Analyzing Differentiable Fuzzy Implications

Emile van Krieken , Erman Acar , Frank van Harmelen
Vrije Universiteit Amsterdam,
De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands
{e.van.krieken, Erman.Acar, Frank.van.Harmelen}@vu.nl

Abstract

Combining symbolic and neural approaches has gained considerable attention in the AI community, as it is argued that their strengths and weaknesses are complementary. One trend in the literature are weakly supervised learning techniques that employ operators from fuzzy logics. They use prior background knowledge described in such logics to help training neural networks from unlabeled and noisy data. By interpreting logical symbols using neural networks (or *grounding* them), this background knowledge can be added to regular loss functions, hence making reasoning a part of learning. We investigate how implications from the fuzzy logic literature behave in a differentiable setting. In this setting, we analyze the differences between the formal properties of these fuzzy implications. It turns out that various fuzzy implications, including some of the most well-known, are highly unsuitable for use in a differentiable learning setting. A further finding shows a strong imbalance between gradients driven by the antecedent and the consequent of the implication. Furthermore, we introduce a new family of fuzzy implications (called sigmoidal implications) to tackle this phenomenon. Finally, we empirically show that it is possible to use Differentiable Fuzzy Logics for semi-supervised learning, and show that sigmoidal implications outperform other choices of fuzzy implications.

1 Introduction

In recent years, integrating symbolic and statistical approaches to AI gained considerable attention (Garcez, Broda, and Gabbay 2012; Besold et al. 2017). This research line has gained further traction due to recent influential critiques on purely statistical deep learning (Marcus 2018; Pearl 2018). While deep learning has brought many important breakthroughs (Brock, Donahue, and Simonyan 2018; Radford et al. 2019; Silver et al. 2017), there are concerns about the massive amounts of data that models need to learn even a simple concept. In contrast, traditional symbolic AI could easily reuse concepts e.g., using only a single logical statement one can already express domain knowledge conveniently.

However, symbolic AI also has its weaknesses. One is scalability: dealing with large amounts of data while performing complex reasoning tasks. Another is not being able to deal with the noise and ambiguity of e.g. sensory data. The latter is also related to the well-known *symbol grounding problem* which (Harnad 1990) defines as how "the semantic interpretation of a formal symbol system can be made intrinsic to the system, rather than just parasitic on the meanings in our heads". In particular, symbols refer to concepts that have an intrinsic meaning to us humans, but computers manipulating these symbols cannot *understand* (or *ground*) this meaning. On the other hand, a properly trained deep learning model excels at modeling complex sensory data. Therefore, several recent approaches (Diligenti, Roychowdhury, and Gori 2017; Garnelo, Arulkumaran, and Shanahan 2016; Serafini and Garcez 2016; Manhaeve et al. 2018; Evans and Grefenstette 2018) interpret symbols that are used in logic-based systems using deep learning models. These implement (Harnad 1990) "a hybrid nonsymbolic/symbolic system [...] in which the elementary symbols are grounded in [...] nonsymbolic representations that pick out, from their proximal sensory projections, the distal object categories to which the elementary symbols refer."

In this article, we introduce *Differentiable Fuzzy Logics* (DFL) which aims to integrate reasoning and learning by using logical formulas expressing background knowledge. In order to ensure loss functions are differentiable, DFL uses fuzzy logic semantics (Klir and Yuan 1995). Moreover, predicate, function and constant symbols are interpreted using a deep learning model. By maximizing the degree of truth of the background knowledge using gradient descent, both learning and reasoning are performed in parallel. The loss function can be used for weakly supervised learning (Zhou 2017), like detecting noisy or inaccurate supervision (Donadello, Serafini, and Garcez 2017), or semi-supervised learning (Xu et al. 2018; Hu et al. 2016). For such problems, DFL corrects the predictions of the deep learning model when it is logically inconsistent with the background knowledge.

Next, we present an analysis of the choice of fuzzy implication. A fuzzy implication generalizes the Boolean implication, and it is usually differentiable, which enables its use in DFL. Interestingly, the derivatives of the implications determine how DFL corrects the deep learning model when its predictions are inconsistent with the background knowledge. We show that the qualitative properties of these derivatives are integral to both the theory and practice of DFL.

More specifically, the main contribution of this article is to answer the following question: *Which fuzzy logic implications have convenient theoretical properties when using them in gradient descent?* To this end,

- we introduce several known implications from fuzzy logic (Section 2) and the framework of Differentiable Fuzzy Logics (Section 3) that uses these implications;
- we analyze the theoretical properties of fuzzy implications and introduce a new family of fuzzy implications called sigmoidal implications (Section 4);
- we perform experiments to compare fuzzy implications in a semi-supervised experiment (Section 5).
- we conclude with several recommendations for choices of fuzzy implications.

2 Background

We will denote predicates using $\mathsf{cushion}$, variables by $x, y, z, x_1, ...$ and objects by $o_1, o_2, ...,$. For convenience, we will be limiting ourselves to function-free formulas in prenex normal form. Formulas in *prenex normal form* start with quantifiers followed by a quantifier-free subformula. An *atom* is $\mathsf{P}(t_1, ..., t_m)$ where $t_1, ..., t_m$ are terms. If $t_1, ..., t_m$ are all constants, we say it is a *ground atom*.

Fuzzy logic is a real-valued logic where truth values are real numbers in $[0, 1]$ where 0 denotes completely false and 1 denotes completely true. We will be looking at predicate fuzzy logics in particular, which extend propositional fuzzy logics with universal and existential quantification. In this text, we limit ourselves to the classic fuzzy negation $N(a) = 1 - a$.

To properly introduce fuzzy implications, we require the notions of t-norms that generalize boolean conjunction, and t-conorms that generalize boolean disjunction. A *t-norm* is a function $T : [0, 1]^2 \to [0, 1]$ that is commutative, associative, increasing, and for all $a \in [0, 1]$, $T(1, a) = a$. A *t-conorm* is a function $S : [0, 1]^2 \to [0, 1]$ that is commutative, associative, increasing, and for all $a \in [0, 1]$, $S(0, a) = a$. T-conorms are constructed from a t-norm using $S(a, b) = 1 - T(1 - a, 1 - b)$.

Fuzzy implications are used to compute the truth value of $p \to q$. p is called the *antecedent* and q the *consequent* of the implication. We follow (Jayaram and Baczynski 2008) and refer to it for details and proofs.

Definition 1. *A* fuzzy implication *is a function* $I : [0, 1]^2 \to [0, 1]$ *so that for all* $a, c \in [0, 1]$*,* $I(\cdot, c)$ *is decreasing,* $I(a, \cdot)$ *is increasing and for which* $I(0, 0) = 1$*,* $I(1, 1) = 1$ *and* $I(1, 0) = 0$.

From this definition follows that $I(0, 1) = 1$. We next introduce several optional properties of fuzzy implications that we will use in our analysis.

