Interpreting Neural Networks as Quantitative Argumentation Frameworks

Nico Potyka
University of Stuttgart,
Universitätsstraße 32,
70569 Stuttgart, Germany,
nico.potyka@ipvs.uni-stuttgart.de

Abstract

We show that an interesting class of feed-forward neural networks can be understood as quantitative argumentation frameworks. This connection creates a bridge between research in Formal Argumentation and Machine Learning. We generalize the semantics of feed-forward neural networks to acyclic graphs and study the resulting computational and semantical properties in argumentation graphs. As it turns out, the semantics gives stronger guarantees than existing semantics that have been tailor-made for the argumentation setting. From a machine-learning perspective, the connection does not seem immediately helpful. While it gives intuitive meaning to some feed-forward-neural networks, they remain difficult to understand due to their size and density. However, the connection seems helpful for combining background knowledge in form of sparse argumentation networks with dense neural networks that have been trained for complementary purposes and for learning the parameters of quantitative argumentation frameworks in an end-to-end fashion from data.

Introduction

In this paper, we establish a relationship between neural networks and abstract argumentation frameworks. More precisely, we study relationships between quantitative bipolar argumentation frameworks (QBAFs) and multilayer perceptrons (MLPs). QBAFs are a knowledge representation formalism that can be used to solve decision problems in a very intuitive way by weighing up pro and contra arguments (Baroni et al. 2015; Rago et al. 2016; Amgoud and Ben-Naim 2017). QBAFs and their variants have been combined with machine learning methods in order to add explainability to problems like product recommendation (Rago, Cocarascu, and Toni 2018), review aggregation (Cocarascu, Rago, and Toni 2019) and stance aggregation in fake news detection (Kotonya and Toni 2019). Multilayer perceptrons (MLPs) (Goodfellow et al. 2016) are a very flexible class of feed-forward neural networks that can be applied in basically all machine learning tasks. This includes applications like classification (Heidari et al. 2019), regression (Hiransha et al. 2018) and function approximation in reinforcement learning (Tesauro 1995).

We explain the basics of QBAFs and MLPs in Sections and , respectively. In Section , we introduce an MLP-based semantics for QBAFs that is based on computing the strength of arguments in an iterative way. In acyclic graphs, the result is equal to the result of the usual evaluation procedure (forward propagation) for MLPs. We give sufficient conditions for convergence of this procedure in cyclic graphs and analyze the convergence rate. Simply put, convergence is guaranteed when the edge weights and the indegree of arguments is not too large. We give an example that demonstrates that our convergence conditions cannot be improved without adding additional assumptions about the structure of the graph. In order to improve the guarantees, we introduce a continuous variant that agrees with its discrete counterpart in the known convergence cases, but still converges in more general cases. Finally, we show that the MLP-based semantics satisfies all properties for QBAF semantics proposed in (Amgoud and Ben-Naim 2017; Potyka 2018a, 2019b) almost perfectly. This is surprising because it actually gives stronger semantical guarantees than some semantics that have been designed specifically for QBAFs. We close the paper with some ideas about how this relationship can be exploited to combine ideas for QBAFs and neural networks fruitfully for both fields.

QBAF Basics

In this work, our conceptual understanding of an argument follows Dung's notion of *abstract argumentation*: "an argument is an abstract entity whose role is solely determined by its relations to other arguments" (Dung 1995). That is, we abstract from the content of arguments and focus on their acceptability dependent on the acceptability of their attackers and supporters. This idea can be formalized in different ways, we refer to (Baroni, Caminada, and Giacomin 2018) for an overview of some classical approaches. Here, we consider quantitative bipolar argumentation frameworks (QBAFs) similar to (Baroni, Rago, and Toni 2018). In general, these frameworks interpret arguments by values from an arbitrary domain $\mathcal{D}$. For simplicity, we assume that $\mathcal{D} = [0, 1]$. Intuitively, the value 0 means that an argument is fully rejected, 1 means that it is fully accepted and values in between balance between these extremes.

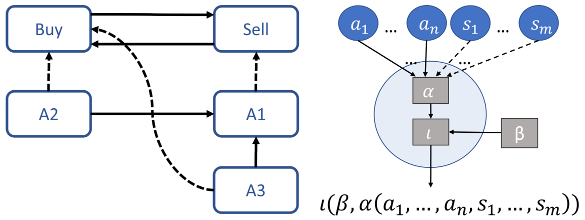

Buy
Sell
A2
A1
A3
a_1 ... a_n s_1 ... s_m
α
ι
β
$\iota(\beta, \alpha(a_1, \ldots, a_n, s_1, \ldots, s_m))$

Figure 1: Example of a QBAF (left) and illustration of local update mechanics under modular semantics (right).

Definition 1 (QBAF)**.** A QBAF (over $\mathcal{D} = [0, 1]$) is a quadruple $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup}, \beta)$ consisting of a set of arguments $\mathcal{A}$, two binary relations Att and Sup called attack and support and a function $\beta : \mathcal{A} \to [0, 1]$ that assigns a *base score* $\beta(a)$ to every argument $a \in \mathcal{A}$.

The base score can be seen as an apriori strength of an argument when it is evaluated independent of its relationships to other arguments. This apriori strength will be adapted dynamically based on the strength of its attackers and supporters. Graphically, we denote attack relations by solid and support relations by dashed edges as illustrated in Figure 1 on the left. The QBAF models part of a decision problem from (Potyka 2018b), where we want to decide whether to buy new or sell existing stocks of a company. A1 corresponds to the statement of an expert that recommends selling. A2 and A3 correspond to statements by experts who contradict the premises of A1 and recommend buying. The selling and the buying decision are simply modeled as arguments that attack each other, so that the confidence in one decision will decrease the confidence in the other.

The main computational problem in QBAFs is to assign a strength value to arguments. We describe this process by interpretations.

Definition 2 (QBAF interpretation)**.** Let Q be a QBAF over $[0, 1]$. An interpretation of Q is a function $\sigma : \mathcal{A} \to [0, 1] \cup \{\bot\}$ and $\sigma(a)$ is called the strength of a for all $a \in \mathcal{A}$. If $\sigma(a) = \bot$ for some $a \in \mathcal{A}$, σ is called *partial*. Otherwise, it is called *fully defined*.

