A neural cognitive model of argumentation with application to legal inference and decision making

Artur S. d'Avila Garcez
Department of Computing
City University London
EC1V 0HB, London, UK.
aag@soi.city.ac.uk

Dov M. Gabbay
Department of Computer Science
King's College London
WC2R 2LS, London, UK
dg@dcs.kcl.ac.uk

Luis C. Lamb
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, 91501-970, Brazil
LuisLamb@acm.org

October 1, 2013

Abstract

Formal models of argumentation have been investigated in several areas, from multi-agent systems and artificial intelligence (AI) to decision making, philosophy and law. In artificial intelligence, logic-based models have been the standard for the representation of argumentative reasoning. More recently, the standard logic-based models have been shown equivalent to standard connectionist models. This has created a new line of research where (i) neural networks can be used as a parallel computational model for argumentation and (ii) neural networks can be used to combine argumentation, quantitative reasoning and statistical learning. At the same time, non-standard logic models of argumentation started to emerge. In this paper, we propose a connectionist cognitive model of argumentation that accounts for both standard and non-standard forms of argumentation. The model is shown to be an adequate framework for dealing with standard and non-standard argumentation, including joint-attacks, argument support, ordered attacks, disjunctive attacks, metalevel attacks, self-defeating attacks, argument accrual and uncertainty. We show that the neural cognitive approach offers an adequate way of modelling all of these different aspects of argumentation. We have applied the framework to the modelling of a public prosecution charging decision as part of a real legal decision making case study containing many of the above aspects of argumentation. The results show that the model can be a useful tool in the analysis of legal decision making, including the analysis of what-if questions and the analysis of alternative conclusions. The approach opens up two new perspectives in the short-term: the use of neural networks for computing prevailing arguments efficiently through the propagation in parallel of neuronal activations, and the use of the same networks to evolve the structure of the argumentation network through learning (e.g. to learn the strength of arguments from data).

Keywords: Argumentation, Neural-Symbolic Reasoning, Legal Decision Making, Cognitive Modeling.

1 Introduction

Formal models of argumentation have been investigated in several areas, from multi-agent systems and artificial intelligence (AI) to decision making, philosophy and law [4, 8, 13, 17, 30, 33]. In artificial intelligence, models of argumentation have been used for commonsense reasoning, modelling chains of *defeasible arguments* to reach a conclusion. Such models are mainly founded on logic-based approaches, which have been the standard for the representation of argumentative reasoning in AI [3].

Recent efforts to bridge the gap between logic-based models of argumentation and cognitive models of computation include [10, 11, 34]. In [10, 11], an equivalence is shown between value-based argumentation [2] and standard connectionist networks [22]. This has created a new line of research in argumentation where (i) neural networks can be used as a cognitive computational model for argumentation and (ii) neural networks can be used to combine argumentation, quantitative reasoning and statistical learning. In [34], behavioural data is used to conclude that in human reasoning, reinstatement does not yield a full recovery of the attacked argument; [1] implements the same idea mathematically through equations that resemble the predator-prey dynamics of species populations. Further work integrating logic and neural networks include [20] where clustering in fuzzy ART networks is used to compute prevailing arguments, and [25] which extends the work in [10] to deal with self-defeating arguments and provides a number of interesting examples. At the same time, some non-standard models of argumentation start to emerge, enriching current models with cognitive abilities; e.g. [15] discusses metalevel attacks, coalitions, disjunctive attacks and argument support, [37] provides an adequate semantics for joint attacks, among much else, [29] seeks to unravel the role of emotions in argumentation, [14, 23] propose to handle uncertainty in argumentation through the assignment of probabilities and weights to arguments, and [12, 26] offer a qualitative method for reasoning about uncertainty and preferences between arguments.

