and, ultimately, the introduction of a hierarchy of Hilbert spaces including the convex cone $F_1$ of INs.

Since $F_1^N \supset R^N$ it is reasonable to expect that the proposed deep learning (T1NN) architectures can achieve at least as much as the conventional T0NN deep learning architectures. Additional advantages of neural computing in $F_1^N$, which summarize the enhancement of T0NN, are enumerated next.

First, partial- (lattice-) order may represent semantics. In the latter sense, the proposed T1NN architectures compute with semantics.

Second, decision-making can be pursued, even at neuron level, by axiomatic logic, for example, Fuzzy Lattice Reasoning (FLR) toward involving logic all along during data/information processing.

Third, significant energy savings may be pursued as conjectured next.

Deep learning, that is a workload that currently accounts for 14% of global Data Center (DC) power, is projected to rise to 27% by 2027 [27].

The good performance demonstrated by a number of Computational Intelligence models, such as "deep learning" models as well as "type-2 fuzzy systems" models in function-approximation problems, has been attributed to their large number of tunable parameters [16]. In particular, the number of tunable parameters of deep learning models has been reported in the order of hundreds of billions [28]. Furthermore, the impressive performance of deep learning models is largely attributed to the capacity of modern digital computer hardware to process vast data fast [15], not because the computation itself is fundamentally new or different.

On the one hand, a conventional T0NN model employs no more than a single parameter between two of its neurons; the aforementioned parameter corresponds to the weight of the link that connects two neurons. On the other hand, a T1NN model uses a parametric, monotone function as a link weight between neurons. Hence, a tunable number of tunable parameters can be introduced with a smaller number of neurons resulting in a potentially smaller deep learning architecture of greater flexibility. Based on the previous, we conjecture that a reduction of the energy required by a DC might be possible, without reducing performance.

Section 2 has developed mathematical tools that satisfy two prerequisites for neural computing, namely (p1) a Hilbert space $G_1$ and (p2) Non-linearities in $G_1$. Prerequisite (p3), that is, the existence of a derivative in $G_1$ toward defining a backpropagated "delta rule" for optimal parameter estimation is a topic for future research. Attention will also be given to the development of T1NNs with more layers as well as large scale comparative experiments by novel algorithms. In addition, the confirmation of the abovementioned conjecture i.e. a reduction of the energy required by a DC, will be tested. Future work extensions will also consider alternative lattices with nonnumerical data elements, represented by strings of 1 s and 0 s, in the context of the Lattice Computing (LC) paradigm, toward enhancing the representation of semantics in deep learning (neural computing) applications. An indispensable component of future work regards the inclusion of human ethics at all levels of data/information processing.

## Acknowledgements

This work was supported by IHU's Research Committee under project no. 81960 entitled "Artificial Intelligence and Applications".

## Thanks

The first author acknowledges the procedural mathematical discussions with Professor Athanasios Kehagias of Aristotle University of Thessaloniki. Of course, the first author assumes full responsibility for possible errors.

## Appendix A: This Appendix includes the proofs of new mathematical results

**Lemma 3**: Assume the Cartesian product $H^N$, where N is integer, of a pre-Hilbert (inner product) space $H$. Let the sum $\bar{x}+\bar{z}$, where $\bar{x} = (x_1, \ldots, x_N)$ and $+\bar{z} = (z_1, \ldots, z_N)$, be defined as $\bar{x}+\bar{z} = (x_1 + z_1, \ldots, x_N + z_N)$; moreover, let the scalar multiple $\kappa\bar{x}$ of $\bar{x}\in H^N$ by $\kappa\in$R be defined as $\kappa\bar{x} = (\kappa x_1, \ldots, \kappa x_N)$. Then, the (real) function $<.,.>: H^N \times H^N \to$ R defined as $<\bar{x}, \bar{y}> = <(x_1, \ldots, x_N), (y_1, \ldots, y_N)> = \sum_1^N \langle x_i, y_i \rangle$ is an inner product in $H^N$.

*Proof.*

The five conditions of Definition 2 are satisfied as shown next.

i. $<\bar{x}+\bar{z}, \bar{y}> = <(x_1, \ldots, x_N) + (z_1, \ldots, z_N), (y_1, \ldots y_N)> = <(x_1 + z_1, \ldots, x_N + z_N), (y_1, \ldots y_N)> = \sum_1^N \langle x_i + z_i, y_i \rangle = \sum_1^N (\langle x_i, y_i \rangle + \langle z_i, y_i \rangle) = \sum_1^N \langle x_i, y_i \rangle + \sum_1^N \langle z_i, y_i \rangle = <\bar{x}, \bar{y}> + <\bar{z}, \bar{y}>$.

ii. $< a\bar{x}, \bar{y}> = <(ax_1, \ldots, ax_N), (y_1, \ldots, y_N)> = \sum_1^N \langle ax_i, y_i \rangle = \sum_1^N a \langle x_i, y_i \rangle = a\sum_1^N \langle x_i, y_i \rangle = a<\bar{x}, \bar{y}>$.

iii. $<\bar{x}, \bar{y}> = <(x_1, \ldots, x_N), (y_1, \ldots, y_N)> = \sum_1^N \langle x_i, y_i \rangle = \sum_1^N \langle y_i, x_i \rangle = <\bar{y}, \bar{x}>$.

iv. $<\bar{x}, \bar{x}> = \sum_1^N \langle x_i, x_i \rangle \geq 0$.

v. $<\bar{x}, \bar{x}> = 0 \Leftrightarrow \sum_1^N \langle x_i, x_i \rangle = 0 \Leftrightarrow < x_i, x_i > = 0, i\in\{1, \ldots, N\} \Leftrightarrow x_i = 0, i\in\{1, \ldots, N\} \Leftrightarrow \bar{x} = \bar{0}$.

**Lemma 4**: Let $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$ be the lattice of *generalized intervals*, where R is the field of real numbers. Let the *addition* of two generalized intervals $[a_1, b_1]$ and $[a_2, b_2]$ be defined as $[a_1, b_1] + [a_2, b_2] = [a_1 + a_2, b_1 + b_2]$; moreover, let the *scalar multiple* of vector $[a,b]\in\bar{R} \times \bar{R}$ by the scalar $\lambda\in$R be defined as $\lambda[a,b] = [\lambda a, \lambda b]$. Then, $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$ is a *vector space* over R.

