

Towards machine learning as AGM-style belief change

Theofanis Aravanis 

Department of Mechanical Engineering, University of the Peloponnese, Patras, 263 34, Greece

ARTICLE INFO

Keywords:

Symbolic/sub-symbolic/hybrid artificial intelligence

Artificial Neural Networks

AGM belief change

Knowledge representation

ABSTRACT

Artificial Neural Networks (ANNs) are powerful computational models that are able to reproduce complex non-linear processes, and are being widely used in a plethora of contemporary disciplines. In this article, we study the statics and dynamics of a certain class of ANNs, called binary ANNs, from the perspective of belief-change theory. A binary ANN is a feed-forward ANN whose inputs and outputs take binary values, and as such, it is suitable for a wide range of practical applications. For this type of ANNs, we point out that their knowledge (expressed via their input-output relationship) can symbolically be represented in terms of a propositional logic language. Furthermore, in the realm of belief change, we identify the process of changing (revising/contracting) an initial belief set to a modified belief set, as a process of a gradual transition of intermediate belief sets — such a gradualist approach to belief change is more congruent with the behaviors of real-world agents. Along these lines, we provide natural metrics for measuring the distance between these intermediate belief sets, effectively quantifying the disparity in their encoded knowledge. Thereafter, we demonstrate that, similar to belief change, the training process of binary ANNs, through backpropagation, can be emulated via a sequence of successive transitions of belief sets, the distance between which is intuitively related through one of the aforementioned metrics. We also prove that the alluded successive transitions of belief sets can be modeled by means of rational revision and contraction operators, defined within the fundamental belief-change framework of Alchourrón, Gärdenfors and Makinson (AGM). Thus, the process of machine learning (specifically, training binary ANNs) is framed as an operation of AGM-style belief change, offering a modular and logically structured perspective on neural learning.

1. Introduction

Artificial Neural Networks (ANNs) have evolved as a cornerstone in the field of Machine Learning (ML), offering robust solutions to a plethora of problems across diverse domains, such as image recognition, natural language processing, and autonomous systems [27,10,47,61,52]. Originating from the ambition to mimic the neuronal structure and functionality of the human brain, ANNs have transcended their biological inspiration to become powerful computational models. While they are primarily recognized for their ability to model complex, non-linear relationships through deep-learning architectures, their capacity to encode, manipulate, and generate *symbolic knowledge* is an area of burgeoning interest and significant implications [45].

Symbolic knowledge, characterized by structured relationships between symbols that represent abstract concepts, has traditionally been the domain of *rule-based systems* [13,71]. However, the integration of symbolic reasoning with sub-symbolic processes, like

E-mail address: taravanis@uop.gr.

<https://doi.org/10.1016/j.ijar.2025.109437>

Received 24 September 2024; Received in revised form 30 March 2025; Accepted 31 March 2025

Available online 2 April 2025

0888-613X/© 2025 The Author. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

those found in ANNs, promises a new *hybrid, neuro-symbolic* paradigm that could leverage the strengths of both symbolic Artificial Intelligence (AI) and neural computation. This convergence is hypothesized to not only enhance the *interpretability* and *transparency* of ANNs, but also to enrich their learning capabilities, by embedding a priori knowledge and logical-reasoning frameworks into their architectures [21,66].

The investigation of how ANNs can represent and utilize symbolic knowledge—whether through embedding symbolic structures within the network layers or interfacing with external symbolic systems—has opened a novel research front. Various methodologies have been explored, including the injection of symbolic rules into ANNs and the extraction of symbolic representations from trained networks [69,2,70,26]. These approaches aim to create more interpretable models that maintain the adaptability and learning efficiency of traditional ANNs, while enhancing them with the ability to reason over learned representations in a human-understandable format. Furthermore, the fusion of symbolic and sub-symbolic AI could potentially address some of the inherent limitations of purely data-driven ANNs, such as their demand for extensive data, vulnerability to adversarial attacks, and difficulty in generalizing from limited samples. By integrating symbolic knowledge directly into the learning process, ANNs could achieve more robust generalizations and provide richer explanations of their decisions and behaviors, thus bridging the gap between neural computation and human-like reasoning [20].

Following a research path that aims at the development of a comprehensive neuro-symbolic AI paradigm, this article studies the *statics* and *dynamics* of a certain family of ANNs—which we shall call *binary* ANNs—from the perspective of *belief-change* theory, a field of study that deals with the process of changing (revising/contracting) beliefs in light of new evidence [24,55,23]. A binary ANN is a feed-forward ANN with all inputs and outputs consisting of *binary values*, and as such, it is suitable for a wide range of practical applications (including, indicatively, image processing and pattern recognition using datasets similar to the benchmark MNIST dataset [48]). Against this background, the following contributions are provided:

- We illustrate how the knowledge of binary ANNs (expressed via their input-output relationship) can *symbolically* be represented in terms of a *propositional logic language*; specifically, by means of a collection of logical theories, also referred to as *belief sets*.
- Furthermore, in the realm of belief change, we identify the process of changing an initial belief set to a modified belief set, as a process of a *gradual transition* of intermediate belief sets. Such a gradualist approach to belief change finds substantial support in research on human development [59,65,72], and is more congruent with the behaviors of real-world agents. Along these lines, we provide two intuitive Hamming-based metrics for measuring the *distance* between these intermediate belief sets, effectively quantifying the disparity in their encoded symbolic knowledge.
- Thereafter, we demonstrate that, similar to belief change, the *training* process of binary ANNs, through the fundamental *back-propagation* algorithm [62], can be emulated via a sequence of *successive transitions* of belief sets, the distance between which is naturally related through one of the aforementioned metrics.
- We also prove that the alluded successive transitions of belief sets can be modeled by means of *rational* revision and contraction operators that implement full-meet belief change [29], a type of change specified within the *AGM framework*, the fundamental belief-change paradigm of Alchourrón, Gärdenfors and Makinson [1,24]. In this way, the process of machine learning (specifically, training binary ANNs) is recast as an operation of AGM-style belief change, offering a modular and conceptually structured perspective on neural learning.

It is noteworthy that, although extensive research has been dedicated to exploring neuro-symbolic approaches to AI, efforts to integrate AGM-style belief change into ML systems, as is attempted herein, remain notably sparse. In fact, to the best of our knowledge, the closest works in the spirit of the present study are those conducted by Coste-Marquis and Marquis [16] and Schwind et al. [63]. In the former [16], the authors discuss the incorporation of symbolic background knowledge into ML-based classifier systems to enhance their accuracy and robustness. This incorporation is framed as a belief-change problem, focusing on adapting a Boolean circuit that mirrors a classifier's behavior to comply with a certain body of background knowledge. It is shown that conventional belief-change operations are inadequate for this nuanced task, prompting the introduction of a specialized operation, called *rectification*—a process that minimally modifies (“rectifies”) the classifier's Boolean circuit to ensure compliance with background knowledge while preserving its classification structure. The article methodically defines rectification operations and investigates their theoretical properties, such as compliance with principal postulates of belief change (including the AGM ones) and computational implications. Similarly, the work by Schwind et al. [63] explores editing Boolean classifiers from a belief-change perspective, directly connecting learning for binary classification and AGM-style revision. The study introduces rational ways of modifying Boolean classifiers when new pieces of evidence must be incorporated, delineating various rationality postulates inspired by belief-revision principles. This approach also underscores the need for specialized edit operations to ensure the modified classifiers' consistency and minimal change.

Moreover, the field of Inductive Logic Programming also takes a symbolic view of ML. Recent work by Morel and Cropper [50] investigates learning logic programs by explaining their failures, which can be viewed as a form of belief expansion—a related AGM-style form of belief change in a more complex setting. Evidently, the aforementioned studies, akin to the present work, incorporate AGM-based belief change into ML models, highlighting a growing interest in the intersection of symbolic belief change and ML.

The remainder of this article is organized as follows: The following section sets the required formal background for our subsequent discussion. Section 3 introduces basic concepts of the AGM framework, whereas, Section 4 presents the process of belief change as a gradual transition of beliefs. Section 5 points out how the input-output relationship of binary ANNs can symbolically be represented by a collection of belief sets. Section 6 briefly discusses backpropagation, the principal method for training feed-forward ANNs. Thereafter, Section 7 points out that the training of binary ANNs, through backpropagation, can be emulated via a sequence of successive transitions of belief sets, which are in turn interrelated through a natural measure of distance. Section 8 proves that the

learning process of binary ANNs is AGM-compatible, in the sense that it can be modeled by means of a special type of AGM-style change operators. The article concludes with a brief conclusion section, which summarizes the established contributions and reports promising avenues for future research.

2. Formal preliminaries

In this section, we set the formal background required for the forthcoming discussion.

Logic Language: In this study, we shall be working with a propositional language \mathbb{L} , built over *finitely many* propositional variables (atoms), using the standard Boolean connectives \wedge (conjunction), \vee (disjunction), \rightarrow (implication), \leftrightarrow (equivalence), \neg (negation), and governed by *classical propositional logic*. The finite, non-empty set of all propositional variables is denoted by \mathcal{P} . The classical inference relation is denoted by \models . The symbol \top denotes an arbitrary tautological sentence of \mathbb{L} .

Sentences and Belief Sets: For a set of sentences Γ of \mathbb{L} , $Cn(\Gamma)$ denotes the set of all logical consequences of Γ ; i.e., $Cn(\Gamma) = \{\varphi \in \mathbb{L} : \Gamma \models \varphi\}$. For sentences $\varphi_1, \dots, \varphi_n$ of \mathbb{L} , we shall write $Cn(\varphi_1, \dots, \varphi_n)$ as an abbreviation of $Cn(\{\varphi_1, \dots, \varphi_n\})$. An agent's set of beliefs will be modeled by a *theory*, also referred to as a *belief set*. A theory K of \mathbb{L} is a set of sentences of \mathbb{L} closed under logical consequence; that is, $K = Cn(K)$. As we shall subsequently introduce formal properties of revision and contraction functions, let us first define the simpler operation of *expansion*. Accordingly, for a theory K and a sentence φ of \mathbb{L} , the expansion of K by φ , denoted by $K + \varphi$, is defined as $K + \varphi = Cn(K \cup \{\varphi\})$.

Possible Worlds: A *literal* is a propositional variable $p \in \mathcal{P}$ or its complement (negation). For a finite set of literals Q , $|Q|$ denotes the cardinality of Q .¹ A *possible world* (abbrev. *world*) r is an inclusion-maximal consistent set of literals, such that, for any propositional variable $p \in \mathcal{P}$, either $p \in r$ or $\neg p \in r$.² For a propositional variable p and a world r , $p \in r$ means that p is assigned *true* in r , whereas, $p \notin r$ means that p is assigned *false* in r . The set of all possible worlds is denoted by \mathbb{M} . For a sentence or set of sentences φ of \mathbb{L} , $[\varphi]$ is the set of worlds at which φ is true. For the sake of readability, possible worlds will sometimes be represented as sequences (rather than sets) of literals, and the negation of a propositional variable p will be represented as \bar{p} , instead of $\neg p$.

Preorders: A *preorder* over a non-empty set M is any reflexive and transitive binary relation on M . A preorder \leq over M is called *total* iff any two elements of M are comparable with respect to \leq ; i.e., for all $r, r' \in M$, $r \leq r'$ or $r' \leq r$. The strict part of \leq is denoted by $<$; i.e., $r < r'$ iff $r \leq r'$ and $r' \not\leq r$. The indifference part of \leq is denoted by \approx ; i.e., $r \approx r'$ iff $r \leq r'$ and $r' \leq r$. The set of all \leq -minimal elements of M is denoted by $\min(M, \leq)$; namely,

$$\min(M, \leq) = \left\{ r \in M : \text{for all } r' \in M, \text{ if } r' \leq r, \text{ then } r \leq r' \right\}.$$

When M contains numbers, we simply write $\min(M)$ to denote the minimum number in M .

Boolean Functions: A (n -ary) *Boolean function* f is a function that maps every possible combination of n input binary variables to a single binary output (0 or 1); in symbols, $f : \{0, 1\}^n \mapsto \{0, 1\}$. An example of a (2-ary) Boolean function is a Boolean function f that implements the logical operation OR , according to which $f(0, 0) = 0$, $f(0, 1) = 1$, $f(1, 0) = 1$, and $f(1, 1) = 1$.

Artificial Neural Networks: A *feed-forward Artificial Neural Network* (ANN) is a computational model that can be formally specified through a directed acyclic graph $G = (V, E)$. In this graph, V represents the set of vertices (neurons/nodes), and E represents the set of directed edges (connections between neurons). The vertices in V are organized into distinct, ordered subsets called *layers*, denoted as V_0, V_1, \dots, V_L , where:

- V_0 is the *input layer* consisting of input nodes X_1, \dots, X_n .
- V_L is the *output layer* consisting of output nodes y_1, \dots, y_m .
- V_1, V_2, \dots, V_{L-1} are the *hidden layers* consisting of intermediate neurons.

A *layer* V_l , for $l = 0, 1, \dots, L$, is a set of neurons such that:

- For $l = 0$, the neurons in V_0 (input layer) receive the external inputs X_1, \dots, X_n .
- For $l = L$, the neurons in V_L (output layer) produce the outputs y_1, \dots, y_m .
- For $l = 1, 2, \dots, L - 1$, the neurons in V_l (hidden layers) receive inputs only from the neurons in V_{l-1} , and send their outputs only to the neurons in V_{l+1} .