Modular semantics define interpretations based on an iterative procedure (Mossakowski and Neuhaus 2018). For every argument, its strength is initialized with its base score. The strength values are then adapted iteratively by applying an *aggregation function* α and an *influence function* ι as illustrated in Figure 1 on the right. The aggregation function α aggregates the strength values of attackers and supporters. Aggregation functions have been based on product (Baroni et al. 2015; Rago et al. 2016), addition (Amgoud and Ben-Naim 2017; Potyka 2018a) and maximum (Mossakowski and Neuhaus 2018). The influence function then takes the aggregate and the base score in order to determine a new strength from the desired domain. Intuitively, supporters increase the strength, while attackers decrease it. If the strength values converge, the limit defines the final strength value. Otherwise, strength values remain undefined and the interpretation is partial. Of course, it would be desirable to always have fully defined interpretations. However, as shown in (Mossakowski and Neuhaus 2018), many update procedures can fail to converge in cyclic QBAFs. Properties for evaluating and comparing different semantics have been discussed in (Amgoud and Ben-Naim 2017; Baroni, Rago, and Toni 2018; Potyka 2018a). We will explain these properties in detail later when we analyze neural networks as QBAFs.

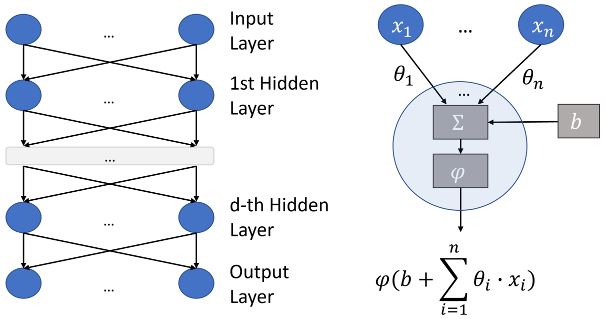


Figure 2: Graphical structure of an MLP (left) and illustration of local forward propagation (right).

MLP Basics

Intuitively, a multilayer perceptron (MLP) is a layered acyclic graph as sketched in Figure 2 on the left. Formally, we describe MLPs as follows.

Definition 3 (MLP)**.** An MLP is a tuple (V, E, B, Θ), where

- (V, E) is a directed graph.
- $V = \uplus_{i=0}^{d+1} V_i$ is the disjoint union of sets of nodes V_i.
- We call V_0 the *input layer*, V_{d+1} the *output layer* and V_i the i-th *hidden layer* for $1 \leq i \leq d$.
- We call d the depth of the network.
- $E \subseteq \bigcup_{i=0}^{d} \left(V_i \times V_{i+1}\right)$ is a set of edges between subsequent layers. If $E = \bigcup_{i=0}^{d} \left(V_i \times V_{i+1}\right)$, the network is called *fully connected*.
- $B : (V \setminus V_0) \to \mathbb{R}$ assigns a *bias* to every non-input node.
- $\Theta : V \to \mathbb{R}$ assigns a *weight* to every edge.

In order to process an example, the input layer of an MLP is initialized with feature values of the example. These inputs are then propagated forward through the network to generate an output in the output layer. For example, in a binary classification task, the output layer could consist of a single node whose value corresponds to the model's confidence that the example belongs to the class. The values at nodes in hidden layers and the output layer are computed by propagating the values from the input layer forward through the network as sketched in Figure 2 on the right. Every edge is associated with a weight. For every ingoing edge e_i, the corresponding weight $\theta_i = \Theta(e_i)$ is multiplied by the value x_i of its source and the resulting values are summed up. The bias $b = B(v_j)$ of the edge's target v_j is added and the result is fed into an activaction function φ. A popular choice to obtain values between 0 and 1 is the logistic activation function that is defined by $\varphi_l(z) = \frac{1}{1+\exp(-z)}$. The logistic function lost popularity since it can slow down gradient-based training due to vanishing derivatives close to 0 and 1. However, recent ideas like batch normalization (Ioffe and Szegedy 2015) can mitigate the problem. In principle, the following ideas can be applied to other activation functions like rectified linear units as well. However, values between 0 and 1 yield a particularly nice and simple interpretation. We will therefore focus on logistic activation functions in the following.

MLP-based Semantics for QBAFs

When comparing the update mechanics of QBAFs as sketched in Figure 1 on the right with the forward propagation mechanics of MLPs as sketched in Figure 2 on the right, we see that they are very similar. Roughly speaking, we can view an MLP as a QBAF where the aggregation function α is based on addition and the influence function ι is based on a neural network activation function. It is then natural to ask, does this QBAF give meaningful guarantees from an argumentation perspective? In order to answer this question, we consider edge-weighted QBAFs as already considered in (Mossakowski and Neuhaus 2018). We consider only one set of edges and regard edges with negative weights as attacks and edges with positive weights as supports. This simplifies making the connection between MLPs and QBAFs, but may not be appropriate in more general settings where the aggregation function is not based on addition.

Definition 4 (Edge-weighted QBAF)**.** An edge-weighted QBAF (over $\mathcal{D} = [0,1]$) is a quadruple $(\mathcal{A}, E, \beta, w)$ consisting of a set of arguments $\mathcal{A}$, edges $E \subseteq \mathcal{A} \times \mathcal{A}$ between these arguments, a function $\beta : \mathcal{A} \to [0,1]$ that assigns a *base score* $\beta(a)$ to every argument $a \in \mathcal{A}$ and a function $w : E \to \mathbb{R}$ that assigns a weight to every edge.

To simplify the presentation, we assume that $\mathcal{A} = \{1, 2, \dots, n\}$ in the following. That is, the names of arguments correspond to numbers. Furthermore, for every argument $a \in \mathcal{A}$, we let $\mathrm{Att}(a) = \{(b,a) \in E \mid w(b,a) < 0\}$ and $\mathrm{Sup}(a) = \{(b,a) \in E \mid w(b,a) > 0\}$.