We argue that the adoption of a cognitive approach to argumentation can offer an adequate framework for dealing with both standard and non-standard argumentation models. In this paper, we show that a cognitive approach can model many different aspects of argumentation in a uniform way, in particular, modelling uncertainty in argumentative reasoning and the accrual of arguments. The approach opens up, through the use of a connectionist system, two new short-term perspectives: (i) the use of neural networks to compute prevailing arguments efficiently through the propagation in parallel of neuronal activation signals and (ii) the use of the same networks to evolve the structure of an argumentation network through learning (e.g. to learn the strength of arguments from data). We believe that this approach also opens a more long-term perspective for the research on argumentation: the use of connectionist models of computation to help investigate and evaluate cognitive models of argumentation. For example, ideas from connectionism about the modeling of attention and emotion could be investigated in the context of argumentation [29, 36].

Argumentation has also been proposed as a method for helping machine learning systems [27] where an expert's arguments, or the *reasons* for some of the learning examples, are used to guide the search for hypotheses. This is related to the body of work on abductive reasoning and combinations of abduction and inductive logic programming [24, 28]. It is said that *the arguments constrain the combinatorial search among possible hypotheses, directing the search towards hypotheses that are more comprehensible in the light of an expert's background knowledge* [27]. We subscribe to this idea. In fact, experimental results on the integration of learning with background knowledge using neural networks have been shown to outperform symbolic and purely-connectionist systems, especially in the presence of noisy data [9]. In this paper, differently from [27], however, learning from data can be used to inform a process of numerical argumentation, allowing different perspectives of human argumentation, including joint attacks, argument support, meta-argumentation and disjunctive attacks, to be modelled in the same framework, as detailed in what follows.

The remainder of the paper is organised as follows. First, we define the concepts of argumentation and neural cognitive models used throughout the paper. Then, we present an algorithm, generalised from [10], which translates standard and non-standard argumentation frameworks into standard connectionist networks. We show that the resulting neural model executes a sound parallel computation of the prevailing arguments according to a number of standard argumentation semantics, and also according to value-based argumentation models [2], abstract dialectical frameworks [5], and other forms of human argumentation. We illustrate the network computation through examples that include joint attacks, support, meta-argumentation and disjunctive attacks. Finally, we apply the framework to a real decision making situation in legal reasoning, which indicates that the network model can be a useful tool in the modelling of non-standard and numerical argumentation, and in the analysis of *what-if* questions that emerge in real situations. The paper concludes with a brief discussion and directions for future work.

2 Background

In this section, we present the concepts of argumentation and neural networks used throughout the paper.

Definition 1 *An argumentation framework has the form $\mathcal{A} = < \alpha, attack >$, where α is a set of arguments, and $attack \subseteq \alpha^2$ is a relation indicating which arguments attack which other arguments.*

In order to record the values associated with arguments, in [2] Bench-Capon has extended Dung's argumentation framework [13] by adding to it a set of values and a function mapping arguments to values. This brings argumentation closer to a numerical, connectionist approach.

Definition 2 *A value-based argumentation framework is a 5-tuple $VAF = < \alpha, attacks, V, val, P >$, where α is a finite set of arguments, $attacks$ is an irreflexive binary relation on α, V is a non-empty set of values, val is a function mapping elements in α to elements in V, and P is a set of possible audiences, where we may have as many audiences as there are orderings on V. For every $A \in \alpha$, $val(A) \in V$.*

Bench-Capon also defines the notions of *objective* and *subjective* acceptability of arguments. The first are arguments acceptable no matter the choice of preferred values for every audience, whereas the second are acceptable to some audiences. Arguments which are neither objectively nor subjectively acceptable are called *indefensible*. A function v from *attack* to $\{0, 1\}$ gives the relative strength of an argument. Given $\alpha_i, \alpha_j \in \alpha$, if $v(\alpha_i, \alpha_j) = 1$ then α_i is said to be stronger than α_j. Otherwise, if $v(\alpha_i, \alpha_j) = 0$ then α_i is weaker than α_j.

We shall also relate the neural approach with abstract dialectical frameworks [5].