The set of edges E consists of directed connections (u, v) , where $u \in V_{l-1}$ and $v \in V_l$, for $l = 1, 2, \dots, L$. This ensures that the graph is acyclic, and that connections only exist between neurons in adjacent layers, not within the same layer or skipping layers.

¹ When Q is a number, then $|Q|$ denotes the absolute value of Q .

² Possible worlds are often called *models* or *interpretations* as well.

Each neuron $v \in V \setminus V_0$ (i.e., all neurons except those in the input layer) computes a *weighted sum* of its inputs, adds a bias term, and then applies a non-linear *activation function* σ . Specifically, for a neuron $v \in V_l$ in layer l (where $l = 1, 2, \dots, L$), the output z_v is given by

$$z_v = \sigma \left(\sum_{u \in V_{l-1}} w_{uv} \cdot x_u + b_v \right).$$

Here, w_{uv} denotes the weight of the edge from neuron u in layer V_{l-1} to neuron v in layer V_l , x_u is the output of neuron u , and b_v is the bias of neuron v . Among the activation functions commonly used in ANNs are the sigmoid, the Rectified Linear Unit (ReLU), and the softmax function.

As we shall discuss in Section 6, training an ANN involves iteratively tuning its parameters (i.e., the w_{uv} 's and b_v corresponding to every neuron) in order to minimize the disparity between the desired/actual outputs and the predictions of the network, thereby improving its predictive accuracy. For a detailed exposition on the architecture of feed-forward ANNs, the interested reader is referred to the classic textbooks by Haykin [32] and Bishop [9].

3. The AGM framework

The process of changing beliefs has been formalized by Alchourrón, Gärdenfors and Makinson, through the introduction of a versatile framework for belief change, now called the *AGM framework*, after the initials of its three originators [1]. Within the AGM framework, the state of belief of an agent is represented by a logical theory K (also referred to as a *belief set*), and the new information (also named *epistemic input*) is represented by a logical formula φ . Between K and φ , the AGM framework examines two fundamental change operations, namely, *belief revision* (or simply revision) and *belief contraction* (or simply contraction). In their seminal article, the AGM trio characterized *axiomatically* both these types of change operations, in terms of a collection of well-accepted *rationality postulates*, whereas, in a subsequent work [37], Katsuno and Mendelzon developed a *possible-worlds* characterization for the process of revision. In this section, we present the axiomatic characterization of the AGM framework (Subsection 3.1), the possible-worlds characterization of Katsuno and Mendelzon (Subsection 3.2), as well as a concrete well-known AGM-style revision operator, proposed by Dalal [17] (Subsection 3.3).

3.1. Axiomatic characterization

In the context of the AGM framework, the process of revision is encoded into a *revision function*. A revision function $*$ is a binary function that maps a belief set K and a sentence φ to a belief set $K * \varphi$, representing the result of revising K by φ . We shall say that a revision function $*$ is an *AGM revision function* iff it respects the following rationality postulates $(K * 1) - (K * 8)$, known as the *AGM revision postulates* [1].

- (K * 1) $K * \varphi$ is a theory.
- (K * 2) $\varphi \in K * \varphi$.
- (K * 3) $K * \varphi \subseteq K + \varphi$.
- (K * 4) If $\neg\varphi \notin K$, then $K + \varphi \subseteq K * \varphi$.
- (K * 5) If φ is consistent, then $K * \varphi$ is also consistent.
- (K * 6) If $Cn(\varphi) = Cn(\psi)$, then $K * \varphi = K * \psi$.
- (K * 7) $K * (\varphi \wedge \psi) \subseteq (K * \varphi) + \psi$.
- (K * 8) If $\neg\psi \notin K * \varphi$, then $(K * \varphi) + \psi \subseteq K * (\varphi \wedge \psi)$.

It is stressed that, in the special case where φ is *consistent* with K (i.e., $\neg\varphi \notin K$), the AGM revision postulates $(K * 3)$ & $(K * 4)$ dictate that the process of revision degenerates to *expansion*, meaning that $K * \varphi = K + \varphi$.

In an analogous manner, the process of *contraction* is encoded into a *contraction function*. A contraction function $\dot{-}$ is a binary function that maps a belief set K and a sentence φ to a belief set $K \dot{-} \varphi$, representing the result of contracting φ from K . We shall say that a contraction function $\dot{-}$ is an *AGM contraction function* iff it respects the following rationality postulates $(K \dot{-} 1) - (K \dot{-} 8)$, known as the *AGM contraction postulates* [1].

- (K $\dot{-}$ 1) $K \dot{-} \varphi$ is a theory.
- (K $\dot{-}$ 2) $K \dot{-} \varphi \subseteq K$.
- (K $\dot{-}$ 3) If $\varphi \notin K$, then $K \dot{-} \varphi = K$.
- (K $\dot{-}$ 4) If φ is not tautological, then $\varphi \notin K \dot{-} \varphi$.
- (K $\dot{-}$ 5) If $\varphi \in K$, then $K \subseteq (K \dot{-} \varphi) + \varphi$.
- (K $\dot{-}$ 6) If $Cn(\varphi) = Cn(\psi)$, then $K \dot{-} \varphi = K \dot{-} \psi$.
- (K $\dot{-}$ 7) $(K \dot{-} \varphi) \cap (K \dot{-} \psi) \subseteq K \dot{-} (\varphi \wedge \psi)$.
- (K $\dot{-}$ 8) If $\varphi \notin K \dot{-} (\varphi \wedge \psi)$, then $K \dot{-} (\varphi \wedge \psi) \subseteq K \dot{-} \varphi$.

A concrete discussion on the rationale behind the AGM revision and contraction postulates has been conducted by [24, Chapter 3] and Peppas [55, Section 8.3]. Herein, we suffice to mention that their guiding principle is the *economy of information*, according to which the belief set K is modified *as little as possible* in response to the epistemic input φ .

It is noteworthy that the change operation identified by the AGM revision postulates and that identified by the AGM contraction postulates are not independent to each other; on the contrary, they are strongly *interrelated*. Such an interrelation was suggested by Harper [31], who proposed a procedure that defines contraction in terms of revision, encoded into the following condition (HR), known as the *Harper Identity*.

$$(HR) \quad K \dot{-} \varphi = (K * \neg\varphi) \cap K.$$

Condition (HR) asserts that, for contracting an epistemic input φ from a belief set K , one should, firstly, revise K by $\neg\varphi$, and then intersect the revised belief set with K . It turns out that, given an AGM revision function $*$, the contraction function $\dot{-}$ produced from $*$, through the Harper Identity, is an AGM contraction function [24].³

3.2. Possible-worlds characterization

It is true that the AGM revision postulates $(K * 1) - (K * 8)$ do *not* suffice to uniquely specify the revised belief set $K * \varphi$, given K and φ alone; the alluded postulates only identify the territory of all different rational ways of performing revision. For an exact specification (construction) of the belief set $K * \varphi$, *constructive models* for belief change are required, namely, appropriate *extra-logical* tools that codify particular modification policies. One such popular constructive model, based on a special kind of total preorders over possible worlds called *faithful preorders*, is the one proposed by Katsuno and Mendelzon [37].⁴

Definition 1 (*Faithful preorder*, [37]). A total preorder over \mathbb{M} , denoted by \leq_K , is faithful to a belief set K iff $[K] \neq \emptyset$ entails $[K] = \min(\mathbb{M}, \leq_K)$.

Intuitively, a faithful preorder \leq_K over \mathbb{M} encodes the *comparative plausibility* of all possible worlds of \mathbb{M} , relative to the belief set K ; the more plausible a world is modulo K , the lower it appears in the ordering \leq_K .

Definition 2 (*Faithful assignment*, [37]). A faithful assignment is a function that maps each belief set K to a total preorder \leq_K over \mathbb{M} , that is faithful to K .

Katsuno and Mendelzon proceed then to the following representation theorem.

Theorem 3 ([37]). A revision function $*$ satisfies postulates $(K * 1) - (K * 8)$ iff there exists a faithful assignment that maps each belief set K to a total preorder \leq_K over \mathbb{M} , such that, for any sentence $\varphi \in \mathbb{L}$:

$$(R) \quad [K * \varphi] = \min([\varphi], \leq_K).$$

Hence, according to condition (R), the revised belief set $K * \varphi$ is specified in terms of the most plausible φ -worlds relative to K .

Combining condition (R) with the Harper Identity (HR) of the preceding subsection, we are able to deduce a possible-worlds characterization for the process of contraction as well (cf. Section 7 of [14]). Specifically, let K be a belief set, and let $*$ be an AGM revision function that assigns at K a faithful preorder \leq_K over \mathbb{M} , via condition (R). Moreover, let $\dot{-}$ be the AGM contraction function induced from $*$, via the Harper Identity. Then, for any epistemic input φ of \mathbb{L} , the following condition (C) holds, suggesting that the contracted belief set $K \dot{-} \varphi$ is specified in terms of the set-theoretic union of the K -worlds and the most plausible $\neg\varphi$ -worlds relative to K .

$$(C) \quad [K \dot{-} \varphi] = [K] \cup \min([\neg\varphi], \leq_K).$$

3.3. Dalal's revision operator

For a belief set K , Dalal specifies the plausibility of possible worlds, encoded into a preorder \leq_K faithful to K , in terms of a *Hamming-based difference* between worlds [17]. In the limiting case where the belief set K is inconsistent (i.e., $[K] = \emptyset$), Dalal defines the belief set resulting from the revision of K by φ to be equal to $Cn(\varphi)$. For the principal case of a consistent belief set K (i.e., $[K] \neq \emptyset$), he proceeds to the following definitions.

Definition 4 (*Difference between worlds*). The difference between two worlds r, r' of \mathbb{M} , denoted by $\text{Diff}(r, r')$, is the set of propositional variables that have different truth values in the two worlds. That is,

³ An analogous procedure that defines revision in terms of contraction is also available by Levi [49].

⁴ Other popular constructive models for belief change are the *epistemic-entrenchment model* [25], and the *partial-meet model* [1].

$$\text{Diff}(r, r') = \left((r \setminus r') \cup (r' \setminus r) \right) \cap \mathcal{P}.$$

Definition 5 (Distance between belief sets and worlds, [17]). The distance between a consistent belief set K and a world r , denoted by $D(K, r)$, is the cardinality-minimum difference between r and the K -worlds. That is,

$$D(K, r) = \min \left(\left\{ \left| \text{Diff}(w, r) \right| : w \in [K] \right\} \right).$$

Definition 6 (Dalal's revision operator, [17]). Dalal's revision operator \star is the revision function induced, via condition (R), from the family of Dalal's preorders $\{\sqsubseteq_K\}_{\forall K}$, where each Dalal's preorder \sqsubseteq_K is uniquely specified such that, for any $r, r' \in \mathbb{M}$,

$$r \sqsubseteq_K r' \quad \text{iff} \quad D(K, r) \leq D(K, r').$$

As noted by Katsuno and Mendelzon [37, p. 269], for each (consistent) belief set K , \sqsubseteq_K is a total preorder faithful to K . Therefore, Dalal's revision operator satisfies all the AGM revision postulates, meaning that it is an AGM revision function. A concrete application of Dalal's revision operator is described in Example 11 of the subsequent section.

We close this section noting that the AGM framework has been extensively studied, and a plethora of augmentations to it have been proposed in the literature; the reader is indicatively referred to the works [38,54,58,57,3,4,19,5,22,41,28].

4. Belief change as a gradual transition of beliefs

Assume that an agent changes (revises or contracts) her initial belief set K_1 and reaches a modified belief set K_2 . The AGM framework, outlined in the previous section, specifies the rational properties that the transition from K_1 to K_2 should respect. However, the AGM framework does *not* address the actual K_1 -to- K_2 transition; it focuses solely on the initial and final states of this process, omitting the dynamics of the transition itself. In this section, we delve into the nature of the K_1 -to- K_2 belief transition. To that end, we firstly introduce in Subsection 4.1 two intuitive measures of distance between belief sets. Thereafter, in Subsection 4.2, we sketch the process of belief change as a gradual alteration of intermediate states of belief, and demonstrate how the aforementioned measures can naturally be utilized for effectively quantifying the disparity of the knowledge encoded into these intermediate states of belief. In Subsection 4.3, we highlight the association of the notion of gradual beliefs with notable formal frameworks of belief change.

4.1. Distance between belief sets

A measure of *distance* between two arbitrary belief sets K_1 and K_2 , essentially, encodes a notion of difference between the knowledge represented by the belief sets K_1 and K_2 . In Definition 7 and Definition 8, we introduce two intuitive *Hamming-based* such measures, namely, *type-A* distance and *type-B* distance between belief sets, respectively.

Definition 7 (Type-A distance). Let K_1, K_2 be two consistent belief sets. The type-A distance between K_1 and K_2 , denoted by $\text{Dist}_A(K_1, K_2)$, is the cardinality-minimum difference between the K_1 -worlds and the K_2 -worlds. That is,

$$\text{Dist}_A(K_1, K_2) = \min \left(\left\{ \left| \text{Diff}(w, w') \right| : w \in [K_1] \text{ and } w' \in [K_2] \right\} \right).$$

Definition 8 (Type-B distance). Let K_1, K_2 be two belief sets. The type-B distance between K_1 and K_2 , denoted by $\text{Dist}_B(K_1, K_2)$, is the cardinality of the symmetric difference between $[K_1]$ and $[K_2]$. That is,

$$\text{Dist}_B(K_1, K_2) = \left| ([K_1] \setminus [K_2]) \cup ([K_2] \setminus [K_1]) \right|.$$

Notice that the type-B distance between two belief sets K_1 and K_2 , essentially, expresses the Hamming distance between the possible worlds satisfied by these belief sets. Example 9 concretely illustrates the usage of both type-A and type-B distances.