In order to interpret the arguments in an edge-weighted QBAF, we consider a modular semantics based on the relationship between QBAFs and MLPs noted earlier. The strength values are computed iteratively. In every iteration, we have a strength vector $s^{(i)} \in [0,1]^n$. Its a-th element $s_a^{(i)}$ is the strength value of argument a in the i-th iteration. For every argument $a \in \mathcal{A}$, we let $s_a^{(0)} := \beta(a)$ be the initial strength value. The strength values are then updated by doing the following two steps repeatedly for all $a \in \mathcal{A}$:

Aggregation: We let $\alpha_a^{(i+1)} := \sum_{(b,a)\in E} w(b,a) \cdot s_b^{(i)}$.

Influence: We let $s_a^{(i+1)} := \varphi_l\big(\ln(\frac{\beta(a)}{1-\beta(a)}) + \alpha_a^{(i+1)}\big)$, where $\varphi_l(z) = \frac{1}{1+\exp(-z)}$ is the logistic function.

Strictly speaking, the influence function is undefined for $\beta(a) \in \{0,1\}$. However, we can complete the definition by using the infinite limits at these points. That is, we let $\ln(0) := -\infty$, $\ln(\frac{1}{0}) := \infty$, $\varphi_l(-\infty) = 0$, $\varphi_l(\infty) = 1$ and for all $x \in \mathbb{R}$, $x - \infty = -\infty$ and $x + \infty = \infty$. In this way, the composition of the aggregation and influence function is continuous and always returns values from the closed interval $[0,1]$. By putting the definition of the aggregation function into the influence function, we obtain the explicit form of the update function $u_{\mathrm{MLP}} : [0,1]^n \to [0,1]^n$ whose i-h component is defined by

$$\frac{1}{1 + \frac{1-\beta(i)}{\beta(i)} \exp(-\sum_{(b,i)\in E} w(b,i) \cdot s_b)}. \tag{1}$$

Note that $s^{(k)} = u_{\mathrm{MLP}}^k(s^{(0)})$, that is, $s^{(k)}$ is obtained from $s^{(0)}$ by applying u_{MLP} k times. The MLP-based semantics is defined based on the result of applying the aggregation and influence function repeatedly.

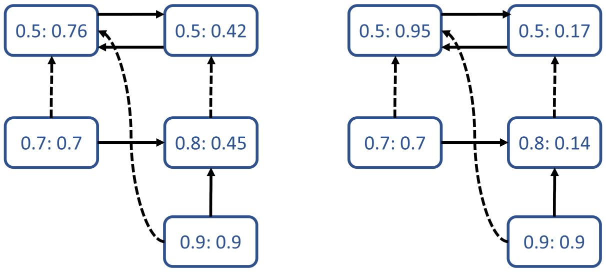


Figure 3: MLP-based interpretation of the QBAF from Figure 1. The nodes are annotated with (base score: strength). The edge weights are s for supports and $-s$ for attacks, where $s = 1$ on the left and $s = 2$ on the right.

Definition 5 (MLP-based Semantics)**.** Let Q be an edge-weighted QBAF over $[0,1]$. The interpretation of Q under MLP-based semantics is defined by

$$\sigma_{\mathrm{MLP}}(a) = \begin{cases} \lim_{k\to\infty} s_a^{(k)} & \text{if the limit exists} \\ \bot & \text{otherwise} \end{cases}$$

for all $a \in \mathcal{A}$.

In order to illustrate the definition, Figure 3 shows the interpretation of our example QBAF from Figure 1 for two different instantiations of edge weights.

As we explain in the following proposition, if the MLP-based semantics is fully defined, then it corresponds to a fixed-point of the update function u_{MLP}. This observation will be important later to study semantical properties.

Proposition 1. *If σ_{MLP} is fully defined, then $s^* = \lim_{k\to\infty} s^{(k)}$ is a fixed-point of u_{MLP}, i.e., $u_{MLP}(s^*) = s^*$.*

Proof. See appendix in (Potyka 2020). □

There are two main questions that we want to answer for a new modular semantics. The first question is, under which conditions does the iterative computation of strength values converge? That is, for which families of QBAFs is the MLP-based semantics fully defined and are there families for which it is not? The second questions is, if the MLP-based semantics defines strength values, do they satisfy meaningful semantical properties? We will look at both questions in turn.

Convergence Guarantees

The following theorem explains some sufficient conditions under which the MLP-based semantics is fully defined. The proofs build up on general results about modular semantics developed in (Potyka 2019a).

Theorem 1. *Let Q be an edge-weighted QBAF over* $[0, 1]$.

1. *If Q is acyclic, then σ_{MLP} is fully defined and, for all $a \in \mathcal{A}$, $\sigma_{MLP}(a)$ can be computed in linear time.*
2. *If all arguments in Q have at most P parents, the weight of all edges is bounded from above by W and we have $W \cdot P < 4$, then σ_{MLP} is fully defined. Furthermore, $|\sigma_{MLP}(a) - s_a^{(n)}| < \epsilon$ whenever $n > \frac{\log \epsilon}{\log W + \log P - \log 4}$.*

Proof. See appendix in (Potyka 2020). □

In the acyclic case in item 1, the strength values can basically be computed by a single forward pass over a topological ordering of the arguments (Potyka 2019a). It is interesting to note that this process is equivalent to the usual forward propagation process in feed-forward networks (because, in an MLP, every layerwise ordering from the input to the output layer corresponds to a topological ordering and vice versa). In this sense, MLPs can indeed be seen as special cases of QBAFs, where the QBAF has an acyclic layered structure, the aggregation function is addition and the influence function is a neural network activation function.