*Proof.*

Following Definition 1, (i) given an arbitrary pair $([a_1,b_1], [a_2,b_2])$ of *generalized intervals* in $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$, the unique element $[a_1,b_1] + [a_2,b_2] = [a_1 + a_2, b_1 + b_2]$ (called *sum* of $[a_1,b_1]$, $[a_2,b_2]$) is in $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$; (ii) given an arbitrary element $\kappa$ in R and an arbitrary element $[a,b]$ in $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$, the unique element $\kappa[a, b] = [\kappa a, \kappa b]$ (called *scalar multiple* of $[a,b]$ by $\kappa$) is in $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$.

The following eight conditions are satisfied:

i. $([a_1,b_1] + [a_2,b_2]) + [a_3,b_3] = [a_1 + a_2, b_1 + b_2] + [a_3, b_3] = [a_1 + a_2 + a_3, b_1 + b_2 + b_3] = [a_1,b_1] + [a_2 + a_3, b_2 + b_3] = [a_1,b_1] + ([a_2,b_2] + [a_3,b_3])$;

ii. There exists an element $[0,0]\in(\bar{R} \times \bar{R}, \geq \times \leq)$, called the *zero element* of $(\bar{R} \times \bar{R}, \geq \times \leq)$, such that $[a, b] + [0,0] = [0,0] + [a, b] = [a, b]$ for all $[a, b]\in(\bar{R} \times \bar{R}, \geq \times \leq)$;

iii. For any $[a, b]\in(\bar{R} \times \bar{R}, \geq \times \leq)$, there exists an element $-[a, b] = [-a, -b]\in(\bar{R} \times \bar{R}, \geq \times \leq)$ satisfying $[a, b] + [-a, -b] = [-a, -b] + [a, b] = [0,0]$;

iv. $[a_1,b_1] + [a_2,b_2] = [a_1 + a_2, b_1 + b_2] = [a_2 + a_1, b_2 + b_1] = [a_2,b_2] + [a_1,b_1]$;

v. $\kappa([a_1,b_1] + [a_2,b_2]) = \kappa[a_1 + a_2, b_1 + b_2] = [\kappa(a_1 + a_2), \kappa (b_1 + b_2)] = [\kappa a_1 + \kappa a_2, \kappa b_1 + \kappa b_2] = [\kappa a_1, \kappa b_1] + [\kappa a_2, \kappa b_2] = \kappa[a_1,b_1] + \kappa[a_2,b_2]$;

vi. For $\lambda\in R$, $(\kappa\lambda)[a_1,b_1] = [(\kappa\lambda)a_1, (\kappa\lambda)b_1] = [\kappa(\lambda a_1), \kappa(\lambda b_1)] = \kappa[\lambda a_1, \lambda b_1] = \kappa(\lambda[a_1, b_1])$;

vii. For $\lambda\in R$, $(\kappa + \lambda)[a, b] = [(\kappa + \lambda)a, (\kappa + \lambda)b] = [\kappa a + \lambda a, \kappa b + \lambda b] = [\kappa a, \kappa b] + [\lambda a, \lambda b] = \kappa[a, b] + \lambda[a, b]$;

viii. $1[a, b] = [1a, 1b] = [a, b]$ (where 1 is the unity element of R).

Therefore, the set $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$ is a linear space (or, vector space) over R.

**Lemma 5**: In vector space $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$ of generalized intervals, the mapping $<.,.>: V_I \times V_I \rightarrow R$ given by $<[a_1, b_1], [a_2, b_2]> = \frac{1}{2}(a_1a_2 + b_1b_2)$ is an *inner product*.

*Proof.*

Following Definition 2, let $\kappa\in R$. Furthermore, to any pair $([a_1,b_1], [a_2,b_2])$ of generalized intervals in $(\bar{R} \times \bar{R}, \geq \times \leq)$, the real number $< [a_1, b_1], [a_2, b_2] > = \frac{1}{2}(a_1a_2+ b_1b_2)$ satisfies the following five conditions:

i. $< [a_1,b_1] + [a_2,b_2], [a_3,b_3] > = < [a_1 + a_2, b_1 + b_2], [a_3,b_3] > = \frac{1}{2}((a_1 + a_2) a_3 + (b_1 + b_2)b_3) = \frac{1}{2}(a_1a_3 + b_1b_3) + \frac{1}{2}(a_2a_3 + b_2b_3) = <[a_1,b_1], [a_3,b_3]> + < [a_2, b_2], [a_3,b_3] >$.

ii. $< \kappa[a_1, b_1], [a_2, b_2] > = < [\kappa a_1, \kappa b_1], [a_2, b_2] > = \frac{1}{2}(\kappa a_1a_2 + \kappa b_1b_2) = \kappa\frac{1}{2}(a_1a_2 + b_1b_2) = \kappa < [a_1, b_1], [a_2, b_2]>$;

iii. $< [a_1, b_1], [a_2, b_2] > = \frac{1}{2}(a_1a_2 + b_1b_2) = \frac{1}{2}(a_2a_1 + b_2b_1) = <[a_2, b_2], [a_1, b_1]>$;

iv. $< [a, b], [a, b] > = \frac{1}{2}\left(a^2 + b^2\right) \geq 0$; and.

v. $< [a, b], [a, b] > = 0 \Leftrightarrow = \frac{1}{2}\left(a^2 + b^2\right) = 0 \Leftrightarrow a = 0 = b \Leftrightarrow [a, b] = [0, 0]$.

Therefore, $(\bar{R} \times \bar{R}, \geq \times \leq)$ is a pre-Hilbert space, and $< [a_1, b_1], [a_2, b_2] >$ is the inner product of $[a_1, b_1]$ and $[a_2, b_2]$.

**Lemma 6**: The inner product space $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$ of generalized intervals is *complete* with respect to the metric distance $\|[a_n, b_n] - [c_m, d_m]\|$.