Example 9 (Distances between belief sets). Let $\mathcal{P} = \{a, b\}$, and let $K_1 = \text{Cn}(a \vee b)$ and $K_2 = \text{Cn}(\neg a \wedge \neg b)$. Then, $[K_1] = \{ab, a\bar{b}, \bar{a}b\}$ and $[K_2] = \{\bar{a}\bar{b}\}$. In view of Definitions 7 and 8, we derive that $\text{Dist}_A(K_1, K_2) = 1$ and $\text{Dist}_B(K_1, K_2) = 4$, respectively.

4.2. Intermediate belief sets during belief change

It is well-accepted that humans often exhibit *resistance* to changing beliefs due to a variety of cognitive and psychological reasons [46,36]. This resistance to belief change implies that it would be plausible to assume that a *realistic* agent would often change her

beliefs *gradually*, rather than instantly; this gradual transition would reflect complex cognitive processes influenced by various factors, including evidence, social interactions, and interconnected belief systems. Indeed, such a gradualist view of belief change has been substantially supported in studies on human development [59,65,72].⁵

In response to this cognitive-based gradualism of beliefs, herein, we outline the transition of an agent, from an initial belief set K_1 to a modified (revised or contracted) belief set K_2 , as a *progressive adjustment* of beliefs. Specifically, we assume that the agent, as changing her state of belief from K_1 to K_2 , adheres to a sequence of *intermediate* belief sets H_1, \dots, H_n . Roughly speaking, the intermediate belief sets H_1, \dots, H_n represent in-between states of belief that are lying somewhere across the span between K_1 and K_2 . In what follows, we shall denote the sequence H_1, \dots, H_n by the tuple $\langle K_1, K_2 \rangle$; i.e., $\langle K_1, K_2 \rangle = (H_1, \dots, H_n)$.

As an indicative realistic example of the alluded intermediate states of belief, consider the subsequent scenario.

Example 10 (Gradual adjustment of beliefs). Bob has two friends, Maria and George, who do not know each other. Bob's initial beliefs is that Maria and George are *not* adopted. Hence, denoting by a and b the propositions "Maria is not adopted" and "George is not adopted", respectively, Bob's initial belief set is $K_1 = Cn(a \wedge b)$. Suppose, now, that Bob receives an epistemic input $\varphi = \neg a \wedge \neg b$, suggesting that both Maria and George are, as a matter of fact, adopted. In response to this new information, Bob revises his beliefs and ultimately adheres to the belief set $K_2 = Cn(\neg a \wedge \neg b)$. It seems reasonable to assume that, prior to adopting his revised belief set K_2 , Bob likely transitioned through an intermediate belief set — either $Cn(\neg a \wedge b)$ or $Cn(a \wedge \neg b)$. This indicates that before Bob came to believe that both of his friends are adopted, he initially believed that just one of the two friends is adopted. Such progression reflects a gradual adjustment in Bob's beliefs in response to new information.

In our formal context, there could be several reasonable ways through which the intermediate belief sets H_1, \dots, H_n are determined. Indicatively, they could be induced by the structure of the faithful preorder \leq_{K_1} that the agent assigns (via (R) and/or (C)) at the initial belief set K_1 ; alternatively, they could be induced by a notion of *distance* between belief sets (see Subsection 4.1), irrespectively of a faithful preorder. For instance, putting $(T_1, \dots, T_l) = (K_1, H_1, \dots, H_n, K_2)$, a plausible relation between the belief sets T_1, \dots, T_l is expressed through the following condition (D), for any $i, j, m \in \{1, \dots, l\}$ such that $i < j < m$,

$$(D) \quad \text{Dist}(T_i, T_m) \geq \text{Dist}(T_j, T_m),$$

where "Dist" denotes either type-A or type-B distance. Condition (D) essentially expresses the intuition that, as the agent modifies her beliefs in order to reach the belief set T_m , she progressively adheres to belief sets whose distances from T_m *gradually shrink*. One can also argue that the gradual transition from K_1 to K_2 could respect other plausible properties as well. For example, given that the shift from K_1 to K_2 is initiated by an epistemic input φ , consider the following four postulates (I1)–(I4).

- (I1) For every belief sets $H_i, H_j \in (H_1, \dots, H_n)$ such that $i < j$, if $\neg\varphi \notin H_i$, then $\neg\varphi \notin H_j$.
- (I2) For every belief sets $H_i, H_j \in (H_1, \dots, H_n)$ such that $i < j$, if $\varphi \in H_i$, then $\varphi \in H_j$.
- (I3) For every belief set $H \in (H_1, \dots, H_n)$, $\neg\varphi \notin H$.
- (I4) For every belief sets $H_i, H_j \in (H_1, \dots, H_n)$ such that $i \leq j$, there are non-tautological sentences $\varphi_i, \varphi_j \in \mathbb{L}$ such that $\varphi \vdash \varphi_j \vdash \varphi_i$, $\varphi_i \in H_i$ and $\varphi_j \in H_j$.

Postulates (I1) and (I2) are in the same spirit and enforce a sort of monotonicity property with respect to the epistemic input φ . Specifically, (I1) asserts that, once $\neg\varphi$ is not a belief of some intermediate belief set $H_i \in (H_1, \dots, H_n)$, it cannot be a belief of *any* subsequent belief set $H_j \in (H_1, \dots, H_n)$ (for $j > i$). Likewise, (I2) states that, once φ is a belief of some intermediate belief set $H_i \in (H_1, \dots, H_n)$, it should remain a belief of *every* subsequent belief set $H_j \in (H_1, \dots, H_n)$ (for $j > i$). Postulate (I3) is a rather radical condition and it is strictly stronger than (I1); it asserts that the negation of the epistemic input φ is *not* believed in any of the intermediate belief sets H_1, \dots, H_n . This implies that, even if the agent initially believes $\neg\varphi$ (i.e., $\neg\varphi \in K_1$), she abandons this belief at the first intermediate step, as $\neg\varphi \notin H_1$. Lastly, postulate (I4) captures the intuition that, as the agent transitions towards the final belief set K_2 , she progressively adheres to *logically stronger* non-tautological beliefs that entail φ . This indicates that the agent gradually accepts the informational content of the epistemic input φ . All postulates (I1)–(I4) express plausible properties that the transition from K_1 to K_2 may exhibit. However, our aim here is not to constrain the potential nature of this transition, therefore, we will refrain from introducing additional properties that could potentially impose limitations.

Example 11 illustrates three concrete scenarios of belief revision, in the context of which reasonable intermediate belief sets are presented — analogous scenarios in the context of belief contraction can be devised in a totally symmetric manner.

Example 11 (Intermediate belief sets). Let $\mathcal{P} = \{a, b, c\}$ and assume that $K = Cn(a \wedge b \wedge c)$ is the initial belief set of the agent; thus, $[K] = \{abc\}$. Firstly, suppose that, for implementing revision, the agent utilizes an AGM revision function $*$ that assigns (via (R)) at K a faithful preorder \leq_K over \mathbb{M} , specified as follows:

$$abc \quad <_K \quad \begin{array}{c} \bar{a}\bar{b}\bar{c} \\ \bar{a}\bar{b}c \end{array} \quad <_K \quad \begin{array}{c} \bar{a}\bar{b}\bar{c} \\ \bar{a}b\bar{c} \end{array} \quad <_K \quad \begin{array}{c} \bar{a}b\bar{c} \\ \bar{a}bc \end{array}$$

⁵ In a formal context, the belief-change process of non-ideal agents constitutes an important domain of study [73].

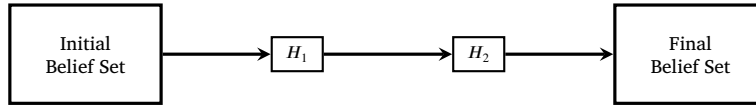


Fig. 1. Belief transitions of the agent described in Example 11.

Now, let $\varphi = (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge c)$ be an epistemic input. Then, according to condition (R) of Subsection 3.2, the $*$ -revision of K by φ produces a belief set $K * \varphi$, such that $[K * \varphi] = \min([\varphi], \leq_K) = \{\bar{a}bc, a\bar{b}c\}$; thus, $K * \varphi = Cn((\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge c))$. Against this background, a sensible \leq_K -generated sequence of intermediate belief sets to which the agent adheres during the transition from K to $K * \varphi$ would be

$$\langle K, K * \varphi \rangle = \left(H_1 = Cn((a \vee b) \wedge \neg c), H_2 = Cn(\neg a \wedge \neg b) \right),$$

for which $[H_1] = \{\bar{a}\bar{b}c, \bar{a}bc, abc\}$ and $[H_2] = \{\bar{a}\bar{b}c, \bar{a}bc\}$. Observe that, for any $r \in [H_1]$, any $r' \in [H_2]$ and any $r'' \in [K * \varphi]$, $r \leq_K r' \leq_K r''$, meaning that the H_1 -worlds are more plausible (modulo K) than the H_2 -worlds, which are in turn more plausible (modulo K) than the $K * \varphi$ -worlds. Essentially, this suggests that, during the process of revision, the agent transitions from initially more plausible states of belief to ones that are less plausible, ultimately arriving at the revised belief set $K * \varphi$.

Next, assume that, for implementing revision, the agent utilizes Dalal's revision operator \star that assigns (via (R)) at K the (uniquely defined) faithful preorder \sqsubseteq_K over \mathbb{M}^6 :

$$abc \quad \sqsubseteq_K \quad \begin{array}{c} \bar{a}bc \\ \bar{a}\bar{b}c \\ ab\bar{c} \end{array} \quad \sqsubseteq_K \quad \begin{array}{c} \bar{a}\bar{b}c \\ \bar{a}\bar{b}\bar{c} \\ a\bar{b}\bar{c} \end{array} \quad \sqsubseteq_K \quad \bar{a}\bar{b}\bar{c}$$

Now, let $\varphi = \neg a \wedge \neg b \wedge \neg c$ be an epistemic input. Then, the \star -revision of K by φ produces a belief set $K \star \varphi$, such that $[K \star \varphi] = \min([\varphi], \leq_K) = \{\bar{a}\bar{b}\bar{c}\}$; thus, $K \star \varphi = Cn(\neg a \wedge \neg b \wedge \neg c)$. Against this background, a sensible \sqsubseteq_K -generated sequence of intermediate belief sets to which the agent adheres during the transition from K to $K \star \varphi$ would be

$$\langle K, K \star \varphi \rangle = \left(H_1 = Cn((\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge c) \vee (a \wedge b \wedge \neg c)), H_2 = Cn((\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c)) \right),$$

for which $[H_1] = \{\bar{a}bc, \bar{a}\bar{b}c, abc\}$ and $[H_2] = \{\bar{a}\bar{b}c, \bar{a}bc, ab\bar{c}\}$. Observe that, for any $r \in [H_1]$, any $r' \in [H_2]$ and any $r'' \in [K \star \varphi]$, $r \sqsubseteq_K r' \sqsubseteq_K r''$. Moreover, putting $(T_1, T_2, T_3, T_4) = (K, H_1, H_2, K \star \varphi)$, it is true that, for any $i, j, m \in \{1, 2, 3, 4\}$ such that $i < j < m$, $\text{Dist}_A(T_i, T_m) > \text{Dist}_A(T_j, T_m)$. This implies that the intermediate belief sets H_1 and H_2 represent in-between states of belief that are lying across the type-A distance between K and $K \star \varphi$, which is $\text{Dist}_A(K, K \star \varphi) = 3$. Observe that the relation of the type-A distances between the involved belief sets is the one encoded into the above-mentioned condition (D).

As a last scenario, assume that $K_1 = Cn((a \vee b) \wedge c)$ is the initial belief set of the agent. Suppose that the agent revises K_1 by $\varphi = \neg a \wedge \neg b \wedge \neg c$, and results in a revised belief set $K_2 = Cn(\neg a \wedge \neg b \wedge \neg c)$. Thus, $[K_1] = \{abc, \bar{a}bc, \bar{a}\bar{b}c\}$ and $[K_2] = \{\bar{a}\bar{b}\bar{c}\}$, meaning that $\text{Dist}_B(K_1, K_2) = 4$. Then, a sequence of intermediate belief sets to which the agent adheres during the transition from K_1 to K_2 could be

$$\langle K_1, K_2 \rangle = \left(H_1 = Cn((\neg a \wedge \neg b \wedge \neg c) \vee (a \wedge \neg b \wedge c) \vee (a \wedge b \wedge \neg c)), H_2 = Cn((\neg a \wedge \neg b \wedge \neg c) \vee (a \wedge \neg b \wedge c)) \right),$$

for which $[H_1] = \{\bar{a}\bar{b}\bar{c}, \bar{a}bc, abc\}$ and $[H_2] = \{\bar{a}\bar{b}\bar{c}, \bar{a}bc\}$. Putting $(T_1, T_2, T_3, T_4) = (K_1, H_1, H_2, K_2)$, it is true that, for any $i, j, m \in \{1, 2, 3, 4\}$ such that $i < j < m$, $\text{Dist}_B(T_i, T_m) > \text{Dist}_B(T_j, T_m)$. This suggests that the intermediate belief sets H_1 and H_2 represent in-between states of belief, with the posterior belief set H_2 being "closer" to K_2 than the prior belief set H_1 . Note that the sequence $\langle K_1, K_2 \rangle$ is irrespective of a faithful preorder; it solely depends on the type-B distances between the involved belief sets, in a way that condition (D) is respected.