Definition 2. *A fuzzy implication I satisfies*

1. *left-neutrality (LN) if for all* $c \in [0, 1]$*,* $I(1, c) = c$ *(generalizes* $(1 \to p) \equiv p$*);*
2. *the exchange principle (EP) if for all* $a, b, c \in [0, 1]$*,* $I(a, I(b, c)) = I(b, I(a, c))$ *(generalizes* $p \to (q \to r) \equiv q \to (p \to r)$*);*
3. *the identity principle (IP) if for all* $a \in [0, 1]$*,* $I(a, a) = 1$ *(generalizes the tautology* $p \to p$*);*
4. *contrapositive symmetry (CP) if for all* $a, c \in [0, 1]$*,* $I(a, c) = I(1 - c, 1 - a)$ *(generalizes* $p \to q \equiv \neg q \to \neg p$*);*
5. *left-contrapositive symmetry (L-CP) if for all* $a, c \in [0, 1]$*,* $I(1 - a, c) = I(1 - c, a)$ *(generalizes* $\neg p \to q \equiv \neg q \to p$*);*
6. *right-contrapositive symmetry (R-CP) if for all* $a, c \in [0, 1]$*,* $I(a, 1 - c) = I(c, 1 - a)$ *(generalizes* $p \to \neg q \equiv q \to \neg p$*).*

R-Implications Using a common construction, we find R-implications. They are the standard choice for implication in t-norm fuzzy logics.

Definition 3. *Let T be a t-norm. The function* $I_T : [0, 1]^2 \to [0, 1]$ *is called an* R-implication *and defined as* $I_T(a, c) = \sup\{b \in [0, 1]|T(a, b) \leq c\}$.

The *supremum* of a set A, denoted $\sup\{A\}$, is the lowest upper bound of A. All R-implications are fuzzy implications, and all satisfy LN, IP and EP. Note that if $a \leq c$ then $I_T(a, c) = 1$.

S-Implications In classical logic, the (material) implication is defined using $p \to q = \neg p \vee q$. Generalizing this definition, we can use a t-conorm S to construct a fuzzy implication.

Definition 4. *Let S be a t-conorm. The function* $I_S : [0, 1]^2 \to [0, 1]$ *is called an* S-implication *and is defined for all* $a, c \in [0, 1]$ *as* $I_S(a, c) = S(1 - a, c)$.

All S-implications I_S are fuzzy implications and satisfy every property from Definition 2 but IP.

Table 1 shows some common differentiable S-implications and R-implications.

3 Differentiable Fuzzy Logics

Differentiable Fuzzy Logics (DFL) are fuzzy logics with differentiable connectives for which differentiable loss functions can be constructed that represent logical formulas. Examples of logics in this family (Serafini and Garcez 2016; Marra et al. 2019; Diligenti, Roychowdhury, and Gori 2017; Marra et al. 2018; Guo et al. 2016) will be discussed in Section 6. They use background knowledge to deduce the truth value of statements in unlabeled or poorly labeled data to be able to use such data during learning. This can be beneficial as unlabeled, poorly labeled and partially labeled data is cheaper and easier to come by.

We motivate the use of DFL with the following scenario: Assume we have an agent M whose goal is to describe the scene on an image. It gets feedback from a supervisor S, who does not have an exact description of these images available. However, S does have a background knowledge base $\mathcal{K}$, encoded in some logical formalism, about the concepts contained on the images. The intuition behind DFL is that S can correct M's descriptions of scenes when they are not consistent with its knowledge base $\mathcal{K}$.

Name	Associated t-norm	Equation	Properties
Kleene-Dienes	Gödel	$I_{KD}(a,c) = \max(1-a,c)$	All but IP, S-implication
Reichenbach	Product	$I_{RC}(a,c) = 1-a+a\cdot c$	All but IP, S-implication
Łukasiewicz	Łukasiewicz	$I_{LK}(a,c) = \min(1-a+c,1)$	All, S-implication, R-implication
Gödel	Gödel	$I_G(a,c) = \begin{cases} 1, & \text{if } a \leq c \\ c, & \text{otherwise} \end{cases}$	LN, EP, IP, R-implication
Goguen	Product	$I_{GG}(a,c) = \begin{cases} 1, & \text{if } a \leq c \\ \frac{c}{a}, & \text{otherwise} \end{cases}$	LN, EP, IP, R-implication

Table 1: Some common differentiable implications.

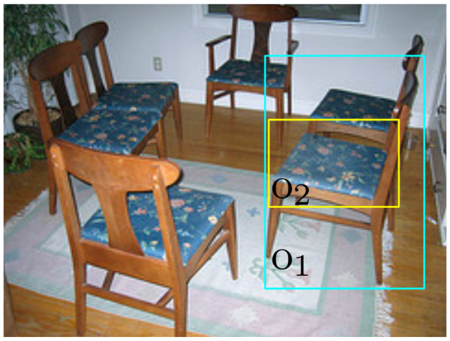

Figure 1: In this running example, we have an image with two objects on it, o_1 and o_2.

Example 1. *'Agent M has to describe the image I in Figure 1 containing two objects, o_1 and o_2. M and the supervisor S only know about the unary class predicates $\{\textsf{chair}, \textsf{cushion}, \textsf{armRest}\}$ and the binary predicate $\{\textsf{partOf}\}$. Since S does not have a description of I, it will have to correct M based on the knowledge base $\mathcal{K}$. M describes the image as follows, where the probability indicates the confidence in an observation:*

$$\begin{aligned} p(\textsf{chair}(o_1)) &= 0.9 & p(\textsf{chair}(o_2)) &= 0.4 \\ p(\textsf{cushion}(o_1)) &= 0.05 & p(\textsf{cushion}(o_2)) &= 0.5 \\ p(\textsf{armRest}(o_1)) &= 0.05 & p(\textsf{armRest}(o_2)) &= 0.1 \\ p(\textsf{partOf}(o_1,o_1)) &= 0.001 & p(\textsf{partOf}(o_2,o_2)) &= 0.001 \\ p(\textsf{partOf}(o_1,o_2)) &= 0.01 & p(\textsf{partOf}(o_2,o_1)) &= 0.95 \end{aligned}$$

Suppose that $\mathcal{K}$ contains the following logic formula which says objects that are a part of a chair are either cushions or armrests:

$$\forall x,y\ \textsf{chair}(x) \wedge \textsf{partOf}(y,x) \rightarrow \textsf{cushion}(y) \vee \textsf{armRest}(y).$$

S might now reason that since M is relatively confident of $\textsf{chair}(o_1)$ and $\textsf{partOf}(o_2,o_1)$ that the antecedent of this formula is satisfied, and thus $\textsf{cushion}(o_2)$ or $\textsf{armRest}(o_2)$ has to hold. Since $p(\textsf{cushion}(o_2)|I,o_2) > p(\textsf{armRest}(o_2)|I,o_2)$, a possible correction would be to tell M to increase its degree of belief in $\textsf{cushion}(o_2)$.

We would like to automate the kind of supervision S performs in the previous example. Therefore, we next formally introduce DFL, in which truth values of ground atoms are in $[0,1]$, and logical connectives are interpreted using fuzzy operators.

DFL defines a new semantics using vector embeddings and functions on such vectors in place of classical semantics. In classical logic, a *structure* consists of a domain of discourse and an interpretation function, and is used to give meaning to the predicates. Similarly, in DFL a structure consists of a probability distribution defined on an embedding space and an *embedded interpretation*:

Definition 5. *A* Differentiable Fuzzy Logics structure *is a tuple $\langle p, \eta_{\boldsymbol{\theta}}\rangle$, where p is a* domain distribution *over d-dimensional objects $o \in \mathbb{R}^d$ whose* domain of discourse *is the* support *of p (i.e., $O = \text{supp}(p) = \{o|p(o) > 0, o \in \mathbb{R}^d\}$), and $\eta_{\boldsymbol{\theta}}$ is an (*embedded*)* interpretation *under $\boldsymbol{\theta}$ (a W-dimensional real vector, also called* parameters*) which maps every predicate symbol $\textsf{P} \in \mathcal{P}$ with arity α to a function that associates α objects to an element in $[0,1]$ (i.e., $\eta_{\boldsymbol{\theta}}(\textsf{P}) : O^\alpha \rightarrow [0,1]$).*

To address the *symbol grounding problem* (Harnad 1990), objects in the domain of discourse are d-dimensional vectors of reals. Their semantics come from the underlying semantics of the vector space as terms are interpreted in a real (valued) world (Serafini and Garcez 2016). Predicates are interpreted as functions mapping these vectors to a fuzzy truth value. Embedded interpretations are implemented using a neural network model with trainable network parameters $\boldsymbol{\theta}$. Different values of $\boldsymbol{\theta}$ will produce different embedded interpretations $\eta_{\boldsymbol{\theta}}$. The domain distribution is used to limit the size of the vector space. For example, p might be the distribution over images representing only the natural images.

