Extraction of Conditional Belief Bases and the System Z Ranking Model From Multilayer Perceptrons for Binary Classification

Marco Wilhelm, Alexander Hahn and Gabriele Kern-Isberner

Dept. of Computer Science, TU Dortmund University, Dortmund, Germany

Abstract

We extract propositional conditional belief bases from multilayer perceptrons, a basic type of feedforward neural networks, and investigate the relation between these two prevalent formalisms from knowledge representation and reasoning (KRR) and machine learning (ML), respectively. The ultimate goal of our work is to imitate with the extracted belief base the main information flow in the original multilayer perceptron detached from specific input data. For this, we introduce a notion of sufficient (in)activators of neurons which reflect the most relevant connections within the multilayer perceptron that lead to the (in)activation of the subsequent neurons. While focusing on the binary multi-class classification task, we show that our approach produces consistent belief bases from which principled inferences can be drawn, for instance under System Z. In particular, no inferences are invented by the System Z ranking model that are not in accordance with the initial neural network.

Keywords

multilayer perceptrons, binary classification, belief base extraction, conditional reasoning, system Z

1. Introduction

Neural networks [1] are formal models studied in the research field of *machine learning (ML)* which have contributed significantly to the recent success of AI. In neural networks, input data is propagated through a network of neurons where neurons weight the received information and process it to the subsequent neurons. Neural networks are used in nearly every application domain with special abilities in *data processing*, *pattern recognition*, *data mining*, and, what is in the focus of this paper, *binary (multi-class) classification* [2]. A drawback of neural networks is that they appear as a black box methodology. Usually, it is not very transparent why input data leads to a specific output.

In contrast to neural networks, *knowledge-based systems* [3] from the field of *knowledge representation and reasoning (KRR)* typically provide a transparent and principled way of drawing inferences. A frequently used inference formalism, *System Z* [4], makes use of *conditionals* $(B|A)$ in order to represent defeasible statements of the form "if A holds, then usually B holds, too" [5, 6]. *Ranking functions* κ [7] like the *System Z ranking function* give such conditionals a clear semantics by assigning (im)plausibility values to sentences while postulating that the *verification* of a conditional $(B|A)$ is more plausible than its *falsification*, in symbols $\kappa(A \wedge B) < \kappa(A \wedge \neg B)$. The κ-ranks according to System Z are gained by penalizing possible worlds for falsifying conditionals, where the penalty points are the greater the more specific the falsified conditionals are. Alternative ranking semantics are provided by *System P* [8] and *c-representations* [9].

In this paper, we extract conditional belief bases from a specific type of neural networks called *multilayer perceptrons*. Multilayer perceptrons are *feedforward networks* in which information is always processed towards the output, hence there are no cycles in the network. In contrast to general feedforward networks, the neurons in multilayer perceptrons are arranged to at least three fully connected layers with neurons connected to the other neurons from the neighboring layers. The extracted belief base reflects the main information flow within such a multilayer perceptron.

The basic idea of our approach is to identify sets of predecessors of a neuron N the *(in)activation* of which is sufficient to (in)activate N. Hereby, the (in)activation of a neuron means that an input of the multilayer perceptron triggers the neuron more (less) than a predefined threshold, i.e., the output value of the neuron is larger (smaller) than this threshold. Therewith, our approach is related to the work in [10] which aims at identifying "most influential" neurons in neural networks, however without establishing logical connections between these neurons.

In more detail, the main contributions of the present paper are as follows:

- We introduce a notion of *sufficient (in)activators* of neurons (Definitions 6 and 7).
- We show that *sufficient (in)activators* are independent of the input of the multilayer perceptron (Propositions 2 and 3).
- Based on the notion of sufficient (in)activators, we extract belief bases from multilayer perceptrons (Definition 9). The extracted belief bases are provably consistent with respect to ranking semantics (Proposition 5).
- We use the extracted belief bases and their System Z ranking models for binary classification and relate their classification behavior to the direct classification with the initial multilayer perceptrons (Proposition 6).

With our approach we abstract from specific input data and also from overlay effects of less relevant connections in the neural networks. The most relevant connections are formalized in form of easy to understand conditionals. Note that establishing such formal bridges between neural- and logic-based models is a very old enterprise and has been pursued in the first papers on neural networks already [11].[1]

The rest of the paper is organized as follows. First we recall basics on multilayer perceptrons, in particular with respect to binary multi-class classification, and conditional

22nd International Workshop on Nonmonotonic Reasoning, November 2-4, 2024, Hanoi, Vietnam

marco.wilhelm@tu-dortmund.de (M. Wilhelm);
alexander.hahn@tu-dortmund.de (A. Hahn);
gabriele.kern-isberner@tu-dortmund.de (G. Kern-Isberner)
0000-0003-0266-2334 (M. Wilhelm); 0009-0008-6114-2594 (A. Hahn);
0000-0001-8689-5391 (G. Kern-Isberner)

[1] We thank the anonymous referees for their valuable comments.

Activation function	Specification	Range
Identity	$\phi(x) = x$	$\mathbb{R}$
Heaviside step	$\phi(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$	$\{0, 1\}$
Logistic function	$\phi(x) = \dfrac{1}{1 + e^{-x}}$	$(0, 1)$
Hyperbolic tangent	$\phi(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$	$(-1, 1)$
ReLU	$\phi(x) = \max(0, x)$	$\mathbb{R}_{\geq 0}$

Table 1
Typical activation functions of neural networks.

reasoning based on ranking functions (Section 2). Then, we discuss related work on extracting belief bases from multilayer perceptrons within a Description Logic context and show that a naïve translation to propositional conditional belief bases works only to a limited extent (Section 3). Eventually, we propose our novel approach on extracting belief bases based on sufficient (in)activators (Section 4) and use this approach for principled binary classification (Section 5). We close the paper with a conclusion that points to future work (Section 6).

2. Preliminaries

In this section, we recall preliminaries on multilayer perceptrons with an application to binary multi-class classification first (Section 2.1). Then, we explain basics on reasoning with conditionals, in particular based on System Z (Section 2.2).

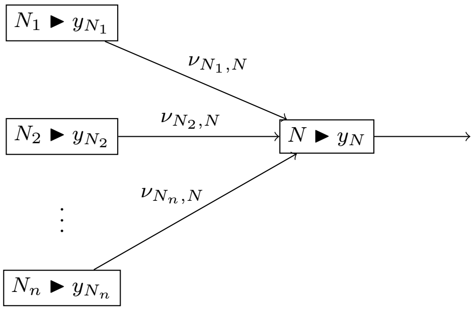
2.1. Multilayer Perceptrons for Binary Multi-Class Classification

Multilayer perceptrons (MLPs) constitute a widely used type of *neural networks* which expand single *perceptrons* to several fully connected layers. We give a brief introduction to neural networks in general and to MLPs in particular. Afterwards, we discuss their application to binary multi-class classification.

Neural Networks *Neural networks* [1] are formal models used to process information in form of data in modern AI systems. In the original sense, neural networks are functions $\mathcal{N} \colon \mathbb{R}^n \to \mathbb{R}^m$ where n is the size of the real-valued input vectors $\vec{x}$, and where m is the size of the real-valued output $\mathcal{N}(\vec{x})$. The computation of $\mathcal{N}(\vec{x})$ is specified by a weighted directed graph the nodes of which are called *neurons*. The functionality of neurons is as follows. Neurons N receive information encoded as real numbers y_{N_i} from their parent nodes/neurons $N_i \in \mathsf{pa}_N$, or the input vector $\vec{x}$ of the network, process this information based on an *activation function* $\phi_N \colon \mathbb{R} \to \mathbb{R}$ and possibly a *bias* $\beta_N \in \mathbb{R}$, and send the processed information

$$y_N = \phi_N(\beta_N + \sum_{N_i \in \mathsf{pa}_N} \nu_{N_i,N} \cdot y_{N_i})$$

to their child nodes/neurons. Hereby, $\nu_{N_i,N} \in \mathbb{R}$ is the weight of the edge from N_i to N (cf. Figure 1). Neurons without child nodes return the output of the neural network. Typical activation functions of neural networks are shown in Table 1. The weights of a neural network and the biases of the neurons are usually derived from *training data*, i.e., input data for which the expected output is known. Here, we solely consider neural networks which are already trained.



$N_1 \blacktriangleright y_{N_1}$
$\nu_{N_1,N}$
$N_2 \blacktriangleright y_{N_2}$
$\nu_{N_2,N}$
$N \blacktriangleright y_N$
$\nu_{N_n,N}$
$\vdots$
$N_n \blacktriangleright y_{N_n}$

Figure 1: Schema of a neuron N.

Multilayer Perceptrons In neural networks, neurons are usually assigned to *layers* with different functionalities. Neurons in the first layer, the *input layer*, receive the input of the network, and neurons in the last layer, the *output layer*, return the output. The layers in-between are called *hidden layers*. If a neural network is represented by an *acyclic* directed graph, it is called a *feedforward network*. In feedforward networks information is always processed towards the output layer. *Multilayer perceptrons* constitute an important subclass of feedforward networks with edges only between adjacent layers and, taking this condition into account, fully connected neurons. Multilayer perceptrons have at least one hidden layer. This hidden layer (as well as a non-linear activation function) is necessary to distinguish data that is not linearly separable [12].

Definition 1 (Multilayer Perceptron)**.** *A* multilayer perceptron $\mathcal{M}_\phi$ *is a special neural network which is represented by a directed graph* $(\mathcal{V}_{\mathcal{M}_\phi}, \mathcal{E}_{\mathcal{M}_\phi})$ *consisting of a set of vertices*

$$\mathcal{V}_{\mathcal{M}_\phi} = \{N_{i,j} \mid i \in [m], j \in [n_i]\},^{2}$$

the neurons in $\mathcal{M}_\phi$*, and a set of edges*

$$\mathcal{E}_{\mathcal{M}_\phi} = \{(N_{i,j}, N_{i+1,k}) \\ \mid i \in [m-1], j \in [n_i], k \in [n_{i+1}]\},$$

where $m \in \mathbb{N}_{\geq 2}$*, and* $n_i \in \mathbb{N}$ *for* $i \in [m]$*. Every edge* $(N_{i,j}, N_{i+1,k}) \in \mathcal{E}_{\mathcal{M}_\phi}$ *is assigned a real-valued weight* $\nu_{i,j,k} = \nu_{N_{i,j},N_{i+1,k}}$*, every neuron* $N_{0,j}$*,* $j \in [n_0]$*, in the input layer is assigned the identity function* $f_{0,j} \colon \mathbb{R} \to \mathbb{R}$ *with* $f_{0,j}(x) = x$*, and every further neuron* $N_{i,j}$ *with* $i > 0$*,* $j \in [n_i]$*, is assigned a function* $f_{N_{i,j}} \colon \mathbb{R}^{n_{i-1}+1} \to \mathbb{R}$ *with*

$$f_{N_{i,j}}(\vec{x}) = \phi(\beta_{i,j} + \sum_{h \in [n_{i-1}]} \nu_{i-1,h,j} \cdot f_{N_{i-1,h}}(\vec{x})), \quad (1)$$

where ϕ *is the activation function of* $\mathcal{M}_\phi$ *and* $\beta_{i,j} \in \mathbb{R}$ *is the* bias *of* $N_{i,j}$*. The* input *of* $\mathcal{M}_\phi$ *is any vector* $\vec{x} \in \mathbb{R}^{n_0+1}$ *whereby the* j*-th component of* $\vec{x}$ *is passed to the neuron* $N_{0,j}$*, and the* output *of* $\mathcal{M}_\phi$ *is*

$$\mathcal{M}_\phi(\vec{x}) = (f_{N_{m,0}}(\vec{x}), \ldots, f_{N_{m,n_m}}(\vec{x})) \in \mathbb{R}^{n_m+1}.$$

Figure 2 shows a schema of a multilayer perceptron with one hidden layer ($m = 2$). For a neuron $N \in \mathcal{M}_\phi$, we will denote the set of its parent nodes by pa_N which will help us to avoid indices.