Item 2 explains more complicated convergence conditions for cyclic QBAFs and gives a guarantee for the convergence rate. Convergence can be guaranteed if the maximum number of parents P of arguments and the maximum edge weight W in the QBAF are not too large. For example, if all edge weights are strictly smaller than $W = 1.3$ and every argument has at most $P = 3$ parents, then the iterative procedure is guaranteed to converge and the interpretation is fully defined. To understand the guarantees for the convergence rate, first note that $\log W + \log P - \log 4 = \log \frac{W \cdot P}{4} < \log(1) = 0$ by the assumption $W \cdot P < 4$. Hence, the denominator in the term $\frac{\log \epsilon}{\log W + \log P - \log 4}$ is always negative. For $\epsilon > 1$, the fraction is negative and, in this case, the bound is trivially true because all strength values are between 0 and 1. Indeed, we are usually interested in small values of ϵ close to 0. In this case, both the numerator and denominator are negative. In particular, $\log \epsilon \to -\infty$ as $\epsilon \to 0$. That is, the number of iterations n needed until the difference between $s_a^{(n)}$ and $\sigma_{\text{MLP}}(a)$ is smaller than a desired accuracy ϵ grows with increasing accuracy as we would naturally expect. Perhaps more surprising, the number of iterations decreases as W and P become larger. An intuitive explanation is that large weights and many parents will move the weights quicker such that convergence occurs faster. Of course, large W and P can also cause divergence of the procedure, but this can only happen if $W \cdot P \geq 4$.

The conditions in Theorem 1 are sufficient, but not necessary for convergence. However, Figure 4 shows a QBAF that demonstrates that the guarantees cannot be improved significantly without adding additional assumptions about the structure of the QBAF. The QBAF in Figure 4 belongs to a family of QBAFs that have been presented in (Mossakowski and Neuhaus 2018) to construct divergence examples for modular semantics. Every blue argument attacks every blue argument (including itself) and supports every green argument. Symmetrically, every green argument attacks every green argument and supports every blue argument. The graph on the right in Figure 4 shows how the strength values evolve over time for green and blue arguments. After approximately 100 iterations, the strength values start cycling between two states. Note that we have $W \cdot P = 0.7 \cdot 6 = 4.2$. The example therefore shows that the condition $W \cdot P < 4$ in Theorem 1 cannot be relaxed significantly. The example can be found in the Java library Attractor[1] (Potyka 2018b) in the folder examples/divergence. The reader can check that the example still diverges for $W = 0.67$ ($W \cdot P = 4.02$). We present the example for $W = 0.7$ mainly because the cycling can easily be illustrated visually for this case.

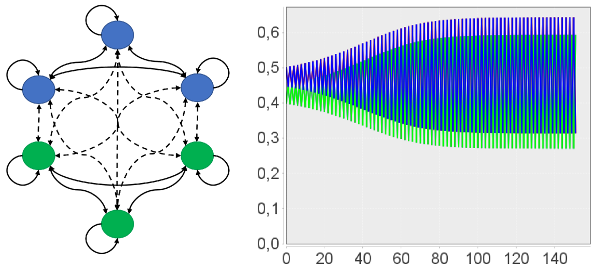


Figure 4: Left: Divergence example with base score 0.5 (blue arguments) and 0.4 (green arguments), edge weights 0.7 (supports) and −0.7 (attacks). Right: evolution of strength values (y-axis) for blue and green arguments plotted against number of iterations (x-axis).

An overview of convergence guarantees for other modular semantics can be found in (Potyka 2019a). The convergence guarantees for MLP-based semantics are similarly strong as the ones for Euler-based semantics (Amgoud and Ben-Naim 2017), which are slightly stronger than the guarantees for DF-QuAD (Rago et al. 2016) and the Quadratic Energy Model (Potyka 2018a). While (Mossakowski and Neuhaus 2018) presented a modular semantics that guarantees convergence in general QBAFs, these guarantees are bought at the expense of open-mindedness (Potyka 2019a). That is, the strength values of arguments cannot be far from their original base scores. There is indeed a tradeoff between convergence guarantees and open-mindedness (Potyka 2019a) and from this perspective, the MLP-based semantics is quite well behaved. Before we start discussing semantical guarantees of MLP-based semantics, we take a detour in order to improve the convergence guarantees.

Continuous MLP-Based Semantics

As discussed in (Potyka 2018a), it is often possible to overcome convergence problems of modular semantics by continuizing their discrete update procedures. To do so, the update function of the modular semantics can be transformed into a system of differential equations.

[1] https://sourceforge.net/projects/attractorproject/

Definition 6 (Continuous MLP-based Semantics)**.** Let Q be an edge-weighted QBAF over $[0, 1]$. The interpretation of Q under Continuous MLP-based Semantics is defined by

$$\sigma_{\text{cMLP}}(a) = \begin{cases} \lim_{t\to\infty} f_a^{\text{MLP}}(t) & \text{if the limit exists} \\ \perp & \text{otherwise} \end{cases}$$

for all $a \in \mathcal{A}$, where $f^{\text{MLP}} : \mathbb{R}_0^+ \to [0, 1]^n$ is the unique solution of the system of differential equations

$$\frac{\mathrm{d}f_i}{\mathrm{d}t} = \frac{1}{1 + \frac{1-\beta(i)}{\beta(i)} \exp(-\sum_{(b,i)\in E} w(b, i) \cdot f_b)} - f_i, \quad (2)$$

$$i = 1, \ldots, n,$$

with initial conditions $f_i(0) = \beta(i)$ for $i = 1, \ldots, n$.

Conceptually, the interpretation σ_{cMLP} is defined by two steps. First, we have to find the solution f^{MLP} of the system of differential equations (2). Then we have to compute the limit of $f^{\text{MLP}}(t)$ as t goes to infinity. Intuitively, $f_a^{\text{MLP}}(t)$ can be understood as the strength of argument a at time t. By the initial condition, we have $f_a^{\text{MLP}}(0) = s_a^{(0)} = \beta(a)$, that is, the strength at time 0 corresponds to the base score. As time progresses, the strength of a continuously evolves. In practice, the solution f^{MLP} is approximated numerically and the two steps can be combined into one. The Java library Attractor (Potyka 2018b) contains an implementation of the Runge-Kutta method RK4 for this purpose.

Intuitively, the i-th partial derivative $\frac{\mathrm{d}f_i}{\mathrm{d}t}$ described in (2) describes the rate of change at a point in time and corresponds to the difference between the desired function value (1) and the actual function value f_i. In particular, if f_i is too large, the derivative will be negative so that the function value will decrease. Symmetrically, it will increase if f_i is too small. The following theorem explains that f^{MLP} is indeed uniquely defined by the system of differential equations (2) and explains some relationships between the discrete and continuous MLP-based semantics. The proofs build up on general results about modular semantics developed in (Potyka 2019a).