With regard to postulates (I1)–(I4), in the first two scenarios, postulates (I1) and (I2) are trivially respected, while postulates (I3) and (I4) are violated. In contrast, the final scenario satisfies all postulates (I1)–(I4), with (I2) being trivially so. Finally, note that the corresponding transitions in the agent's beliefs are abstractly illustrated in Fig. 1.

4.3. Interesting associations

The concept of gradual beliefs discussed earlier parallels, to some extent, the idea of *non-prioritized revision*, a form of belief revision designed to weaken postulate $(K * 2)$. Prominent types of non-prioritized revision include *credibility-limited revision* by Hansson et al. [30], where new information is either fully accepted or entirely rejected, and *selective revision* by Fermé and Hansson [22], where an agent may accept only a portion of the new information, embodying the essence of postulate (I4). Similarly, *improvement operators* by Konieczny et al. [40] and *promotion operators* by Schwind et al. [64] employ distinct methods to implement revision by increasing

⁶ \sqsubseteq_K denotes the strict part of \sqsubseteq_K .

the plausibility/firmness of the epistemic input within the state of belief. These types of belief change are indeed connected to the idea of gradual beliefs, though they primarily concern the process of revision, whereas, gradual beliefs also encompass the process of contraction. A closer investigation of these associations promises to yield intriguing insights and is suggested for future research.

Beyond the above-mentioned non-prioritized and incremental approaches, a line of research has explored how sequential belief revision can be viewed as a learning process that converges toward *truth* in the limit. Kelly [39] investigates the “learning power” of revision methods, analyzing in what sense they can reliably identify correct hypotheses across multiple environments. Similarly, work by Baltag, Gierasimczuk, and Smets [7,8] focuses on “truth-tracking”, showing how repeated revisions of one’s state of belief can lead to eventual convergence on correct beliefs. These frameworks resonate with the gradualist perspective introduced here, insofar as they cast belief revision itself as a step-by-step adjustment that can stabilize on the truth under suitable conditions.

On a separate note, assume that the agent assigns (via (R) and/or (C)) the faithful preorder \leq_{K_1} at the initial belief set K_1 and the faithful preorder \leq_{K_2} at the final belief set K_2 . Our analysis identifies the transition from K_1 to K_2 as a gradual transition of beliefs. Similarly, it would be of interest to sketch the transition from the initial preorder \leq_{K_1} to the final preorder \leq_{K_2} as a *gradual transition of preorders*. In this context, the intuition behind the well-known postulates proposed by Darwiche and Pearl [19], along with works by Booth and Meyer [12] or Spohn [68], can provide valuable guidance. Alternatively, the transition from \leq_{K_1} to \leq_{K_2} could be based on a notion of distance between preorders, analogous to the one expressed via condition (D). A promising candidate for this distance is the cardinality of the symmetric difference between \leq_{K_1} and \leq_{K_2} . Under this premise, as the agent modifies her initial preorder \leq_{K_1} to reach the final preorder \leq_{K_2} , she will progressively adhere to preorders whose distances from \leq_{K_2} gradually shrink.

Having discussed some basic concepts of AGM-style belief change, the remainder of this article is devoted to the establishment of a connection between AGM-style belief change and a special type of feed-forward ANNs, which we shall call *binary ANNs*.

5. Binary Artificial Neural Networks

In this work, we solely consider feed-forward ANNs whose inputs X_1, \dots, X_n take *binary values* (either 0 or 1). Given that *any* collection of input data can be converted into a collection of binary values, the previous assumption does not restrict the type of input of the considered ANNs. We also assume that each neuron i (with $i \in \{1, \dots, m\}$) of the *output layer* of an ANN produces a real value $y_i \in [0, 1]$. Notice that an ANN with these properties places *no* constraints on the values of its weights and biases. Thus, an ANN with a single neuron in its output layer that implements the sigmoid activation function, used for binary classification, or an ANN that uses the softmax activation function in its output layer, common for multi-class classification, are both representative examples of ANNs with the aforementioned properties, provided that the networks accept binary values in their input layers.

Now, given a (real-valued) threshold $\tau_i \in [0, 1]$, each real-valued output y_i of an ANN of the above type identifies a *binary output* Y_i , such that:

$$Y_i = \begin{cases} 1 & \text{if } y_i \geq \tau_i \\ 0 & \text{if } y_i < \tau_i \end{cases}$$

Thus, an ANN with the above-mentioned properties, along with a collection of thresholds, form an (augmented) ANN whose inputs X_1, \dots, X_n and outputs Y_1, \dots, Y_m *all* take binary values. We shall call an ANN of this type a *binary ANN*. The topology of a binary ANN, with a single hidden layer, is depicted in Fig. 2.

Evidently, the limited assumptions/constraints characterizing binary ANNs make them suitable for a plethora of real-world applications. Indicatively, the well-known MNIST (Modified National Institute of Standards and Technology) dataset [48]—which contains a large collection of handwritten digits that are commonly used as training-benchmark for various classification algorithms [60,44]—can be utilized for training a binary ANN. As the images in the MNIST dataset consist of grayscale pixel values ranging from 0 to 255, the only modification required for feeding the MNIST dataset to a binary ANN is the conversion of each grayscale value (0-255) to an 8-bit binary number, or even a binarization into strictly black-and-white pixels.⁷

The following crucial observation, grounded in the universal-approximation capabilities of ANNs [33], allows the operation of an arbitrary binary ANN with *multiple* outputs to be reduced to a collection of binary ANNs, each with a *single* output.

Remark 12. The input-output relationship of any binary ANN with multiple outputs can be emulated by means of a collection of binary ANNs, each with a single output.

In view of Remark 12, and for ease of presentation, our analysis will primarily focus on *single-output* binary ANNs.

Firstly, it is stressed that a binary ANN with a single binary output Y induces a *Boolean function*.⁸ We shall refer to the Boolean function f corresponding to the binary output Y as the Boolean function of Y ; thus,

⁷ The use of an Autoencoder [42,43] to compress the MNIST dataset into a lower-dimensional representation could facilitate more efficient learning by the binary ANN.

⁸ Notable indicative works that study the induced Boolean functions of ANNs are the works [51,15,18,67]. Moreover, recent works have demonstrated how various ML classifiers can be associated with Boolean functions, exhibiting the same input-output behaviors. For instance, Izza et al. [35] explore the case of decision trees, Ignatiev and Marques-Silva [34] investigate the case of decision lists, while Audemard et al. [6] examine the case of several families of ML classifiers.

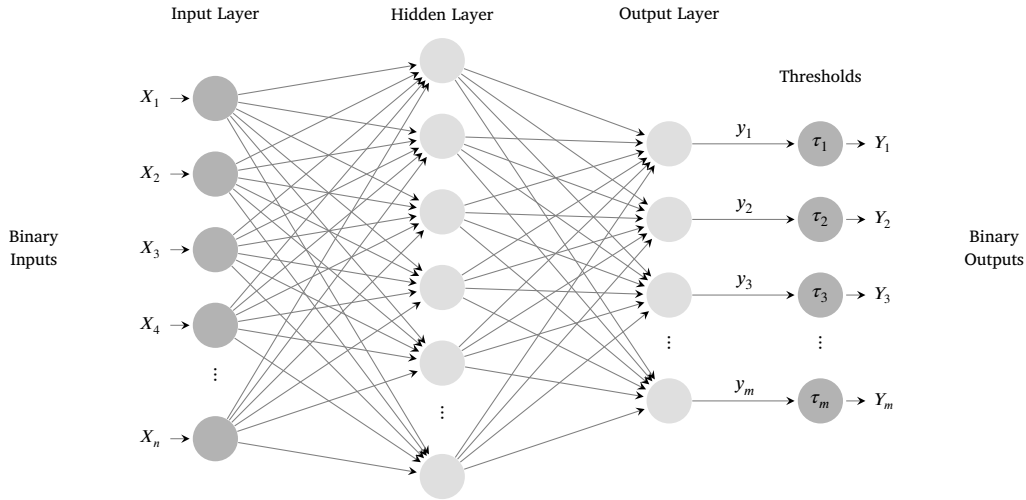


Fig. 2. The topology of a binary ANN, with a single hidden layer, which receives binary inputs X_1, \dots, X_n and generates binary outputs Y_1, \dots, Y_m . Each real value y_i (with $i \in \{1, \dots, m\}$) of the output layer of the ANN passes through the threshold τ_i and generates a binary output Y_i .

$$Y = f(X_1, \dots, X_n).$$

The Boolean function f of Y can be represented as a propositional formula ψ of \mathbb{L} , in the sense that f and ψ respect identical truth tables. To illustrate the relation between f and ψ , assume that the number of propositional variables of \mathcal{P} is the same as the number of the binary inputs X_1, \dots, X_n of the ANN; i.e., $|\mathcal{P}| = n$. Moreover, let w be a possible world of \mathbb{M} such that, for any $p_i \in \mathcal{P}$ with $i \in \{1, \dots, n\}$, $p_i \in w$ whenever $X_i = 1$ and $\neg p_i \in w$ whenever $X_i = 0$. Obviously then, there is a *one-to-one correspondence* between the propositional variables of \mathcal{P} and the binary inputs X_1, \dots, X_n . Hence, for the binary output Y , the following statement is true:

$$f(X_1, \dots, X_n) = 1 \quad \text{iff} \quad w \in [\psi].$$

That is to say, the possible world w satisfies ψ iff the Boolean function f of Y maps to 1 the truth values of the propositional variables assigned in w . Given that, for the belief set $K = Cn(\psi)$, it is true that $[K] = [\psi]$, it follows that the Boolean function f of the binary output Y can, equivalently, be represented by the belief set K .

The following concrete example presents a simple, yet representative, binary ANN with a single output, whose input-output relationship can symbolically be represented by means of a propositional formula or, equivalently, by means of a belief set.

Example 13 (Symbolic knowledge of a binary ANN). Consider a binary ANN that has two binary inputs X_1, X_2 and one binary output Y , and assume that it has been trained so that it implements the logical operation OR . Let a, b be the propositional variables corresponding to the binary inputs X_1, X_2 , respectively, and let f be the Boolean function of the binary output Y . Given that $f(0, 0) = 0$, $f(0, 1) = 1$, $f(1, 0) = 1$ and $f(1, 1) = 1$, we derive that the propositional formula $\psi = a \vee b \in \mathbb{L}$, such that $[\psi] = \{\bar{a}b, a\bar{b}, ab\}$, represents the Boolean function f of Y , and hence, the input-output relationship of the ANN. Equivalently, the Boolean function f of Y is also represented by the belief set $K = Cn(a \vee b)$. Observe that the sentence ψ is a disjunction of the propositional variables a and b , and thereby it expresses the logical operation OR , as expected.

It follows from Remark 12 that the input-output relationship of a binary ANN with multiple binary outputs Y_1, \dots, Y_m can be represented by means of a non-empty tuple $S = \langle K_1, \dots, K_m \rangle$ of belief sets, where each belief set of S represents the Boolean function corresponding to a binary ANN with a single output. Hence, as illustrated in Fig. 3, a tuple S , which we shall call *belief state*, can represent the state of belief (i.e., the input-output relationship) of any binary ANN.⁹

6. Machine learning through backpropagation

Backpropagation is a core algorithm essential for training feed-forward ANNs. The algorithm leverages calculus' principles in order to optimize the network's parameters (weights and biases), aiming to *reduce* the discrepancies between the predicted/estimated outputs and the actual/desired target values. In this section, we shall present a brief overview of the backpropagation algorithm — for further details, the interested reader is referred to the seminal article by Rumelhart et al. [62], or to the classic textbooks by Haykin [32] and Bishop [9].

⁹ In [11], a collection of belief corpora is referred to as a *flock*.

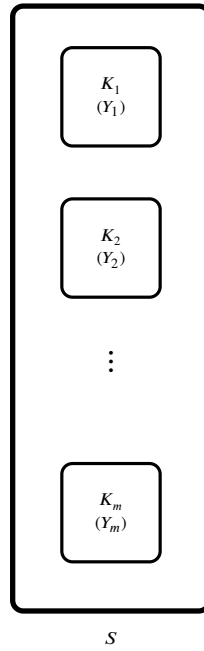


Fig. 3. A belief state $S = \langle K_1, K_2, \dots, K_m \rangle$ that represents the state of belief of a binary ANN, with binary outputs Y_1, Y_2, \dots, Y_m . Each belief set of S represents the Boolean function corresponding to a binary ANN with a single output.

6.1. Overview of backpropagation

In a typical ANN, each neuron outputs a signal that is a non-linear function of the weighted sum of its inputs. The parameters defining these relationships are the weights and biases. Weights are arranged in matrices, with each matrix corresponding to connections between two successive layers, whereas biases are similarly organized into vectors.