Definition 3 *An abstract dialectical framework (ADF) is a tuple $D = (S, L, C)$ where S is a set of nodes, $L \subseteq S \times S$ is a set of links, $C = \{C_s\}_{s \in S}$ is a set of total functions $C_s : 2^s \to \{in, out\}$, one for each node s.*

Consider an example. A person is innocent (i), unless she is a murderer (m). A killer (k) is a murderer (m), unless she acted in self-defence (s). There must be evidence for self-defence, e.g. a witness (w) who is not known to be a liar (l). An ADF can model the above example by stating that s is in if w is in and l is out. Similarly, m is in if k is in and s is out. In other words, like neural networks, ADFs include the concept of support. If k and w are in and l is out then s will be in; s then defeats m regardless of k so that i prevails. Every argumentation framework has an associated ADF. Also, normal logic programs have associated ADFs [5]. Since every logic program also has an associated neural network [9], this fact will be used later to show the required correspondences.

Other established definitions of argumentation semantics will be considered as well [6, 7, 13]. In all the definitions that follow, an argumentation framework is a pair $\mathcal{A} = < \alpha, attack >$, as above. First, let us define a function $F : 2^\alpha \to 2^\alpha$, such that $F(Args) = \{A \mid A \text{ is defended by } \alpha\}$, where $Args \subseteq \alpha$. The function F computes the arguments accepted in the sense of [13] or defended by a set of arguments, in the sense of [7]. In this way, we can define conflict-free sets of arguments. A set of arguments is *conflict-free* if and only if (iff, for short) it does not contain any arguments A and B such that A defeats B. Let $Args$ be a conflict-free set of arguments. $Args$ is said to be a *complete extension* iff $Args = F(Args)$.

In another useful argumentation semantics, the *grounded semantics*, only one extension is yielded by making use of the function F and defining the *grounded extension* as the minimal fixed point of F [31]. A grounded extension is conflict-free [7]. In the *preferred semantics* [13], a more credulous approach is used, which maximizes the number of *accepted* arguments. In order to define *preferred semantics*, Dung introduces the notion of admissible sets of arguments. A set of arguments is *admissible* iff it is conflict-free and $Args \subseteq F(Args)$. The set $Args$ is a *preferred extension* iff $Args$ is maximal w.r.t. set inclusion. In the *stable semantics* of argumentation [18], a set of arguments is called *stable* iff it defeats each argument that does not belong to this set.

In semi-stable semantics, a set of arguments $Args$ is *semi-stable* iff $Args$ is a complete extension of which $Args^+ = \alpha \setminus Args$ is maximal and defines a set of arguments which are defeated by an argument in $Args$.

In order to illustrate the different argumentation semantics, we borrow an example from [7]. Consider the abstract argumentation framework depicted as a directed graph in Fig. 1, where each node is an argument and an arrow from argument X to argument Y denotes an attack from X on Y. This framework has grounded extension {E,F}; complete extensions {E,F},{B,C,E,F} and {A,E,F}; preferred extensions {B,C,E,F} and {A,E,F}; stable extension {B,C,E,F}; and semi-stable extension {B,C,E,F}. As pointed out in [7], semi-stable and stable extensions will coincide whenever the framework has at least one stable extension.

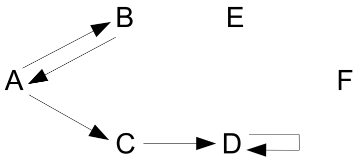

Figure 1: An abstract argumentation framework with arguments A, B, C, D, E, and F; D is a self-defeating argument

Finally, we shall also consider meta-argumentation [15]. In a meta-argumentation network, let argument a attack b in the usual way. It is possible to define an argument c as an attack on a's attack. This makes the framework more fine-grained in that c's attack does not propagate throughout the network, but is targeted at one specific attack in the network. Meta-argumentation can be reduced to argumentation frameworks with the addition of a node denoting (c, a) and a careful re-organization of the network [15].