[2] For $m \in \mathbb{N}$, we abbreviate $[m] = \{0, 1, \ldots, m\}$.

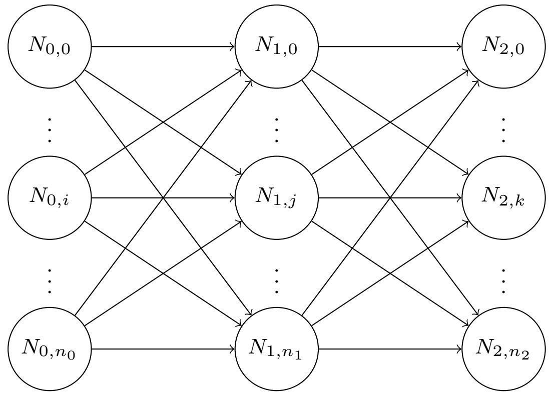


Figure 2: Multilayer perceptron with one hidden layer.

Binary Multi-Class Classification A possible application of neural networks in general and multilayer perceptrons in particular is binary (multi-class) classification [2]. For instance, the input $\vec{x}$ of a multilayer perceptron $\mathcal{M}_\phi$ could represent medical patient data, and we could ask for therapies that are suited to cure the patient. In the easiest case, the neurons in the output layer of $\mathcal{M}_\phi$ represent the different therapies and are equipped with the Heaviside step function as activation function ϕ such that $\mathcal{M}_\phi(\vec{x}) \in \{0,1\}^m$ for some $m \in \mathbb{N}$. Then, $y_i = 1$, where y_i is the outcome of neuron N_i in the output layer, can be interpreted as "the therapy N_i is suited to cure the patient represented by $\vec{x}$," and $y_1 = 0$ can be understood as the opposite.

In practice, one usually uses sigmoid functions like the logistic function (cf. Table 1) for classification, instead, which range over the interval $(0,1)$ and, thus, allow for a gradual answer behavior. Furthermore, the Heaviside function cannot be used for gradient-based training because it is not differentiable at 0 and the derivative is 0 at all other points, while the logistics function can be differentiated any number of times which makes it particularly suited for numerical methods. In this paper, we equip multilayer perceptrons with the logistic function as an activation function and denote this by $\mathcal{M}_{\log}$. Our approach works with any sigmoid function, though. We consider the following three-valued interpretation of the output of neurons in $\mathcal{M}_{\log}$.

Definition 2 ((In)active Neurons)**.** *Let $\mathcal{M}_{\log}$ be a multilayer perceptron, let N be a neuron in $\mathcal{M}_{\log}$, let $\vec{x}$ be an input vector of $\mathcal{M}_{\log}$, and let $\tau \in [0, 0.5)$. We call τ a* tolerance factor, *and say that neuron N is (cf. (1))*

- activated by $\vec{x}$ *wrt. τ, or* active *for short, iff*

$$f_N(\vec{x}) \geq 1 - \tau,$$

- inactivated by $\vec{x}$ *wrt. τ, or* inactive *for short, iff*

$$f_N(\vec{x}) \leq \tau,$$

- ambiguous *otherwise.*

With Definition 2, we can say that an input vector $\vec{x}$ of $\mathcal{M}_{\log}$ is *classified as an instance of class $\mathcal{C}_N$*, represented by the neuron N in the output layer of $\mathcal{M}_{\log}$, if N is activated by $\vec{x}$, and $\vec{x}$ is *declassified as an instance of class $\mathcal{C}_N$* if N is inactivated by $\vec{x}$. Otherwise, the membership to $\mathcal{C}_N$ is ambiguous. We give an example.

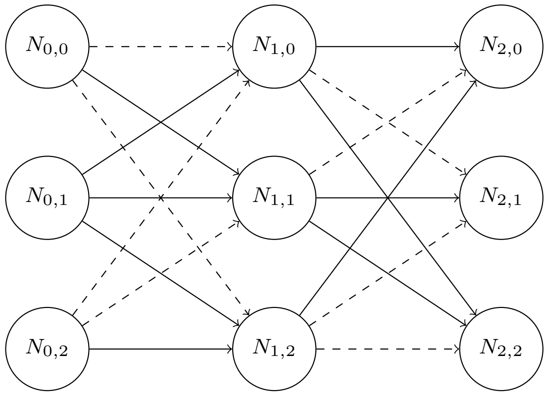


Figure 3: Multilayer perceptron from Example 1. Edges with negative weights are dashed.

$N_{i,j}$	$\nu_{i,j,0}$	$\nu_{i,j,1}$	$\nu_{i,j,2}$
$N_{0,0}$	-1.27	0.91	-0.44
$N_{0,1}$	1.23	0.81	0.27
$N_{0,2}$	-0.91	-0.09	1.96
$N_{1,0}$	1.62	-0.96	1.31
$N_{1,1}$	-1.19	1.15	1.46
$N_{1,2}$	0.14	-1.18	-0.14

Table 2
Weights of the multilayer perceptron from Example 1.

Example 1. *We consider the multilayer perceptron $\mathcal{M}^{\text{ex}}_{\log}$ from Figure 3 with the edge weights from Table 2 as a running example. Further, we assume that the neurons in $\mathcal{M}^{\text{ex}}_{\log}$ are unbiased ($\beta_{N_{i,j}} = 0$), and let $\tau = 0.3$. For instance, for the input vector $\vec{x} = (0.9, 0.8, 0.1)$, we obtain*

$$y_{N_{2,2}} \approx 0.844$$

so that $\vec{x}$ is classified as an instance of class $\mathcal{C}_{N_{2,2}}$ when the tolerance factor τ is equal to or greater than 0.156.

Besides the fact that sigmoid functions like the logistic function are common activation functions for classification tasks, we will utilize in some proofs that logistic functions are bounded between 0 and 1 (cf. the proofs of Propositions 2 and 3).

Definition 3 (Classification Scheme)**.** *Let $\mathcal{M}_{\log}$ be a multilayer perceptron with the logistic function as activation function, and let τ be a tolerance factor. Then, we call $(\mathcal{M}_{\log}, \tau)$ a* classification scheme.

Within our approach on extracting conditional belief bases from multilayer perceptrons, we will focus on the task of binary multi-class classification.

2.2. Conditionals and System Z

Within the field of *nonmonotonic reasoning*, *conditionals* [13] constitute a widely used representation of defeasible knowledge resp. beliefs. Here, we consider conditionals defined over a *propositional language* and interpret them via so-called *ranking functions*, in particular the System Z ranking model.

Conditional Reasoning Let $\mathcal{L}(\Sigma)$ be a *propositional language* defined over a finite signature Σ as usual.[3] A *conditional* $(B|A)$ with $A, B \in \mathcal{L}(\Sigma)$ is a formal representation of the defeasible statement: "If A holds, then usually B holds, too." Finite sets of conditionals serve as *belief bases*. The semantics of conditionals is based on *possible worlds*. Here, *possible worlds* $\omega \in \Omega(\Sigma)$ are the propositional interpretations of $\mathcal{L}(\Sigma)$ represented as complete conjunctions of literals. That is, every atom from Σ occurs in a possible world once, either positive or negated. A *ranking function* $\kappa\colon \Omega(\Sigma) \to \mathbb{N}_0 \cup \{\infty\}$ [7] maps possible worlds to a *degree of implausibility* while satisfying the normalization condition $\kappa^{-1}(0) \neq \emptyset$. The higher the *rank* $\kappa(\omega)$, the less plausible the possible world ω is. Hence, $\kappa^{-1}(0)$ is the set of the most plausible possible words. Ranking functions are extended to propositions via

$$\kappa(A) = \min_{\omega \in \Omega(\Sigma)\colon\, \omega \models A} \kappa(\omega)$$

and *accept* a conditional $(B|A)$ if $\kappa(AB) < \kappa(A\overline{B})$. A ranking function κ is a *ranking model* of a belief base Δ if κ accepts all conditionals in Δ. If Δ has a ranking model, then it is called *consistent*. Ranking models κ of Δ yield a nonmonotonic inference relation between Δ and conditionals $(B|A)$ in the following sense:

$$\Delta \mathrel{|\!\sim}_{\kappa} (B|A) \;\text{ iff }\; \kappa(AB) < \kappa(A\overline{B}) \;\text{ or }\; \kappa(A) = \infty.$$

System Z A sophisticated ranking model of consistent belief bases is provided by *System Z* [4] which is based on the notion of *tolerance*. A conditional $(B|A)$ is *tolerated* by a belief base Δ if there is a possible world ω such that $\omega \models AB$ ("the conditional $(B|A)$ is *verified* in ω") and $\omega \models A'B' \vee \overline{A'}$ for all conditionals $(B'|A')$ in Δ ("the conditional $(B'|A')$ is *verified* or *not applicable* in ω"). An ordered partition $(\Delta_0, \Delta_1, \ldots, \Delta_m)$ of Δ is called a *tolerance partition* of Δ if every conditional in Δ_0 is tolerated by Δ and $(\Delta_1, \ldots, \Delta_m)$ is a tolerance partition of $\Delta \setminus \Delta_0$. It is a well-known result that Δ is consistent iff Δ has a tolerance partition. If the partitioning sets are chosen inclusion maximally, beginning from Δ_0, then the resulting tolerance partition $Z(\Delta) = (\Delta_0, \Delta_1, \ldots, \Delta_m)$ is unique and called *Z-partition* of Δ. Via the *Z ranks* $Z_\Delta(\delta) = i$ of conditionals $\delta \in \Delta$ where i is the index of the partitioning set from $Z(\Delta)$ with $\delta \in \Delta_i$, the Z-partition of Δ allows one to define the following *System Z ranking model* of consistent belief bases Δ:

$$\kappa^Z_\Delta(\omega) = \begin{cases} 0 & \mathsf{fal}_\Delta(\omega) = \emptyset \\ 1 + \max_{\delta \in \mathsf{fal}_\Delta(\omega)} Z_\Delta(\delta) & \text{otherwise} \end{cases},$$

where $\omega \in \Omega(\Sigma)$, and $\mathsf{fal}_\Delta(\omega) = \{(B|A) \in \Delta \omega \models A\overline{B}\}$ is the set of conditionals *falsified* in ω.