Next, we define how to compute the truth value of formulas of DFL, which generalizes the computation of Real Logic (Serafini and Garcez 2016). An *aggregation operator* is a function $A : [0,1]^n \rightarrow [0,1]$ that is symmetric and increasing with respect to each argument, and for which $A(0,...,0) = 0$ $A(1,...,1) = 1$. A *variable assignment* μ maps variable symbols x to objects $o \in O$. $\mu(x)$ retrieves the object $o \in O$ assigned to x in μ.

Definition 6. *Let $\langle p, \eta_{\boldsymbol{\theta}}\rangle$ be a DFL structure, T a t-norm, S a t-conorm, I a fuzzy implication and A an aggregation operator. Then the* valuation function *$e_{\eta_{\boldsymbol{\theta}},p,T,S,I,A}$ (or, for brevity, $e_{\boldsymbol{\theta}}$) computes the truth value of a formula φ in $\mathcal{L}$ given a variable assignment μ. It is defined inductively as* *follows:*

$$e_{\boldsymbol{\theta}}\left(P(x_1, ..., x_m)\right) = \eta_{\boldsymbol{\theta}}(P)\left(\mu(x_1), ..., \mu(x_m)\right) \tag{1}$$

$$e_{\boldsymbol{\theta}}(\neg\phi) = 1 - e_{\boldsymbol{\theta}}(\phi) \tag{2}$$

$$e_{\boldsymbol{\theta}}(\phi \wedge \psi) = T(e_{\boldsymbol{\theta}}(\phi), e_{\boldsymbol{\theta}}(\psi)) \tag{3}$$

$$e_{\boldsymbol{\theta}}(\phi \vee \psi) = S(e_{\boldsymbol{\theta}}(\phi), e_{\boldsymbol{\theta}}(\psi)) \tag{4}$$

$$e_{\boldsymbol{\theta}}(\phi \rightarrow \psi) = I(e_{\boldsymbol{\theta}}(\phi), e_{\boldsymbol{\theta}}(\psi)) \tag{5}$$

$$e_{\boldsymbol{\theta}}(\forall x\ \phi) = A_{o\in O} e_{\boldsymbol{\theta}}(\phi), \textit{with } x \textit{ assigned to } o \textit{ in } \mu. \tag{6}$$

Equation 1 defines the fuzzy truth value of an atomic formula. μ finds the objects assigned to the terms $x_1, ..., x_m$ resulting in a list of d-dimensional vectors. These are the inputs to the interpretation of the predicate symbol $\eta_{\boldsymbol{\theta}}(\mathsf{P})$ to get a fuzzy truth value. Equations 2 - 5 define the truth values of the connectives using the operators T, S and I.

Equation 6 defines the truth value of universally quantified formulas $\forall x\ \phi$. This is done by enumerating the domain of discourse $o \in O$, computing the truth value of ϕ with o assigned to x in μ, and combining the truth values using an aggregation operator A. When enumerating the objects is not viable, we can choose to sample a batch of objects to approximate the computation of the valuation.

It is commonly assumed in Machine Learning (Goodfellow et al. 2016)(p.109) that a dataset contains independent samples from the domain distribution p and thus using such samples approximates sampling from p. Unfortunately, by relaxing quantifiers in this way we lose soundness of the logic.

In DFL, the parameters $\boldsymbol{\theta}$ are learned using *fuzzy maximum satisfiability* (Donadello, Serafini, and Garcez 2017), which finds parameters that maximize the valuation of the knowledge base $\mathcal{K}$.

Definition 7. *Let $\mathcal{K}$ be a knowledge base of formulas, $\langle p, \eta_{\boldsymbol{\theta}}\rangle$ a DFL structure for the predicate symbols in $\mathcal{K}$ and $e_{\eta_{\boldsymbol{\theta}},p,T,S,I,A}$ a valuation function. Then the* Differentiable Fuzzy Logics loss $\mathcal{L}_{DFL}$ *of a knowledge base of formulas $\mathcal{K}$ is computed using*

$$\mathcal{L}_{DFL}(\boldsymbol{\theta}; O, \mathcal{K}) = -w_{DFL} \sum_{\varphi\in\mathcal{K}} \cdot e_{\eta_{\boldsymbol{\theta}},p,T,S,I,A}(\varphi), \tag{7}$$

where w_{DFL} is a weight for this loss. The fuzzy maximum satisfiability problem *is the problem of finding parameters $\boldsymbol{\theta}^*$ that minimize Equation 7:*

$$\boldsymbol{\theta}^* = \mathit{argmin}_{\boldsymbol{\theta}}\ \mathcal{L}_{DFL}(\boldsymbol{\theta}; O, \mathcal{K}). \tag{8}$$

This optimization problem can be solved using a gradient descent method. If the operators T, S, I and A are all differentiable, we can repeatedly apply the chain rule, i.e. reverse-mode differentiation, on the DFL loss $\mathcal{L}_{DFL}(\boldsymbol{\theta}_n; O, \mathcal{K})$, $n = 0, ..., N$. This procedure finds the derivative with respect to the truth values of the ground atoms $\frac{\partial\mathcal{L}_{DFL}(\boldsymbol{\theta}_n; O, \mathcal{K})}{\partial\eta_{\boldsymbol{\theta}_n}(\mathsf{P})(o_1, ..., o_m)}$. We can use these partial derivatives to update the parameters $\boldsymbol{\theta}_n$ using the chain rule, resulting in a different embedded interpretation $\eta_{\boldsymbol{\theta}_{n+1}}$.

One particularly interesting property of Differentiable Fuzzy Logics is that the partial derivatives of the subformulas with respect to the satisfaction of the knowledge base have a somewhat explainable meaning. For example, turning back to Example 1, the computed partial derivatives reflect whether we should increase $p(\mathsf{cushion}(o_2))$, that is, increase the agents belief in $\mathsf{cushion}(o_2)$.

4 Differentiable Fuzzy Implications

A significant proportion of background knowledge is written as universally quantified implications of the form $\forall x\ \phi(x) \rightarrow \psi(x)$, like 'all humans are mortal'.

The implication is used in two well known rules of inference. *Modus ponens* inference says that if $\forall x\ \phi(x) \rightarrow \psi(x)$ and we know that $\phi(x)$, then also $\psi(x)$. *Modus tollens* inference says that if $\forall x\ \phi(x) \rightarrow \psi(x)$ and we know that $\neg\psi(x)$, then also $\neg\phi(x)$, as if $\phi(x)$ were true, $\psi(x)$ should also have been.

When the learning agent predicts a scene in which an implication is false, the supervisor has multiple choices to correct it. Consider the implication 'all ravens are black'. There are 4 categories for this formula: *black ravens* (BR), *non-black non-ravens* (NBNR), *black non-ravens* (BNR) and *non-black ravens* (NBR). Assume our agent observes an NBR, which is inconsistent with the background knowledge. There are four options to consider.