We shall use a standard definition of neural networks, as follows. A neural network is a directed graph with the following structure: a unit (or neuron) in the graph is characterized, at time t, by its *input vector* $I_i(t)$, its *input potential* $U_i(t)$, its *activation state* $A_i(t)$, and its *output* $O_i(t)$. The units of the network are interconnected via a set of directed and weighted connections such that if there is a connection from unit i to unit j then $W_{ji} \in \mathbb{R}$ denotes the *weight* of this connection. The input potential of neuron i at time t ($U_i(t)$) is obtained by computing a weighted sum for neuron i such that $U_i(t) = \sum_j W_{ij} I_i(t)$. The activation state $A_i(t)$ of neuron i at time t is then given by the neuron's *activation function* h_i such that $A_i(t) = h_i(U_i(t))$. Typically, h_i is either a linear function, a non-linear step function, or a sigmoid function such as $tanh(x)$. In this paper, we use $tanh(x)$ as activation function and inputs values in the range $[-1, 1]$. In addition, θ_i (an extra weight with input always fixed at 1) is known as the *bias* of neuron i. We say that neuron i is *active* at time t if $A_i(t) > -\theta_i$. Finally, the neuron's output value $O_i(t)$ is given by its output function $f_i(A_i(t))$. Usually, f_i is the identity function so that $O_i(t) = A_i(t)$.

3 Neural Cognitive Argumentation Frameworks

We start by considering the relationship between argumentation and neural networks informally. If we represent an argument by a neuron then a connection from neuron i to neuron j can indicate that argument i either attacks or supports argument j, the weight of the connection corresponding to the strength of the attack or support. Since real numbers are used as weights in a neural network, we associate negative weights with attacks, positive weights with support, and zero weight with the lack of an attack or support.

Definition 4 *We say that an argument prevails at time t when the activation state of its associated neuron is greater than a predefined value A_{min} at time t, $0 < A_{min} < 1$. We say that an argument is defeated at time t when the neuron's activation is smaller than $-A_{min}$. Otherwise, i.e. for activations in the interval $[-A_{min}, A_{min}]$, we say that it is unknown whether an argument prevails or not.*

There are different ways in which an argument may support other arguments. For example, an argument i may support argument j by attacking an argument k that attacks j, or argument i may support j directly, e.g. by strengthening the value of j, or even i and j may *get together* to attack k. Generally, an argument i supports an argument j if the coordination of i and j reduces the likelihood of j being defeated [39]. A general neural network structure capable of accounting for the above combinations of attack and support and the necessary computations of prevailing arguments would be a recurrent network having an input layer, a single hidden layer, and an output layer with feedback from the output to the input layer [9]. At time t_0, input values are provided to the network. Neuronal activation is then propagated in parallel from the input to the hidden layer at time t_1, and to the output of the network at time t_2. At time t_3, the output values can be fed back to the input of the network, and this process can be repeated until a stable state is obtained, when input and output values will be the same for each pair of neurons with feedback. Arguments for which the associated neuron is activated at the stable state are said to prevail.

Consider the neural network of Fig. 2(b), which implements the argumentation network of Fig. 2(a). Arguments A, B and C are encoded in the network's input and output layers. In this example, arguments do not get together to attack another argument so that each input neuron is connected directly to a hidden neuron and the weights from the input to the hidden layer simply serve to send the input information forward. Support and attack information is encoded by the weights leading from the hidden to the output layer of the network. Support for A is encoded by the positive weight going from neuron h_1 to output neuron A. Similarly, support for B (resp. C) is encoded by the weight going from h_2 (resp. h_3) to B (resp. C). A's attack on B is represented by the dashed arrow going from h_1 to B, which should have negative weight, as specified in the algorithm below. Similarly, B's attack on C is represented by the dashed arrow going from h_2 to C, with a negative weight.