The process of training an ANN involves several key steps, initialized by what is known as *forward propagation*. In that context, starting from the input layer and moving through to the output layer, each neuron's output is calculated based on the current set of weights and biases. The core of backpropagation begins after this forward pass, when the algorithm calculates the *gradient* of the *loss function* \mathcal{L} —a measure of prediction error— relative to each parameter (weight and bias). This process involves the following sub-steps:

1. **Error Estimation at Output:** The difference between the actual output and the desired output *for all samples* is first determined at the output layer. This difference forms the basis of the loss function \mathcal{L} , which quantifies the error at the network's output. Note that a *sample* is a pair (\mathbf{x}, y) , where \mathbf{x} is a (typically vector-valued) collection of *feature* values representing a single observation, and y is the ground-truth *label* (actual target value) associated with that observation. The sample represents one instance from the dataset used to train or evaluate the ANN.
2. **Error Propagation Backwards:** The aforementioned error, expressed via the loss function \mathcal{L} , is then propagated backward through the network. This backward pass efficiently computes the gradients of \mathcal{L} using the chain rule, a fundamental rule in calculus, enabling the estimation of error contributions from all neurons in the network.
3. **Gradient Calculation:** For each parameter, a gradient (partial derivative) of the loss function \mathcal{L} is calculated which signifies the direction and rate at which the error would decrease the fastest. These gradients indicate how much a change in each parameter will impact \mathcal{L} .
4. **Parameters Update:** The weights and biases are then adjusted in the *opposite direction* of their respective gradients from \mathcal{L} , scaled by a small factor α , known as the *learning rate*. This step is aimed at reducing the value of the loss function \mathcal{L} , by updating the parameters to decrease the error.

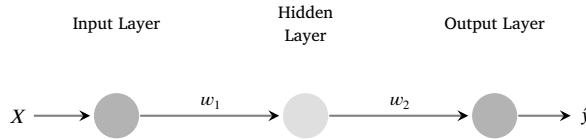
By repeating the above steps across multiple iterations (epochs), the ANN gradually learns the optimal parameters that *minimize* the loss function \mathcal{L} , thereby enhancing its accuracy in making predictions.

It is important to note that the type of loss function \mathcal{L} is intentionally left unspecified to allow for *flexibility* in its choice. Common types of loss functions include Mean Squared Error for regression tasks and cross-entropy loss for classification tasks. For the purposes of this study, it suffices to bear in mind that, regardless of its specific form and some basic properties that it needs to respect (such as

differentiability), the loss function \mathcal{L} is a well-behaved function that quantifies the difference between the predicted outputs and the actual target values across all samples.¹⁰

The following toy example captures the essence of how backpropagation flows gradients backward to adjust weights based on prediction error, even in the simplest architectures.

Example 14 (Backpropagation). Consider the minimalistic feed-forward ANN illustrated below, consisting of a single neuron in the input layer, a hidden layer with one neuron, and a single neuron in the output layer. Let X denote the input, and y the ground-truth label. Let w_1 denote the weight connecting the input and hidden layer neurons, and w_2 the weight connecting the hidden and output layer neurons. For simplicity, we assume no biases. The two neurons of the hidden and output layer use the sigmoid activation function $\sigma(x) = \frac{1}{1+e^{-x}}$.



During forward propagation, the output of the hidden-layer neuron is $h = \sigma(w_1 \cdot X)$, and the network's prediction is $\hat{y} = \sigma(w_2 \cdot h)$. The binary cross-entropy loss function is given by $\mathcal{L} = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y}))$.

In the backpropagation phase, using the chain rule and the derivative of the sigmoid function $\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$, the gradient (partial derivative) of \mathcal{L} with respect to each weight is computed as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_2} &= \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_2} = \left(\frac{\hat{y} - y}{\hat{y} \cdot (1 - \hat{y})} \right) \cdot (\hat{y} \cdot (1 - \hat{y}) \cdot h) = (\hat{y} - y) \cdot h \\ \frac{\partial \mathcal{L}}{\partial w_1} &= \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h} \cdot \frac{\partial h}{\partial w_1} = \left(\frac{\hat{y} - y}{\hat{y} \cdot (1 - \hat{y})} \right) \cdot (\sigma'(w_2 \cdot h) \cdot w_2) \cdot (\sigma'(w_1 \cdot X) \cdot X) = \\ &= \left(\frac{\hat{y} - y}{\hat{y} \cdot (1 - \hat{y})} \right) \cdot (\hat{y} \cdot (1 - \hat{y}) \cdot w_2) \cdot (\sigma'(w_1 \cdot X) \cdot X) = (\hat{y} - y) \cdot w_2 \cdot \sigma'(w_1 \cdot X) \cdot X \end{aligned}$$

Finally, the weights are updated via gradient descent as follows:

$$w_i := w_i - \alpha \cdot \frac{\partial \mathcal{L}}{\partial w_i}, \quad \text{for } i = 1, 2.$$

6.2. The smoothness and monotonicity assumptions

In the course of this work, we adopt the subsequent two simplifying *assumptions* that characterize the training process of any single-output ANN.

- 1. Smoothness:** In each iteration of the parameters update using the backpropagation algorithm, the loss function \mathcal{L} *strictly decreases*. Formally, this means that, for any two *consecutive* values $\mathcal{L}_1, \mathcal{L}_2$ of the loss function, the following condition (S) holds:

$$(S) \quad \mathcal{L}_2 < \mathcal{L}_1.$$

The smoothness assumption, encoded into condition (S), allows us to focus on the convergence properties of the backpropagation algorithm under idealized conditions. While this assumption does not generally hold in practice, due to factors such as the stochastic nature of the optimization, learning rate selection, and the presence of non-convex loss landscapes, it simplifies the analysis and enables us to derive certain theoretical guarantees about the learning process. It is important to note that, in practical applications, techniques such as learning-rate schedules, momentum, or adaptive learning-rate methods are employed to mitigate issues that may prevent the loss from decreasing after every update [27].

- 2. Monotonicity:** Given that the ANN is trained over N individual samples, its loss function \mathcal{L} is *monotonically related* to the sum $\sum_{s=1}^N |\hat{y}_s - y_s|$, where \hat{y}_s is the (real-valued) prediction of the output-layer neuron for the s -th sample, and y_s is the corresponding label.¹¹ This monotonicity assumption implies that the ANN is closely tied to the sum of the absolute errors across all predictions; hence, reducing these errors would directly reduce the value of the loss function \mathcal{L} , leading to better model performance.

¹⁰ Details on types and properties of loss functions can be found in the textbook by Goodfellow et al. [27].

¹¹ Two functions f and g (defined on the same domain) are monotonically related whenever, for all x, y in the domain, $f(x) \leq f(y)$ iff $g(x) \leq g(y)$. Thus, f and g have a consistent direction of increase or decrease over their common domain.

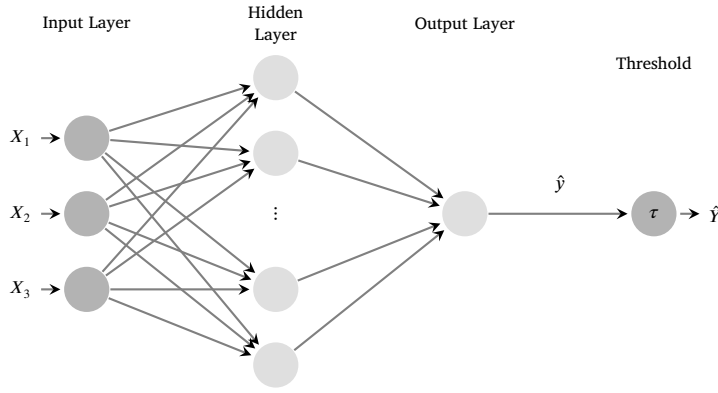


Fig. 4. The topology of the binary ANN of Example 17 (cf. Fig. 2 of Section 5).

The following interesting observations can be easily verified, and shall prove useful in our subsequent discussion. Remark 15 concerns the relation between the predictions of the output-layer neuron of a single-output binary ANN and the corresponding predictions of its binary output, whereas, Remark 16 connects the predictions of the binary output of the ANN with the type-B distance between belief sets (see Definition 8).

Remark 15. Consider a binary ANN with a single output that is trained over N individual samples. Let \hat{y}_s, \hat{y}'_s be two (real-valued) predictions of the output-layer neuron of the ANN for the s -th sample, let \hat{Y}_s, \hat{Y}'_s be the corresponding (binary) predictions of the binary output of the ANN, and let y_s be the corresponding label. Then, $\sum_{s=1}^N |\hat{y}_s - y_s| > \sum_{s=1}^N |\hat{y}'_s - y_s|$ entails $\sum_{s=1}^N |\hat{Y}_s - y_s| \geq \sum_{s=1}^N |\hat{Y}'_s - y_s|$.

In view of Remark 15, it follows that, if the sum $\sum_{s=1}^N |\hat{y}_s - y_s|$ decreases (resp., increases), then the sum $\sum_{s=1}^N |\hat{Y}_s - y_s|$ cannot increase (resp., cannot decrease).

Remark 16. Let Y be the binary output of a single-output binary ANN, which is trained over N individual samples. Let \hat{Y}, \hat{Y}' be two binary estimations that Y generates during training, and let K, K' be the belief sets that represent the Boolean functions corresponding to the estimations \hat{Y}, \hat{Y}' , respectively. Then,

$$\sum_{s=1}^N |\hat{Y}_s - \hat{Y}'_s| = \text{Dist}_B(K, K')$$

6.3. Illustrative examples: training binary ANNs

Let us now present Example 17, which illustrates the development of a binary ANN trained to perform a certain logical operation, while adhering to smoothness and monotonicity assumptions throughout its training process. Example 17 will serve as a running example in the subsequent sections of this article.

Example 17 (Training a binary ANN for a logical operation). A binary ANN is built with the aid of Keras Python library. The ANN will be trained in order to implement the logical operation “at least one but not all”, and has the following topology, as visualized in Fig. 4:

- Three neurons in its input layer, thus, three binary inputs X_1, X_2, X_3 .
- One hidden layer with 50 neurons, each one equipped with a Rectified Linear Unit (ReLU) activation function.
- One neuron in its output layer, equipped with a sigmoid activation function, producing a real-valued estimation \hat{y} . The estimation \hat{y} passes through a threshold $\tau = 0.5$, thus, the binary output Y of the ANN generates a single binary estimation \hat{Y} .

As the ANN should implement the logical operation “at least one but not all”, the Boolean function f of the binary output Y should be such that:

Table 1

Successive transitions of the binary output Y of the ANN, during training, for all (8) possible combinations of input. The sum $\sum_{s=1}^8 |\hat{Y}_s - y_s|$, the loss function \mathcal{L} and the accuracy of the ANN, during training, are also reported.

Input	1-st Output (\hat{Y}_s)	2-nd Output (\hat{Y}_s)	3-rd Output (\hat{Y}_s)	4-th Output (\hat{Y}_s)	5-th Output (\hat{Y}_s)	6-th Output (\hat{Y}_s)	Label (y_s)
(0,0,0)	1	1	1	1	0	0	0
(0,0,1)	0	0	1	1	1	1	1
(0,1,0)	0	0	0	1	1	1	1
(0,1,1)	0	0	1	1	1	1	1
(1,0,0)	0	1	1	1	1	1	1
(1,0,1)	0	0	1	1	1	1	1
(1,1,0)	0	0	0	1	1	1	1
(1,1,1)	0	0	1	1	1	0	0
$\sum_{s=1}^8 \hat{Y}_s - y_s $	7	6	4	2	1	0	—
\mathcal{L}	0.8324	0.7101	0.6766	0.6593	0.6461	0.1479	—
Accuracy	12.5%	25%	50%	75%	87.5%	100%	—

$$f(0,0,0) = 0$$

$$f(0,0,1) = 1$$

$$f(0,1,0) = 1$$

$$f(0,1,1) = 1$$

$$f(1,0,0) = 1$$

$$f(1,0,1) = 1$$

$$f(1,1,0) = 1$$

$$f(1,1,1) = 0$$

The above mapping forms the training dataset of the ANN, which consists of 8 samples. The co-domain of f contains the labels that shall be used during the training of the ANN. The ANN is compiled using a binary cross-entropy loss function \mathcal{L} (commonly used for binary classification), which is defined as follows:

$$\mathcal{L} = -\frac{1}{8} \sum_{s=1}^8 \left(y_s \cdot \log(\hat{y}_s) + (1 - y_s) \cdot \log(1 - \hat{y}_s) \right),$$

where \hat{y}_s is the (real-valued) prediction of the output-layer neuron for the s -th sample, and y_s is the corresponding (binary) label. The network parameters were initialized randomly prior to training.

On that basis, Table 1 shows the successive transitions of the binary output Y of the ANN, during training, for all (8) possible combinations of input. The table also reports the sum $\sum_{s=1}^8 |\hat{Y}_s - y_s|$, as well as the values of the loss function \mathcal{L} and accuracy of the ANN, during training — note that the accuracy is defined as $\frac{8 - \sum_{s=1}^8 |\hat{Y}_s - y_s|}{8} \cdot 100\%$. It is evident that, as the network is being trained, the loss function \mathcal{L} and the sum $\sum_{s=1}^8 |\hat{Y}_s - y_s|$ strictly decrease. This implies that the network respects the smoothness and monotonicity assumptions during its training. On the other hand, the accuracy strictly increases, reaching a value of 100%, meaning that the network has been properly trained.