*Proof.*

Recall that a sequence $x_1, x_2, x_3, \ldots$ of elements from $X$ of a metric space $(X, d)$ is called *Cauchy* if for every positive real number $\varepsilon > 0$, there is a positive integer $N$ such that for all integers $m, n > N$, it is $d(x_m, x_n) < \varepsilon$. A metric space $(X, d)$ is called *complete* if every Cauchy sequence in $X$ converges in $X$, that is it has a limit in $X$. Next, it is shown that every Cauchy sequence $[a_n, b_n]$ of $V_I$ elements has a limit in $V_I$.

Let $\|[a_n, b_n] - [a_m, b_m]\| \to 0$ $(m, n \to \infty)$, that is, $\|[a_n - a_m, b_n - b_m]\| \to 0$ $(m, n \to \infty)$, $\sqrt{\frac{1}{2}\left((a_n - a_m)^2 + (b_n - b_m)^2\right)} \to 0$ $(m, n \to \infty)$, $\left((a_n - a_m)^2 + (b_n - b_m)^2\right) \to 0$ $(m, n \to \infty)$, both $(a_n - a_m)^2 \to 0$ $(m,n \to \infty)$ and $(b_n - b_m)^2 \to 0$ $(m, n \to \infty)$. As it is known that every Cauchy sequence in the Hilbert space of real numbers R converges, with respect to the corresponding norm, to a real number, it follows that $\lim_{n\to\infty} [a_n, b_n] = [a, b] \in (\bar{R} \times \bar{R}, \geq \times \leq)$, where $a = \lim_{n\to\infty} a_n$ and $b = \lim_{n\to\infty} b_n$.

**Theorem 7**: Consider the space G of GINs and let R be the field of real numbers. Let (a) the *addition* of two GINs $E_1$ and $E_2$ be defined as $E_s(h) = E_1(h) + E_2(h), h\in[0,1]$, and (b) the *scalar multiplication* of a scalar $\lambda\in$R times a GIN $E$ be defined as $E_p(h) = \lambda E(h), h\in[0,1]$. Then, G is a *vector space* over R.

*Proof.*

Lemma 4 has proven that the complete lattice $V_I = (\bar{R} \times \bar{R}, \geq \times \leq)$ of generalized intervals is a vector space over R. Given INs $E$, $F$, and $G$ in G as well as $\kappa\in$R, the eight conditions of Definition 1 are satisfied as shown next.

i. $(E + F) + G = (E_h + F_h) + G_h, h\in[0,1] = E_h + (F_h + G_h), h\in[0,1] = E+(F + G)$.

ii. There exists an element $0 = [0,0]_h, h\in[0,1]$, called the *zero element* of G such that $E+0 = E_h + [0,0]_h, h\in[0,1] = [0,0]_h + E_h, h\in[0,1] = 0 + E = E$ for all $E\in$G;

iii. For any $E\in$G, there exists an element $X = (-E)\in$G, specifically $-E = -E_h$ $\forall h\in[0,1]$, satisfying $E + X = E_h + (-E_h), h\in[0,1] = (-E_h) + E_h$, $h\in[0,1] = X + E = [0,0]_h$ $\forall h\in[0,1] = 0$;

iv. $E + F = E_h + F_h, h\in[0,1] = F_h + E_h, h\in[0,1] = F + E$;

v. $\kappa(E + F) = \kappa(E_h + F_h), h\in[0,1] = \kappa E_h + \kappa F_h, h\in[0,1] = \kappa E + \kappa F$.

vi. For $\lambda\in$R, $(\kappa\lambda)E = (\kappa\lambda)E_h, h\in[0,1] = \kappa(\lambda E_h), h\in[0,1] = \kappa(\lambda E)$.

vii. For $\lambda\in$R, $(\kappa + \lambda)E = (\kappa + \lambda)E_h, h\in[0,1] = \kappa E_h + \lambda E_h, h\in[0,1] = \kappa E + \lambda E$.

viii. $1E = 1E_h, h\in[0,1] = E_h, h\in[0,1] = E$ (where 1 is the unity element of R).

**Theorem 10**: The inner product space $G_1$ of GINs is complete with respect to the metric distance $\|E_n - E_m\|$, where $E_i$, $i = 1,2,\ldots$ is a Cauchy sequence.

*Proof.*

This Theorem extends the completeness of Lemma 6 to the inner product space $G_1$, based on the fact that every Cauchy sequence $E_n$ in the Hilbert space of real numbers R converges, with respect to the corresponding norm, to a real number.

In particular, $\|E_n - E_m\| \to 0$ $(n, m \to \infty)$ implies $\sqrt{\langle E_n - E_m, E_n - E_m \rangle} = \sqrt{\int_0^1 \langle (E_n - E_m)_h, (E_n - E_m)_h \rangle dh} \to 0$ $(n,m \to \infty)$, that is, $\int_0^1 \langle (E_n)_h - (E_m)_h, (E_n)_h - (E_m)_h \rangle dh \to 0$ $(n, m \to \infty)$. Given 1. $<x\text{-}y, x\text{-}y> = <x, x> + <y, y> - 2<x, y>$; 2. $(E_n)_h = [a_n(h), b_n(h)]$; and 3. $(E_m)_h = [a_m(h), b_m(h)]$; it follows $\int_0^1 (\langle (E_n)_h, (E_n)_h \rangle + \langle (E_m)_h, (E_m)_h \rangle - 2\langle (E_n)_h, (E_m)_h \rangle) dh$ $(n, m \to \infty)$; hence, $\frac{1}{2}\int_0^1 (a_n(h) - a_m(h))^2 dh + \frac{1}{2}\int_0^1 (b_n(h) - b_m(h))^2 dh \to 0$ $(n, m \to \infty)$. The latter implies both $\int_0^1 (a_n(h) - a_m(h))^2 dh \to 0$ $(n, m \to \infty)$ and $\int_0^1 (b_n(h) - b_m(h))^2 dh \to 0$ $(n, m \to \infty)$. Therefore, both $\lim_{n,m\to\infty} |a_n(h) - a_m(h)|$ and $\lim_{n,m\to\infty} |b_n(h) - b_m(h)|$ are zero *almost everywhere* for $h \in [0,1]$. In the latter sense, $E_n \to E \in \mathrm{G}_1$.