Theorem 2. *For every QBAF Q, we have that*

1. *the system of differential equations in Definition 6 has a unique solution f^{MLP}.*
2. *If the limit $s^* = \lim_{t\to\infty} f^{MLP}(t)$ exists, then s^* is a fixed-point of u_{MLP}, that is, $u_{MLP}(s^*) = s^*$.*
3. *If $\lim_{t\to\infty} f^{MLP}(t)$ converges and Q satisfies any of the convergence conditions from Theorem 1, then $\sigma_{cMLP} = \sigma_{MLP}$.*

Proof. See appendix in (Potyka 2020). □

Item 2 explains that whenever the continuous MLP-based semantics defines strength values, these strength values correspond to a fixed-point of the discrete update function. Note that the same is true for the discrete semantics as explained in Proposition 1. Unfortunately, it is not obvious that the fixed-points are equal because u_{MLP} may have several fixed-points. However, item 3 explains that if the continuous MLP-based semantics defines strength values, and any of the convergence conditions from Theorem 1 are met, then the fixed-points and thus the strength values are equal. Note that this applies, in particular, to acyclic graphs and graphs with small indegree or small weights. What makes this relationship particularly interesting is that the continuous model can still converge to a meaningful limit when the discrete model does not. Since this limit is guaranteed to be a fixed-point of the discrete model, it is, in a way, consistent with the discrete semantics.

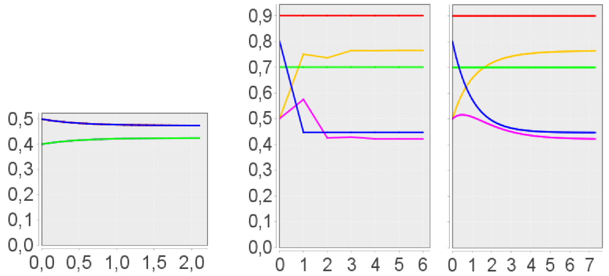


Figure 5: Evolution of strength values (y-axis) for QBAF from Figure 4 under continuous MLP-based semantics (left) and comparison of evolution of strength values for QBAF from Figure 1 with $s = 1$ under discrete and continuous MLP-based semantics (right)

Figure 5 shows on the left how the strength values under continuous MLP-based semantics evolve for the QBAF from Figure 4. As opposed to the iterative update procedure, the continuous update process changes the strength values continuously and does indeed converge. This example demonstrates that the continuous model offers strictly stronger convergence guarantees than the discrete one. The intuitive reason is that every discrete modular semantics with smooth aggregation and influence function can be seen as a coarse approximation of a continuous counterpart (Potyka 2018a). From this perspective, the convergence problems for discrete semantics occur because the step-size of the approximation is too large. It is actually an open question if there are QBAFs for which continuized semantics diverge as well. Until now, neither divergence examples nor general convergence proofs have been found. To illustrate the general relationship between discrete and continuous MLP-based semantics further, Figure 5 shows, on the right, the evolution of strength values under discrete and continuous semantics for the QBAF from Figure 1.

Semantical Guarantees

We will now look at semantical guarantees for MLP-based semantics. We know from Proposition 1 and Theorem 2 that the strength values under both semantics correspond to fixed-points of u_{MLP} if they are defined. Therefore, we can study the properties of both semantics simultaneously by studying properties that hold in a fixed-point of u_{MLP}. In (Amgoud and Ben-Naim 2017), 12 desirable properties have been presented that should be satisfied by quantitative argumentation semantics. We consider two additional properties from (Potyka 2018a, 2019b) that have been motivated by shortcomings of existing semantics. Since the properties have been phrased for QBAFs without edge-weights, we assume that the weights of all supports are 1 and the weights of all attacks are -1. To phrase the properties, we let Att^+ and Sup^+ denote the subsets of arguments in Att and Sup that the fixed-point assigns a non-zero strength to. The last property *Almost Open-Mindedness* is a slightly weaker form of *Open-Mindedness* from (Potyka 2019b). The only difference to the original definition is that it excludes the base scores 0 and 1.

Theorem 3. *Consider edge-weighted QBAFs $Q = (\mathcal{A}, E, \beta, w)$ and $Q' = (\mathcal{A}', E', \beta', w')$ with $w(e), w'(e') \in \{-1, 1\}$ for all $e \in E, e' \in E'$ and corresponding interpretations σ and σ' under discrete or continuous MLP-based semantics. Then the following properties are satisfied:*

Anonymity: *If Q and Q' are ismomorphic, then $\sigma = \sigma'$.*

Independence: *If $\mathcal{A} \cap \mathcal{A}' = \emptyset$, then for $Q'' = (\mathcal{A} \cup \mathcal{A}', E \cup E', \beta \cup \beta', w \cup w')$, σ'' is fully defined, $\sigma''(a) = \sigma(a)$ for $a \in \mathcal{A}$ and $\sigma''(a) = \sigma'(a)$ for $a \in \mathcal{A}'$.*

Directionality: *If $\mathcal{A} = \mathcal{A}'$ and $E = E' \cup \{(a, b)\}$, then for all $c \in \mathcal{A}$ such that there is no directed path from b to c, we have $\sigma(c) = \sigma'(c)$.*

Equivalence: *If there are $a, b \in \mathcal{A}$ such that $\beta(a) = \beta(b)$ and there are bijections $h : \mathrm{Att}(a) \rightarrow \mathrm{Att}(b)$, $h' : \mathrm{Sup}(a) \rightarrow \mathrm{Sup}(b)$ such that $\sigma(x) = \sigma(h(x))$ and $\sigma(y) = \sigma(h'(y))$ for all $x \in \mathrm{Att}(a), y \in \mathrm{Sup}(a)$, then $\sigma(a) = \sigma(b)$.*

Stability: *If there is an $a \in \mathcal{A}$ such that $\mathrm{Att}(a) = \mathrm{Sup}(a) = \emptyset$, then $\sigma(a) = \beta(a)$.*

