Analyzing Differentiable Fuzzy Implications

Emile van Krieken[a,*], Erman Acar[a], Frank van Harmelen[a]

[a] *VU University Amsterdam*

1. Implementation of Differentiable Fuzzy Logics

Algorithm 1 Computation of the Differentiable Fuzzy Logics loss. First it computes the fuzzy Herbrand interpretation g given the current embedded interpretation $\eta_{\boldsymbol{\theta}}$. This performs a forward pass through the neural networks that are used to interpret the predicates. Then it computes the valuation of each formula φ in the knowledge base $\mathcal{K}$.

1: **function** $e_{N,T,S,I,A}(\varphi, g, C, \mu)$ ▷ The valuation function computes the Fuzzy truth value of φ.
2:     **if** $\varphi = \mathsf{P}(x_1, ..., x_m)$ **then**
3:         **return** $g[\mathsf{P}, (\mu(x_1), ..., \mu(x_m)]$ ▷ Find the truth value of a ground atom using the dictionary g.
4:     **else if** $\varphi = \neg\phi$ **then**
5:         **return** $N(e_{N,T,S,I,A}(\phi, g, C, \mu))$
6:     **else if** $\varphi = \phi \wedge \psi$ **then**
7:         **return** $T(e_{N,T,S,I,A}(\phi, g, C, \mu), e_{N,T,S,I,A}(\psi, g, C, \mu))$
8:     **else if** $\varphi = \phi \vee \psi$ **then**
9:         **return** $S(e_{N,T,S,I,A}(\phi, g, C, \mu), e_{N,T,S,I,A}(\psi, g, C, \mu))$
10:     **else if** $\varphi = \phi \rightarrow \psi$ **then**
11:         **return** $I(e_{N,T,S,I,A}(\phi, g, C, \mu), e_{N,T,S,I,A}(\psi, g, C, \mu))$
12:     **else if** $\varphi = \forall x\ \phi$ **then** ▷ Apply the aggregation operator as a quantifier.
13:         **return** $\mathrm{A}_{o \in C}\, e_{N,T,S,I,A}(\phi, g, C, \mu \cup \{x/o\})$ ▷ Each assignment can be seen as an instance of φ.
14:     **end if**
15: **end function**
16:
17: **procedure** DFL$(\eta_{\boldsymbol{\theta}}, \mathcal{P}, \mathcal{K}, O, N, T, S, I, A)$ ▷ Computes the Differentiable Fuzzy Logics loss.
18:     $C \leftarrow o_1, ..., o_b$ sampled from O ▷ Sample b constants to use this pass.
19:     $g \leftarrow dict()$ ▷ Collects truth values for ground atoms.
20:     **for** $\mathsf{P} \in \mathcal{P}$ **do**
21:         **for** $o_1, ..., o_{\alpha(\mathsf{P})} \in C$ **do**
22:             $g[\mathsf{P}, (o_1, ..., o_{\alpha(\mathsf{P})})] \leftarrow \eta_{\boldsymbol{\theta}}(\mathsf{P})(o_1, ..., o_{\alpha(\mathsf{P})})$ ▷ Calculate the truth values of the ground atoms.
23:         **end for**
24:     **end for**
25:     **return** $\mathrm{A}_{\varphi \in \mathcal{K}}\, w_\varphi \cdot e_{N,T,S,I,A}(\varphi, g, C, \emptyset)$ ▷ Calculate valuation of the formulas φ. Start with an empty variable assignment..
26: **end procedure**

The computation of the satisfaction is shown in pseudocode form in Algorithm 1. By first computing the dictionary g that contains truth values for all ground atoms,[1] we can reduce the amount of forward passes through the computations of the truth values of the ground atoms that are required to compute the satisfaction.

This algorithm can fairly easily be parallelized for efficient computation on a GPU by noting that the individual terms that are aggregated over in line 12 (the different *instances* of the universal quantifier) are not dependent on each other. By noting that formulas are in prenex normal form, we can set up the dictionary g using tensor operations so that the recursion has to be done only once for each formula. This can be done by applying the fuzzy operators elementwise over vectors of truth values instead of a single truth value, where each element of the vector represents a variable assignment.