Example 2. *A typical example to illustrate System Z is the Tweety example. Let* $\Delta = \{\delta_1, \delta_2, \delta_3\}$ *with*

$$\delta_1 = (b|p), \qquad \delta_2 = (f|b), \qquad \delta_3 = (\overline{f}|p),$$

state that penguins like Tweety are usually birds and birds usually fly, but penguins usually do not fly. The System Z tolerance partition of Δ *is* $Z(\Delta) = (\Delta_0, \Delta_1)$ *with*

$$\Delta_0 = \{\delta_2\}, \qquad \Delta_1 = \{\delta_1, \delta_3\}.$$

The resulting System Z ranking model is

$$\kappa^Z_\Delta(\omega) = \begin{cases} 0, & \omega \in \{bf\bar{p},\ \bar{b}f\bar{p},\ \bar{b}\overline{f}\bar{p}\} \\ 1, & \omega \in \{b\overline{f}\bar{p},\ b\overline{f}p\} \\ 2, & \omega \in \{bfp,\ \bar{b}fp,\ \bar{b}\overline{f}p\} \end{cases}.$$

System Z coincides with *rational closure* [14].

3. Related Work and Synaptic Conditionals

In this section, we briefly recall the extraction of beliefs from neural networks as presented in [15] and provide a naïve translation of this approach to propositional conditionals. We also discuss why this naïve translation is too simple to capture the essential streams of information of a neural network.

In [15], an extraction of belief bases from neural networks is proposed where the belief bases are defined over defeasible subsumptions of Description Logic concepts.[4] Neurons N_i are represented as atomic concepts C_i, and an edge from a neuron N_i to a neuron N_j is represented as the defeasible subsumption $\mathbf{T}(C_i) \sqsubseteq C_j$, expressing that input vectors $\vec{x}$ that typically activate N_i also activate N_j. This notion of representing the structure of a neural network using uncertain connections between atoms can be carried over to propositional conditional logic, utilizing atomic propositions A_i to represent neurons and conditionals $(A_i|A_j)$ to encode connections between them. Then, a (partial) possible world ω encodes a possible state of the neural network, with $\omega \models A_i$ ($\omega \models \overline{A_i}$) meaning that the neuron N_i is active (inactive) in the neural network. From another point of view, ω can be seen as a representation of all input vectors $\vec{x}$ that cause the same neurons to be (in)active. Together, the possible worlds in $\Omega(\Sigma)$ partition the set of input vectors based on their (abstracted) activation of neurons.

We formalize the extraction of propositional conditionals in analogy to the defeasible subsumptions in [15] now. For this, and in the rest of this paper, we will use the same symbol N to denote both a neuron in the neural network and the atomic proposition representing the neuron. Moreover,

$$\mathsf{pa}^+_N = \{N' \in \mathsf{pa}_N \mid \nu_{N',N} > 0\},$$
$$\mathsf{pa}^-_N = \{N' \in \mathsf{pa}_N \mid \nu_{N',N} < 0\},$$

denote the sets of the parent nodes N' of N within a neural network $\mathcal{N}$ with positive and negative weights $\nu_{N',N}$, respectively.

Definition 4 (Synaptic Conditionals). *Let* $\mathcal{N}$ *be a neural network. Then we define for each neuron* $N \in \mathcal{N}$ *the* backward synaptic conditionals *as follows:*

$$\Delta^+_{\leftarrow}(N) = \left\{(N'|N) \mid N' \in \mathsf{pa}^+_N\right\},$$
$$\Delta^-_{\leftarrow}(N) = \left\{(\overline{N'}|N) \mid N' \in \mathsf{pa}^-_N\right\}.$$

Analogously, we define forward synaptic conditionals*:*

$$\Delta^+_{\rightarrow}(N) = \left\{(N|N') \mid N' \in \mathsf{pa}^+_N\right\},$$
$$\Delta^-_{\rightarrow}(N) = \left\{(\overline{N}|N') \mid N' \in \mathsf{pa}^-_N\right\}.$$

Note that backward synaptic conditionals are abductive in nature. The idea of backward synaptic conditionals is that

[3] In order to shorten logical expressions, we use the abbreviations AB for conjunctions $A \wedge B$ and $\overline{A}$ for negations $\neg A$ where $A, B \in \mathcal{L}(\Sigma)$.

[4] Please see [16] for an introduction to Description Logics.

if a neuron N is active, the positive inputs of N must have outweighed the negative inputs of N (modulo the bias β_N). Therefore, it is plausible to assume that parents with positive connections are generally active, while parents with negative connections are generally inactive, even if exceptions are possible (and likely). Forward synaptic conditionals, on the other hand, are predictive: Given that a neuron N has an active parent with a positive connection (and without any additional information about the other parents), it is plausible to assume that this positive influence will cause N to be active as well.

We can now define belief bases containing synaptic conditionals.

Definition 5 (Synaptic Belief Bases). *Let $\mathcal{N}$ be a neural network. We define the* backward/forward synaptic belief bases *as the union of all synaptic conditionals that share the same direction, i.e.,*

$$\Delta^{\leftarrow}_{\mathcal{N}} = \bigcup_{N \in \mathcal{N}} \left(\Delta^{+}_{\leftarrow}(N) \cup \Delta^{-}_{\leftarrow}(N)\right),$$

$$\Delta^{\rightarrow}_{\mathcal{N}} = \bigcup_{N \in \mathcal{N}} \left(\Delta^{+}_{\rightarrow}(N) \cup \Delta^{-}_{\rightarrow}(N)\right).$$

The synaptic belief bases capture the information that is immediately available from the structure of the neural network, namely the positive or negative influence neurons have on each other based on the trained synaptic weights. From a formal perspective, the direction of the conditionals is arbitrary. As long as the two directions are not mixed, the synaptic belief base extracted from a multilayer perceptron is consistent.

Proposition 1. *For every multilayer perceptron $\mathcal{M}_\phi$, the synaptic belief bases $\Delta^{\leftarrow}_{\mathcal{M}_\phi}$ and $\Delta^{\rightarrow}_{\mathcal{M}_\phi}$ are consistent.*

Proof. We prove the proposition for $\Delta^{\leftarrow}_{\mathcal{M}_\phi}$ by showing that the layers of the multilayer perceptron $\mathcal{M}_\phi$ induce a tolerance partition of $\Delta^{\leftarrow}_{\mathcal{M}_\phi}$. Let $(m+1) \in \mathbb{N}$ be the number of layers in $\mathcal{M}_\phi$ and let $\mathcal{N}_i$ be the set of neurons in the i-th layer of $\mathcal{M}_\phi$. Then, $(\Delta_0, \ldots, \Delta_{m-1})$ defined by

$$\Delta_k = \{(\dot{N}'|N) \in \Delta^{\leftarrow}_{\mathcal{M}} \mid N \in \mathcal{N}_{k+1}\}$$

partitions $\Delta^{\leftarrow}_{\mathcal{M}_\phi}$. Now, we show that every conditional in Δ_k is tolerated by $\bigcup_{l:\ k \leq l < m} \Delta_l$. Let Δ_k and $N \in \mathcal{N}_{k+1}$ be arbitrary but fixed. We choose a possible world ω with the following properties: (1) $\omega \models N$, (2) $\omega \models N'$ if $(N'|N) \in \Delta_k$ for every $N' \in \mathcal{N}_k$, and (3) $\omega \models \overline{N''}$ for every $N'' \in \mathcal{N}_p$ with $k < p \leq m$ and $N \neq N''$. It can be quickly checked that all three properties concern different neurons and, hence, can be satisfied by ω at the same time. The properties (1) and (2) together ensure that ω verifies all conditionals with antecedent N; property (3) ensures that ω is indifferent with respect to all other conditionals in all Δ_l with $k \leq l < m$. Since Δ_k and N were chosen arbitrarily, this proves that every conditional in every Δ_k is tolerated by all Δ_l (with $0 \leq k \leq l < m$).

The proof for $\Delta^{\rightarrow}_{\mathcal{M}}$ is analogous; only the order of the partition needs to be reversed. □

In contrast to [15], which makes use of fuzzy Description Logics, the synaptic belief bases are purely qualitative representations of the connections in neural networks. Naturally, this means that all information about how strong individual connections between neurons are missing. The following example shows that this can lead to different inferences.