Next, let us denote by a, b, c the propositional variables corresponding to the binary inputs X_1, X_2, X_3 , respectively. Let $K_1, K_2, K_3, K_4, K_5, K_6$ be the belief sets that represent the six different Boolean functions to which the binary output Y successively adheres during training, as illustrated in Table 1. It follows then from the table that:

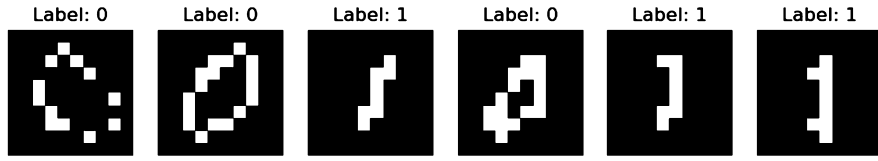


Fig. 5. Representative MNIST images used in the experimental setup of Example 18, along with their corresponding (binary) labels.

$$\begin{aligned}
 [K_1] &= \{\bar{a}\bar{b}\bar{c}\} && \iff K_1 = Cn(\neg a \wedge \neg b \wedge \neg c) \\
 [K_2] &= \{\bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}c\} && \iff K_2 = Cn(\neg b \wedge \neg c) \\
 [K_3] &= \{\bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}c, \bar{a}b\bar{c}, \bar{a}bc, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}c, abc\} && \iff K_3 = Cn(\neg b \vee c) \\
 [K_4] &= \mathbb{M} && \iff K_4 = Cn(\emptyset) \\
 [K_5] &= \{\bar{a}\bar{b}c, \bar{a}\bar{b}\bar{c}, \bar{a}b\bar{c}, \bar{a}bc, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}c, \bar{a}b\bar{c}, abc\} && \iff K_5 = Cn(a \vee b \vee c) \\
 [K_6] &= \{\bar{a}\bar{b}c, \bar{a}\bar{b}\bar{c}, \bar{a}b\bar{c}, \bar{a}bc, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}c, \bar{a}b\bar{c}, abc\} && \iff K_6 = Cn((a \vee b \vee c) \wedge \neg(a \wedge b \wedge c))
 \end{aligned}$$

Therefore, the Boolean function f of the binary output Y of the trained ANN is represented by the belief set $K_6 = Cn((a \vee b \vee c) \wedge \neg(a \wedge b \wedge c))$, or equivalently, by the propositional formula $\psi = (a \vee b \vee c) \wedge \neg(a \wedge b \wedge c)$. As expected due to the proper training of the ANN, both K_6 and ψ express the logical operation “at least one but not all”.

Finally, we note that alternative initializations of the ANN’s parameters may result in different transition patterns for the binary output Y . Nonetheless, as long as the network adheres to the smoothness and monotonicity assumptions throughout training, the validity of our analysis is preserved.

Example 18 concludes this section by demonstrating the training of a binary ANN on the MNIST dataset. As noted in Section 5, the MNIST dataset is a standard benchmark of handwritten digits, with each image represented by grayscale pixel values [48].

Example 18 (Training a binary ANN on the MNIST dataset). A binary ANN is built using the Keras Python library to recognize handwritten digits 0 and 1 from the MNIST dataset. Our training and testing datasets comprise 700 and 100 samples, respectively, with no overlap. Each MNIST image is down-sampled from its original 28×28 resolution to 10×10 pixels, and subsequently binarized into black-and-white pixels. Hence, each image can be effectively encoded as a sequence of $10 \times 10 = 100$ binary numbers. Fig. 5 displays a representative selection of images used in the experimental setup, along with their corresponding (binary) labels.

The binary ANN consists of the following layers:

- An input layer with 100 neurons, thus, 100 binary inputs X_1, \dots, X_{100} , each receiving one binary pixel from an image.
- A hidden layer with 10 neurons, each utilizing a ReLU activation function.
- An output layer with a single neuron, equipped with a sigmoid activation function, producing a real-valued estimation \hat{y} . The expected output is $\hat{y} = 0$ for images representing the digit 0, and $\hat{y} = 1$ for images representing the digit 1. The prediction \hat{y} is thresholded at $\tau = 0.5$, so that the binary output Y of the ANN generates a single binary estimation \hat{Y} .



The ANN is trained using the binary cross-entropy loss function \mathcal{L} , as defined in Example 17, with network parameters randomly initialized prior to training.

Within this setting, the following analysis will be exclusively conducted on the 100 samples of the testing dataset. Let us begin with Table 2, which illustrates the successive transitions of the binary output Y of the ANN, during training, for two indicative input images of the testing dataset. The table also reports the sum $\sum_{s=1}^{100} |\hat{Y}_s - y_s|$, the loss function \mathcal{L} and the accuracy of the network, during training. Notably, as training progresses, the loss function \mathcal{L} and the sum $\sum_{s=1}^{100} |\hat{Y}_s - y_s|$ on the testing dataset strictly decrease, implying that the network adheres to the smoothness and monotonicity assumptions (on the testing dataset) during its training. Simultaneously, accuracy steadily improves, reaching an impressive 99% on the testing dataset.

Thereafter, let us denote by p_1, \dots, p_{100} the propositional variables corresponding to the binary inputs X_1, \dots, X_{100} , respectively. Clearly, there is a one-to-one-correspondence between the $10 \times 10 = 100$ pixels of an image and the propositional variables p_1, \dots, p_{100} . On that basis, each image uniquely corresponds to a possible world $w = \{l_1, \dots, l_{100}\}$ (where l_1, \dots, l_{100} are literals), such that, for any $p_i \in \{p_1, \dots, p_{100}\}$, $l_i = p_i$ if the i -th pixel of the image has a value of 1, and $l_i = \neg p_i$ if the i -th pixel of the image has a value of 0.

Table 2

Successive transitions of the binary output Y of the ANN, during training, for two indicative input MNIST images of the testing dataset. As seen by the corresponding labels, Image A represents the digit 0, whereas Image B represents the digit 1. The sum $\sum_{s=1}^{100} |\hat{Y}_s - y_s|$, the loss function \mathcal{L} and the accuracy of the ANN on the testing dataset, during training, are also reported.

Input	1-st Output (\hat{Y}_s)	2-nd Output (\hat{Y}_s)	3-rd Output (\hat{Y}_s)	4-th Output (\hat{Y}_s)	5-th Output (\hat{Y}_s)	Label (y_s)
Image A 	1	1	1	0	0	0
Image B 	0	1	1	1	1	1
$\sum_{s=1}^{100} \hat{Y}_s - y_s $	6	2	2	1	1	—
\mathcal{L}	0.4942	0.3833	0.2953	0.2266	0.1769	—
Accuracy	94%	98%	98%	99%	99%	—

0. Now, let us denote by \mathbb{I} the set of possible worlds corresponding to the images of the training and testing datasets. Moreover, let K_1, K_2, K_3, K_4, K_5 be the belief sets that represent, respectively, the five different Boolean functions f_1, f_2, f_3, f_4, f_5 , to which the binary output Y successively adheres during training, as illustrated in Table 2. Then, for any $i \in \{1, \dots, 5\}$, it is true that:

- $\mathbb{I}_{in}^i = \left\{ \{l_1, \dots, l_{100}\} \in \mathbb{I} : f_i(X_1, \dots, X_{100}) = 1 \right\}$ and $\mathbb{I}_{in}^i \subseteq [K_i]$.
- $\mathbb{I}_{out}^i = \left\{ \{l_1, \dots, l_{100}\} \in \mathbb{I} : f_i(X_1, \dots, X_{100}) = 0 \right\}$ and $\mathbb{I}_{out}^i \cap [K_i] = \emptyset$.

It follows from the above statements that the propositional knowledge satisfying the set \mathbb{I}_{in}^i of possible worlds entails the belief set K_i , whereas the belief set K_i entails the negation of the propositional knowledge satisfying the set \mathbb{I}_{out}^i of possible worlds.

7. Machine learning as a gradual transition of beliefs

The concept of intermediate states of belief, discussed in Section 4 within the context of belief change, can also take meaning in the realm of the training process of ANNs. Indeed, given the binary output Y of a single-output binary ANN, there is a sequence of belief sets K_1, \dots, K_n such that, for any $i \in \{1, \dots, n-1\}$, K_i and K_{i+1} represent, respectively, the Boolean functions of Y right before and right after the i -th update of weights and biases. Hence, K_1 and K_n are the belief sets that represent the Boolean functions of Y *before* and *after* the whole process of training, respectively, and K_2, \dots, K_{n-1} are the *intermediate* belief sets that represent the consecutive Boolean functions of Y *during* training. Obviously then, Remark 12 of Section 5 entails that the training of a binary ANN with multiple outputs leads to a gradual transition of ANN's belief states.

Against this background, and in the presence of the smoothness and monotonicity assumptions of Section 6, we can formulate Theorem 19 that highlights an interesting and natural *relation* of type-B distances between the aforementioned belief sets K_1, \dots, K_n of the single-output binary ANN; the alluded relation is in the spirit of the relation encoded into condition (D) of Subsection 4.2.

Theorem 19. Assume that the training process of a binary ANN with a single output respects the smoothness and monotonicity assumptions. Moreover, let Y be the binary output of the ANN, let K_n be the belief set that represents the Boolean function of Y corresponding to the labels, and let K_1, \dots, K_n be belief sets such that, for any $i \in \{1, \dots, n-1\}$, K_i and K_{i+1} represent, respectively, the Boolean functions of Y right before and right after the i -th update of parameters, that is implemented during training. Then, for any $i, j \in \{1, \dots, n\}$ such that $i < j$,

$$Dist_B(K_i, K_n) \geq Dist_B(K_j, K_n).$$

Proof. Assume that the binary ANN is trained over N individual samples, and let y be the neuron of the output layer that feeds the binary output Y . Let $i, j \in \{1, \dots, n\}$ such that $i < j$, let \hat{y}_s^i, \hat{y}_s^j be the (real-valued) predictions of y for the s -th sample that correspond to the belief sets K_i, K_j , respectively, and let y_s be the label for the s -th sample. Moreover, let \hat{Y}_s^i, \hat{Y}_s^j be the (binary) predictions of Y for the s -th sample that correspond to the belief sets K_i, K_j , respectively.

From condition (S) of Section 6, and the fact that the loss function \mathcal{L} of the ANN is *monotonically related* to the sums $\sum_{s=1}^N |\hat{y}_s^j - y_s|$ and $\sum_{s=1}^N |\hat{y}_s^j - y_s|$, it follows that

$$\sum_{s=1}^N |\hat{y}_s^j - y_s| > \sum_{s=1}^N |\hat{y}_s^j - y_s|.$$

Hence, in view of Remark 15 of Section 6, we derive that

$$\sum_{s=1}^N |\hat{Y}_s^i - y_s| \geq \sum_{s=1}^N |\hat{Y}_s^j - y_s|.$$

Given Remark 16 of Section 6, it is true that $\sum_{s=1}^N |\hat{Y}_s^i - y_s| = \text{Dist}_B(K_i, K_n)$ and $\sum_{s=1}^N |\hat{Y}_s^j - y_s| = \text{Dist}_B(K_j, K_n)$. Combining the above, we deduce that $\text{Dist}_B(K_i, K_n) \geq \text{Dist}_B(K_j, K_n)$, as desired. \square

Example 20 builds upon Example 17 of the previous section, and points out the relation of type-B distances between the symbolic states of belief to which a single-output binary ANN adheres during training, as highlighted in Theorem 19.

Example 20 (*Distances in a binary ANN, cont'd Example 17*). Recall that $K_1, K_2, K_3, K_4, K_5, K_6$ are the belief sets that represent the six different Boolean functions to which the binary output Y of the binary ANN successively adheres during training, and the belief set K_6 represents the Boolean function of Y corresponding to the label. Given that $[K_1] = \{\bar{a}\bar{b}\bar{c}\}$, $[K_2] = \{\bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}\}$, $[K_3] = \{\bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}\}$, $[K_4] = \mathbb{M}$, $[K_5] = \{\bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}\}$ and $[K_6] = \{\bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c}\}$, we have the following list of type-B distances:

$$\text{Dist}_B(K_1, K_6) = 7$$

$$\text{Dist}_B(K_2, K_6) = 6$$

$$\text{Dist}_B(K_3, K_6) = 4$$

$$\text{Dist}_B(K_4, K_6) = 2$$

$$\text{Dist}_B(K_5, K_6) = 1$$

It is obvious that

$$\text{Dist}_B(K_1, K_6) \geq \text{Dist}_B(K_2, K_6) \geq \text{Dist}_B(K_3, K_6) \geq \text{Dist}_B(K_4, K_6) \geq \text{Dist}_B(K_5, K_6),$$

as expected due to Theorem 19.

8. Machine learning is AGM-compatible

As described in Section 6, *backpropagation* is a principal (gradient-based) algorithm used for training feed-forward ANNs. The algorithm iteratively updates the parameters (weights and biases) of the ANN, until its loss function \mathcal{L} is minimized to an acceptable value. As previously noted, during that update of its parameters, a binary ANN passes through a *sequence* of belief states. Focusing on the binary output of a single-output binary ANN, there exists a *sequence* of belief sets that represents the *successive* Boolean functions of that binary output. Against this background, we can formulate Theorem 21, which proves that the transitions between the above-mentioned belief sets of the single-output binary ANN can be modeled/implemented by means of a *single* pair of an AGM revision function and an AGM contraction function.

Theorem 21. *Let Y be the binary output of a single-output binary ANN. Moreover, let K_1, K_2 be belief sets that represent, respectively, the Boolean functions of Y right before and right after an arbitrary update of the parameters of the ANN, that is implemented during its training. There exist an AGM revision function $*$, an AGM contraction function $\dot{-}$, and two sentences $\varphi_1, \varphi_2 \in \mathbb{L}$ depending on K_1, K_2 , such that $K_2 = (K_1 * \varphi_1) \dot{-} \varphi_2$.*

Proof. Let $*$ be an AGM revision function that assigns (via (R)) at every belief set K of the language a faithful preorder \leq_K over \mathbb{M} , specified as follows:

- $\min(\mathbb{M}, \leq_K) = [K]$.
- $r \approx_K r'$, for all $r, r' \notin [K]$.