1. *Modus Ponens* (MP): The antecedent is true, so by modus ponens, the consequent is also true. We trust the agent's observation of a raven and believe it was a black raven (BR).
2. *Modus Tollens* (MT): The consequent is false, so by modus tollens, the antecedent is also false. We trust the agent's observation of a non-black object and believe it was not a raven (NBNR).
3. *Distrust*: We believe the agent is wrong both about observing a raven and a non-black object and it was a black object which is non-raven (BNR).
4. *Exception*: We trust the agent and ignore the fact that its observation goes against the background knowledge. Hence, it has to be a non-black raven (NBR).

The distrust option seems somewhat useless. The exception option can be correct, but we cannot know when there is an exception from the agent's observations alone. In such cases, DFL would not be very useful since it would not teach the agent anything new.

We can safely assume that there are far more non-black objects which are not ravens than there are ravens. Thus, from a statistical perspective, it is most likely that the agent observed an NBNR. This shows the imbalance associated with the implication, which was first noted in (van Krieken, Acar, and van Harmelen 2019) for the Reichenbach implication. It is quite similar to the *class imbalance problem* in Machine Learning (Japkowicz and Stephen 2002) in that the real world has far more 'negative' (or *contrapositive*) examples than positive examples of the background knowledge.

This problem is closely related to the Raven paradox (Hempel 1945; Vranas 2004) from the field of confirmation theory which ponders what evidence can confirm a statement like 'ravens are black'. It is usually stated as follows:

- Premise 1: Observing examples of a statement contributes positive evidence towards that statement.
- Premise 2: Evidence for some statement is also evidence for all logically equivalent statements.
- Conclusion: Observing examples of non-black non-ravens is evidence for 'all ravens are black'.

The conclusion follows from the fact that 'non-black objects are non-ravens' is logically equivalent to 'ravens are black'. Although we are considering logical validity instead of confirmation, we note that for DFL a similar thing happens. When we correct the observation of an NBR to a BR, the difference in truth value is equal to when we correct it to NBNR. More precisely, representing 'ravens are black' as $I(a, b)$, where, for example, $I(1, 1)$ corresponds to BR:

$$\begin{aligned}&A(x_1, ..., I(1, 0), ..., x_n) - A(x_1, ..., I(1, 1), ..., x_n)\\ =&A(x_1, ..., I(1, 0), ..., x_n) - A(x_1, ..., I(0, 0), ..., x_n)\end{aligned}$$

as $I(0, 0) = I(1, 1) = 1$. Furthermore, when one agent observes a thousand BR's and a single NBR, and another agent observes a thousand NBNR's and a single NBR, their truth value for 'ravens are black' is equal. This seems strange, as the first agent has actually seen many ravens of which only a single exception was not black, while the second only observed many non ravens which were not black, among which a single raven that was not black either. Intuitively, the first agent's beliefs seem to be more in line with the background knowledge. We will now proceed to analyse a number of implication operators in light of this discussion.

4.1 Analyzing the Implication Operators

We define two functions for a fuzzy implication I:

$$d_{Ic}(a, c) = \frac{\partial I(a, c)}{\partial c} \tag{9}$$

$$d_{I\neg a}(a, c) = -\frac{\partial I(a, c)}{\partial a} = \frac{\partial I(a, c)}{\partial \neg a}. \tag{10}$$

d_{Ic} is the derivative with respect to the consequent and $d_{I\neg a}$ is the derivative with respect to the *negated* antecedent. We choose to take the derivative with respect to the negated antecedent as it makes it easier to compare them.

Definition 8. *A fuzzy implication I is called* contrapositive differentiable symmetric *if $d_{Ic}(a, c) = d_{I\neg a}(1 - c, 1 - a)$ for all $a, c \in [0, 1]$.*

A consequence of contrapositive differentiable symmetry is that if $c = 1 - a$, then the derivatives are equal since $d_{Ic}(a, c) = d_{I\neg a}(1 - c, 1 - a) = d_{I\neg a}(1 - (1 - a), c) = d_{I\neg a}(a, c)$. This could be seen as the 'distrust' option as it increases the consequent and negated antecedent equally.

Proposition 1. *If a fuzzy implication I is contrapositive symmetric, it is also contrapositive differentiable symmetric.*

Proof. Say we have an implication I that is contrapositive symmetric. We find that $d_{Ic}(a, c) = \frac{\partial I(a,c)}{\partial c}$ and $d_{I\neg a}(1 - c, 1 - a) = -\frac{\partial I(1-c,1-a)}{\partial 1-c}$. Because I is contrapositive symmetric, $I(1 - c, 1 - a) = I(a, c)$. Thus, $d_{I\neg a}(1 - c, 1 - a) = -\frac{\partial I(a,c)}{\partial 1-c} = \frac{\partial I(a,c)}{\partial c} = d_{Ic}(a, c)$. □

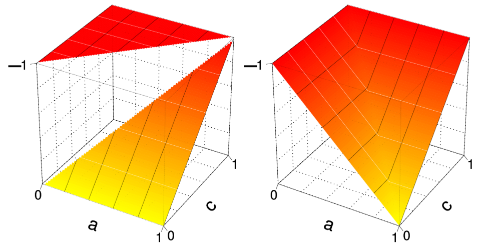

Figure 2: Left: The Gödel implication. Right: The Kleene Dienes implication.

In particular, by this proposition all S-implications are contrapositive differentiable symmetric. This says that there is no difference in how the implication handles the derivatives with respect to the consequent and antecedent.

Proposition 2. *If an implication I is left-neutral, then $d_{Ic}(1, c) = 1$. If, in addition, I is contrapositive differentiable symmetric, then $d_{I\neg a}(a, 0) = 1$.*

Proof. Assume I is left-neutral. Then for all $c \in [0, 1]$, $I(1, c) = c$. Taking the derivative with respect to c, it turns out that $d_{Ic}(1, c) = 1$. Next, assume I is contrapositive differentiable symmetric. Then, $d_{Ic}(1, c) = d_{I\neg a}(1 - c, 1 - 1) = d_{I\neg a}(1 - c, 0) = 1$. As $1 - c \in [0, 1]$, $d_{I\neg a}(a, 0) = 1$. □

All S-implications and R-implications are left-neutral, but only S-implications are all also contrapositive differentiable symmetric. The derivatives of R-implications vanish when $a \leq c$, that is, on no less than half of the domain. Note that the plots in this section are rotated so that the smallest value is in the front. In particular, plots of the derivatives of the implications are rotated 180 degrees compared to the implications themselves.

Gödel-based Implications Implications based on the Gödel t-norm ($T_G(a, b) = \min(a, b)$) make strong discrete choices and increase at most one of their outputs. The two associated implications are shown in Figure 2. The Gödel implication is a simple R-implication with the following derivatives:

$$d_{I_G c}(a, c) = \begin{cases} 1, & \text{if } a > c \\ 0, & \text{otherwise} \end{cases}, \quad d_{I_G \neg a}(a, b) = 0. \tag{11}$$

The Gödel implication increases the consequent whenever $a > c$, and the antecedent is never changed. This makes it a poorly performing implication in practice. For example, consider $a = 0.1$ and $c = 0$. Then the Gödel implication increases the consequent, even if the agent is fairly certain that neither is true. Furthermore, as the derivative with respect to the negated antecedent is always 0, it can never choose the modus tollens correction, which, as we argued, is actually often the best choice.