**Theorem 11**: The convex cone $\mathrm{F}_1$ of Intervals' Numbers (INs) is complete.

*Proof.*

Theorem 10 implies that any Cauchy sequence $E_1, E_2, E_3, \ldots$ of INs converges in the inner product space $\mathrm{G}_1$ of GINs – for definition of a Cauchy sequence see in the Proof of Lemma 6. In the following, it is proven that $\lim_{i\to\infty} E_i$ is an IN by showing that $\lim_{i\to\infty} E_i$ satisfies the definition of an IN.

Let $E_i(h) = [a_i(h), b_i(h)]$, $i = 1,2,3, \ldots$ and $h \in [0,1]$. For a specific $h \in [0,1]$, Theorem 10 has shown that $a_i(h)$, $i = 1,2,3, \ldots$ is a Cauchy sequence that converges to, say, $a_h$; likewise, $b_i(h)$, $i = 1,2,3, \ldots$ is a Cauchy sequence that converges to, say, $b_h$.

a. It has to be $a_h \le b_h$; otherwise, there follows a contradiction as shown next.

   Assume $a_h > b_h$. As both $\lim_{i\to\infty} a_i(h) \to a_h$ and $\lim_{i\to\infty} b_i(h) \to b_h$, it follows that $\forall \varepsilon = (a_h - b_h)/\delta$, where $\delta > 2$, $\exists N$: $\forall i > N$, it holds both $|a_i(h) - a_h| < \varepsilon$ and $|b_i(h) - b_h| < \varepsilon$. Hence, $a_i(h) > b_i(h)$; in other words, $F_i$ is not an IN – contradiction. Therefore, the assumption $a_h > b_h$ is false. It logically follows $a_h \le b_h$.

b. Let $h_1 \le h_2$. It has to be $[a_{h_1}, b_{h_1}] \supseteq [a_{h_2}, b_{h_2}]$, otherwise there follows a contradiction as shown next.

   Assume $[a_{h_1}, b_{h_1}] \subset [a_{h_2}, b_{h_2}] \Leftrightarrow$ either $a_{h_2} < a_{h_1} \le b_{h_1} \le b_{h_2}$ or $a_{h_2} \le a_{h_1} \le b_{h_1} < b_{h_2}$. Given that $\lim_{i\to\infty} a_i(h_2) \to a_{h_2}$ and $\lim_{i\to\infty} a_i(h_1) \to a_{h_1}$ and $\lim_{i\to\infty} b_i(h_1) \to b_{h_1}$ and $\lim_{i\to\infty} b_i(h_2) \to b_{h_2}$, it follows $\forall \varepsilon = \max\{(a_{h_1} - a_{h_2})/\delta, (b_{h_1} - b_{h_2})/\delta\}$, where $\delta > 2$, $\exists N$: $\forall i > N$, it holds that $\max\{|a_i(h_2) - a_{h_2}|, |a_i(h_1) - a_{h_1}|, |b_i(h_1) - b_{h_1}|, |b_i(h_2) - b_{h_2}|\} < \varepsilon$. Hence, $[a_i(h_1), b_i(h_1)] \subset [a_i(h_2), b_i(h_2)]$; in other words, $F_i$ is not an IN – contradiction. Therefore, the assumption $[a_{h_1}, b_{h_1}] \subset [a_{h_2}, b_{h_2}]$ is false. It follows $[a_{h_1}, b_{h_1}] \supseteq [a_{h_2}, b_{h_2}]$.

c. $\forall S \subseteq [0,1]$: $\bigcap_{h\in S} [a(h), b(h)] = \wedge_{h\in S} [a(h), b(h)] = [\vee_{h\in S}\, a(h), \wedge_{h\in S}\, b(h)] = [a(\vee S), b(\vee S)]$.

In conclusion, the limit $E = \lim_{i\to\infty} E_i$ of any Cauchy sequence $E_1, E_2, E_3, \ldots$ of INs is an IN because (i) $a_h \le b_h$, (ii) $h_1 \le h_2 \Rightarrow [a_{h_1}, b_{h_1}] \supseteq [a_{h_2}, b_{h_2}]$ and (ii) $\forall S \subseteq [0,1]$: $\bigcap_{h\in S} [a_h, b_h] = [a_{\vee S}, b_{\vee S}]$. In other words, the convex cone $\mathrm{F}_1$ of Intervals' Numbers (INs) is complete.

**Lemma 12**: Let $v(.)$ be a strictly increasing real function $v$: R → R. Then, lattices $(\mathrm{I}_1, \subseteq)$ and $(v(\mathrm{I}_1), \subseteq)$ are *order-isomorphic*, symbolically $(\mathrm{I}_1, \subseteq) \approx (v(\mathrm{I}_1), \subseteq)$, in the sense that $\forall A,B \in \mathrm{I}_1$, 1) $v(.)$ is bijective (i.e. one-to-one and onto), 2) $v(A \wedge B) = v(A) \wedge v(B)$ and 3) $v(A \vee B) = v(A) \vee v(B)$.

*Proof.*

Let $A = [a_1,b_1]$ and $B = [a_2,b_2]$ be intervals in $\mathrm{I}_1$.

1. As function $v(.)$ is strictly increasing, it is bijective.

   Next, the following two cases are considered:

   a. $a_1 \le a_2 \Rightarrow v(a_1) \le v(a_2) \Rightarrow \begin{cases} v(a_1 \vee a_2) = v(a_2). & v(a_1 \wedge a_2) = v(a_1). \\ v(a_1) \vee v(a_2) = v(a_2). & v(a_1) \wedge v(a_2) = v(a_1). \end{cases}$.

   b. $a_1 > a_2 \Rightarrow v(a_1) > v(a_2) \Rightarrow \begin{cases} v(a_1 \vee a_2) = v(a_1). & v(a_1 \wedge a_2) = v(a_2). \\ v(a_1) \vee v(a_2) = v(a_1). & v(a_1) \wedge v(a_2) = v(a_2). \end{cases}$.