Example 3. *We consider the multilayer perceptron $\mathcal{M}^{\text{ex}}_{\log}$ from Example 1. The synaptic belief bases extracted from $\mathcal{M}^{\text{ex}}_{\log}$ are*

$$\begin{aligned}\Delta^{\leftarrow}_{\mathcal{M}^{\text{ex}}_{\log}} = \{&(\overline{N_{0,0}}|N_{1,0}), (N_{0,1}|N_{1,0}), (\overline{N_{0,2}}|N_{1,0}),\\ &(N_{0,0}|N_{1,1}), (N_{0,1}|N_{1,1}), (\overline{N_{0,2}}|N_{1,1}),\\ &(\overline{N_{0,0}}|N_{1,2}), (N_{0,1}|N_{1,2}), (N_{0,2}|N_{1,2}),\\ &(N_{1,0}|N_{2,0}), (\overline{N_{1,1}}|N_{2,0}), (N_{1,2}|N_{2,0}),\\ &(\overline{N_{1,0}}|N_{2,1}), (N_{1,1}|N_{2,1}), (\overline{N_{1,2}}|N_{2,1}),\\ &(N_{1,0}|N_{2,2}), (N_{1,1}|N_{2,2}), (\overline{N_{1,2}}|N_{2,2})\},\end{aligned}$$

and

$$\begin{aligned}\Delta^{\rightarrow}_{\mathcal{M}^{\text{ex}}_{\log}} = \{&(\overline{N_{1,0}}|N_{0,0}), (N_{1,1}|N_{0,0}), (\overline{N_{1,2}}|N_{0,0}),\\ &(N_{1,0}|N_{0,1}), (N_{1,1}|N_{0,1}), (N_{1,2}|N_{0,1}),\\ &(\overline{N_{1,0}}|N_{0,2}), (\overline{N_{1,1}}|N_{0,2}), (N_{1,2}|N_{0,2}),\\ &(N_{2,0}|N_{1,0}), (\overline{N_{2,1}}|N_{1,0}), (N_{2,2}|N_{1,0}),\\ &(\overline{N_{2,0}}|N_{1,1}), (N_{2,1}|N_{1,1}), (N_{2,2}|N_{1,1}),\\ &(N_{2,0}|N_{1,2}), (\overline{N_{2,1}}|N_{1,2}), (\overline{N_{2,2}}|N_{1,2})\}.\end{aligned}$$

In both cases (backward/forward), the Z-partition collapses:

$$Z(\Delta^{\leftarrow}_{\mathcal{M}^{\text{ex}}_{\log}}) = (\Delta^{\leftarrow}_{\mathcal{M}^{\text{ex}}_{\log}}), \quad Z(\Delta^{\rightarrow}_{\mathcal{M}^{\text{ex}}_{\log}}) = (\Delta^{\rightarrow}_{\mathcal{M}^{\text{ex}}_{\log}}),$$

and we have, with $\psi(N_{2,2}) = N_{0,0} \wedge N_{0,1} \wedge \overline{N_{0,2}}$,

$$\Delta \not\mid\!\sim_{\kappa^Z_\Delta} (N_{2,2}|\psi(N_{2,2}))$$

regardless of whether $\Delta = \Delta^{\leftarrow}_{\mathcal{M}^{\text{ex}}_{\log}}$ or $\Delta = \Delta^{\rightarrow}_{\mathcal{M}^{\text{ex}}_{\log}}$ because

$$\kappa^Z_\Delta(N_{2,2} \wedge \psi(N_{2,2})) = 1 \not< 0 = \kappa^Z_\Delta(\overline{N_{2,2}} \wedge \psi(N_{2,2}))$$

for $\Delta = \Delta^{\leftarrow}_{\mathcal{M}^{\text{ex}}_{\log}}$, and

$$\kappa^Z_\Delta(N_{2,2} \wedge \psi(N_{2,2})) = 1 \not< 1 = \kappa^Z_\Delta(\overline{N_{2,2}} \wedge \psi(N_{2,2}))$$

for $\Delta = \Delta^{\rightarrow}_{\mathcal{M}^{\text{ex}}_{\log}}$. Thus, In both cases this contradicts the fact that the input vector $\vec{x} = (0.9, 0.8, 0.1)$ triggers the neurons $N_{0,0}$, $N_{0,1}$, and $\overline{N_{0,2}}$ and is classified as an instance of$\mathcal{C}_{N_{2,2}}$ by $\mathcal{M}^{\text{ex}}_{\log}$ (cf. Example 1). Hence, we come to different conclusions if we either classify $\vec{x} = (0.9, 0.8, 0.1)$ by $\mathcal{M}^{\text{ex}}_{\log}$ directly or classify $\vec{x}$ based on the synaptic belief bases.

The example above shows that belief bases consisting of synaptic conditionals (only) are too basic to give any guarantees with respect to reasoning behavior when using System Z. It is to be expected that a qualitative belief base cannot provide inferences on the same level of detail like the original neural network. The example also shows that the belief base introduces new inferences which cannot be obtained from the neural network. This can be considered undesirable. Therefore, in order to make better use of the quantitative information learned by the neural network, we make the extracted conditionals more complex to capture relevant influences among the neurons better in the next section.

4. Sufficient (In)activators for Belief Base Extraction

Now, we propose a more sophisticated approach than synaptic conditionals for extracting conditional belief bases from multilayer perceptrons. On the one hand, this means an abstraction from specific input data to generalized defeasible rules, here conditionals. On the other hand, the embedding of the essential information flow of multilayer perceptrons into a logical framework allows us to draw principled inferences of verifiable quality.

4.1. Basic Idea and Preconditions

The basic idea of our method is to extract conditionals $\delta_N^{\tau;+} = (N|\psi_N^{\tau;+})$ from a multilayer perceptron $\mathcal{M}_{\log}$ where the *consequence* N refers to a neuron from $\mathcal{M}_{\log}$ and the *premise* $\psi_N^{\tau;+}$ to sets of parent nodes of N which are (in combination) "most relevant" for the activation of N. Relevance here means that the conditional $(N|\psi_N^{\tau;+})$ is effective, i.e., $\psi_N^{\tau;+}$ is true, only if it is guaranteed that the neuron N is sufficiently highly activated. Hence, it is reliably justified to infer N. Analogously, we extract conditionals $\delta_N^{\tau;-} = (\overline{N}|\psi_N^{\tau;-})$ wrt. the inactivation of N. The "most relevant" parents nodes of neurons in $\mathcal{M}_{\log}$ are identified based on the notion of *sufficient (in)activators.*

We assume that the input of the multilayer perceptron $\mathcal{M}_{\log}$ is normalized to $\vec{x} \in [0,1]^n$ and that the activation function used in $\mathcal{M}_{\log}$ is the logistic function which ensures that the output of all neurons in $\mathcal{M}_{\log}$ is within the range $[0,1]$ again. Given a tolerance factor τ, this allows for an interpretation of the activation of all neurons in $\mathcal{M}_{\log}$ as in Definition 2.

4.2. Sufficient (In)activators

Based on the concept of active and inactive neurons, we define (sets of) parent nodes of neurons in a multilayer perceptron $\mathcal{M}_{\log}$ which are sufficient to activate resp. deactivate the neurons, independent of the specific input vector $\vec{x}$.

Definition 6 (Sufficient Activator). *Let $(\mathcal{M}_{\log}, \tau)$ be a classification scheme. Further, let N be a neuron in $\mathcal{M}_{\log}$ from a hidden layer or the output layer. We call a tuple $(A^+, A^-) \subseteq \mathsf{pa}_N^2$ with $A^+ \cap A^- = \emptyset$ a* sufficient activator *of N wrt. τ, if the activation of the neurons in A^+ and the inactivation of the neurons in A^- implies the activation of N; formally, if $y_{N'} \geq 1-\tau$ for $N' \in A^+$ and $y_{N'} \leq \tau$ for $N' \in A^-$ implies*

$$\phi(\beta_N + \sum_{N' \in \mathsf{pa}_N} \nu_{N',N} \cdot y_{N'}) \geq 1 - \tau.$$

We denote the set of the sufficient activators of N wrt. τ by $\mathcal{SA}^\tau(N)$.

The idea of the sufficient activators in $\mathcal{SA}^\tau(N)$ is that the output of the neurons $N' \in \mathsf{pa}_N$ with $N' \notin A^+ \cup A^-$ is irrelevant for the activation of N, regardless of the concrete input of $\mathcal{M}_{\log}$, as captured in the next proposition.

Proposition 2. *Let $(\mathcal{M}_{\log}, \tau)$ be a classification scheme, and let N be a neuron in $\mathcal{M}_{\log}$ from a hidden layer or the output layer. Then, $(A^+, A^-) \subseteq \mathsf{pa}_N^2$ with $A^+ \cap A^- = \emptyset$ is a sufficient activator of N iff*

$$\begin{aligned}\phi(\beta_N &+ (1-\tau) \cdot \textstyle\sum_{N' \in \mathsf{pa}_N^+ \cap A^+} \nu_{N',N} \\ &+ \tau \cdot \textstyle\sum_{N' \in \mathsf{pa}_N^- \cap A^-} \nu_{N',N} \\ &+ \textstyle\sum_{N' \in \mathsf{pa}_N^- \setminus A^-} \nu_{N',N} \qquad) \geq 1 - \tau.\end{aligned} \tag{2}$$

Proof. ($\Leftarrow$) Assume that (2) and $y_{N'} \geq 1-\tau$ for $N' \in A^+$ and $y_{N'} \leq \tau$ for $N' \in A'$ hold. Then,

$$\begin{aligned}&\phi(\beta_N + \textstyle\sum_{N' \in \mathsf{pa}_N} \nu_{N',N} \cdot y_{N'}) \\ =\ &\phi(\beta_N + \textstyle\sum_{N' \in \mathsf{pa}_N^+ \cap A^+} \nu_{N',N} \cdot y_{N'} \\ &\qquad + \textstyle\sum_{N' \in \mathsf{pa}_N^+ \setminus A^+} \nu_{N',N} \cdot y_{N'} \\ &\qquad + \textstyle\sum_{N' \in \mathsf{pa}_N^- \cap A^-} \nu_{N',N} \cdot y_{N'} \\ &\qquad + \textstyle\sum_{N' \in \mathsf{pa}_N^- \setminus A^-} \nu_{N',N} \cdot y_{N'}) \\ \geq\ &\phi(\beta_N + (1-\tau) \cdot \textstyle\sum_{N' \in \mathsf{pa}_N^+ \cap A^+} \nu_{N',N} \\ &\qquad + \tau \cdot \textstyle\sum_{N' \in \mathsf{pa}_N^- \cap A^-} \nu_{N',N} \\ &\qquad + \textstyle\sum_{N' \in \mathsf{pa}_N^- \setminus A^-} \nu_{N',N}) \\ \geq\ &1 - \tau.\end{aligned}$$

Hereby, we used $\sum_{N' \in \mathsf{pa}_N^+ \setminus A^+} \nu_{N',N} \cdot y_{N'} \geq 0$. Thus, (A^+, A^-) is a sufficient activator of N.