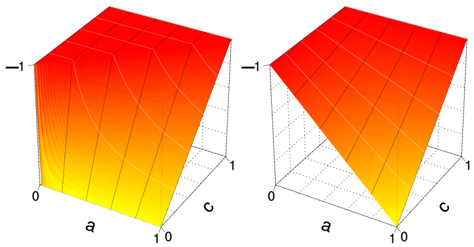

Figure 3: Left: The Goguen implication. Right: The Reichenbach implication.

The derivatives of the Kleene-Dienes implication are

$$d_{I_{KD}c}(a,c) = \begin{cases} 1, & \text{if } 1-a < c \\ 0, & \text{if } 1-a > c \end{cases}, \tag{12}$$

$$d_{I_{KD}\neg a}(a,b) = \begin{cases} 1, & \text{if } 1-a > c \\ 0, & \text{if } 1-a < c \end{cases}. \tag{13}$$

Or, simply put, if we are more confident in the truth of the consequent than in the truth of the negated antecedent, increase the truth of the consequent. Otherwise, decrease the truth of the antecedent. This decision can be somewhat arbitrary and does not take into account the imbalance of modus ponens and modus tollens.

Łukasiewicz Implication The Łukasiewicz implication is both an S- and an R-implication. It has the simple derivatives

$$d_{I_{LK}c}(a,c) = d_{I_{LK}\neg a}(a,c) = \begin{cases} 1, & \text{if } a > c \\ 0, & \text{if } a < c. \end{cases} \tag{14}$$

Whenever the implication is not satisfied because the antecedent is higher than the consequent, it simply increases the negated antecedent and the consequent until it is lower. This could be seen as the 'distrust' choice as both observations of the agent are equally corrected, and so does not take into account the imbalance between modus ponens and modus tollens cases. The derivatives of the Gödel implication I_G are equal to those of I_{LK} except that I_G always has a zero derivative for the negated antecedent.

Product-based Implications The product t-norm is given as $T_P(a,b) = a \cdot b$. The associated R-implication is called the Goguen implication. We plot this implication in Figure 3. The derivatives of I_{GG} are

$$d_{I_{GG}c}(a,c) = \begin{cases} 0, & \text{if } a \leq c \\ \frac{1}{a}, & \text{otherwise} \end{cases}, \tag{15}$$

$$d_{I_{GG}\neg a}(a,c) = \begin{cases} 0, & \text{if } a \leq c \\ \frac{c}{a^2}, & \text{otherwise} \end{cases}. \tag{16}$$

We plot these in Figure 4. This derivative is not very useful. First of all, both the modus ponens and modus tollens derivatives increase with $\neg a$. This is opposite of the modus ponens rule as when the antecedent is *low*, it increases the consequent most. For example, if raven is 0.1 and black is 0, then the derivative with respect to black is 10, because of the singularity when a approaches 0.

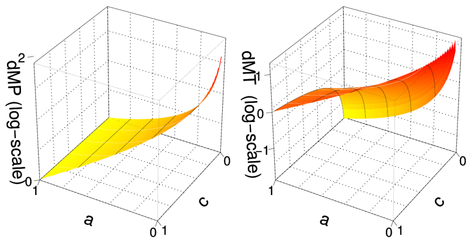

Figure 4: The derivatives of the Goguen implication. Note that we plot these in log scale.

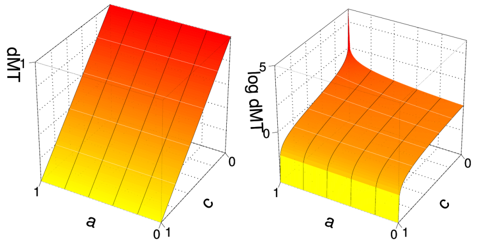

Figure 5: Left: The antecedent derivative of the Reichenbach implication. Right: The antecedent derivative of the Reichenbach implication with the log-product aggregator.

The derivatives of the Reichenbach implication are:

$$d_{I_{RC}c}(a,c) = a, \quad d_{I_{RC}\neg a}(a,c) = 1 - c. \tag{17}$$

These derivatives closely follow modus ponens and modus tollens inference: When the antecedent is high, increase the consequent, and when the consequent is low, decrease the antecedent. However, around $(1-a) = c$, the derivative is equal and the 'distrust' option is chosen. This can result in counter-intuitive behaviour. For example, if the agent predicts 0.6 for raven and 0.5 for black and we use gradient descent, we could end up at 0.3 for raven and 1 for black. We would end up increasing our confidence in black as raven was high. However, because of the negated antecedent derivatives, raven is barely true.

Furthermore, if the agent mostly predicts values around $a = 0$, $c = 0$ as a result of the modus tollens case being the most common, then a majority of the gradient decreases the antecedent as $d_{I_{RC}\neg a}(0,0) = 1$. We next identify two methods that counteract this behavior.

Log product aggregator The first method for counteracting the 'corner' behavior notes that different aggregators change how the derivatives of the implications behave. Note that the aggregator based on the product t-norm is $A_P(x_1, ..., x_n) = \prod_{i=1}^n x_i$. As formulas are in prenex normal form, maximizing this aggregator is equivalent to maximizing the logarithm of this aggregator, which gives $A_{\log P}(x_1, ..., x_n) = \sum_{i=1}^n \log(x_i)$ that is reminiscent of the cross-entropy loss function. Using the chain rule, we find that the negated antecedent derivative becomes:

$$\frac{\partial A_{\log P}(I(a_1,c_1), ..., I(a_n,c_n))}{\partial 1 - a_i} = \frac{d_{I\neg a}(a_i,c_i)}{I(a_i,c_i)} \tag{18}$$

As this divides by the truth value of the implication, implications that do not have a high truth value get stronger

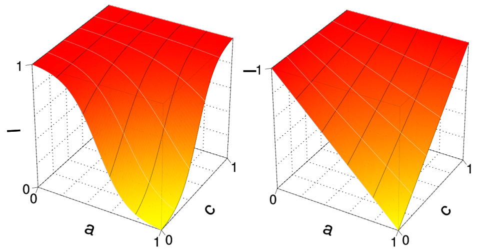

Figure 6: The Reichenbach-sigmoidal implication for different values of s. Left: $s = 9$. Right: $s = 0.01$.

derivatives. We plot the negated antecedent derivative for the Reichenbach implication when using the log-product aggregator in Figure 5. Note that the derivative with respect to the negated antecedent in $a_i = 0, c_i = 0$ is still 1. By differentiable contrapositive symmetry, the consequent derivative is 0. Therefore, when using the log-product aggregator, one of antecedent and consequent will still have a gradient.

Sigmoidal Implications For the second method for tackling the corner problem, we introduce a new class of fuzzy implications formed by transforming other fuzzy implications using the sigmoid function and translating it so that the boundary conditions still hold.[1]

Definition 9. *If I is a fuzzy implication, then the I-sigmoidal implication σ_I is given for some $s > 0$ as*

$$\sigma_I(a,c) = \frac{\left(1 + e^{\frac{s}{2}}\right) \cdot \sigma\left(s \cdot I(a,c) - \frac{s}{2}\right) - 1}{e^{\frac{s}{2}} - 1} \tag{19}$$

where $\sigma(x) = \frac{1}{1+e^x}$ denotes the sigmoid function.