If the absolute value of the weight going from h_1 to B (call it W_1) is larger than the value of the weight from h_2 to B (call it W_2) then A defeats B. This produces $\{A, C\}$ as prevailing arguments and is identical to the usual (non-value based) interpretation of argumentation frameworks [13]. The prevailing arguments are computed by the network as follows: suppose that A, B and C are all present at time t_0, denoted by input vector $[1, 1, 1]$. Hidden neurons h_1, h_2 and h_3 all become activated; h_1 activates A, and blocks the activation of B from h_2 (because $W_1.h_1 + W_2.h_2 < 0$). At the same time, C is blocked by h_2 (by default, we assume that attacks are stronger than support, i.e. the weight from h_2 to C is greater in absolute value than the weight from h_3 to C, unless stated otherwise). Thus, at time t_2, only output neuron A is activated (i.e. only the output of neuron A is greater than A_{min}). This is then represented as a new input vector $[1, -1, -1]$ at time t_3. Given this new input, A continues to prevail, B is defeated as before, but C is reinstated because B now fails to defeat it, since B is not present in the input anymore. Thus, at time t_5, output neurons A and C become activated. At time t_6, a new input $[1, -1, 1]$ is produced. Finally, with $[1, -1, 1]$ given as input, output neurons A and C become activated again, producing a stable state in the network. Recall that network outputs are real values in the range $(-1, 1)$. To compute a stable state, each output value in the range $(A_{min}, 1)$ is mapped to 1, output values in the range $(-1, -A_{min})$ are mapped to -1, and values in the range $[-A_{min}, A_{min}]$ are mapped to 0. After this is done, the network's new input vector can be compared with its previous input. In this example, the input vector at time t_9 will be identical to the input vector at time t_6, that is $[1, -1, 1]$ is a stable state. The network's stable activations indicate the prevailing arguments $\{A, C\}$. This result also coincides with the standard ADF interpretation of support (Definition 3). In the case of VAF (Definition 2), if B is preferred over A, all that needs changing in the network is the value of W_1 or W_2 so that $|W_1| < W_2$, denoting that the attack from A on B should not be strong sufficiently to defeat B [10]. In this case, input $[1, 1, 1]$ produces new input $[1, 1, -1]$, which is a stable state, given the neural network's new set of weights. This new network, therefore, computes prevailing arguments $\{A, B\}$, as expected in the case of the VAF under consideration.

As another example, consider the network of Fig. 3(b). Here, a cycle exists in the argumentation network. This may create an infinite loop in the computation of the stable state of the associated neural network; e.g. if the network were to be started on input vector $[1, 1, 1]$, it would oscillate between that state and state $[-1, -1, -1]$ indefinitely, with the arguments all being attacked in parallel and defeated in a single pass through the network, and then reinstated in the next pass through the network. In order to handle this as intended by the usual argumentation semantics, whenever the neural network reaches a state $[-1, -1, ..., -1]$, the computation stops. In this case, the neural network computes the grounded extension of the argumentation network, as discussed in more detail later. Summarising, our policy is that the network computation stops when either a stable state is obtained or when $[-1, -1, ..., -1]$ is reached. We call $[-1, -1, ..., -1]$ a terminal state. Notice that by following a value-based approach, and changing the weights according to some preference relation, one might eliminate the loop [10]. For example, if as before, B is preferred over A then the attack from A on B will not be successful, with input $[1, 1, 1]$ producing stable state $[1, 1, -1]$. When the weights are different, the neural network is less likely to enter into an infinite loop (see [10] for a discussion on how weight learning given new information following a value-based approach can be useful at resolving loops).

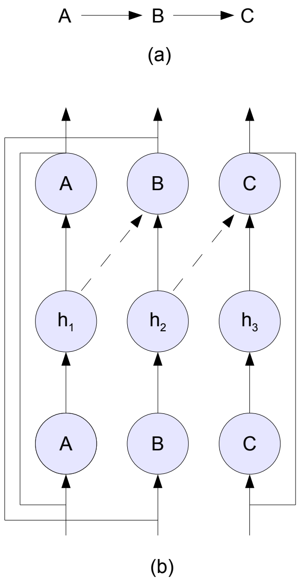

Figure 2: (a) Illustration of an argumentation network in which argument A attacks argument B, which in turn attacks argument C, and (b) its neural implementation, where solid lines receive positive weights and dashed lines, which represent the attacks, receive negative weights