Hence, for any belief set K , the faithful preorder \leq_K attributes equal plausibility (modulo K) to all possible worlds outside of $[K]$. By the construction of \leq_K and condition (R), it follows that, for any sentence $\chi \in \mathbb{L}$ such that $\neg\chi \in K$ (i.e., $[K] \cap [\chi] = \emptyset$), $[K * \chi] = \min([\chi], \leq_K) = [\chi]$.

Moreover, let \div be the AGM contraction function induced from $*$, via the Harper Identity (HR) of Subsection 3.1. Then, the construction of \leq_K and condition (C) entail that, for any sentence $\chi \in \mathbb{L}$ such that $\chi \in K$ (i.e., $[K] \cap [\neg\chi] = \emptyset$), $[K \div \chi] = [K] \cup \min([\neg\chi], \leq_K) = [K] \cup [\neg\chi]$.

Now, we consider the two cases according to whether the successive belief sets K_1 and K_2 are mutually inconsistent or not, i.e., according to whether $[K_1] \cap [K_2] = \emptyset$ or $[K_1] \cap [K_2] \neq \emptyset$, respectively.

- Assume firstly that $[K_1] \cap [K_2] = \emptyset$. Let φ_1 be a sentence of \mathbb{L} such that $[\varphi_1] = [K_2]$, and let $\varphi_2 = \top$. Since $[K_1] \cap [K_2] = \emptyset$, it follows that $[K_1] \cap [\varphi_1] = \emptyset$. Hence, we derive that $[K_1 * \varphi_1] = [\varphi_1] = [K_2]$; therefore, $K_2 = K_1 * \varphi_1$. Since $\varphi_2 = \top$, we deduce from postulates $(K \div 2)$ & $(K \div 5)$ that $(K_1 * \varphi_1) \div \varphi_2 = K_1 * \varphi_1$, and consequently, $K_2 = (K_1 * \varphi_1) \div \varphi_2$.
- Thereafter, assume that $[K_1] \cap [K_2] \neq \emptyset$. Let φ_1, φ_2 be two sentences of \mathbb{L} such that $[\varphi_1] = [K_1] \cap [K_2]$ and $[\neg\varphi_2] = [K_2] \setminus [K_1]$. Since $[\varphi_1] \subseteq [K_1]$, it follows that the sentence φ_1 is consistent with the belief set K_1 , meaning that $K_1 * \varphi_1 = K_1 + \varphi_1 = Cn(\varphi_1)$. This in turn entails that $[K_1 * \varphi_1] = [K_1] \cap [K_2]$.
Next, since $[K_1 * \varphi_1] \cap [\neg\varphi_2] = \emptyset$, it is true that $\varphi_2 \in K_1 * \varphi_1$. Therefore, $[(K_1 * \varphi_1) \div \varphi_2] = [K_1 * \varphi_1] \cup [\neg\varphi_2] = ([K_1] \cap [K_2]) \cup ([K_2] \setminus [K_1]) = [K_2]$. Consequently, $K_2 = (K_1 * \varphi_1) \div \varphi_2$.

Thus, in any case, there exist an AGM revision function $*$, an AGM contraction function \div , and two sentences $\varphi_1, \varphi_2 \in \mathbb{L}$ that depend on the belief sets K_1, K_2 , such that $K_2 = (K_1 * \varphi_1) \div \varphi_2$, as desired. \square

Some comments on the preceding result are in order. Firstly, it is noteworthy that the AGM revision function $*$ and the AGM contraction function \div of Theorem 21 implement a special type of belief change, called *full-meet* belief change [1,24]. Hansson in [29] emphasizes that full-meet operations are essential in the study of belief dynamics. He argues that full-meet contraction serves not only as a “point of reference”, but also as a foundational component in constructing composite contraction operators, similar to how the basic operation of expansion is crucial in developing more sophisticated revision operators. Theorem 21 reinforces this perspective, by demonstrating that full-meet operations can also act as a building block in effectively modeling the dynamics of the belief sets that represent the (training-dependent) symbolic knowledge of a binary ANN.

Furthermore, it is true that the epistemic inputs φ_1 and φ_2 in the proof of Theorem 21 are specified in terms of the belief sets K_1 and K_2 . Since the “driving force” behind the transformation from K_1 to K_2 is the target labels used during training, it follows that φ_1 and φ_2 fundamentally capture the influence of these labels in effecting the change from K_1 to K_2 . Now, suppose that T is the belief set that represents the Boolean function of the binary ANN of Theorem 21, as specified by the labels of the training dataset. Assuming that the training process of the ANN respects the smoothness and monotonicity assumptions, it follows from Theorem 19 that the type-B distance of the belief set K_2 from T is *smaller* than the type-B distance of the belief set K_1 from T . Roughly speaking, this means that K_2 contains information that is “closer” to the informational content represented by the labels, than K_1 . Therefore, one could argue that φ_1 and φ_2 carry labels-based information that take the symbolic knowledge of the binary ANN “closer” to the target knowledge. Undoubtedly however, further research is essential to gain a deeper and more comprehensive understanding of the relationship between the training labels of the ANN and the epistemic inputs φ_1 and φ_2 .

On a related note, Theorem 21 points out that the sequence of belief sets representing the successive Boolean functions of a single-output binary ANN can be modeled by an *iterative* application of AGM-style change operations (revisions and contractions). While the original AGM framework introduced by Alchourrón, Gärdenfors and Makinson does not address such multi-step changes [1], they have been extensively explored in numerous subsequent studies on iterated belief change, as reviewed by Peppas in his survey [56].

In light of the above contributions, Fig. 6 illustrates the training process of a binary ANN with multiple outputs, represented as a sequence of modifications of belief states S_1, S_2, \dots, S_k , by means of a *single* pair $(*, \div)$ of AGM revision and contraction functions that implement full-meet belief change. As pointed out in Remark 12 of Section 5, each output of the alluded binary ANN can be emulated via a binary ANN with a single output. Modeling the training of binary ANNs using AGM-style operators enables a modular perspective, wherein each individual training step can be independently analyzed and interpreted as a logical transformation. This abstraction not only offers a conceptually appealing view of neural learning, but may also contribute to contexts such as explainability by providing a symbolic trace of how a network’s internal representations evolve throughout training.

This section closes with Example 22, that builds upon Example 20 of the previous section, and points out the AGM-style belief change in the context of a binary ANN, as highlighted in Theorem 21.

Example 22 (Belief change in a binary ANN, cont’d Example 20). Recall that $K_1, K_2, K_3, K_4, K_5, K_6$ are the belief sets that represent the six different Boolean functions to which the binary output Y of the binary ANN successively adheres during training. Let $*$ and \div be

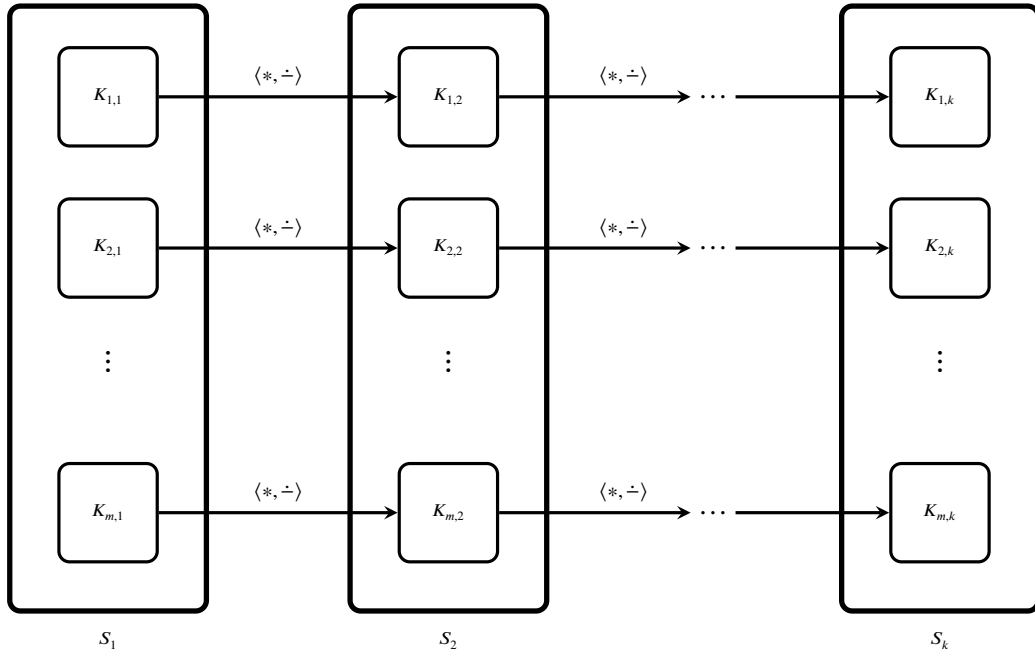


Fig. 6. The training process of a binary ANN with multiple (m) outputs, represented as a sequence of modifications of belief states S_1, S_2, \dots, S_k , by means of a single pair $\langle *, \dot{-} \rangle$ of AGM revision and contraction functions that implement full-meet belief change. Each output of the alluded binary ANN can be emulated via a binary ANN with a single output, whereas, each belief state S_i (with $i \in \{1, \dots, k\}$) is modeled as a tuple of belief sets $K_{1,i}, K_{2,i}, \dots, K_{m,i}$.

the AGM revision and contraction functions, respectively, that implement full-meet belief change.¹² Then, according to Theorem 21, there exist sentences $\varphi_i \in \mathbb{L}$ with $i \in \{1, \dots, 10\}$, such that:

$$\begin{aligned} K_2 &= (K_1 * \varphi_1) \dot{-} \varphi_2 \\ K_3 &= (K_2 * \varphi_3) \dot{-} \varphi_4 \\ K_4 &= (K_3 * \varphi_5) \dot{-} \varphi_6 \\ K_5 &= (K_4 * \varphi_7) \dot{-} \varphi_8 \\ K_6 &= (K_5 * \varphi_9) \dot{-} \varphi_{10} \end{aligned}$$

Based on the proof of Theorem 21, it also follows that:

$$\begin{array}{ll} \varphi_1 = \neg a \wedge \neg b \wedge \neg c & \text{and } \varphi_2 = \neg a \vee \neg b \vee \neg c \\ \varphi_3 = \neg a \wedge \neg b & \text{and } \varphi_4 = \neg c \\ \varphi_5 = \neg b \vee c & \text{and } \varphi_6 = \neg b \vee c \\ \varphi_7 = a \vee b \vee c & \text{and } \varphi_8 = \top \\ \varphi_9 = (a \vee b \vee c) \wedge \neg(a \wedge b \wedge c) & \text{and } \varphi_{10} = \top \end{array}$$

9. Conclusion

In this study, we investigated the statics and dynamics of binary Artificial Neural Networks (ANNs), from the perspective of belief-change theory. A binary ANN is a feed-forward ANN whose inputs and outputs take binary values, and thereby is well-suited for a plethora of practical applications (including, indicatively, image processing and pattern recognition using datasets similar to the benchmark MNIST dataset). For this type of ANNs, we pointed out that their knowledge (expressed via their input-output relationship) can symbolically be represented in terms of a propositional logic language; specifically, by means of a collection (tuple) of belief sets.

¹² Hence, the faithful preorders that $*$ and $\dot{-}$ assign at the belief sets of the language, via conditions (R) and (C), respectively, are in the spirit of the faithful preorder described in the proof of Theorem 21.

Furthermore, in the realm of belief change, we identified the process of changing (revising/contracting) an initial belief set to a modified belief set, as a process of a gradual transition of intermediate belief sets. Such a gradualist approach to belief change has been supported in several studies on human development, and better aligns with the operation of realistic agents. Along these lines, we provided two natural Hamming-based measures of distances between these intermediate belief sets (i.e., type-A and type-B distances), that quantify the difference between their encoded symbolic knowledge.

Following that, we demonstrated that, similar to belief change, the training process of binary ANNs, through backpropagation, can be emulated via a sequence of successive transitions of belief sets, the distance between which is intuitively related through the aforementioned type-B distance. We also proved that the alluded successive alterations of belief sets can be modeled by means of a single pair of an AGM revision function and an AGM contraction function, that both implement full-meet belief change. Thus, we have sketched the process of machine learning (specifically, training binary ANNs) as an operation of AGM-style belief change.