Here s controls the 'spread' of the curve. σ_I is the function $\sigma\left(s \cdot \left(I(a,c) - \frac{1}{2}\right)\right)$ linearly transformed so that its codomain is the closed interval $[0, 1]$. σ_I is a fuzzy implication in the sense of Definition 1. Furthermore, σ_I satisfies the identity principle if I does, and is contrapositive (differentiable) symmetric if I is. We plot the Reichenbach-sigmoidal implication $\sigma_{I_{RC}}$ in Figure 6 for two values of s. Note that for $s = 0.01$, the plotted function is indiscernible from the plot of the Reichenbach implication in Figure 3 as the interval on which the sigmoid acts is extremely small and the sigmoidal transformation is almost linear. The derivative is computed as

$$\begin{aligned}\frac{\partial \sigma_I(a,c)}{\partial I(a,c)} =& \frac{s \cdot \left(1 + e^{\frac{s}{2}}\right)}{e^{\frac{s}{2}} - 1} \cdot \sigma\left(s \cdot I(a,c) - \frac{s}{2}\right) \cdot \\ & \left(1 - \sigma\left(s \cdot I(a,c) - \frac{s}{2}\right)\right).\end{aligned} \tag{20}$$

The derivative keeps the properties of the original function but smoothes the gradient for higher values of s. As the derivative of the sigmoid function (that is, $\sigma(x) \cdot (1 - \sigma(x))$) cannot be zero, this derivative vanishes only when $\frac{\partial I(a,c)}{\partial \neg a} = 0$ or $\frac{\partial I(a,c)}{\partial c} = 0$.

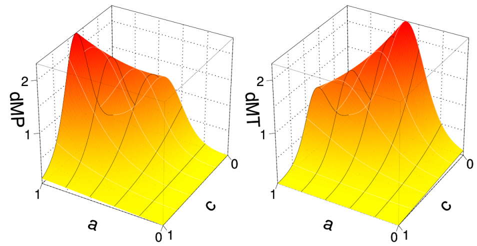

Figure 7: The derivatives of the Reichenbach-sigmoidal implication for $s = 9$.

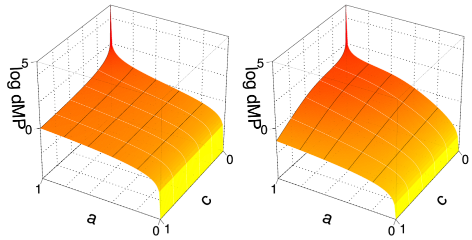

Figure 8: The consequent derivatives of the log-Reichenbach and log-Reichenbach-sigmoidal (with $s = 9$) implications. The figure is plotted in log scale.

We plot the derivatives for the Reichenbach-sigmoidal implication $\sigma_{I_{RC}}$ in Figure 7. As expected, it is clearly differentiable contrapositive symmetric. Compared to the derivatives of the Reichenbach implication it has a small gradient in all corners. In Figure 8 we compare the consequent derivative of the normal Reichenbach implication with the Reichenbach-sigmoidal implication when using the $\log$ product aggregator. The sigmoidal variant is less 'flat' than the normal Reichenbach implication. This can be useful, as this means there is a larger gradient for values of c that make the implication less true. In particular, the gradient at the modus ponens case ($a = 1,\ c = 1$) and the modus tollens case ($a = 0,\ c = 0$) are far smaller, which could help balancing the effective total gradient by solving the 'corner' problem of the Reichenbach implication. These derivatives are smaller for for higher values of s.

5 Experiments

To get an idea of the practical behavior of these implications we now perform experiments using the MNIST dataset of handwritten digits (LeCun and Cortes 2010) to investigate the behavior of different fuzzy implications.

5.1 Measures

To investigate the performance of the different configurations of DFL, we first introduce several useful metrics. In this section, we assume we are dealing with formulas of the form $\varphi = \forall x_1, ..., x_m\ \phi \to \psi$.

Definition 10. *The* consequent magnitude $|\text{cons}|$ *and the* antecedent magnitude $|\text{ant}|$ *for a knowledge base $\mathcal{K}$ is defined as the sum of the partial derivatives of the consequent and*

[1] The derivation, along with several proofs of properties, can be found at https://github.com/HEmile/differentiable-fuzzy-logics/blob/master/appendix_sigmoidal_implications.pdf.

antecedent with respect to the DFL loss:

$$|\text{cons}| = \sum_{\varphi\in\mathcal{K}} \sum_{\mu\in M_\varphi} \frac{\partial e_{\boldsymbol{\theta}}(\varphi)}{\partial e_{\boldsymbol{\theta}}(\psi)}, \tag{21}$$

$$|\text{ant}| = \sum_{\varphi\in\mathcal{K}} \sum_{\mu\in M_\varphi} \frac{-\partial e_{\boldsymbol{\theta}}(\varphi)}{\partial e_{\boldsymbol{\theta}}(\phi)}, \tag{22}$$

where M_φ is the set of instances of the universally quantified formula φ and ψ and ϕ are evaluated under instantiation μ. The consequent ratio cons% *is the sum of consequent magnitudes divided by the sum of consequent and antecedent magnitudes:* $\text{cons}\% = \frac{|\text{cons}|}{|\text{cons}|+|\text{ant}|}$.

Definition 11. *Given a* labeling function l *that returns the truth value of a formula according to the data for instance μ, the* consequent and antecedent correctly updated magnitudes *are the sum of partial derivatives for which the consequent or the negated antecedent is true:*

$$cu_{\text{cons}} = \sum_{\varphi\in\mathcal{K}} \sum_{\mu\in M_\varphi} l(\psi,\mu)\cdot\frac{\partial e_{\boldsymbol{\theta}}(\varphi)}{\partial e_{\boldsymbol{\theta}}(\psi)}, \tag{23}$$

$$cu_{\text{ant}} = \sum_{\varphi\in\mathcal{K}} - \sum_{\mu\in M_\varphi} l(\neg\phi,\mu)\cdot\frac{\partial e_{\boldsymbol{\theta}}(\varphi)}{\partial e_{\boldsymbol{\theta}}(\phi)}. \tag{24}$$

The correctly updated ratio for consequent and antecedent *are* $cu_{\text{cons}}\% = \frac{cu_{\text{cons}}}{|\text{cons}|}$ *and* $cu_{\text{ant}}\% = \frac{cu_{\text{ant}}}{|\text{ant}|}$.

That is, if the consequent is true in the data, we measure the magnitude of the derivative with respect to the consequent. The correctly updated ratios quantify what fraction of the updates are going in the right direction. When they approach 1, DFL will always increase the truth value of the consequent or negated antecedent correctly. When it is not close to 1, we are increasing truth values of subformulas that are wrong, thus ideally, we want these measures to be high.

5.2 Experimental Setup

We use a knowledge base $\mathcal{K}$ of universally quantified logic formulas. There is a predicate for each digit (zero, ..., nine). For example, $\mathsf{zero}(x)$ is true whenever x is a digit labeled with 0. Secondly, there is the binary predicate same that is true whenever both its arguments are the same digit. We next describe the types of formulas we use.

1. $\forall x, y\ \mathsf{zero}(x) \wedge \mathsf{zero}(y) \to \mathsf{same}(x,y)$. If both x and y are handwritten zeros, then they represent the same digit.
2. $\forall x, y\ \mathsf{zero}(x) \wedge \mathsf{same}(x,y) \to \mathsf{zero}(y)$. If x and y represent the same digit and one of them represents zero, then the other one does as well.
3. $\forall x, y\ \mathsf{same}(x,y) \to \mathsf{same}(y,x)$. This formula encodes the symmetry of the same predicate.