The algorithm below generalises the algorithm first introduced in [10], which was made for VAF only. It translates argumentation frameworks into single-hidden layer neural networks that behave as exemplified, and can be used to compute prevailing arguments in parallel. It does so by defining the structure and set of weights of a neural network as a function of A_{min}, using activation function $tanh(x)$ and inputs in $\{-1, 1\}$. Each argument has a strength given by positive weight W_l. Certain arguments may attack other arguments with negative weight W', and certain arguments may support other arguments with positive weight W_s. The weights are such that activations in the interval $[-A_{min}, A_{min}]$ are guaranteed not to occur for now (we shall consider uncertainty in the next section). The neural network produces outputs in the intervals $(-1, -A_{min})$, which is mapped to *false* and denotes that an argument is defeated, and $(A_{min}, 1)$, which is mapped to *true*, denoting that an argument prevails.

By default, Algorithm 1 implements Dung's argumentation frameworks. If an argument α_i attacks an argument α_j, and α_i is itself not attacked, then neuron α_i should block neuron α_j. However, if α_i is deemed weaker than α_j, and no other argument attacks α_j, then neuron α_i should not block neuron α_j. To achieve this, and account for a number of other, alternative semantics and modes of argumentation in a neural network [16], the constraints on W_l and W' can be modified, as will become clearer in Section 4. Also, in Algorithm 1, a single supporting argument is deemed sufficient to defeat any attack; the other alternatives, leading to different constraints on W_s and

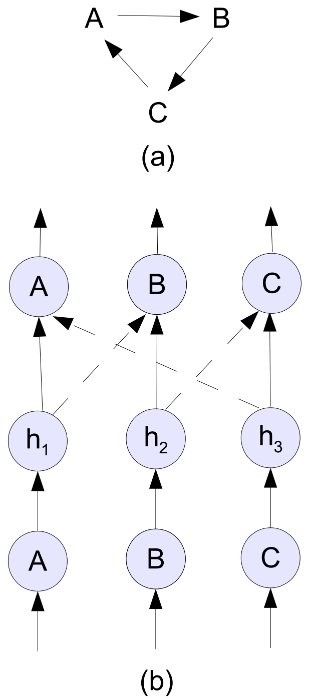

Figure 3: (a) An example of a cyclic argumentation network and (b) its neural implementation whose parallel computation produces, in a single pass through the network, a terminal state [-1,-1,-1] with no prevailing argument, following a grounded argumentation semantics

θ_j, will be considered in Section 4. A novelty of the algorithm is that arguments may have strengths W_l, which may vary from one attack to another. Before we consider the extensions of Algorithm 1, however, let us make the ideas developed so far more precise.

Definition 5 *($\mathcal{N}$ computes $\mathcal{A}$) Let $(\alpha_i, \alpha_j) \in attacks$. Let $A_\alpha(t)$ denote the activation of neuron α at time t. We say that a neural network $\mathcal{N}$ computes an argumentation framework $\mathcal{A}$ if whenever $A_{\alpha_i}(t) > A_{min}$ and $A_{\alpha_j}(t) > A_{min}$ then $A_{\alpha_j}(t+1) < -A_{min}$, and whenever $A_{\alpha_i}(t) < -A_{min}$ for every α_i such that $(\alpha_i, \alpha_j) \in attacks$ then $A_{\alpha_j}(t+1) > A_{min}$ (reinstatement).*

Proposition 6 *For each argumentation framework $\mathcal{A}_v$ there exists a single-hidden layer neural network $\mathcal{N}$ such that $\mathcal{N}$ computes $\mathcal{A}_v$.*
***Proof**: First, we show that if $A\alpha_i(t) > A_{min}$ in the input layer of $\mathcal{N}$ then $A\alpha_i(t+1) > A_{min}$ in the output layer of $\mathcal{N}$ whenever there are no attacks on α_i. In the worst case, the input potential of hidden neuron N_i is $A_{min}W_l$, and the output of N_i is* $\tanh(A_{min}W_l)$. *We want* $\tanh(A_{min}W_l) > A_{min}$. *Again in the worst case, the input*