   In either aforementioned case, it is both $v(a_1 \vee a_2) = v(a_1) \vee v(a_2)$ and $v(a_1 \wedge a_2) = v(a_1) \wedge v(a_2)$. Hence,

2. $v(A \wedge B) = v([a_1,b_1] \wedge [a_2,b_2]) = v([a_1 \vee a_2, b_1 \wedge b_2]) = [v(a_1 \vee a_2), v(b_1 \wedge b_2)] = [v(a_1) \vee v(a_2), v(b_1) \wedge v(b_2)] = [v(a_1), v(b_1)] \wedge [v(a_2), v(b_2)] = v([a_1,b_1]) \wedge v([a_2,b_2]) = v(A) \wedge v(B)$.

3. $v(A \vee B) = v([a_1,b_1] \vee [a_2,b_2]) = v([a_1 \wedge a_2, b_1 \vee b_2]) = [v(a_1 \wedge a_2), v(b_1 \vee b_2)] = [v(a_1) \wedge v(a_2), v(b_1) \vee v(b_2)] = [v(a_1), v(b_1)] \vee [v(a_2), v(b_2)] = v([a_1,b_1]) \vee v([a_2,b_2]) = v(A) \vee v(B)$.

**Lemma 14**: Let $v(.)$ be a strictly increasing (positive valuation) function $v$: R → R such that $v(-x) = -v(x)$, let $\theta(x) = -x$ and let $A,B \in (\mathrm{I}_1, \subseteq)$. Then, (a) $d_1(A,B; v,\theta) = d_1(v(A), v(B); x,\theta)$; (b) $\sigma_\sqcap(A,B; v,\theta) = \sigma_\sqcap(v(A),v(B); x,\theta)$; (c) $\sigma_\sqcup(A,B; v,\theta) = \sigma_\sqcup(v(A),v(B); x,\theta)$.

*Proof.*

Let $A = [a_1,b_1]$ and $B = [a_2,b_2] \Rightarrow v(A) = [v(a_1), v(b_1)]$ and $v(B) = [v(a_2), v(b_2)]$.

a. It follows

   1. $d_1(A, B; v,\theta) = [v(\theta(a_1 \wedge a_2)) - v(\theta(a_1 \vee a_2))] + [v(b_1 \vee b_2) - v(b_1 \wedge b_2)] = [v(-a_1 \wedge a_2) - v(-a_1 \vee a_2)] + [v(b_1 \vee b_2) - v(b_1 \wedge b_2)] = [v(a_1 \vee a_2) - v(a_1 \wedge a_2)] + [v(b_1 \vee b_2) - v(b_1 \wedge b_2)] = |v(a_1) - v(a_2)| + |v(b_1) - v(b_2)|$, and

   2. $d_1(v(A), v(B); x,\theta) = d_1([v(a_1),v(b_1)], [v(a_2),v(b_2)]) = [\theta(v(a_1) \wedge v(a_2)) - \theta(v(a_1) \vee v(a_2))] + [v(b_1) \vee v(b_2) - v(b_1) \wedge v(b_2))] = [-v(a_1) \wedge v(a_2) + v(a_1) \vee v(a_2)] + [v(b_1) \vee v(b_2) - v(b_1) \wedge v(b_2)] = |v(a_1) - v(a_2)| + |v(b_1) - v(b_2)|$,

   Hence, $d_1(A,B; v,\theta) = d_1(v(A),v(B); x,\theta)$.

b. It follows

   1. $\sigma_\sqcap(A, B; v, \theta) = \frac{v(\theta(a_1 \vee a_2)) + v(b_1 \wedge b_2)}{v(\theta(a_1)) + v(b_1)} = \frac{v(-(a_1 \vee a_2)) + v(b_1 \wedge b_2)}{v(-a_1) + v(b_1)} = \frac{-v(a_1 \vee a_2) + v(b_1 \wedge b_2)}{-v(a_1) + v(b_1)}$,

$$2.\sigma_{\sqcap}(v(A),v(B);x,\theta)=\frac{\theta(v(a_1)\vee v(a_2))+v(b_1)\wedge v(b_2)}{\theta(v(a_1))+v(b_1)}=\frac{-v(a_1)\vee v(a_2)+v(b_1)\wedge v(b_2)}{-v(a_1)+v(b_1)}.$$

There are two cases, regarding the comparison of $v(a_1 \vee a_2)$ and $v(a_1)\vee v(a_2)$:

1.$a_1 \le a_2 \Rightarrow$ [ (i) $v(a_1 \vee a_2) = v(a_2)$ and (ii) $v(a_1) \le v(a_2) \Rightarrow v(a_1)\vee v(a_2) = v(a_2)$].

2.$a_1 > a_2 \Rightarrow$ [ (i) $v(a_1 \vee a_2) = v(a_1)$ and (ii) $v(a_1) > v(a_2) \Rightarrow v(a_1)\vee v(a_2) = v(a_1)$].

Hence, $v(a_1 \vee a_2) = v(a_1)\vee v(a_2)$.
There are two cases, regarding the comparison of $v(b_1 \wedge b_2)$ and $v(b_1)\wedge v(b_2)$:

1.$b_1 \le b_2 \Rightarrow$ [ (i) $v(b_1 \wedge b_2) = v(b_1)$ and (ii) $v(b_1) \le v(b_2) \Rightarrow v(b_1)\wedge v(b_2) = v(b_1)$].

2.$b_1 > b_2 \Rightarrow$ [ (i) $v(b_1 \wedge b_2) = v(b_2)$ and (ii) $v(b_1) > v(b_2) \Rightarrow v(b_1)\wedge v(b_2) = v(b_2)$].

Hence, $v(b_1 \wedge b_2) = v(b_1)\wedge v(b_2)$.
In conclusion, $\sigma_{\sqcap}(A,B;\, v,\theta) = \sigma_{\sqcap}(v(A),v(B);\, x,\theta)$.

c.It follows

$$1.\sigma_{\sqcup}(A,B;v,\theta)=\frac{v(\theta(a_2))+v(b_2)}{v(\theta(a_1\wedge a_2))+v(b_1\vee b_2)}=\frac{v(-a_2)+v(b_2)}{v(-(a_1\wedge a_2))+v(b_1\vee b_2)}=\frac{-v(a_2)+v(b_2)}{-v(a_1\wedge a_2)+v(b_1\vee b_2)},$$

$$2.\sigma_{\sqcup}(v(A),v(B);x,\theta)=\frac{\theta(v(a_2))+v(b_2)}{\theta(v(a_1)\wedge v(a_2))+v(b_1)\vee v(b_2)}=\frac{-v(a_2)+v(b_2)}{-v(a_1)\wedge v(a_2)+v(b_1)\vee v(b_2)}$$

There are two cases, regarding the comparison of $v(a_1 \wedge a_2)$ and $v(a_1)\wedge v(a_2)$:

1.$a_1 \le a_2 \Rightarrow$ [ (i) $v(a_1 \wedge a_2) = v(a_1)$ and (ii) $v(a_1) \le v(a_2) \Rightarrow v(a_1)\wedge v(a_2) = v(a_1)$].

