Reduced Implication-bias Logic Loss for Neuro Symbolic Learning

Presenter: Haoyuan He / Paper: Link

Introduction

Begin of our story.

Currently, machine learning methods are achieving significant success in perception. However, real-world learning tasks usually require not only the perception ability but also the logical reasoning ability. It is reasonable to believe that the next generation of advanced artificial intelligence will be seamlessly constructed by combining logical reasoning with machine learning.

Neuro-Symbolic Learning

Neuro-Symbolic (NeSy) AI aims to combine neural networks and symbolic reasoning.

Some people try to implement a hybrid system by implementing an interface between neural networks and symbolic reasoning systems. Zhou and Dai 2019 introduce the Abductive Learning (ABL) framework to integrate first-order logic with machine learning models and abductive reasoning. Manhaeve 2018 propose DeepProbLog, which integrates probabilistic logic programming with deep learning by interfacing them with neural predicates.

However, these models are difficult to train because of the high complexity caused by the joint optimisation of the neural and symbolic modules. Therefore, many researchers propose to approximate logical reasoning with differentiable operators, thereby making it possible to turn symbolic knowledge into loss functions and train an end-to-end model by gradient descent.

Implication Bias [breif intro]

In the process of approximating implication rules such as $p_1 \wedge p_2 \wedge \cdots \wedge p_n \rightarrow q$ with differentiable operations, a phenomenon that we called **Implication Bias** could degrade the model performance. Informally, this rule could be satisfied via vacuous truth, i.e., by negating the premises $p_1 \wedge p_2 \wedge \cdots \wedge p_n$. Under ideal circumstances, samples that do not satisfy the rule premises, their loss values and gradients computed from this rule should be zero. However, we found this is not the case for semantic loss and many other widely used logic loss functions derived from fuzzy operators. Furthermore, the shortcuts problem of neural network training would amplify this bias and make the model tends to negate the premise literals when the training data is insufficient.

We point out that the major problem behind this phenomenon is that the model keeps inferring unwanted negations during the training process. In logic programming, people employ Closed World Assumption and Negation As Failure (NAF) to prevent this issue and guarantee the soundness of inference. Nevertheless, in NeSy systems, there lacks such a method to reduce this bias.

Preliminaries

First Order Logic

A First Order Logic (FOL) $\mathcal{L}$ consists of a set $\mathcal{C}$ of constant symbols, a set $\mathcal{F}$ of functional symbols, a set $\mathcal{P}$ of predicates (relations) symbols, and a set $\mathcal{V}$ of variable symbols, a set $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ of connectives, a set $\{\forall, \exists\}$ of quantifiers.

The connectives set can be reduced to $\{\neg, \rightarrow\}$ because other operations can be replaced with a combination of these two operations. For example, $p \wedge q \equiv \neg(p \rightarrow \neg q), p \leftrightarrow q \equiv \neg((p \rightarrow q) \rightarrow \neg(q \rightarrow p)), p \vee q \equiv \neg p \rightarrow q$. Atom is the basic component of logic formulas, such as $\mathrm{BiggerThan}(x, \mathrm{Add10}(y)), \mathrm{Equal}(\mathrm{Succ}(x), y)$. Denote the set of atoms as $\mathcal{A}$, the set of formulas as $\mathbb{F}$.

Closed World Assumption

The closed-world assumption (CWA) is a presumption that a true statement is also known to be true. Conversely, what is not currently known to be true, is false. The opposite of the closed-world assumption is the open-world assumption (OWA), holding that lack of knowledge does not imply falsity. The CWA is a very natural rule to use in a database context. Information not explicitly presented in the database is taken to be false. For example:

Statement	Bob and Cam are students.
Question	Is Alice a student?
Closed world answer	No.
Open world answer	Unknown.

Negation as Failure

Suppose we wish to infer $\neg A$ from a program P. To use the CWA, we have to show that A is not a logical consequence of P. Unfortunately, because of the undecidability of the validity problem of first-order logic, there is no algorithm which will take an arbitrary A as input and respond in a finite amount of time with the answer whether A is or is not a logical consequence of P. If A is not a logical consequence, it may loop forever.