($\Rightarrow$) We prove the contraposition. Assume that

$$\begin{aligned}\phi(\beta_N + (1-\tau) \cdot &\sum_{N' \in \mathsf{pa}_N^+ \cap A^+} \nu_{N',N} \\ + \tau \cdot \sum_{N' \in \mathsf{pa}_N^- \cap A^-} \nu_{N',N} + &\sum_{N' \in \mathsf{pa}_N^- \setminus A^-} \nu_{N',N}) < 1 - \tau\end{aligned}$$

holds. We have to show that there is $y_{N'} \in [0,1]$ for $N' \in \mathsf{pa}_N$ with $y_{N'} \geq 1-\tau$ for $N' \in A^+$ and $y_{N'} \leq \tau$ for $N' \in A^-$ such that

$$\phi(\beta_N + \sum_{N' \in \mathsf{pa}_N} \nu_{N',N} \cdot y_{N'}) < 1 - \tau.$$

With

$$y_{N'} = \begin{cases} 1-\tau & \text{if } N' \in \mathsf{pa}_N^+ \cap A^+ \\ 0 & \text{if } N' \in \mathsf{pa}_N^+ \setminus A^+ \\ \tau & \text{if } N' \in \mathsf{pa}_N^- \cap A^- \\ 1 & \text{if } N' \in \mathsf{pa}_N^- \setminus A^- \\ 0 & \text{if } N' \in \mathsf{pa}_N \setminus (\mathsf{pa}_N^+ \cup \mathsf{pa}_N^-) \end{cases}$$

it follows that

$$\begin{aligned}&\phi(\beta_N + \textstyle\sum_{N' \in \mathsf{pa}_N} \nu_{N',N} \cdot y_{N'}) \\ =\ &\phi(\beta_N + \textstyle\sum_{N' \in \mathsf{pa}_N^+ \cap A^+} \nu_{N',N} \cdot y_{N'} \\ &\qquad + \textstyle\sum_{N' \in \mathsf{pa}_N^+ \setminus A^+} \nu_{N',N} \cdot y_{N'} \\ &\qquad + \textstyle\sum_{N' \in \mathsf{pa}_N^- \cap A^-} \nu_{N',N} \cdot y_{N'} \\ &\qquad + \textstyle\sum_{N' \in \mathsf{pa}_N^- \setminus A^-} \nu_{N',N} \cdot y_{N'}) \\ =\ &\phi(\beta_N + (1-\tau) \cdot \textstyle\sum_{N' \in \mathsf{pa}_N^+ \cap A^+} \nu_{N',N} \\ &\qquad + \tau \cdot \textstyle\sum_{N' \in \mathsf{pa}_N^- \cap A^-} \nu_{N',N} \\ &\qquad + \textstyle\sum_{N' \in \mathsf{pa}_N^- \setminus A^-} \nu_{N',N}) \\ <\ &1 - \tau,\end{aligned}$$

which finishes the proof. Note that the choice of $y_{N'} = 0$ in case of $N' \in \mathsf{pa}_N \setminus (\mathsf{pa}_N^+ \cup \mathsf{pa}_N^-)$ is not mandatory because $\nu_{N',N} = 0$ holds in this case anyway.

In this proof of Proposition 2 we have exploited that the logistic function is non-negative. If one wants to apply similar techniques to arbitrary sigmoid functions which are not necessarily non-negative but bounded by $(a,b) \subset \mathcal{R}$ one can rewrite $\phi(\beta_N + \sum_{N' \in \mathsf{pa}_N} \nu_{N',N} \cdot y_{N'})$ beforehand to

$$\phi_N((b-a)(\beta'_N + \sum_{N' \in \mathsf{pa}_N} \nu_{N',N} \cdot y'_{N'}))$$

with $\beta'_N = \frac{1}{b-a} \cdot (\beta_N + a \cdot \sum_{N_i \in \mathsf{pa}_N} \nu_{N',N})$ and $y'_{N'} = \frac{y_{N'} - a}{b-a}$ where $y'_{N'}$ is bounded by $(0,1)$ for all $N_i \in \mathsf{pa}_N$.

Note that in this case the thresholds for neurons being (in)active have to be adjusted from $1 - \tau$ and τ to $b - \tau$ and $a + \tau$ as well, now with $\tau \in [0, \frac{b-a}{2})$. □

Proposition 2 can be used to compute sufficient activators. For a neuron N one generates each pair (A^+, A^-) with $A^+ \in \mathsf{pa}_N^+$ and $A^- \in \mathsf{pa}_N^-$ and tests whether (2) holds or not.

Example 4. *We consider the multilayer perceptron $\mathcal{M}_{\log}^{\mathsf{ex}}$ from Example 1 (cf. Table 2) and the tolerance factor $\tau = 0.3$. Then, for instance, $(\{N_{0,0}, N_{0,1}\}, \emptyset)$ is a sufficient activator of $N_{1,1}$ because*

$$\phi(0.7 \cdot (0.91 + 0.81) - 0.09) \approx 0.753 \geq 0.7,$$

where ϕ is the logistic function (cf. Table 1). Note that $(\{N_{0,0}\}, \emptyset)$ is not a sufficient activator of $N_{1,1}$, instead, because

$$\phi(0.7 \cdot (0.91) - 0.09) \approx 0.633 < 0.7.$$

Analogously to sufficient activators, we can define *sufficient inactivators.*

Definition 7 (Sufficient Inactivator)**.** *Let $(\mathcal{M}_{\log}, \tau)$ be a classification scheme, and let N be a neuron in $\mathcal{M}$ from a hidden layer or the output layer. We call a tuple $(I^+, I^-) \subseteq \mathsf{pa}_N^2$ with $I^+ \cap I^- = \emptyset$ a* sufficient inactivator *of N wrt. τ if, the activation of the neurons in I^+ and the inactivation of the neurons in I^- implies the inactivation of N; formally, if $y_{N'} \geq 1 - \tau$ for $N' \in I^+$ and $y_{N'} \leq \tau$ for $N' \in I^-$ implies*

$$\phi(\beta_N + \sum_{N' \in \mathsf{pa}(N)} \nu_{N',N} \cdot y_{N'}) \leq \tau.$$

We denote the set of the sufficient inactivators of N wrt. τ by $\mathcal{SI}^\tau(N)$.

Similar to sufficient activators, the idea of sufficient inactivators (I^+, I^-) of neurons N is that the output of the neurons $N' \in \mathsf{pa}_N$ with $N' \notin I^+ \cup I^-$ is irrelevant for the inactivation of N, regardless of the concrete input of $\mathcal{M}_{\log}$.

Proposition 3. *Let $(\mathcal{M}_{\log}, \tau)$ be a classification scheme, and let N be a neuron in $\mathcal{M}_{\log}$ from a hidden layer or the output layer. Then, $(I^+, I^-) \subseteq \mathsf{pa}_N^2$ with $I^+ \cap I^- = \emptyset$ is a sufficient inactivator of N iff*

$$\begin{aligned} \phi(\beta_N &+ \tau \cdot \textstyle\sum_{N' \in \mathsf{pa}_N^+ \cap I^-} \nu_{N',N} \\ &+ \textstyle\sum_{N' \in \mathsf{pa}_N^+ \setminus I^-} \nu_{N',N} \\ &+ (1 - \tau) \cdot \textstyle\sum_{N' \in \mathsf{pa}_N^- \cap I^+} \nu_{N',N} \,) \leq \tau. \end{aligned} \tag{3}$$

Proof. The proof is similar to the proof of Proposition 2. For the direction ($\Leftarrow$) note that $\sum_{N' \in \mathsf{pa}_N^- \setminus I^+} \nu_{N',N} \cdot y_{N'} \leq 0$. For the proof of the contraposition of ($\Rightarrow$), we select

$$y_{N'} = \begin{cases} \tau & \text{if } N' \in \mathsf{pa}_N^+ \cap I^- \\ 1 & \text{if } N' \in \mathsf{pa}_N^+ \setminus I^- \\ 1 - \tau & \text{if } N' \in \mathsf{pa}_N^- \cap I^+ \\ 0 & \text{if } N' \in \mathsf{pa}_N^- \setminus I^+ \\ 0 & \text{if } N' \in \mathsf{pa}_N \setminus (\mathsf{pa}_N^+ \cup \mathsf{pa}_N^-) \end{cases} \quad □$$

Again, this proposition can be used to compute sufficient inactivators as the next example shows.

Example 5. *Again, we consider the multilayer perceptron $\mathcal{M}_{\log}^{\mathsf{ex}}$ from Example 1 (cf. Table 2) and the tolerance factor $\tau = 0.3$. Then, $(\{N_{0,0}, N_{0,2}\}, \{N_{0,1}\})$ is a sufficient inactivator of $N_{1,0}$ because*

$$\phi(0.7 \cdot (-1.27 - 0.91) + 0.3 \cdot 1.23) \approx 0.239 \leq 0.3,$$

where ϕ is the logistic function (cf. Table 1). Note that $(\{N_{0,0}, N_{0,2}\}, \emptyset)$ is not a sufficient inactivator of $N_{1,0}$ because

$$\phi(0.7 \cdot (-1.27 - 0.91) + 1.23) \approx 0.427 > 0.3.$$

For tuples of sets (S_1, S_2) and (T_1, T_2) we write $(S_1, S_2) \sqsubseteq (T_1, T_2)$ iff $S_1 \subseteq T_1$ and $S_2 \subseteq T_2$. Obviously, if (A^+, A^-) is a sufficient activator of N and, for $(A'^+, A'^-) \in \mathsf{pa}(N)^2$, $(A^+, A^-) \sqsubseteq (A'^+, A'^-)$ holds, then (A'^+, A'^-) is a sufficient activator of N, too. A similar result holds for sufficient inactivators.

Proposition 4. *Let $(\mathcal{M}_{\log}, \tau)$ be a classification scheme, and let N be a neuron in $\mathcal{M}_{\log}$ from a hidden layer or the output layer. Then,*

- *if (A^+, A^-) is a sufficient activator of N, then $(A'^+, A'^-) \in \mathsf{pa}_N^2$ with $(A^+, A^-) \sqsubseteq (A'^+, A'^-)$ is a sufficient activator of N, too,*
- *if (I^+, I^-) is a sufficient inactivator of N, then $(I'^+, I'^-) \in \mathsf{pa}_N^2$ with $(I^+, I^-) \sqsubseteq (I'^+, I'^-)$ is a sufficient inactivator of N, too.*

Proof. Let (A^+, A^-) be a sufficient activator of N, and let $(A'^+, A'^-) \in \mathsf{pa}_N^2$ with $(A^+, A^-) \sqsubseteq (A'^+, A'^-)$. Further, let $y_{N'} \geq 1 - \tau$ for $N' \in A'^+$ and $y_{N'} \leq \tau$ for $N' \in A'^-$. From $A^+ \subseteq A'^+$ and $A^- \subseteq A'^-$ it follows that $y_{N'} \geq 1 - \tau$ for $N' \in A^+$ and $y_{N'} \leq \tau$ for $N' \in A^-$ holds as well. Then, because (A^+, A^-) is a sufficient activator,

$$\phi(\beta_N + \sum_{N' \in \mathsf{pa}_N} \nu_{N',N} \cdot y_{N'}) \geq 1 - \tau.$$

The proof for sufficient inactivators is analogous. □

Proposition 4 suggests to define *minimal* sufficient (in)activators.