2.$a_1 > a_2 \Rightarrow$ [ (i) $v(a_1 \wedge a_2) = v(a_2)$ and (ii) $v(a_1) > v(a_2) \Rightarrow v(a_1)\wedge v(a_2) = v(a_2)$].

Hence, $v(a_1 \wedge a_2) = v(a_1)\wedge v(a_2)$.
There are two cases, regarding the comparison of $v(b_1 \vee b_2)$ and $v(b_1)\vee v(b_2)$:

1.$b_1 \le b_2 \Rightarrow$ [ (i) $v(b_1 \vee b_2) = v(b_2)$ and (ii) $v(b_1) \le v(b_2) \Rightarrow v(b_1)\vee v(b_2) = v(b_2)$].

2.$b_1 > b_2 \Rightarrow$ [ (i) $v(b_1 \vee b_2) = v(b_1)$ and (ii) $v(b_1) > v(b_2) \Rightarrow v(b_1)\vee v(b_2) = v(b_1)$].

Hence, $v(b_1 \vee b_2) = v(b_1)\vee v(b_2)$.
In conclusion, $\sigma_{\sqcup}(A,B;\, v,\theta) = \sigma_{\sqcup}(v(A),v(B);\, x,\theta)$.

**Theorem 15**: Let $v(.)$ be a strictly increasing (positive valuation) $v$: R → R such that $v(-x) = -v(x)$, let $\theta(x) = -x$ and let $E,F \in (\mathrm{F}_1, \preccurlyeq)$. Then, (a) $D_1(E,F;\, v,\theta) = D_1(v(E),v(F);\, x,\theta)$; (b) $\sigma_{\curlywedge}(E,F;\, v,\theta) = \sigma_{\curlywedge}(v(E),v(F);\, x,\theta)$; (c) $\sigma_{\curlyvee}(E,F;\, v,\theta) = \sigma_{\curlyvee}(v(E),v(\mathrm{F});\, x,\theta)$.

*Proof.*

From Lemma 14, it follows.

(a) $D_1(E,\, F;\, v,\theta) = \int_0^1 d_1(E_h,F_h;v,\theta)dh = \int_0^1 d_1(v(E_h),v(F_h);x,\theta)dh = D_1(v(E), v(F);\, x,\theta)$.

(b) $\sigma_{\curlywedge}(E,\, F;\, v,\theta) = \int_0^1 \sigma_{\sqcap}(E_h,F_h;v,\theta)dh = \int_0^1 \sigma_{\sqcap}(v(E_h),v(F_h);x,\theta)dh = \sigma_{\curlywedge}(v(E), v(F);\, x,\theta)$.

(c) $\sigma_{\curlyvee}(E, F; v,\theta) = \int_0^1 \sigma_{\sqcup}(E_h, F_h; v, \theta)dh = \int_0^1 \sigma_{\sqcup}(v(E_h), v(F_h); x, \theta)dh = \sigma_{\curlyvee}(v(E), v(F); x,\theta)$.

**Theorem 16**: Let $v(.)$ be a strictly increasing (positive valuation) real function $v$: R → R such that $v(0) \neq 0$. Let $v_0(x) = v(x) - v(0)$. If (either $v(x)$ or $v_0(x)$) and $\theta(x) = -x$ is used then $D_1(E,F; v,\theta) = D_1(E,F; v_0,\theta)$.

*Proof.*

Let $A = [a_1,b_1]$ and $B = [a_2,b_2] \Rightarrow v(A) = [v(a_1), v(b_1)]$ and $v(B) = [v(a_2), v(b_2)]$. From the Proof of Lemma 14 it follows,

$d_1(A, B; v_0,\theta) = |v_0(a_1) - v_0(a_2)| + |v_0(b_1) - v_0(b_2)| = |v(a_1) - v(a_2)| + |v(b_1) - v(b_2)| = d_1(A, B; v,\theta)$. Hence,

$D_1(E, F; v,\theta) = \int_0^1 d_1(E_h, F_h; v, \theta)dh = \int_0^1 d_1(E_h, F_h; v_0, \theta)dh = D_1(E, F; v_0,\theta)$.