To alleviate this problem, Clark 1978 proposed the *negation as failure* rule. It states that if A is in the finite failure set of P, then infer $\neg A$. We can see that the negation as a failure rule is less powerful than the CWA. However, implementing anything beyond negation as a failure rule is difficult in practice.

Semantics of Continuous-valued Logic

To transform logic rules into loss functions, we have to assign every grounding term in logic rules a continuous value between 0 and 1. In most cases, this assignment will be given by the machine learning model's output.

Definition. *Valuation of Atom*

A valuation of atom is a function $G(\cdot)\colon \mathcal{A} \to [0, 1]$.

It is natural to define the valuation of negation of an atom $G(\neg A)$ as $1 - G(A)$.

Definition. *Implication Likelihood*

A function $I(\cdot, \cdot)\colon [0, 1] \times [0, 1] \to [0, 1]$ is implication likelihood if it is differentiable.

Implication likelihood is a function that is used to estimate the degree of satisfiability of implication rules, which is closely related to fuzzy t-norms. For example, $I(G(\mathrm{Raven}(x)), G(\mathrm{Black}(x))) = 0.95$ says the satisfiability of this rule : $\mathrm{Raven}(x) \to \mathrm{Black}(x)$ is 0.95.

Semantics of Continuous-valued Logic (Con't)

Since computing the satisfiability of negation and implication rules being available, as mentioned before, in FOL, $\{\neg, \rightarrow\}$ as connectives are enough. It is easy to calculate any logic rule's satisfiability by $\rightarrow$ and $\neg$ recursively.

Definition. *Logic Likelihood*

Logic Likelihood $s(\cdot, \cdot)$:$\mathbb{F} \times [0, 1]^n \rightarrow [0, 1]$ is a function that used to estimate the degree of satisfiablity of logic formula. $s(\cdot, \cdot)$ takes two inputs, the first one is a logic formula, and the second one is the valuation vector of this logic formula. For example:$s(P \rightarrow Q, (x, y)) = I(x, y)$

Semantics of Continuous-valued Logic (Con't)

Definition. *Confidence Monotonic*

A Logic Likelihood is called δ-Confidence Monotonic if its Implication Likelihood satisfies:

- For any $y \in [0, \delta)$, $I(x, y)$ is monotonically decreasing on x.
- For any $x \in (1 - \delta, 1]$, $I(x, y)$ is monotonically increasing on y.

For example, the Implication Likelihood used most commonly is defined as $I(x, y) = 1 - x + x \cdot y$. It is a 1 -Confidence Monotonic Implication Likelihood.

An example for understanding this definition may be helpful. Suppose we have an implication rule, $\mathrm{Raven}(x) \to \mathrm{Black}(x)$. When $G(\mathrm{Black}(x))$ is nearly 0, that is to say, x is not black w.h.p., the higher confidence we trust $\mathrm{Raven}(x)$, the less satisfiability of this rule, vice versa.

Definition. *Logic Loss*

A Logic Loss ℓ_{logic} is a function that estimates the degree of unsatisfiability of logic formula r. It is derived from a Logic Likelihood: $\ell_{\text{logic}}(r, z) = g(s(r, z))$ where g is a monotonically decreasing function, w.l.o.g.,

assume $g(1) = 0$.

Impact of additional logic loss

During the training process of NeSy system which using logic loss as inference of knowledge bases, the optimization object should be set as:

$$\ell = \ell_{\text{task}} + \lambda \cdot \ell_{\text{logic}}$$

Combining logic loss with a task-specific loss for machine learning should be mutually beneficial. In one way, the task-specific loss can direct the model to achieve a pretty good performance so it can give out valid logic primitive facts. In the other way, the logic loss can help shrink the empirical optimal space that can make the model easier to achieve the optimum.

However, a logic loss with δ-confidence monotonic implication likelihood may introduce bias into NeSy systems during the training process.

Proposed Method