Definition 8 (Minimal Sufficient (In)activators)**.** *Let $(\mathcal{M}_{\log}, \tau)$ be a classification scheme, and let N be a neuron in $\mathcal{M}_{\log}$ from a hidden layer or the output layer. Then,*

- *A sufficient activator (A^+, A^-) of N is* minimal *if no $(A'^+, A'^-) \in \mathsf{pa}_N^2$ with $(A'^+, A'^-) \sqsubseteq (A^+, A^-)$ and $(A'^+, A'^-) \neq (A^+, A^-)$ is a sufficient activator of N,*
- *A sufficient inactivator (I^+, I^-) of N is* minimal *if no $(I'^+, I'^-) \in \mathsf{pa}_N^2$ with $(I^+, I^-) \sqsubseteq I'^+, I'^-)$ and $(I^+, I^-) \neq (I'^+, I'^-)$ is a sufficient inactivator of N.*

We denote the set of the minimal sufficient activators of N wrt. τ with $\mathcal{SA}^\tau_{\min}(N)$ and the set of the minimal sufficient inactivators of N wrt. τ with $\mathcal{SI}^\tau_{\min}(N)$.

We consider our running example.

Example 6. *The minimal sufficient (in)activators of the neurons in $\mathcal{M}_{\log}^{\mathsf{ex}}$ from Example 1 (cf. Table 2) with respect to the tolerance factor $\tau = 0.3$ are shown in Table 3 resp. Table 4. Minimal sufficient (in)activators allow for a graphical representation (cf. Figure 4). For instance, the minimal*

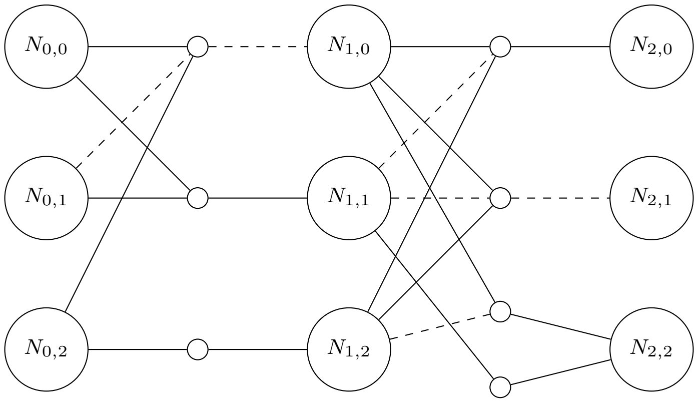


Figure 4: Minimal sufficient (in)activators of the neurons in $\mathcal{M}^{\text{ex}}_{\text{log}}$ from Example 1. Solid lines indicate activation and dashed lines inactivation.

$N_{i,j}$	$\mathcal{SA}^{\tau}_{\min}(N_{i,j})$
$N_{1,0}$	$\emptyset$
$N_{1,1}$	$\{(\{N_{0,0}, N_{0,1}\}, \emptyset)\}$
$N_{1,2}$	$\{(\{N_{0,2}\}, \emptyset)\}$
$N_{2,0}$	$\{(\{N_{1,0}, N_{1,2}\}, \{N_{1,1}\})\}$
$N_{2,1}$	$\emptyset$
$N_{2,2}$	$\{(\{N_{1,0}\}, \{N_{1,2}\}), (\{N_{1,1}\}, \emptyset)\}$

Table 3
Minimal sufficient activators of the neurons in the hidden resp. output layer of $\mathcal{M}^{\text{ex}}_{\text{log}}$ from Example 1 wrt. $\tau = 0.3$.

$N_{i,j}$	$\mathcal{SI}^{\tau}_{\min}(N_{i,j})$
$N_{1,0}$	$\{(\{N_{0,0}, N_{0,2}\}, \{N_{0,1}\})\}$
$N_{1,1}$	$\emptyset$
$N_{1,2}$	$\emptyset$
$N_{2,0}$	$\emptyset$
$N_{2,1}$	$\{(\{N_{1,0}, N_{1,2}\}, \{N_{1,1}\})\}$
$N_{2,2}$	$\emptyset$

Table 4
Minimal sufficient inactivators of the neurons in the hidden resp. output layer of $\mathcal{M}^{\text{ex}}_{\text{log}}$ from Example 1 wrt. $\tau = 0.3$.

sufficient inactivator $(\{N_{0,0}, N_{0,2}\}, \{N_{0,1}\})$ *of* $N_{1,0}$ *can be visualized as three outgoing edges from* $N_{0,0}$*,* $N_{0,1}$*, and* $N_{0,2}$*, respectively, which conjointly result in* $N_{1,0}$*. The dashed line in Figure 4 after these three edges have met indicates that* $(\{N_{0,0}, N_{0,2}\}, \{N_{0,1}\})$ *is a sufficient inactivator (and not an activator) of* $N_{1,0}$ *and the dashed line from* $N_{0,1}$ *indicates that* $N_{0,1}$ *has a negative influence on the inactivation of* $N_{1,0}$ *(because the weight* $\nu_{0,1,0}$ *is positive).*

Altogether, (minimal) sufficient activators and inactivators make it possible to abstract from the concrete input data of a multilayer perceptron $\mathcal{M}_{\text{log}}$ and reveal the essential streams of information within $\mathcal{M}_{\text{log}}$. This is the motivation for our following extraction of conditional belief bases from multilayer perceptrons.

4.3. Belief Base Extraction

Now, we describe our approach on extracting a conditional belief base $\Delta^{\tau}_{\mathcal{M}_{\text{log}}}$ from a multilayer perceptron $\mathcal{M}_{\text{log}}$ based on sufficient (in)activators. In $\Delta^{\tau}_{\mathcal{M}_{\text{log}}}$ we formalize for every neuron N in $\mathcal{M}_{\text{log}}$ its relationship to its sufficient (in)activators by a conditional which states that if the neurons in one of the sufficient activators (inactivators) of N are (in)activated, then the neuron N is usually active (inactive), too.

Definition 9 (Belief Base $\Delta^{\tau}_{\mathcal{M}_{\text{log}}}$). *Let* $(\mathcal{M}_{\text{log}}, \tau)$ *be a classification scheme, and let* N *be a neuron from a hidden layer or the output layer of* $\mathcal{M}_{\text{log}}$*. Then, we define the conditionals* $\delta^{\tau;+}_N = (N|\psi^{\tau;+}_N)$ *and* $\delta^{\tau;-}_N = (\overline{N}|\psi^{\tau;-}_N)$ *via*

$$\psi^{\tau;+}_N = \bigvee_{(A^+,A^-)\in\mathcal{SA}^{\tau}_{\min}(N)} \left(\bigwedge_{N'\in A^+} N' \wedge \bigwedge_{N'\in A^-} \overline{N'} \right),$$

$$\psi^{\tau;-}_N = \bigvee_{(I^+,I^-)\in\mathcal{SI}^{\tau}_{\min}(N)} \left(\bigwedge_{N'\in I^+} N' \wedge \bigwedge_{N'\in I^-} \overline{N'} \right),$$

provided that

$$\begin{aligned} &\mathcal{SA}^{\tau}_{\min}(N) \neq \emptyset \textit{ in case of } \delta^{\tau;+}_N, \\ &\mathcal{SI}^{\tau}_{\min}(N) \neq \emptyset \textit{ in case of } \delta^{\tau;-}_N. \end{aligned} \qquad (*)$$

Note that the conditionals depend on the tolerance factor τ *because the sets of (minimal) sufficient (in)activators depend on* τ*. However, the conditionals are not dependent on any input vector of* $\mathcal{M}_{\text{log}}$*, since* τ *abstracts from that. Based on that, we define the extraction of the belief base* $\Delta^{\tau}_{\mathcal{M}_{\text{log}}}$ *from* $\mathcal{M}_{\text{log}}$ *via*

$$\Delta^{\tau}_{\mathcal{M}_{\text{log}}} = \{\delta^{\tau;+}_N \mid N \in \mathcal{N}^+\} \cup \{\delta^{\tau;-}_N \mid N \in \mathcal{N}^-\},$$

where $\mathcal{N}^{\tau,+}$ *is the set of neurons* N *for which the conditional* $\delta^{\tau;+}_N$ *exists, and where* $\mathcal{N}^{\tau,-}$ *is the set of neurons* N *for which the conditional* $\delta^{\tau;-}_N$ *exists, i.e.,* $(*)$ *applies.*

The number of conditionals in $\Delta^{\tau}_{\mathcal{M}_{\text{log}}}$ is bounded by the number of neurons in $\mathcal{M}_{\text{log}}$ (minus the input layer) which means a higher degree of abstraction than prevalent in synaptic belief bases (cf. Definition 5) the cardinality of which is bounded by the number of edges in $\mathcal{M}_{\text{log}}$. Furthermore, the condition $(*)$ in Definition 9 ensures that the conditionals $\delta^{\tau;+}_N$ (resp. $\delta^{\tau;-}_N$) are added to $\Delta^{\tau}_{\mathcal{M}_{\text{log}}}$ only if N has sufficient activators (inactivators). This prevents from conditionals of the form $(N|\bot)$ and $(\overline{N}|\bot)$ in $\Delta^{\tau}_{\mathcal{M}_{\text{log}}}$ which would cause inconsistencies according to our acceptance definition of conditionals. If there is a neuron N with $\delta^{\tau;+}_N, \delta^{\tau;-}_N \notin \Delta^{\tau}_{\mathcal{M}_{\text{log}}}$, then one can increase τ in order to improve the chance of obtaining such a conditional.

Example 7. *We consider $\mathcal{M}_{\log}^{\mathsf{ex}}$ from Example 1 and the tolerance factor $\tau = 0.3$. The minimal sufficient (in)activators of the neurons in $\mathcal{M}_{\log}^{\mathsf{ex}}$ are shown in Table 3 resp. Table 4 from which we can derive the belief base $\Delta_{\mathcal{M}_{\log}}^{0.3}$. The conditionals in $\Delta_{\mathcal{M}_{\log}}^{0.3}$ are*

$$\begin{aligned}
\delta_{N_{1,0}}^{0.3,-} &= (\overline{N_{1,0}}|N_{0,0} \wedge N_{0,2} \wedge \overline{N_{0,1}}),\\
\delta_{N_{1,1}}^{0.3,+} &= (N_{1,1}|N_{0,0} \wedge N_{0,1}),\\
\delta_{N_{1,2}}^{0.3,+} &= (N_{1,2}|N_{0,2}),\\
\delta_{N_{2,0}}^{0.3,+} &= (N_{2,0}|N_{1,0} \wedge N_{1,2} \wedge \overline{N_{1,1}}),\\
\delta_{N_{2,1}}^{0.3,-} &= (\overline{N_{2,1}}|N_{1,0} \wedge N_{1,2} \wedge \overline{N_{1,1}}),\\
\delta_{N_{2,2}}^{0.3,+} &= (N_{2,2}|N_{1,0} \wedge \overline{N_{1,2}} \vee N_{1,1}).
\end{aligned}$$

In particular, note the disjunction in the premise of $\delta_{N_{2,2}}^{0.3,+}$ because of the two (different) minimal sufficient activators of $N_{2,2}$.