*Corresponding author

Email addresses: e.van.krieken@vu.nl (Emile van Krieken), erman.acar@vu.nl (Erman Acar), Frank.van.Harmelen@vu.nl (Frank van Harmelen)

[1] The dictionary g could be seen as a 'fuzzy Herbrand interpretation', in that it assigns a truth value to all ground atoms.

The complexity of this computation then is $O(|\mathcal{K}| \cdot P \cdot b^d)$, where $\mathcal{K}$ is the set of formulas, P is the amount of predicates used in each formula and d is the maximum depth of nesting of universal quantifiers in the formulas in $\mathcal{K}$ (known as the **quantifier rank**). This is exponential in the amount of quantifiers, as every object from the constants C has to be iterated over in line 12, although as mentioned earlier this can be mitigated somewhat using efficient parallelization. Still, computing the valuation for transitive rules (such as. $\forall x\, y, z\ \mathsf{Q}(x, z) \wedge \mathsf{R}(z, y) \rightarrow \mathsf{P}(x, y)$) will for example be far more demanding than for antisymmetry formulas (such as $\forall x, y\ \mathsf{P}(x, y) \rightarrow \neg\mathsf{P}(y, x)$).

2. Proofs

2.1. Sigmoidal Functions

In Machine Learning, the logistic function or sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ is a common activation function (Goodfellow et al., 2016)(p.65-66). This inspired (šourek et al., 2018) to introduce parameterized families of aggregation functions they call Max-Sigmoid activation functions:

$$A'_{\sigma+\wedge}(x_1, ..., x_n) = \sigma\left(s \cdot \left(\sum_{i=1}^{n} x_i - n + 1 + b_0\right)\right), \quad A'_{\sigma+\vee}(x_1, ..., x_n) = \sigma\left(s \cdot \left(\sum_{i=1}^{n} x_i + b_0\right)\right) \tag{1}$$

We apply this transformation for any fuzzy implication $I : [0, 1]^2 \rightarrow [0, 1]$:

$$\sigma'_I(a, c) = \sigma(s \cdot (I(a, c) + b_0)) \tag{2}$$

This cannot be an aggregation function as $\sigma \in (0, 1)$ and so the boundary conditions $\sigma'_I(1, 1) = 1$, $\sigma'_I(0, 0) = 1$ and $\sigma'_I(1, 0) = 0$ do not hold. We can solve this by adding two linear parameters w and h, redefining σ_I as

$$\sigma_I(a, c) = w \cdot \sigma(s \cdot (I(a, c) + b_0)) - h \tag{3}$$

For this, we need to make sure the lowest value of I maps to 0 and the highest to 1:

$$\sigma_I(1, 0) = w \cdot \sigma(s \cdot (0 + b_0)) - h = 0 \tag{4}$$
$$\sigma_I(1, 1) = w \cdot \sigma(s \cdot (1 + b_0)) - h = 1 \tag{5}$$

First solve both equations for w, starting with Equation 4:

$$\begin{aligned} w \cdot \sigma(s \cdot b_0) - h &= 0 \\ \frac{1}{1 + e^{-s \cdot b_0}} &= \frac{h}{w} \\ w &= h \cdot (1 + e^{-s \cdot b_0}) \end{aligned} \tag{6}$$

Likewise for Equation 5:

$$\begin{aligned} w \cdot \sigma(s \cdot (1 + b_0)) - h &= 1 \\ \frac{1}{1 + e^{-s \cdot (1+b_0)}} &= \frac{1 + h}{w} \\ w &= (1 + h) \cdot (1 + e^{-s \cdot (1+b_0)}) \end{aligned} \tag{7}$$

Now we can solve for h by equating Equations 7 and 6. Then

$$\begin{aligned} (1 + h) \cdot 1 + e^{-s \cdot (1+b_0)} &= h \cdot 1 + e^{-s \cdot b_0} \\ h &= \frac{1 + e^{-s \cdot (1+b_0)}}{1 + e^{-s \cdot b_0} - 1 + e^{-s \cdot (1+b_0)}} \end{aligned}$$