## Appendix B: This Appendix includes MATLAB software that produced Figure 5

```
x = −8:0.01:8;
subplot(2,1,1);
yd. = (2./(1 + exp.(+x)))-1;
pts. = [0 0.5 0.9 1.1 1.5 2.17 2.3 2.4 2.5 2.6 3.5 3.8 4.1 4.5 5.7 5.8 6]';
[pts1, val1] = fin(pts - 7); val1(1) = 0;
pts. = [0 0.6 0.7 0.75 1.01 1.53 1.87 2.17 2.3 2.4 2.45 2.5 2.6 2.9 3.5 3.8 4.1 4.2 4.5 5.7
5.8 5.9 6]';
[pts2, val2] = fin((pts/3)-1); val2(1) = 0;
pts. = [0 0.5 0.9 1.1 1.15 1.2 1.5 1.6 1.65 2.17 2.3 2.4 2.5 2.55 2.6 2.66 2.76 3.5 3.8 4.1
4.5 5.7 5.8 6]';
[pts3, val3] = fin((pts/2) + 2); val3(1) = 0;
plot(x,yd.,'k-',pts1,val1,'k-',pts2,val2,'k-',pts3,val3,'k-'); hold on;
plot([pts1(length(pts1)) pts1(length(pts1))],[0 1],'k-'); hold on;
plot([pts2(length(pts2)) pts2(length(pts2))],[0 1],'k-'); hold on;
plot([pts3(length(pts3)) pts3(length(pts3))],[0 1],'k-'); hold on;
grid.
axis([−8 8–1.2 1.5]);
set(gca,'XTick',-10:1:10);
set(gca,'YTick',-1:0.5:2);
text(0.7,-0.8,'\it{w(x)}');
text(−1.7,1.2,'\it{F}_1(x)');
text(0.4,1.2,'\it{F}_2(x)');
text(4.3,1.2,'\it{F}_3(x)');
xlabel('\it{x}');
ylabel('\it{Range}: [−1, 1]');
subplot(2,1,2);
y1 = (2./(1 + exp.(+pts1)))-1; y1 = sort(y1,'ascend');
y2 = (2./(1 + exp.(+pts2)))-1; y2 = sort(y2,'ascend');
y3 = (2./(1 + exp.(+pts3)))-1; y3 = sort(y3,'ascend');
plot(y1,val1,'k-',y2,val2,'k-',y3,val3,'k-'); hold on;
plot([y1(length(y1)) y1(length(y1))],[0 1],'k-'); hold on;
plot([y2(length(y2)) y2(length(y2))],[0 1],'k-'); hold on;
plot([y3(length(y3)) y3(length(y3))],[0 1],'k-'); hold on;
```

```
grid.
axis([−1.1 1.1 0 1.2]);
set(gca,'XTick',-1:0.25:1);
set(gca,'YTick',0:0.2:1);
text(−0.90, 1.1,'\it{G_3 = w(F_3)}');
text(0.30, 1.1,'\it{G_2 = w(F_2)}');
text(0.76, 1.1,'\it{G_1 = w(F_1)}');
xlabel('\it{Domain}: [−1, 1]');
ylabel('\it{h}');
________________________________________________________________________.
function [pts, val] = fin(x).
% [pts, val] = fin(x) computes an IN out of samples in vector 'x'.
% Column vector 'pts' returns the (IN's) domain points.
% Column vector 'val' returns the values on the aforementioned domain points.
epsilon = 0.001;
% no identical points in vector 'x' are allowed.
x = sort(x);
if min(abs(diff(x))) == 0, %condition for identifying identical numbers.
for i = 1:length(x),
j = i + 1;
while (j < = length(x))&(x(i) == x(j)),
x(j) = x(i) + epsilon;
j = j + 1;
end %while.
end %for.
end %if.
pts. = sort(x);
val = [];
Len = length(x);
for i = 1:Len,
val = [val; i/Len];
end
```

## Notation table

| | |
|---|---|
| $\bar{\mathrm{R}}$ | The set R of real numbers augmented by "-∞" and "+∞" |
| $(\mathrm{L}, \leq)$ | A complete mathematical lattice, where $\mathrm{L} \subseteq \bar{\mathrm{R}} = \mathrm{R}\cup\{-\infty,+\infty\}$ |
| $v$: $\mathrm{L}\rightarrow\mathrm{R}$ | A positive valuation (strictly increasing) function in $(\mathrm{L}, \sqsubseteq)$ |
| $\sigma$: $\mathrm{L} \times \mathrm{L}\rightarrow[0,1]$ | An order measure function |
| FLR | Fuzzy lattice reasoning |
| $\theta$: $\mathrm{L}\rightarrow\mathrm{L}$ | A dual (strictly decreasing) isomorphic function in $(\mathrm{L}, \sqsubseteq)$ |
| $<.,.>$: $H\times H\rightarrow\mathrm{R}$ | The inner product $<x, y>$ of $x$ and $y$ in vector space $H$ |
| $\|\|.\|\|$: $H\rightarrow\mathrm{R}$ | The norm $\|\|x\|\|$ of $x$ |
| $V_{\mathrm{I}}$ | The lattice $(\bar{\mathrm{R}} \times \bar{\mathrm{R}}, \geq\times\leq)$ of generalized intervals |
| $\rho$: $V\times V\rightarrow[-1,1]$ | A similarity cosine function in an inner product space $V$ |
| $E$: $[0,1]\rightarrow V_{\mathrm{I}}$ | A Generalized Intervals' Number (GIN) |
| G | The set of GINs |
| $L_2$ | The Lebesgue space of square-integrable functions |
| $\mathrm{G}_1$ | $\mathrm{G}_1 = \{[a(h), b(h)]: a,b\in L_2\}$. It is $\mathrm{G}_1 \subset \mathrm{G}$ |

| | |
|---|---|
| $(I_1, \subseteq)$ | The complete lattice of intervals in $(L, \leq)$ |
| $[i, o]$ | The empty interval in $(I_1, \subseteq)$ |
| $E$: $[0,1] \to I_1$ | A (type 1) Intervals' Number (IN) |
| $F_1$ | The set of INs. It is $F_1 \subset G_1$ |
| distrIN | An algorithm that induces an IN interpreted probabilistically |
| CALFIN | An algorithm that induces an IN interpreted possibilistically |
| CDF | Cumulative Distribution Function |
| $L$ | The number of intervals in an IN's interval representation |
| $\bar{d}_p$: $X \times X \to R_0^+$ | Metric distance in a Cartesian product $X = X_1 \times \ldots \times X_N$, $p > 1$ |
| $d_1$: $I_1 \times I_1 \to R_0^+$ | Metric distance in the complete lattice $(I_1, \subseteq)$ of intervals |
| $D_1$: $F_1 \times F_1 \to R_0^+$ | Metric distance function in the convex cone $F_1$ of INs |
| $\sigma_{\sqcap}$: $I_1 \times I_1 \to [0,1]$ | The sigma meet in the lattice $(I_1, \subseteq)$ of intervals |
| $\sigma_{\sqcup}$: $I_1 \times I_1 \to [0,1]$ | The sigma join in the lattice $(I_1, \subseteq)$ of intervals |
| $\sigma_{\wedge}$: $F_1 \times F_1 \to [0,1]$ | The sigma meet in the lattice $(F_1, \preccurlyeq)$ of INs |
| $\sigma_{\vee}$: $F_1 \times F_1 \to [0,1]$ | The sigma join in the lattice $(F_1, \preccurlyeq)$ of INs |

## Author details

Vassilis Kaburlasos[1]*, George Siavalas[1] and Lola Bris[1,2]

1 Department of Informatics, Computer and Telecommunications Engineering, School of Engineering, International Hellenic University (IHU), Serres, Greece