The belief base $\Delta_{\mathcal{M}_{\log}}^{\tau}$ is consistent. To show this, we make use of the following lemma.

Lemma 1. *Let $(\mathcal{M}_{\log}, \tau)$ be a classification scheme. Then, for every neuron N from a hidden layer or the output layer of $\mathcal{M}_{\log}$ it holds that (cf. Definition 9)*

$$\psi_N^{\tau,+} \wedge \psi_N^{\tau,-} \equiv \bot.$$

Proof. Assume that $\psi_N^{\tau,+} \wedge \psi_N^{\tau,-} \not\equiv \bot$ holds. Then, there is a possible world ω, a sufficient activator (A^+, A^-) of N wrt. τ, and a sufficient inactivator (I^+, I^-) of N wrt. τ such that

$$\omega \models \bigwedge_{N' \in A^+} N' \wedge \bigwedge_{N' \in A^-} \overline{N'} \wedge \bigwedge_{N' \in I^+} N' \wedge \bigwedge_{N' \in I^-} \overline{N'}.$$

It follows that $(A^+ \cup I^+) \cap (A^- \cup I^-) = \emptyset$. Otherwise, ω would mention an atom both negated and positive. From this and Proposition 4 it follows that $(A^+ \cup I^+, A^- \cup I^-)$ is both a sufficient activator and a sufficient inactivator of N wrt. τ because $(A^+, A^-) \sqsubseteq (A^+ \cup I^+, A^- \cup I^-)$ and $(I^+, I^-) \sqsubseteq (A^+ \cup I^+, A^- \cup I^-)$ hold. According to the definitions of sufficient (in)activators, for appropriate values $y_{N'}$ for $N' \in \mathsf{pa}(N)$,

$$1 - \tau \leq \phi(\beta_N + \sum_{N' \in \mathsf{pa}_N} \nu_{N',N} \cdot y_{N'}) \leq \tau$$

follows. This implies $1 - \tau \leq \tau$ or, equivalent $0.5 \leq \tau$, which contradicts $\tau \in [0, 0.5)$. □

Lemma 1 states that there is no neuron N in $\mathcal{M}_{\log}$ for which both $\delta_N^{\tau,+}$ (supporting N) and $\delta_N^{\tau,-}$ (supporting $\overline{N}$) can be applicable at the same time.

Proposition 5. *Let $(\mathcal{M}_{\log}, \tau)$ be a classification scheme. Then, the belief base $\Delta_{\mathcal{M}_{\log}}^{\tau}$ extracted from $\mathcal{M}_{\log}$ is consistent.*

Proof. We show that $\Delta_{\mathcal{M}_{\log}}^{\tau}$ has a tolerance partition from which its consistency follows. Let $m + 1$ be the number of layers in $\mathcal{M}_{\log}$ and, for $j = 0, 1, \ldots, m$, let $\mathcal{N}_j$ be the set of neurons in the j-th layer. Then, $(\Delta_1, \ldots, \Delta_m)$ with

$$\begin{aligned}
\Delta_j = \{\delta_N^{\tau,+} \in \Delta_{\mathcal{M}_{\log}}^{\tau} \mid N \in \mathcal{N}_j\}&\\
\cup \{\delta_N^{\tau,-} \in \Delta_{\mathcal{M}_{\log}}^{\tau} \mid N \in \mathcal{N}_j\}&
\end{aligned}$$

for $j = 1, \ldots, m$ is a partition of $\Delta_{\mathcal{M}_{\log}}^{\tau}$ (modulo empty sets). Let $\delta \in \Delta_j$, provided hat $\Delta_j \neq \emptyset$. We have to show that δ is tolerated by $\bigcup_{k=j}^{m} \Delta_k$. For this, let δ be of the form $\delta_N^{\tau,+}$ for some $N \in \mathcal{N}_j$. The proof for δ of the form $\delta_N^{\tau,-}$ is analogous. By construction of $\delta_N^{\tau,+}$, there is $(A^+, A^-) \in \mathcal{SA}_{\min}^{\tau}(N)$ and a (partial) possible world $\omega \in \Omega(\mathcal{N}_{j-1})$ with $\omega \models \bigwedge_{N' \in A^+} N' \wedge \bigwedge_{N' \in A^-} \overline{N'}$ (A^+ and A^- are disjoint).

Thanks to Lemma 1, we can extend ω to a (partial) possible world $\omega' \in \Omega(\mathcal{N}_{j-1} \cup \mathcal{N}_j)$ such that all conditionals in Δ_j are either not applicable or verified by concatenating N' to ω in case of $\omega \models \psi_{N'}^{\tau,+}$ or concatenating $\overline{N'}$ to ω in case of $\omega \models \psi_{N'}^{\tau,-}$ for $N' \in \mathcal{N}_j$. In particular, ω' verifies $\delta_N^{\tau,+}$. By a repeated application of this argument, we can construct a (partial) possible world $\omega'' \in \Omega(\bigcup_{k=j-1}^{m} \mathcal{N}_k)$ which verifies $\delta_N^{\tau,+}$ and falsifies no conditional from $\bigcup_{k=j}^{m} \Delta_k$. Eventually, this (partial) possible world can be extended to a possible world in $\Omega(\bigcup_{k=0}^{m} \mathcal{N}_k)$ by the concatenation of the remaining ground atoms, either positive or negated which can be chosen freely. □

Note that the belief base $\Delta_{\mathcal{M}_{\log}}^{\tau}$ might be empty, namely if for all neurons in $\mathcal{M}_{\log}$ there is no sufficient (in)activator. On the contrary, if a neuron N can be (in)activated, then there is a sufficient (in)activator of N so that there is a conditional wrt. N in $\Delta_{\mathcal{M}_{\log}}^{\tau}$. Thus, $\Delta_{\mathcal{M}_{\log}}^{\tau}$ reflects the most important information flow in $\mathcal{M}_{\log}$.

5. Binary Classification with $\Delta_{\mathcal{M}_{\log}}^{\tau}$

Now, we discuss how to perform binary (multi-class) classification based on the belief base $\Delta_{\mathcal{M}_{\log}}^{\tau}$ which we have extracted from a multilayer perceptron $\mathcal{M}_{\log}$ (cf. Definition 9). Recall that, following Definition 2, we can say that an input vector $\vec{x}$ of $\mathcal{M}_{\log}$ is *classified* (resp. *declassified*) as $\mathcal{C}_N$ represented by the neuron N from the output layer of $\mathcal{M}_{\log}$ if $\mathcal{M}_{\log}(\vec{x}) \geq 1 - \tau$ (resp. $\mathcal{M}_{\log}(\vec{x}) \leq \tau$) where τ is a tolerance factor. We denote this with

$$\begin{aligned}
\mathcal{M}_{\log}, \vec{x} \mathrel{|\!\!\sim}_{\tau} N \;\text{ iff }\; \mathcal{M}_{\log}(\vec{x}) \geq 1 - \tau\\
\mathcal{M}_{\log}, \vec{x} \mathrel{|\!\!\sim}_{\tau} \overline{N} \;\text{ iff }\; \mathcal{M}_{\log}(\vec{x}) \leq \tau.
\end{aligned}$$

We lift this idea of classifying $\vec{x}$ from $\mathcal{M}_{\log}$ to the belief base $\Delta_{\mathcal{M}_{\log}}^{\tau}$. Thereby, we make use of the System Z ranking model $\kappa_{\Delta_{\mathcal{M}_{\log}}^{\tau}}^{Z}$ of $\Delta_{\mathcal{M}_{\log}}^{\tau}$.

Definition 10 (Z-Classification)**.** *Let $(\mathcal{M}_{\log}, \tau)$ be a classification scheme, let $\Delta_{\mathcal{M}_{\log}}^{\tau}$ be the belief base extracted from $\mathcal{M}_{\log}$, and let $\kappa_{\mathcal{M}_{\log},\tau}^{Z} = \kappa_{\Delta_{\mathcal{M}_{\log}}^{\tau}}^{Z}$ be its System Z ranking model. With $A_{\vec{x}}^{\tau}$ we denote the set of neurons from the input layer of $\mathcal{M}_{\log}$ which are activated by $\vec{x}$ wrt. τ, and with $I_{\vec{x}}^{\tau}$ the set of neurons which are inactivated. Then, we say that an input vector $\vec{x}$ of $\mathcal{M}_{\log}$ is*

- Z-classified as $\mathcal{C}_N$ *wrt. a neuron N from the output layer of $\mathcal{M}_{\log}$, denoted by*

$$\begin{aligned}
\Delta_{\mathcal{M}_{\log}}^{\tau}, \vec{x} \mathrel{|\!\!\sim}_{\tau}^{Z} N, \;\; \textit{iff } \kappa_{\mathcal{M}_{\log},\tau}^{Z} \;\textit{accepts}&\\
(N| \bigwedge_{N' \in \mathcal{A}_{\vec{x}}^{\tau}} N' \wedge \bigwedge_{N' \in \mathcal{I}_{\vec{x}}^{\tau}} \overline{N'}),&
\end{aligned}$$

- Z-declassified as $\mathcal{C}_N$, *denoted by*

$$\Delta^{\tau}_{\mathcal{M}_{\log}}, \vec{x} \mathrel{|\!\sim}^{Z}_{\tau} \overline{N}, \text{ iff } \kappa^{Z}_{\mathcal{M}_{\log},\tau} \text{ accepts } (\overline{N} | \bigwedge_{N' \in \mathcal{A}^{\tau}_{x}} N' \wedge \bigwedge_{N' \in \mathcal{I}^{\tau}_{x}} \overline{N'}).$$

We obtain the following central result stating that $\kappa^{Z}_{\mathcal{M}_{\log},\tau}$ does not "invent" inferences but yields inferences that can be drawn from $\mathcal{M}_{\log}$ only. Instead, inferences drawn from $\kappa^{Z}_{\mathcal{M}_{\log},\tau}$ can be understood, in some sense, as the most reliable inferences from $\mathcal{M}_{\log}$.