Neutrality: *If there are $a, b \in \mathcal{A}$ such that $\beta(a) = \beta(b)$, $\mathrm{Att}(a) \subseteq \mathrm{Att}(b)$, $\mathrm{Sup}(a) \subseteq \mathrm{Sup}(b)$, $\mathrm{Att}(a) \cup \mathrm{Sup}(a) = \mathrm{Att}(b) \cup \mathrm{Sup}(b) \cup \{d\}$ and $\sigma(d) = 0$, then $\sigma(a) = \sigma(b)$.*

Monotony: *If there are $a, b \in \mathcal{A}$ such that $0 < \beta(a) = \beta(b) < 1$, $\mathrm{Att}(a) \subseteq \mathrm{Att}(b)$, $\mathrm{Sup}(a) \supseteq \mathrm{Sup}(b)$, then*

1. *$\sigma(a) \geq \sigma(b)$. (Monotony)*
2. *if furthermore ($\sigma(a) > 0$ or $\sigma(b) < 1$) and ($\mathrm{Att}(a)^+ \subset \mathrm{Att}(b)^+$ or $\mathrm{Sup}(a)^+ \supset \mathrm{Sup}(b)^+$), then $\sigma(a) > \sigma(b)$. (Strict Monotony)*

Reinforcement: *If there are $a, b \in \mathcal{A}$ such that $0 < \beta(a) = \beta(b) < 1$, $\mathrm{Att}(a) \setminus \{x\} = \mathrm{Att}(b) \setminus \{y\}$, $\mathrm{Sup}(a) \setminus \{x'\} = \mathrm{Sup}(b) \setminus \{y'\}$, $\sigma(x) \leq \sigma(y)$ and $\sigma(x') \geq \sigma(y')$, then*

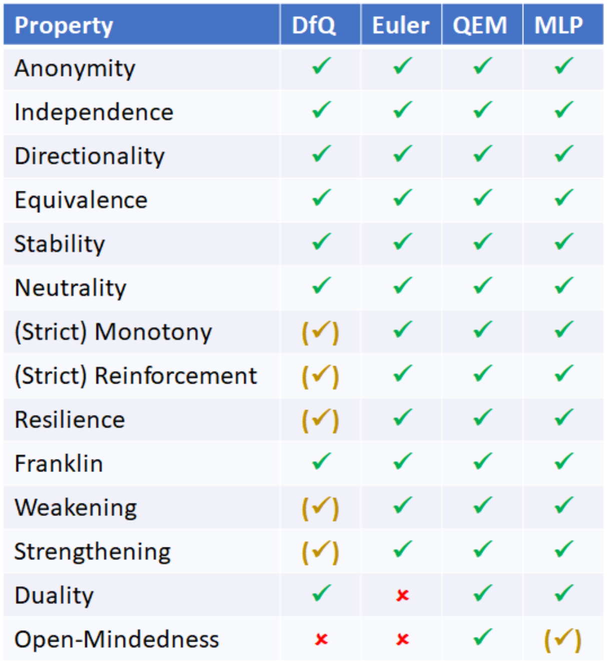
1. *$\sigma(a) \geq \sigma(b)$. (Reinforcement)*
2. *if ($\sigma(a) > 0$ or $\sigma(b) < 1$) and ($\sigma(x) < \sigma(y)$ or $\sigma(x') > \sigma(y')$), then $\sigma(a) > \sigma(b)$. (Strict Reinforcement)*

Resilience: *If $a \in \mathcal{A}$ is such that $0 < \beta(a) < 1$, then $0 < \sigma(a) < 1$.*

Franklin: *If there are $a, b \in \mathcal{A}$ such that $\beta(a) = \beta(b)$, $\mathrm{Att}(a) = \mathrm{Att}(b) \cup \{x\}$, $\mathrm{Sup}(a) = \mathrm{Sup}(b) \cup \{y\}$ and $\sigma(x) = \sigma(y)$, then $\sigma(a) = \sigma(b)$.*

Weakening: *Assume that there is an $a \in \mathcal{A}$ with $\beta(a) > 0$. Assume further that $g : \mathrm{Sup}(a) \rightarrow \mathrm{Att}(a)$ is an injective function such that $\sigma(x) \leq \sigma(g(x))$ for all $x \in \mathrm{Sup}(a)$ and $(\mathrm{Att}(a)^+ \setminus g(\mathrm{Sup}(a)) \neq \emptyset$ or there is an $x \in \mathrm{Sup}(a)$ such that $\sigma(x) < \sigma(g(x)))$. Then $\sigma(a) < \beta(a)$.*

Strengthening: *Assume that there is an $a \in \mathcal{A}$ with $\beta(a) < 1$. Assume further that $b : \mathrm{Att}(a) \rightarrow \mathrm{Sup}(a)$ is an injective function such that $\sigma(x) \leq \sigma(b(x))$ for all $x \in \mathrm{Att}(a)$ and $(\mathrm{Sup}(a)^+ \setminus b(\mathrm{Att}(a)) \neq \emptyset$ or there is an $x \in \mathrm{Att}(a)$ such that $\sigma(x) < \sigma(b(x)))$. Then $\sigma(a) > \beta(a)$.*



Property	DfQ	Euler	QEM	MLP
Anonymity	✓	✓	✓	✓
Independence	✓	✓	✓	✓
Directionality	✓	✓	✓	✓
Equivalence	✓	✓	✓	✓
Stability	✓	✓	✓	✓
Neutrality	✓	✓	✓	✓
(Strict) Monotony	(✓)	✓	✓	✓
(Strict) Reinforcement	(✓)	✓	✓	✓
Resilience	(✓)	✓	✓	✓
Franklin	✓	✓	✓	✓
Weakening	(✓)	✓	✓	✓
Strengthening	(✓)	✓	✓	✓
Duality	✓	×	✓	✓
Open-Mindedness	×	×	✓	(✓)

Figure 6: Semantical properties that are satisfied (✓), satisfied when excluding base scores 0 and 1 ((✓)) or not satisfied even when excluding base scores 0 or 1 (x) by DfQuAD (DfQ), Euler-based Semantics (Euler), Quadratic Energy Model (QEM) and MLP-based Semantics (MLP).