We split the MNIST dataset so that 1% of it is labeled and 99% is unlabeled. We use two models.[2] Given a handwritten digit $\boldsymbol{x}$, the first model $p_{\boldsymbol{\theta}}(y|\boldsymbol{x})$ computes the distribution over the 10 possible labels. We use 2 convolutional layers with max pooling, the first with 10 and the second with 20 filters, and two fully connected hidden layers with 320 and 50 nodes and a softmax output layer, which is trained using cross entropy. The probability that $\mathsf{same}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ for two handwritten digits $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ holds is modeled by $p_{\boldsymbol{\theta}}(\mathsf{same}|\boldsymbol{x}_1, \boldsymbol{x}_2)$. This takes the 50-dimensional embeddings of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ of the fully connected hidden layer $e_{\boldsymbol{x}_1}$ and $e_{\boldsymbol{x}_2}$. These are used in a Neural Tensor Network (Socher et al. 2013) with a hidden layer of size 50. It is trained using binary cross entropy on the cross product of the labeled dataset. As there are far more negative examples than positive examples, we undersample the negative examples.

	Accuracy	cons%	$cu_{\text{cons}}\%$	$cu_{\text{ant}}\%$
I_{KD}	96.1	0.10	**0.88**	0.97
I_{LK}	**97.0**	0.5	0.03	0.97
I_{RC}	96.9	0.08	0.85	**0.99**
I_G	90.6	1	0.07	–
I_{GG}	94.0	0.86	0.01	0.97

Table 2: The results using the introduced Fuzzy Implications.

We add the DFL loss to the other two losses. For all experiments, we use the product aggregator with DFL weight of $w_{dfl} = 10$, and optimize the logarithm of the truth value. For conjunction, we use the Yager t-norm with $p = 2$, defined as $T_Y(a, b) = \max(1 - ((1-a)^p + (1-b)^p)^{\frac{1}{p}}, 0)$.

5.3 Results

We analyze the results for different implication operators. The purely supervised baseline has a test accuracy of $95.0\% \pm 0.001$ (3 runs). We report the accuracy of recognizing digits in the test set. We do learning for at most 100.000 iterations (or until convergence). We also report the consequent ratio cons% and the consequent and antecedent correctly updated ratios $cu_{\text{cons}}\%$ and $cu_{\text{ant}}\%$. We compute these values during the backwards pass of the DFL loss on the ‘unlabeled’ dataset. Because it is a split of MNIST, we can access the labels for evaluation.

Implications In Table 2, we compare different fuzzy implications. The Reichenbach implication and the Łukasiewicz implication work well, both having an accuracy around 97%. Using the Kleene Dienes implication surpasses the baseline as well. As hypothesized, the Gödel implication and Goguen implication have worse performance than the supervised baseline. While the derivatives of I_{LK} and I_G only differ in that I_G disables the derivatives with respect to negated antecedent, I_{LK} performs among the best but I_G performs among the worst, suggesting that the derivatives with respect to the negated antecedent are required to successfully applying DFL. All well performing implications are S-implications, which inherently balance derivatives with respect to the consequent and negated antecedent by being contrapositive differentiable symmetric.

Reichenbach-Sigmoidal Implication The newly introduced Reichenbach-sigmoidal implication $\sigma_{I_{RC}}$ is a promising candidate. In Figure 9 we plot the results when we experiment with the parameter s. Note that as s approaches 0, the Reichenbach-sigmoidal implication is I_{RC}. $s = 9$ gives

[2] Code: https://github.com/HEmile/differentiable-fuzzy-logics.

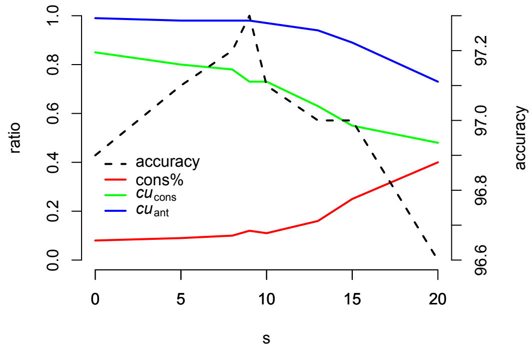

Figure 9: The results using the Reichenbach-sigmoidal implication $\sigma_{I_{RC}}$ for various values of s.

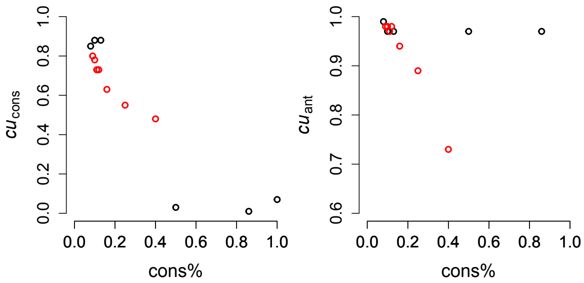

Figure 10: Left: Plot of $\text{cons}\%$ to $cu_{\text{cons}}\%$. Right: Plot of $\text{cons}\%$ to $cu_{\text{ant}}\%$. Red dots represent runs using $\sigma_{I_{RC}}$.

the best results, with 97.3% accuracy. Interestingly, there seem to be clear trends in the values of $\text{cons}\%$, $cu_{\text{cons}}\%$ and $cu_{\text{ant}}\%$. Increasing s seems to increase $\text{cons}\%$. This is because the antecedent derivative around the corner $a = 0,\ c = 0$ will be low, as argued before. When s increases, the corners will be more smoothed out. Furthermore, both $cu_{\text{cons}}\%$ and $cu_{\text{ant}}\%$ decrease when s increases. This could be because around the corners the derivatives become small. Updates in the corner will likely be correct as the model is already confident about those. For a higher value of s, most of the gradient magnitude is at instances on which the model is less confident. Regardless, the best parameter value clearly is not the one for which the values of $cu_{\text{cons}}\%$ and $cu_{\text{ant}}\%$ are highest, namely the Reichenbach implication itself.

Analysis We plot the experimental values of $\text{cons}\%$ to the values of $cu_{\text{cons}}\%$ and $cu_{\text{ant}}\%$ in Figure 10. For both, there seems to be a negative correlation: larger consequent ratios decrease the correctness of the updates. We argued that this could be because for lower values of $\text{cons}\%$, a smaller portion of the reasoning happens in the corners around $a = 0,\ c = 0$ and $a = 1,\ c = 1$, and more for instances that the agent is less certain about. As S-implications have strong derivatives at both these corners (Proposition 2), this phenomenon is likely present in other S-implications. Although DFL significantly improves the supervised baseline, it is currently not competitive with state-of-the-art methods like Ladder Networks (Rasmus et al. 2015).

6 Related Work

DFL is in the field of Statistical Relational Learning (Getoor and Taskar 2007), which concerns models that reason under uncertainty and learn relational structures. Special cases of DFL have been researched under different names. Real Logic (Serafini and Garcez 2016) implements function symbols and uses a model called Logic Tensor Networks to interpret predicates. It uses S-implications. Real Logic is applied to weakly supervised learning on Semantic Image Interpretation (Donadello, Serafini, and Garcez 2017; Donadello and Serafini 2019) and transfer learning in Reinforcement Learning (Badreddine and Spranger 2019). Semantic-based regularization (SBR) (Diligenti, Gori, and Sacca 2017) applies DFL to kernel machines. They use R-implications, like (Marra et al. 2019) which simplifies the satisfiability computation and finds generalizations of common loss functions. In (Marra et al. 2018), which employs the Goguen implication, DFL is applied to image generation. The Reichenbach implication is used in (Rocktäschel, Singh, and Riedel 2015) for relation extraction using a matrix embedding of the rules.