Proposition 6. *Let $(\mathcal{M}_{\log}, \tau)$ be a classification scheme, let $\Delta^{\tau}_{\mathcal{M}_{\log}}$ be the belief base extracted from $\mathcal{M}_{\log}$, let $\kappa^{Z}_{\mathcal{M}_{\log},\tau}$ be its System Z ranking model, and let $\vec{x}$ be an input vector of $\mathcal{M}_{\log}$. Then,*

$$\Delta^{\tau}_{\mathcal{M}_{\log}}, \vec{x} \mathrel{|\!\sim}^{Z}_{\tau} N \text{ implies } \mathcal{M}_{\log}, \vec{x} \mathrel{|\!\sim}_{\tau} N,$$

and, analogously,

$$\Delta^{\tau}_{\mathcal{M}_{\log}}, \vec{x} \mathrel{|\!\sim}^{Z}_{\tau} \overline{N} \text{ implies } \mathcal{M}_{\log}, \vec{x} \mathrel{|\!\sim}_{\tau} \overline{N}.$$

Proof. Let $\mathcal{A}^{\tau}_{\vec{x}}$ and $\mathcal{I}^{\tau}_{\vec{x}}$ be the sets of the neurons from the input layer of $\mathcal{M}_{\log}$ which are activated resp. inactivated by the input $\vec{x}$ wrt. τ (cf. Definition 10). Further, let $m + 1$ be the number of layers in $\mathcal{M}_{\log}$, and, for $j = 0, 1, \ldots, m$, let $\mathcal{N}_j$ be the set of neurons in the j-th layer of $\mathcal{M}_{\log}$. We prove that $\Delta^{\tau}_{\mathcal{M}_{\log}}, \vec{x} \mathrel{|\!\sim}^{Z}_{\tau} N$ implies $\mathcal{M}_{\log}, \vec{x} \mathrel{|\!\sim}_{\tau} \overline{N}$. The proof that $\Delta^{\tau}_{\mathcal{M}_{\log}}, \vec{x} \mathrel{|\!\sim}^{Z}_{\tau} \overline{N}$ implies $\mathcal{M}_{\log}, \vec{x} \mathrel{|\!\sim}_{\tau} \overline{N}$ is analogous.

Let $\Delta^{\tau}_{\mathcal{M}_{\log}}, \vec{x} \mathrel{|\!\sim}^{Z}_{\tau} N$, i.e., by definition, $\kappa^{Z}_{\mathcal{M}_{\log},\tau}$ accepts the conditional $(N|\chi_N)$ with

$$\chi_N = \bigwedge_{N' \in \mathcal{A}^{\tau}_{\vec{x}}} N' \wedge \bigwedge_{N' \in \mathcal{I}^{\tau}_{\vec{x}}} \overline{N'}.$$

Following the construction of possible worlds in the proof of Proposition 5, every (partial) possible world $\omega \in \Omega(\mathcal{N}_0)$ with $\omega \models \chi_N$ can be extended to a possible world $\omega' \in \Omega(\bigcup_{j=0}^{m} \mathcal{N}_j)$ such that no conditional from $\Delta^{\tau}_{\mathcal{M}_{\log}}$ is falsified. Hence, $\kappa^{Z}_{\mathcal{M}_{\log},\tau}(\omega') = 0$. Because $\kappa^{Z}_{\mathcal{M}_{\log},\tau}$ accepts the conditional $(N|\chi_N)$, none of these extensions ω' satisfies $\overline{N}$. Otherwise, $\kappa^{Z}_{\mathcal{M}_{\log},\tau}(\overline{N} \wedge \chi_N) = 0$ would hold which contradicts the acceptance of $(N|\chi_N)$. As a consequence, the conditional $\delta^{\tau;+}_{N}$ (cf. Definition 9) must be in $\Delta^{\tau}_{\mathcal{M}_{\log}}$ which is the only possibility to exclude $\overline{N}$ from the extensions ω' (and which is also accepted in all the extensions ω'). Otherwise, there is no reason why not to have an extension ω' with $\omega' \models \overline{N}$.

In more detail, either there is an extension ω' of ω with $\omega' \models \overline{N}$ and $\kappa^{Z}_{\mathcal{M}_{\log},\tau}(\omega') = 0$ which contradicts the acceptance of $(N|\chi_N)$, or $\kappa^{Z}_{\mathcal{M}_{\log},\tau}(\omega') > 0$ for all such extension ω' which requires a conditional in $\Delta^{\tau}_{\mathcal{M}_{\log}}$ that is falsified in ω'. The only candidate for such a conditional would be $\delta^{\tau;+}_{N}$. As a consequence of the acceptance of $\delta^{\tau;+}_{N}$, the input vector $\vec{x}$ activates at least one sufficient activator of N. From this, it follows that $\vec{x}$ also activates N in $\mathcal{M}_{\log}$. □

We recall our running example to illustrate this proposition.

Example 8. *We consider the same scenario as in Example 1, i.e., the multilayer perceptron $\mathcal{M}^{\text{ex}}_{\log}$, the tolerance factor $\tau = 0.3$, and the input vector $\vec{x} = (0.9, 0.8, 0.1)$. Then,*

$$\mathcal{A}^{0.3}_{\vec{x}} = \{N_{0,0},\ N_{0,1}\}, \quad \mathcal{I}^{0.3}_{\vec{x}} = \{N_{0,2}\}.$$

Further, the Z-partition of $\Delta^{0.3}_{\mathcal{M}_{\log}}$ is $Z(\Delta^{0.3}_{\mathcal{M}^{\text{ex}}_{\log}}) = (\Delta^{0.3}_{\mathcal{M}^{\text{ex}}_{\log}})$, so that, for $(N_{2,2}|\chi_{N_{2,2}})$ with $\chi_{N_{2,2}} = N_{0,0} \wedge N_{0,1} \wedge \overline{N_{0,2}}$, we have, with $\Delta = \Delta^{0.3}_{\mathcal{M}^{\text{ex}}_{\log}}$,

$$\kappa^{Z}_{\Delta}(N_{2,2} \wedge \chi_{N_{2,2}}) = 0 < 1 = \kappa^{Z}_{\Delta}(\overline{N_{2,2}} \wedge \chi_{N_{2,2}})$$

Thus, we classify $\vec{x}$ as an instance of $\mathcal{C}_{N_{2,2}}$ in accordance with the result from Example 1.

Our approach focuses attention on the main dependencies among the neurons in multilayer perceptrons. In contrast to the synaptic conditionals in Section 3, the influence of several parent nodes on a neuron N is aggregated, with the guarantee that the aggregated parent nodes are able to (in)active N. A depiction of these aggregated influences is shown in Figure 4 for our running example. Figure 4 can be understood as a visualization of the main information flow in $\mathcal{M}^{\text{ex}}_{\log}$.

6. Conclusions

We proposed an approach on extracting propositional conditional belief bases from multilayer perceptrons (MLPs) for binary multi-class classification. The conditionals relate to the main information flow in the multilayer perceptron detached from specific input vectors. Therewith, our approach abstracts from both the input data as well as overlay effects in the network and rebuilds the backbone of the network within a prevalent KRR formalism. The main idea of our approach is to exploit sufficient (in)activators of neurons N the (in)activation of which guarantees that N is (in)activated as well. The extracted conditional belief base allows for drawing inferences in a principled way, for instance, under System Z. It is guaranteed that the belief base is consistent and does not invent inferences that cannot be drawn from the multilayer perceptron.

In recent work [17] it has been shown that there is a tight connection between multilayer perceptrons and *quantitative bipolar argumentation frameworks*. Roughly speaking, MLPs can be seen as specific argumentation frameworks under a so-called *MLP-semantics*. To make this connection useful for explanations, some ideas on sparsification have been considered [18]. In future work, we want to investigate the connections between our approach and the approaches from [17, 18]. Exploiting sparsified networks may simplify the computation of conditional belief bases. The other way round, the qualitative conditionals could perhaps be used to construct argumentation frameworks in order to simulate the MLPs that are easier to interpret than the argumentation frameworks obtained from the current approaches.

Also in future work, we want to extract conditionals from multilayer perceptrons that are based on "necessary (in)activators" and can be used for explaining classifications that are made by the multilayer perceptrons. Therewith, we expect to be able to bound all possible classifications from two directions (upper and lower bound) which, as we hope, can help to better understand the essence of binary multi-class classification based on multilayer perceptrons. Further research directions could be to investigate how the choice of the tolerance factor influences the shape of the conditional belief base and how different inference operators, e.g., based on System P [8], *lexicographic closure* [19], or c-representations [9], relate to the binary multi-class classification with multilayer perceptrons.

Acknowledgments

This work was supported by DFG Grant KE 1413/14-1 of the German Research Foundation (DFG) awarded to Gabriele Kern-Isberner.

References

[1] K. Gurney, An Introduction to Neural Networks, UCL Press, 1997.

[2] S. Yang, C. Zhang, W. Wu, Binary output layer of feedforward neural networks for solving multi-class classification problems, IEEE Access 7 (2019) 5085–5094.

[3] A. Rajendra, P. Sajja, Knowledge-Based Systems, Jones & Bartlett Learning, 2009.

[4] J. Pearl, System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning, in: R. Parikh (Ed.), Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge, Pacific Grove, CA, USA, March 1990, Morgan Kaufmann, 1990, pp. 121–135.

[5] B. d. Finetti, La logique de la probabilité, The Journal of Symbolic Logic 2 (1937) 31–39.

[6] E. W. Adams, The Logic of Conditionals, Springer, 1975.

[7] W. Spohn, The Laws of Belief - Ranking Theory and Its Philosophical Applications, Oxford UP, 2014.

[8] S. Kraus, D. Lehmann, M. Magidor, Nonmonotonic reasoning, preferential models and cumulative logics, Artif. Intell. 44 (1990) 167–207.

[9] G. Kern-Isberner, A thorough axiomatization of a principle of conditional preservation in belief revision, Ann. Math. Artif. Intell. 40 (2004) 127–164.

[10] A. Ghorbani, J. Y. Zou, Neuron shapley: Discovering the responsible neurons, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[11] W. S. McCulloch, W. H. Pitts, A logical calculus of the ideas immanent in nervous activity, The bulletin of mathematical biophysics 5 (1943) 115–133.

[12] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control. Signals Syst. 2 (1989) 303–314.

[13] D. Nute, Topics in Conditional Logic, Springer, 2011.

[14] M. Goldszmidt, J. Pearl, On the Relation Between Rational Closure and System Z, CSD (Series), UCLA Computer Science Department, 1991.

[15] L. Giordano, D. Theseider Dupré, Weighted defeasible knowledge bases and a multipreference semantics for a deep neural network model, in: Logics in Artificial Intelligence, Springer International Publishing, 2021, pp. 225–242.

[16] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge UP, 2017.

[17] N. Potyka, Interpreting neural networks as quantitative argumentation frameworks, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, 2021, pp. 6463–6470.

[18] H. Ayoobi, N. Potyka, F. Toni, Sparx: Sparse argumentative explanations for neural networks, in: K. Gal, A. Nowé, G. J. Nalepa, R. Fairstein, R. Radulescu (Eds.), ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023), volume 372 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2023, pp. 149–156.

[19] D. Lehmann, Another perspective on default reasoning, Ann. Math. Artif. Intell. 15 (1995) 61–82.