Duality: *Assume that there are $a, b \in \mathcal{A}$ such that $\beta(a) = 0.5 + \epsilon$, $\beta(b) = 0.5 - \epsilon$ for some $\epsilon \in [0, 0.5]$. If there are bijections $h : \mathrm{Att}(a) \rightarrow \mathrm{Sup}(b)$, $h' : \mathrm{Sup}(a) \rightarrow \mathrm{Att}(b)$ such that $\sigma(x) = \sigma(f(x))$ and $\sigma(y) = \sigma(g(y))$ for all $x \in \mathrm{Att}(a), y \in \mathrm{Sup}(a)$, then $\sigma(a) - \beta(a) = \beta(b) - \sigma(b)$.*

Almost Open-Mindedness: *For all $k \in \mathbb{N}$ and $p \in \{-1, 1\}$, let $Q_k^p = (\mathcal{A}_k^p, E_k^p, \beta_k^p, w_k^p)$ be constructed from Q by letting $\mathcal{A}_k^p = \mathcal{A} \cup \{A_1, \ldots, A_k\}$, $E_k^p = E \cup \{(A_1, a), \ldots, (A_k, a)\}$, $\beta_k^p(b) = \beta(b)$ for all $b \in \mathcal{A}$ and $\beta_k^p(A_i) = p$ for $1 \leq i \leq k$. Then for every $a \in \mathcal{A}$ with $0 < \beta(a) < 1$ and for every $\epsilon > 0$, there is an $N \in \mathbb{N}$ such that the interpretation $\sigma^{k,p}$ corresponding to Q_k^p satisfies*

1. *$\sigma^{k,p}(a) < \epsilon$ whenever $p = -1$ and $k > N$ and*
2. *$\sigma^{k,p}(a) > 1 - \epsilon$ whenever $p = 1$ and $k > N$.*

Proof. See appendix in (Potyka 2020). □

The first 12 properties have been introduced in (Amgoud and Ben-Naim 2017). *Anonymity* is a fairness condition and intuitively states that the strength values should not depend on the identity of the argument. *Independence* says that disconnected subgraphs should not affect each other. *Directionality* demands that the strength of an argument depends only on its predecessors in the graph. *Equivalence* says that arguments with equal status should be evaluated equally. *Stability* states that the final strength is just the initial weight if an argument does not have any parents. *Neutrality* demands that arguments with strength 0 do not affect other arguments. *Monotony* makes a quantitative statement: adding attackers or removing supporters can only weaken an argument. *Reinforcement* makes a similiar qualitative statement: strengthening attackers or weakening supporters can only weaken an argument. *Resilience* demands that the extreme values 0 and 1 can never be taken unless the base score was already an extreme value. *Franklin* says that an attacker and a supporter with equal strength cancel their effects. *Weakening* states that an argument's strength must be smaller than its base score when the attackers dominate the supporters. Symmetrically, Strengthening says that its strength must be larger when the supporters dominate. *Duality* from (Potyka 2018a) demands that attacks and supports are treated equally. Roughly speaking, the positive effect of a support should correspond to the negative effect of an attack. *Open-mindedness* (Potyka 2019b) says that the strength of an argument can become arbitrarily close to 0 or 1 independent of its base score if there is only a sufficient number of strong attackers or supporters. As we explain in the appendix, the MLP-based semantics satisfies this property in almost all cases except if base scores are set to 0 and 1. In this case, they can actually never change under MLP-based semantics.

Figure 6 gives an overview about which properties are satisfied by different semantics. Df-QuAD (Rago et al. 2016) had been introduced first and already fixed a problem of the QuAD model proposed in (Baroni et al. 2015). However, it does not completely satisfy several properties because of the way how it aggregates strength values. Roughly speaking, if an argument has both an attacker and a supporter with strength 1, its strength will necessarily be the base score no matter what other attackers and supporters there are. The Euler-based semantics (Amgoud and Ben-Naim 2017) had been introduced to overcome these problems. However, it introduced some other problems that are reflected by the fact that it satisfies neither duality nor open-mindedness. In particular, it treats attacks and supports in a rather random asymmetrical fashion. The quadratic energy model (Potyka 2018a) had been introduced to fix these issues. Therefore, it is not surprising that it satisfies all properties. Perhaps more surprising is that the MLP-based semantics satisfies all properties almost perfectly even though it has not been designed for this purpose. Its mechanics are actually very similar to the Euler-based semantics, but it fixes the Euler-based semantics' asymmetry between attacks and supports. As we explain in the appendix, the MLP-based semantics violates *Open-Mindedness* only when the base scores are set to the extreme values 0 or 1. It is a little bit odd that these values cannot change since they basically render such arguments redundant (their effect could directly be encoded in the base score of their children). However, it is not a big drawback since there is usually not a big practical difference between the base scores 0.99 and 1 or 0.01 and 0, respectively.

Conclusions and Related Work

We viewed MLPs as QBAFs to analyze their mechanics from an argumentation perspective. As it turns out, the MLP-based semantics offers comparatively good convergence guarantees in cyclic QBAFs and satisfies the common-sense properties from the literature almost perfectly. Recent combinations of machine learning methods and QBAFs often use variants of Df-QuAD and Euler-based semantics (Cocarascu, Rago, and Toni 2019; Kotonya and Toni 2019). It may be interesting to evaluate these approaches with MLP-based semantics. In particular, the generated QBAFs are acyclic in many applications, so that the resulting model under MLP-based semantics is a sparse MLP. For applications, this is interesting because it allows to retrain the weights by the usual backpropagation procedure in an end-to-end fashion (base score β translates to bias $\ln(\beta/(1-\beta))$ and bias θ translates to base score $\varphi_l(\theta)$). From a machine learning perspective, this is interesting because there has been growing interest in learning sparse neural networks (Louizos, Welling, and Kingma 2018; Frankle and Carbin 2018; Mocanu et al. 2018), not only to improve their interpretability, but also to tame their learning complexity. We may create sparse MLPs by building an acyclic sparse QBAFs from data like in (Cocarascu, Rago, and Toni 2019; Kotonya and Toni 2019) and translating it into an MLP.

It seems, more generally, interesting to view an acyclic QBAF with sum for aggregation as an MLP with a particular activation function to learn base scores and edge weights of QBAFs from data. If the influence function is differentiable, we can indeed just use the usual backpropagation procedure that is implemented in libraries like PyTorch and Tensorflow.