We thus get the following formula:

$$\sigma_I(a, c) = \frac{1 + e^{-s \cdot (1+b_0)}}{e^{-s \cdot b_0} - e^{-s \cdot (1+b_0)}} \cdot \left(\left(1 + e^{-s \cdot b_0}\right) \cdot \sigma\left(s \cdot (I(a, c) + b_0)\right) - 1\right) \tag{8}$$

The most straightforward choice of b_0 is $-\frac{1}{2}$. This translates the outputs of I to $[-\frac{1}{2}, \frac{1}{2}]$ and so it uses a symmetric part of the sigmoid function. With this value, we find the following simplification:

$$\sigma_I(a,c) = \frac{1+e^{-s(1-\frac{1}{2})}}{e^{s(0-\frac{1}{2})} - e^{-s(1-\frac{1}{2})}} \cdot$$
$$\left(\left(1+e^{-s(0-\frac{1}{2})}\right)\cdot\sigma\left(s\cdot\left(I(a,c)-\frac{1}{2}\right)\right)-1\right) \tag{9}$$
$$= \frac{1+e^{-\frac{s}{2}}}{e^{\frac{s}{2}}-e^{-\frac{s}{2}}}\cdot\frac{e^{\frac{s}{2}}-1}{e^{\frac{s}{2}}-1}\cdot\left(\left(1+e^{\frac{s}{2}}\right)\cdot\sigma\left(s\cdot\left(I(a,c)-\frac{1}{2}\right)\right)-1\right) \tag{10}$$
$$= \frac{e^{\frac{s}{2}}-e^{-\frac{s}{2}}}{(e^{\frac{s}{2}}-e^{-\frac{s}{2}})(e^{\frac{s}{2}}-1)}\cdot\left(\left(1+e^{\frac{s}{2}}\right)\cdot\sigma\left(s\cdot\left(I(a,c)-\frac{1}{2}\right)\right)-1\right) \tag{11}$$
$$= \frac{1}{e^{\frac{s}{2}}-1}\cdot\left(\left(1+e^{\frac{s}{2}}\right)\cdot\sigma\left(s\cdot\left(I(a,c)-\frac{1}{2}\right)\right)-1\right) \tag{12}$$
$$= \frac{\left(1+e^{\frac{s}{2}}\right)\cdot\sigma\left(s\cdot I(a,c)-\frac{s}{2}\right)-1}{e^{\frac{s}{2}}-1} \tag{13}$$

Next, we proof several properties of the sigmoidal implication.

Proposition 1. For all $a_1, c_1, a_2, c_2 \in [0,1]$,

1. if $I(a_1,c_1) < I(a_2,c_2)$, then also $\sigma_I(a_1,c_1) < \sigma_I(a_2,c_2)$;
2. if $I(a_1,c_1) = I(a_2,c_2)$, then also $\sigma_I(a_1,c_1) = \sigma_I(a_2,c_2)$.

Proof. 1. We note that σ_I can be written as $\sigma_I(a,c) = w\cdot\sigma\left(s\cdot\left(I(a,c)+\frac{1}{2}\right)\right)-h$ for constants $w = \frac{\left(1+e^{-\frac{s}{2}}\right)^2}{e^{\frac{s}{2}}-e^{-\frac{s}{2}}}$ and $h = \frac{1+e^{-\frac{s}{2}}}{e^{\frac{s}{2}}-e^{-\frac{s}{2}}}$. As $s > 0$, $\frac{s}{2} > -\frac{s}{2}$. Therefore, $e^{\frac{s}{2}} - e^{-\frac{s}{2}} > 0$. Furthermore, as $e^{-\frac{s}{2}} > 0$ then certainly $\left(1+e^{-\frac{s}{2}}\right)^2 > 0$. As both $e^{\frac{s}{2}} - e^{-\frac{s}{2}} > 0$ and $\left(1+e^{-\frac{s}{2}}\right)^2 > 0$, then also $w > 0$. As $s > 0$, $s\cdot(I(a_1,c_1)+b_0) < s\cdot\left(I(a_2,c_2)+\frac{1}{2}\right)$ as by assumption $I(a_1,c_1) < I(a_2,c_2)$. Next, note that the sigmoid function σ is a monotonically increasing function. Using $w > 0$ we find that $\sigma_I(a_1,c_1) = w\cdot\sigma(s\cdot\left(I(a_1,c_1)+\frac{1}{2}\right)) < w\cdot\sigma(s\cdot\left(I(a_2,c_2)+\frac{1}{2}\right)) = \sigma_I(a_2,c_2)$.