The reported contributions align fundamental machine-learning operations with formalized logic theories. This alignment not only propels forward the understanding of neural computation, but also sets a precedent for future research in the integration of Artificial Intelligence with logical frameworks, paving the way for more interpretable and robust (neuro-symbolic) intelligent systems. Building on this groundwork, future investigation will focus on the following key issues:

- In view of the universal-approximation capabilities of ANNs [33], binary ANNs with sufficient depth and capacity can, in principle, approximate any classification function, including those implemented by more complex and contemporary architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [27]. As future work, we aim to explore such sophisticated ANN architectures through the lens of belief-change theory, examining how the symbolic representations and belief-change processes described in this study can be adequately enriched, adapted and scaled.
- Another critical research direction is the integration of logical constraints directly into the learning process [26]. As noted in the Introduction, this integration could enhance interpretability and robustness by guiding neural models with explicit knowledge, reduce the amount of required training data, and improve generalization. By combining neural and logical methods, we envision architectures where belief-change mechanisms operate alongside standard backpropagation, ensuring logical consistency and domain-specific rule adherence throughout training. Future studies will focus on these challenges, aiming to bridge the gap between purely data-driven approaches and more interpretable, knowledge-guided models.
- Considering that full-meet belief change has been criticized for not adhering to Parikh's notion of relevance [53,54,58] and to Darwiche and Pearl's approach for iterated belief change [19], it is valuable to explore alternative types of AGM-style operators that can effectively model the training of ANNs.
- As highlighted in the previous section regarding Theorem 21, further research is crucial for gaining a deeper understanding of the relationship between the training labels of a binary ANN and the epistemic inputs φ_1 and φ_2 .
- For the training of binary ANNs, we primarily employed datasets that fully characterize the corresponding input-output relationships (see the running Example 17). Future research will focus on scenarios involving incomplete datasets, exploring how partial data affects the learning process.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The author expresses his gratitude to the late Pavlos Peppas for the fruitful discussions on the topic of this study. His insights and contributions were invaluable, and his presence will be deeply missed. This work is dedicated to his memory. The author also thanks the anonymous reviewers for their constructive feedback on previous versions of this article.

Data availability

No data was used for the research described in the article.

References

- [1] Carlos Alchourrón, Peter Gärdenfors, David Makinson, On the logic of theory change: partial meet contraction and revision functions, *J. Symb. Log.* 50 (2) (1985) 510–530.
- [2] Robert Andrews, Joachim Diederich, Alan B. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, *Knowl.-Based Syst.* 8 (1995) 373–389.
- [3] Theofanis Aravanis, Generalizing Parikh's criterion for relevance-sensitive belief revision, *ACM Trans. Comput. Log.* 24 (18) (2023) 1–29.
- [4] Theofanis I. Aravanis, Collective belief revision, *J. Artif. Intell. Res.* 78 (2023) 1221–1247.
- [5] Theofanis I. Aravanis, Pavlos Peppas, Mary-Anne Williams, Incompatibilities between iterated and relevance-sensitive belief revision, *J. Artif. Intell. Res.* 68 (2020) 85–108.
- [6] Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche, Jean-Marie Lagniez, Pierre Marquis, On the computational intelligibility of Boolean classifiers, in: *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR 2021)*, Special Session on KR and Machine Learning, 2021, pp. 74–86.

- [7] Alexandru Baltag, Nina Gierasimczuk, Sonja Smets, Truth-tracking by belief revision, *Stud. Log.* 107 (2019) 917–947.
- [8] Alexandru Baltag, Sonja Smets, Keep changing your beliefs, aiming for the truth, *Erkenntnis* 75 (2011) 255–270.
- [9] Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 2006.
- [10] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [11] Alexander Bochman, *A Logical Theory of Nonmonotonic Inference and Belief Change*, Springer, 2001.
- [12] Richard Booth, Thomas Meyer, How to revise a total preorder, *J. Philos. Log.* 40 (2011) 193–238.
- [13] Ronald Brachman, Hector Levesque, *Knowledge Representation and Reasoning*, Morgan Kaufmann, 2004.
- [14] Thomas Caridroit, Sébastien Konieczny, Pierre Marquis, Contraction in propositional logic, *Int. J. Approx. Reason.* 80 (2017) 428–442.
- [15] Arthur Choi, Weijia Shi, Andy Shih, Adnan Darwiche, Compiling neural networks into tractable Boolean circuits, in: *AAAI Spring Symposium on Verification of Neural Networks (VNN)*, 2019.
- [16] Sylvie Coste-Marquis, Pierre Marquis, On belief change for multi-label classifier encodings, in: *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021)*, 2021, pp. 1829–1836.
- [17] Mukesh Dalal, Investigations into theory of knowledge base revision: preliminary report, in: *Proceedings of the 7th National Conference of the American Association for Artificial Intelligence (AAAI 1988)*, 1988, pp. 475–479.
- [18] Adnan Darwiche, Auguste Hirth, On the reasons behind decisions, in: *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020)*, 2020, pp. 712–720.
- [19] Adnan Darwiche, Judea Pearl, On the logic of iterated belief revision, *Artif. Intell.* 89 (1997) 1–29.
- [20] Artur S. d'Ávila Garcez, Luís C. Lamb, Neurosymbolic AI: the 3rd wave, *Artif. Intell. Rev.* 56 (2023) 12387–12406.
- [21] Artur S. d'Ávila Garcez, Luís C. Lamb, Dov M. Gabbay, *Neural-Symbolic Cognitive Reasoning*, Springer, 2009.
- [22] Eduardo Fermé, Sven Ove Hansson, Selective revision, *Stud. Log.* 63 (1999) 331–342.
- [23] Eduardo Fermé, Sven Ove Hansson, *Belief Change: Introduction and Overview*, Springer International Publishing, 2018.
- [24] Peter Gärdenfors, *Knowledge in Flux – Modeling the Dynamics of Epistemic States*, MIT Press, Cambridge, Massachusetts, 1988.
- [25] Peter Gärdenfors, David Makinson, Revisions of knowledge systems using epistemic entrenchment, in: *Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning About Knowledge (TARK 1988)*, Morgan Kaufmann, Pacific Grove, California, 1988, pp. 83–95.
- [26] Eleonora Giunchiglia, Mihaela Catalina Stoian, Thomas Lukasiewicz, Deep learning with logical constraints, in: *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI 2022)*, Survey Track, 2022, pp. 5478–5485.
- [27] Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, The MIT Press, 2016.
- [28] Sven Ove Hansson, Kernel contraction, *J. Symb. Log.* 59 (1994) 845–859.
- [29] Sven Ove Hansson, In praise of full meet contraction, *Anal. Filos.* 26 (2006) 134–146.
- [30] Sven Ove Hansson, Eduardo Fermé, John Cantwell, Marcelo A. Falappa, Credibility limited revision, *J. Symb. Log.* 66 (4) (2001) 1581–1596.
- [31] William L. Harper, Rational conceptual change, in: *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 1977, pp. 462–494.
- [32] Simon Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, 1994.
- [33] Kurt Hornik, Maxwell Stinchcombe, Halbert White, Multilayer feedforward networks are universal approximators, *Neural Netw.* 2 (1989) 359–366.
- [34] Alexey Ignatiev, Joao Marques-Silva, SAT-based rigorous explanations for decision lists, in: Chu-Min Li, Felip Manyà (Eds.), *Theory and Applications of Satisfiability Testing – SAT 2021*, in: *Lecture Notes in Computer Science*, Springer, 2021, pp. 251–269.
- [35] Yacine Izza, Alexey Ignatiev, Joao Marques-Silva, On explaining decision trees, *arXiv:2010.11034*, 2020.
- [36] Daniel Kahneman, *Thinking, Fast and Slow*, Farrar, Straus and Giroux, 2013.
- [37] Katsuno Hirofumi, Alberto Mendelzon, Propositional knowledge base revision and minimal change, *Artif. Intell.* 52 (3) (1991) 263–294.
- [38] Katsuno Hirofumi, Alberto Mendelzon, On the difference between updating a knowledge base and revising it, in: Peter Gärdenfors (Ed.), *Belief Revision*, Cambridge University Press, 1992, pp. 183–203.
- [39] Kevin T. Kelly, The learning power of belief revision, in: *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 1998)*, ACM, 1998, pp. 111–124.
- [40] Sébastien Konieczny, Mattia Medina Grespan, Ramón Pino Pérez, Taxonomy of improvement operators and the problem of minimal change, in: *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR 2010)*, 2010, pp. 161–170.
- [41] Sébastien Konieczny, Ramón Pino Pérez, Logic based merging, *J. Philos. Log.* 40 (2011) 239–270.
- [42] Mark A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, *AIChE J.* 37 (2) (1991) 233–243.
- [43] Mark A. Kramer, Autoassociative neural networks, *Comput. Chem. Eng.* 16 (4) (1992) 313–328.
- [44] Ernst Kussul, Tatiana Baidyk, Improved method of handwritten digit recognition tested on MNIST database, *Image Vis. Comput.* 22 (2004) 971–981.
- [45] Guillaume Lample, François Charton, Deep learning for symbolic mathematics, in: *International Conference on Learning Representations (ICLR 2020)*, 2020.
- [46] Joseph R. Lao, Jason Young, *Resistance to Belief Change*, Routledge, 2019.
- [47] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Deep learning, *Nature* 521 (2015) 436–444.
- [48] Yann LeCun, Corinna Cortes, *The MNIST database of handwritten digits*, <http://yann.lecun.com/exdb/mnist/index.html>, 1998.
- [49] Isaac Levi, *The Enterprise of Knowledge: An Essay on Knowledge, Credal Probability and Chance*, MIT Press, 1980.
- [50] Rolf Morel, Andrew Cropper, Learning logic programs by explaining their failures, *Mach. Learn.* 112 (2023) 3917–3943.
- [51] Nina Narodytska, Shiva Kasiviswanathan, Leonid Ryzhyk, Mooly Sagiv, Toby Walsh, Verifying properties of binarized deep neural networks, in: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, 2018, pp. 6615–6624.
- [52] Luigi Nele, Giulio Mattered, Emily W. Yap, Mario Vozza, Silvestro Vespoli, Towards the application of machine learning in digital twin technology: a multi-scale review, *Discov. Appl. Sci.* 6 (502) (2024).
- [53] Rohit Parikh, Beliefs, belief revision, and splitting languages, in: Lawrence S. Moss, Jonathan Ginzburg, Maarten de Rijke (Eds.), *Logic, Language and Computation*, vol. 2, CSLI Publications, 1999, pp. 266–278.
- [54] Rohit Parikh, Beth definability, interpolation and language splitting, *Synthese* 179 (2011) 211–221.
- [55] Pavlos Peppas, Belief revision, in: Frank van Harmelen, Vladimir Lifschitz, Bruce Porter (Eds.), *Handbook of Knowledge Representation*, Elsevier Science, 2008, pp. 317–359.
- [56] Pavlos Peppas, A panorama of iterated revision, in: Sven Ove Hansson (Ed.), *David Makinson on Classical Methods for Non-Classical Problems*, Springer, Netherlands, 2014, pp. 71–94.
- [57] Pavlos Peppas, Mary-Anne Williams, Grigoris Antoniou, Revision operators with compact representations, *Artif. Intell.* 329 (2024).
- [58] Pavlos Peppas, Mary-Anne Williams, Samir Chopra, Norman Foo, Relevance in belief revision, *Artif. Intell.* 229 (2015) 126–138.
- [59] Jean Piaget, The theory of stages in cognitive development, in: Donald Ross Green, Marguerite Prentice Ford, George Browning Flamer (Eds.), *Proceedings of the CTB/McGraw-Hill Conference on Ordinal Scales of Cognitive Development*, McGraw-Hill, New York, 1971.
- [60] John C. Platt, Using analytic QP and sparseness to speed training of support vector machines, in: *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems*, 1999, pp. 557–563.
- [61] M. Mazhar Rathore, Syed Attique Shah, Dharendra Shukla, Spiridon Bakiras, The role of AI, Machine Learning, and Big Data in Digital Twinning: a systematic literature review, challenges, and opportunities, *IEEE Access* 9 (2021) 32030–32052.
- [62] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams, Learning representations by back-propagating errors, *Nature* 323 (1986) 533–536.

- [63] Nicolas Schwind, Katsumi Inoue, Pierre Marquis, Editing Boolean classifiers: a belief change perspective, in: The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2023), 2023, pp. 6516–6524.
- [64] Nicolas Schwind, Sébastien Konieczny, Pierre Marquis, On belief promotion, in: Proceedings of the Sixteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2018), 2018, pp. 297–306.
- [65] Eric Schwitzgebel, Gradual belief change in children, *Hum. Dev.* 42 (1999) 283–296.
- [66] Paulo Shakarian, Chitta Baral, Gerardo I. Simari, Bowen Xi, Lahari Pokala, *Neuro Symbolic Reasoning and Learning*, Springer, 2023.
- [67] Weijia Shi, Andy Shih, Adnan Darwiche, Arthur Choi, On tractable representations of binary neural networks, in: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020), Special Session on KR and Machine Learning, 2020, pp. 882–892.
- [68] Wolfgang Spohn, Ordinal conditional functions: a dynamic theory of epistemic states, in: William L. Harper, Brian Skyrms (Eds.), *Causation in Decision, Belief Change, and Statistics*, in: The University of Western Ontario Series in Philosophy of Science, vol. 42, Springer, Netherlands, 1988, pp. 105–134.
- [69] Geoffrey G. Towell, Jude W. Shavlik, Knowledge-based artificial neural networks, *Artif. Intell.* 70 (1994).
- [70] Son N. Tran, Artur S. d’Ávila Garcez, Deep logic networks: inserting and extracting knowledge from deep belief networks, *IEEE Trans. Neural Netw. Learn. Syst.* 29 (2016) 246–258.
- [71] Frank van Harmelen, Vladimir Lifschitz, Bruce Porter, *Handbook of Knowledge Representation*, 1st edition, Elsevier, 2008.
- [72] Stella Vosniadou, Capturing and modeling the process of conceptual change, *Learn. Instr.* 4 (1994) 45–69.
- [73] Renata Wassermann, Resource-bounded belief revision, *Erkenntnis* 50 (1999) 429–446.