Let us note that there has been previous work on using neural networks for argumentation. For example, the authors in (Garcez, Gabbay, and Lamb 2005) showed how *value-based argumentation frameworks* (Bench-Capon 2003) can be encoded as MLPs. In these frameworks, every argument is associated with a *value* and there is a set of audiences with different preferences over the values. Arguments can then be *subjectively accepted* by one or *objectively accepted* by all audiences. The authors in (Garcez, Gabbay, and Lamb 2005) showed that an MLP with a single hidden layer and a *semi-linear activation function* can compute the *prevailing arguments* in these frameworks. More recently, there have also been attempts to use neural networks to approximately compute labellings of classical argumentation frameworks (Riveret et al. 2015; Kuhlmann and Thimm 2019).

Argumentation technology has also been considered as a more immediate tool for interpretable machine learning. Thimm and Kersting proposed to solve classification problems by means of *structured argumentation* (Thimm and Kersting 2017). As opposed to the abstract argumentation setting that we considered here, structured argumentation explicitly takes the premises and conclusions of arguments into account. Thimm and Kersting suggest learning structured arguments by rule mining algorithms. The rules can then be fed into a structured argumentation solver that can then derive a label for given inputs and explain the outcome.

Acknowledgments

This work was supported by the DFG through the projects EVOWIPE (STA572/15-1) and COFFEE (STA572/15-2).

References

Amgoud, L.; and Ben-Naim, J. 2017. Evaluation of arguments in weighted bipolar graphs. In *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*, 25–35. Springer.

Baroni, P.; Caminada, M.; and Giacomin, M. 2018. Abstract argumentation frameworks and their semantics. *Handbook of Formal Argumentation* 1: 157–234.

Baroni, P.; Rago, A.; and Toni, F. 2018. How many properties do we need for gradual argumentation? In *AAAI Conference on Artificial Intelligence (AAAI)*, 1736–1743. AAAI.

Baroni, P.; Romano, M.; Toni, F.; Aurisicchio, M.; and Bertanza, G. 2015. Automatic evaluation of design alternatives with quantitative argumentation. *Argument & Computation* 6(1): 24–49.

Bench-Capon, T. J. 2003. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation* 13(3): 429–448.

Cocarascu, O.; Rago, A.; and Toni, F. 2019. Extracting Dialogical Explanations for Review Aggregations with Argumentative Dialogical Agents. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 1261–1269.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence* 77(2): 321–357.

Frankle, J.; and Carbin, M. 2018. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations (ICLR)*.

Garcez, A. S.; Gabbay, D. M.; and Lamb, L. C. 2005. Value-based argumentation frameworks as neural-symbolic learning systems. *Journal of Logic and Computation* 15(6): 1041–1058.

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

Heidari, A. A.; Faris, H.; Aljarah, I.; and Mirjalili, S. 2019. An efficient hybrid multilayer perceptron neural network with grasshopper optimization. *Soft Computing* 23(17): 7941–7958.

Hiransha, M.; Gopalakrishnan, E. A.; Menon, V. K.; and Soman, K. 2018. NSE stock market prediction using deep-learning models. *Procedia computer science* 132: 1351–1362.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Bach, F. R.; and Blei, D. M., eds., *International Conference on Machine Learning (ICML)*, volume 37 of *JMLR Workshop and Conference Proceedings*, 448–456. JMLR.org.

Kotonya, N.; and Toni, F. 2019. Gradual Argumentation Evaluation for Stance Aggregation in Automated Fake News Detection. In *Workshop on Argument Mining*, 156–166.

Kuhlmann, I.; and Thimm, M. 2019. Using Graph Convolutional Networks for Approximate Reasoning with Abstract Argumentation Frameworks: A Feasibility Study. In *International Conference on Scalable Uncertainty Management (SUM)*, 24–37. Springer.

Louizos, C.; Welling, M.; and Kingma, D. P. 2018. Learning Sparse Neural Networks through L_0 Regularization. In *International Conference on Learning Representations (ICLR)*.

Mocanu, D. C.; Mocanu, E.; Stone, P.; Nguyen, P. H.; Gibescu, M.; and Liotta, A. 2018. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications* 9(1): 1–12.

Mossakowski, T.; and Neuhaus, F. 2018. Modular Semantics and Characteristics for Bipolar Weighted Argumentation Graphs. *arXiv preprint arXiv:1807.06685* .

Potyka, N. 2018a. Continuous Dynamical Systems for Weighted Bipolar Argumentation. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 148–157.

Potyka, N. 2018b. A Tutorial for Weighted Bipolar Argumentation with Continuous Dynamical Systems and the Java Library Attractor. International Workshop on Non-Monotonic Reasoning (NMR).

Potyka, N. 2019a. Extending Modular Semantics for Bipolar Weighted Argumentation. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 1722–1730.

Potyka, N. 2019b. Open-Mindedness of Gradual Argumentation Semantics. In *Scalable Uncertainty Management (SUM)*, volume 11940 of *Lecture Notes in Computer Science*, 236–249. Springer.

Potyka, N. 2020. Interpreting Neural Networks as Quantitative Argumentation Frameworks. *arXiv preprint arXiv:2012.05738* .

Rago, A.; Cocarascu, O.; and Toni, F. 2018. Argumentation-based recommendations: Fantastic explanations and how to find them. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1949–1955.

Rago, A.; Toni, F.; Aurisicchio, M.; and Baroni, P. 2016. Discontinuity-Free Decision Support with Quantitative Argumentation Debates. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 63–73.

Riveret, R.; Pitt, J. V.; Korkinof, D.; and Draief, M. 2015. Neuro-Symbolic Agents: Boltzmann Machines and Probabilistic Abstract Argumentation with Sub-Arguments. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 1481–1489.

Tesauro, G. 1995. Temporal difference learning and TD-Gammon. *Communications of the ACM* 38(3): 58–68.

Thimm, M.; and Kersting, K. 2017. Towards argumentation-based classification. In *Logical Foundations of Uncertainty and Machine Learning Workshop*, volume 17.