2. $$\sigma_I(a_1,c_1) = w\cdot\sigma\left(s\cdot\left(I(a_1,c_1)+\frac{1}{2}\right)\right)-h = w\cdot\sigma\left(s\cdot\left(I(a_2,c_2)+\frac{1}{2}\right)\right)-h = \sigma_I(a_2,c_2)$$

□

Proposition 2. $\sigma_I(a,c)$ is 1 if and only if $I(a,c) = 1$. Similarly, $\sigma_I(a,c)$ is 0 if and only if $I(a,c) = 0$.

Proof. Assume there is some $a, c \in [0,1]$ so that $I(a,c) = 1$. By construction, $\sigma_I(a,c)$ is 1 (see 2.1).

Now assume there is some $a_1, c_1 \in [0,1]$ so that $\sigma_I(a_1,c_1) = 1$. Now consider some a_2, c_2 so that $I(a_2,c_2) = 1$. By the construction of σ_I, $\sigma_I(a_2,c_2) = 1$. For the sake of contradiction assume $I(a_1,c_1) < 1$. However, by Proposition 1 as $I(a_1,c_1) < I(a_2,c_2)$ then $\sigma_I(a_1,c_1) < \sigma_I(a_2,c_2)$ has to hold. This is in contradiction with $\sigma_I(a_1,c_1) = \sigma_I(a_2,c_2) = 1$ so the assumption that $I(a_1,c_1) < 1$ has to be wrong and $I(a_1,c_1) = 1$.

The proof for $I(a,c) = 0$ is analogous. □

Proposition 3. For all fuzzy implications I, σ_I is also a fuzzy implication.

Proof. By the definition of fuzzy implications, $I(\cdot,c)$ is decreasing and $I(a,\cdot)$ is increasing. Therefore, by Proposition 1.1, $\sigma_I(\cdot,c)$ is also decreasing and $\sigma_I(a,\cdot)$ is also increasing. Furthermore, $I(0,0) = 1$, $I(1,1) = 1$ and $I(1,0) = 0$. We find by Proposition 2 that then also $\sigma_I(0,0) = 1$, $\sigma_I(1,1) = 1$ and $\sigma_I(1,0) = 0$. □

I-sigmoidal implications only satisfy left-neutrality if I is left-neutral and s approaches 0.

Proposition 4. If a fuzzy implication I is contrapositive symmetric with respect to N, then σ_I also is.

Proof. Assume we have an implication I that is contrapositive symmetric and so for all $a, c \in [0,1]$, $I(a,c) = I(N(c),N(a))$. By Proposition 1.2, $\sigma_I(a,c) = \sigma_I(N(c),N(a))$. Thus, σ_I is also contrapositive symmetric with respect to N. □

By this proposition, if I is an S-implication, σ_I is contrapositive symmetric and thus also contrapositive differentiable symmetric.

Proposition 5. If I satisfies the identity principle, then σ_I also satisfies the identity principle.

Proof. Assume we have a fuzzy implication I that satisfies the identity principle. Then $I(a,a) = 1$ for all a. By Proposition 2 it holds that $\sigma_I(a,a)$ is also 1. □

References

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

šourek, G., Aschenbrenner, V., Železný, F., Schockaert, S., Kuželka, O., 2018. Lifted relational neural networks: Efficient learning of latent relational structures. Journal of Artificial Intelligence Research 62, 69–100. doi:10.1613/jair.1.11203.