2 Department of Information Technology, CESI Engineering School, Toulouse Campus, France

*Address all correspondence to: vgkabs@ihu.gr

## References

[1] LeCun Y, Bengio Y, Hinton F. Deep learning. Nature. 2015;**521**:436-444

[2] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 2002;**86**(11): 2278-2324

[3] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L. Attention Is all you Need. 2017. Available from: https://arxiv.org/abs/1706.03762

[4] Newton I. The Mathematical Principles of Natural Philosophy (Philosophiae Naturalis Principia Mathematica). North Charleston, South Carolina, USA: Createspace Independent Publishing Platform; 2018

[5] Okeke, F. Top 7 AI Chip Makers in 2024 – Who Are They and Why Should You Care? Edmonton, Alberta, Canada: Techopedia; 2024. Available from: https://www.techopedia.com/top-ai-chipmakers

[6] Birkhoff G. Lattice Theory. Vol. 25. Providence, RI: American Mathematical Soc; 1967

[7] Kaburlasos VG. Granular enhancement of fuzzy-ART/SOM neural classifiers based on lattice theory. In: Kaburlasos VG, Ritter GX, editors. Computational Intelligence Based on Lattice Theory. Series: Studies in Computational Intelligence 67. Heidelberg, Germany: Springer; 2007. p. 3-22. DOI: 10.1007/978-3-540-72687-6_1

[8] Ritter GX, Urcid G. Introduction to Lattice Algebra with Applications in AI, Pattern Recognition, Image Analysis, and Biomimetic Neural Networks. Boca Raton, FL, USA: Chapman and Hall/CRC; 2021

[9] Kaburlasos VG. Lattice computing: A mathematical modelling paradigm for cyber-physical system applications. Mathematics. 2022;**10**(2):271. DOI: 10.3390/math10020271

[10] Papadakis SE, Kaburlasos VG. Piecewise-linear approximation of nonlinear models based on probabilistically/possibilistically interpreted intervals' numbers (INs). Information Sciences. 2010;**180**(24): 5060-5076

[11] Kaburlasos VG. Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory - Computational Intelligence and Soft Computing Applications. Heidelberg, Germany: Springer; 2006. DOI: 10.1007/978-3-540-34170-3

[12] Kaburlasos VG, Kehagias A. Fuzzy inference system (FIS) extensions based on the lattice theory. IEEE Transactions on Fuzzy Systems. 2014;**22**(3):531-546

[13] Kaburlasos VG, Papakostas GA. Learning distributions of image features by interactive fuzzy lattice reasoning in pattern recognition applications. IEEE Computational Intelligence Magazine. 2015;**10**(3):42-51

[14] Kaburlasos VG, Papakostas GA. Introduction to Computational Intelligence – A Holistic Approach. Athens, Greece: Hellenic Academic Ebooks; 2016. Available from: www.kallipos.gr. DOI: 10.57713/kallipos-661

[15] Kaburlasos V, Siavalas G. Computing with semantics in a Hilbert space: A proposal for enhancing deep-learning. In: Ramos F, Liu T, Samsonovich AV, editors. Proceedings of the 16th Annual Meeting of the Biologically Inspired Cognitive Architectures Society (BICA 2025); 6-10 August 2025; Puerto Vallarta, Mexico.

Cham, Switzerland: Springer Nature; 2025. ISSN: 1860-949X. (expected in December)

[16] Kaburlasos VG, Lytridis C, Vrochidou E, Bazinas C, Papakostas GA, Lekova A, et al. Granule-based-classifier (GbC): A lattice computing scheme applied on tree data structures. Mathematics. 2021;**9**(22):2889

[17] Kaburlasos VG, Petridis V. Fuzzy lattice neurocomputing (FLN) models. Neural Networks. 2000;**13**(10): 1145-1170

[18] Mathematical Society of Japan. Encyclopedic Dictionary of Mathematics (EDM2). 2nd ed. Cambridge, MA: The MIT Press; 1993

[19] Rudin W. Functional analysis. In: International Series in Pure and Applied Mathematics. 2nd ed. Hamburg, Germany: McGraw-Hill; 1991

[20] Kaburlasos VG, Vrochidou E, Panagiotopoulos F, Aitsidis C, Jaki A. Time Series Classification in Cyber-Physical System Applications by Intervals' Numbers Techniques. In: Proceedings of the IEEE Intl. Conf. On Fuzzy Systems (FUZZ-IEEE); 23–26 June 2019; New Orleans, LA. IEEE Xplore: Piscataway, NJ; 2019

[21] Kaburlasos VG, Kehagias A. Novel fuzzy inference system (FIS) analysis and design based on lattice theory. IEEE Transactions on Fuzzy Systems. 2007; **15**(2):243-260

[22] Kaburlasos VG, Papakostas GA, Pachidis T, Athinellis A. Intervals' Numbers (INs) Interpolation/ Extrapolation. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2013); 7–10 July 2013; Hyderabad, India. IEEE Xplore: Piscataway, NJ; 2013

[23] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Communications of the ACM. 2017;**60** (6):84-90. DOI:10.1145/3065386

[24] Betancourt D, Muhanna R. Interval Deep Learning for Uncertainty Quantification in Safety Applications. 2021. Available from: https://arxiv.org/abs/2105.06438

[25] Wang K, Shariatmadar K, Manchingal SK, Cuzzolin F, Moens D, Hallez H. Creinns: Credal-set interval neural networks for uncertainty estimation in classification tasks. Neural Networks. 2025;**185**:107198

[26] Esmi E, de Barros LC, Pedro FS, Laiate B. Banach spaces generated by strongly linearly independent fuzzy numbers. Fuzzy Sets and Systems. 2021; **417**:110-129. DOI: 10.1016/j.fss.2020.09.010

[27] Golden Sachs. 2025. Available from: https://www.goldmansachs.com/insights/articles/ai-to-drive-165-increase-in-data-center-power-demand-by-2030?utm_source=chatgpt.com

[28] Athlur S, Saran N, Sivathanu M, Ramjee R, Kwatra N. Varuna: Scalable, Low-Cost Training of Massive Deep Learning Models. 2021. Available from: https://arxiv.org/abs/2111.04007