(Demeester, Rocktäschel, and Riedel 2016) use a regularization technique that is equivalent to the Łukasiewicz implication. It finds a loss function which does not iterate over objects, yet guarantees that rules hold. (Minervini et al. 2017; Minervini and Riedel 2018) extend this using adversarial sets of objects from the domain that do not satisfy the knowledge base, which are probably the most informative objects.

Instead of fuzzy logics, DeepProbLog (Manhaeve et al. 2018) and Semantic Loss (Xu et al. 2018) use probabilistic logics with neural predicates that compute the probabilities of ground atoms. Like DFL, they ground predicates and back-propagate from the loss to deep learning models.

7 Conclusion

We analyzed fuzzy implications in Differentiable Fuzzy Logics in order to understand how reasoning using implications behaves in a differentiable setting. We found substantial differences between the properties of a large number of fuzzy implications, and showed that many of them, including some of the most popular, are highly unsuitable for use in a differentiable learning setting.

The Reichenbach implication has derivatives that are intuitive and correspond to inference rules from classical logic. The Łukasiewicz implication is the best R-implication in our experiments. The Gödel and Goguen implications, however, were not successful, performing worse than the supervised baseline. The newly introduced Reichenbach-sigmoidal implication performs best on the MNIST experiments. Its spread can be tweaked to decrease the imbalance of derivatives with respect to the negated antecedent and consequent. This imbalance exists because the modus tollens case is much more common. We conclude that a large part of the useful inferences are made by decreasing the antecedent, or by 'modus tollens reasoning'. Furthermore, we found that derivatives with respect to the consequent often increase the truth value of something that is false as the consequent is false in the majority of times. Therefore, we argue that 'modus tollens reasoning' should be embraced.

Acknowledgements

The authors thank the editors and all anonymous reviewers. This work is partly funded by the MaestroGraph research programme with project number 612.001.552, which is financed by the Dutch Research Council (NWO). This work is supported by the DAS-5 distributed supercomputer (Bal et al. 2016).

References

Badreddine, S., and Spranger, M. 2019. Injecting Prior Knowledge for Transfer Learning into Reinforcement Learning Algorithms using Logic Tensor Networks. *arXiv preprint arXiv:1906.06576*.

Bal, H.; Epema, D.; de Laat, C.; van Nieuwpoort, R.; Romein, J.; Seinstra, F.; Snoek, C.; and Wijshoff, H. 2016. A medium-scale distributed system for computer science research: Infrastructure for the long term. *Computer* 49(5):54–63.

Besold, T. R.; Garcez, A. d.; Bader, S.; Bowman, H.; Domingos, P.; Hitzler, P.; Kuehnberger, K.-U.; Lamb, L. C.; Lowd, D.; Lima, P. M. V.; de Penning, L.; Pinkas, G.; Poon, H.; and Zaverucha, G. 2017. Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. *arXiv preprint arXiv:1711.03902*.

Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Demeester, T.; Rocktäschel, T.; and Riedel, S. 2016. Lifted Rule Injection for Relation Embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1389–1399. Association for Computational Linguistics.

Diligenti, M.; Gori, M.; and Sacca, C. 2017. Semantic-based regularization for learning and inference. *Artificial Intelligence* 244:143–165.

Diligenti, M.; Roychowdhury, S.; and Gori, M. 2017. Integrating Prior Knowledge into Deep Learning. In *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*, 920–923. IEEE.

Donadello, I., and Serafini, L. 2019. Compensating Supervision Incompleteness with Prior Knowledge in Semantic Image Interpretation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Donadello, I.; Serafini, L.; and Garcez, A. d. 2017. Logic Tensor Networks for Semantic Image Interpretation. In *IJCAI International Joint Conference on Artificial Intelligence*, 1596–1602.

Evans, R., and Grefenstette, E. 2018. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research* 61:65–170.

Garcez, A. S. d.; Broda, K. B.; and Gabbay, D. M. 2012. *Neural-symbolic learning systems: foundations and applications*. Springer Science & Business Media.

Garnelo, M.; Arulkumaran, K.; and Shanahan, M. 2016. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*.

Getoor, L., and Taskar, B. 2007. *Introduction to statistical relational learning*, volume 1. MIT press Cambridge.

Goodfellow, I.; Bengio, Y.; Courville, A.; and Bengio, Y. 2016. *Deep learning*, volume 1. MIT press Cambridge.

Guo, S.; Wang, Q.; Wang, L.; Wang, B.; and Guo, L. 2016. Jointly embedding knowledge graphs and logical rules. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 192–202.

Harnad, S. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena* 42(1-3):335–346.

Hempel, C. G. 1945. Studies in the Logic of Confirmation (II.). *Mind* 54(214):97–121.

Hu, Z.; Ma, X.; Liu, Z.; Hovy, E.; and Xing, E. 2016. Harnessing Deep Neural Networks with Logic Rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2410–2420. Association for Computational Linguistics.

Japkowicz, N., and Stephen, S. 2002. The class imbalance problem: A systematic study. *Intelligent data analysis* 6(5):429–449.

Jayaram, B., and Baczynski, M. 2008. *Fuzzy Implications*, volume 231. Springer, Berlin, Heidelberg.

Klir, G., and Yuan, B. 1995. *Fuzzy sets and fuzzy logic*, volume 4. Prentice hall New Jersey.

LeCun, Y., and Cortes, C. 2010. MNIST handwritten digit database.

Manhaeve, R.; Dumančić, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2018. DeepProbLog: Neural Probabilistic Logic Programming. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*.

Marcus, G. 2018. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.

Marra, G.; Giannini, F.; Diligenti, M.; and Gori, M. 2018. Constraint-Based Visual Generation. *arXiv preprint arXiv:1807.09202*.

Marra, G.; Giannini, F.; Diligenti, M.; Maggini, M.; and Gori, M. 2019. Learning and T-Norms Theory. *arXiv preprint arXiv:1907.11468*.

Minervini, P., and Riedel, S. 2018. Adversarially Regularising Neural NLI Models to Integrate Logical Background Knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 65–74.

Minervini, P.; Demeester, T.; Rocktäschel, T.; and Riedel, S. 2017. Adversarial sets for regularising neural link predictors. In *Uncertainty in Artificial Intelligence-Proceedings of the 33rd Conference, UAI 2017*.

Pearl, J. 2018. Theoretical Impediments to Machine Learning With Seven Sparks from the Causal Revolution. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 3. ACM.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners.

Rasmus, A.; Berglund, M.; Honkala, M.; Valpola, H.; and Raiko, T. 2015. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, 3546–3554.

Rocktäschel, T.; Singh, S.; and Riedel, S. 2015. Injecting Logical Background Knowledge into Embeddings for Relation Extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1119–1129.

Serafini, L., and Garcez, A. D. 2016. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. *CEUR Workshop Proceedings* 1768.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; and others. 2017. Mastering the game of Go without human knowledge. *Nature* 550(7676):354.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. Y. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. *Proc.\ of NIPS'13* 1–10.

van Krieken, E.; Acar, E.; and van Harmelen, F. 2019. Semi-Supervised Learning using Differentiable Reasoning. *IFCoLog Journal of Logic and its Applications* 6(4):633–653.

Vranas, P. B. 2004. Hempel's raven paradox: A lacuna in the standard Bayesian solution. *British Journal for the Philosophy of Science* 55(3):545–560.

Xu, J.; Zhang, Z.; Friedman, T.; Liang, Y.; and den Broeck, G. 2018. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In Dy, J., and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 5502–5511. Stockholmsmässan, Stockholm Sweden: PMLR.

Zhou, Z.-H. 2017. A brief introduction to weakly supervised learning. *National Science Review* 5(1):44–53.