Algorithm 1: General Neural Argumentation Algorithm

```
Given an argumentation framework $\mathcal{A}$ with arguments $\alpha_1, \alpha_2, ..., \alpha_n$:
1. Create an input and output layer of a neural network $\mathcal{N}$ with $n$ neurons each such that
the i-th neuron corresponds to argument $\alpha_i$;
2. Given $0 < A_{min} < 1$, calculate $W = \frac{1}{A_{min}} tanh^{-1}(A_{min})$;
3. For each argument $\alpha_l$ in $\mathcal{A}$ $(1 \leq l \leq n)$ do:
    (a) Add a neuron $N_l$ to the hidden layer of $\mathcal{N}$;
    (b) Connect input neuron $\alpha_l$ to neuron $N_l$ and set the connection weight to:
        $W_l > W$;
    (c) Connect neuron $N_l$ to output neuron $\alpha_l$ and set the connection weight to $W_l$;
4. For each $(\alpha_i, \alpha_j) \in attack$, do:
    (a) Connect neuron $N_i$ to output neuron $\alpha_j$;
    (b) Set the connection weight to $W' < \frac{2(tanh^{-1}(-A_{min}))+(A_{min}-1)W_l}{(n+1)A_{min}-n+1}$;
    (c) Set the bias $\theta_j$ of output neuron $\alpha_j$ such that
    $(tanh^{-1}(A_{min}) - A_{min}W_l + nA_{min}W') < \theta_j <$
    $(tanh^{-1}(-A_{min}) - W_l + (n - 1 - A_{min})W')$,
    where $n$ is the number of attacks on $\alpha_j$.
5. If an argument $\alpha_i$ supports argument $\alpha_j$, do:
    (a) Connect neuron $N_i$ to output neuron $\alpha_j$;
    (b) Set the connection weight to $W_s > \frac{2(tanh^{-1}(-A_{min}))+(A_{min}-1)W_l}{(n+1)A_{min}-n+1}$;
    (c) Set the bias $\theta_j$ of output neuron $\alpha_j$ such that
    $(tanh^{-1}(A_{min}) - nW' + W_l - A_{min}W_s) < \theta_j <$
    $(tanh^{-1}(-A_{min}) - A_{min}W' + (n-1)W' + A_{min}W_l + A_{min}W_s)$,
    where $n$ is the number of attacks on $\alpha_j$;
6. Set the bias of any other neuron to zero.
```

potential of output neuron α_i will be $A_{min}W_l$, and we need $\tanh(A_{min}W_l) > A_{min}$. As a result, $W_l > \tanh^{-1}(A_{min})/A_{min}$ needs to be satisfied. When there is an attack on α_i, the activation of output neuron α_j needs to be smaller than $-A_{min}$ if hidden neuron N_i is active. In the worst case, N_i has activation A_{min}, N_j has activation 1, and any other attacking neuron has activation -1. Hence, $tanh(A_{min}W' + W_l - (n-1)W' + \theta) < -A_{min}$ has to be satisfied, where n is the number of attacks. Dually, when there are no attacks on α_i, the following inequality has to be satisfied: $tanh(A_{min}W_l - nA_{min}W' + \theta) > A_{min}$, which gives $W' < \frac{2(tanh^{-1}(-A_{min}))+(A_{min}-1)W_l}{(n+1)A_{min}-n+1}$ and the constraint on θ_j, as shown in Algorithm 1, step 4. When at least one argument α_i supports argument α_j, the following inequality has to be satified (again, in the worst case analysis): $tanh(nW' - W_l + A_{min}W_s + \theta) > A_{min}$. Dually, in the worst case, $tanh(A_{min}W' - (n-1)W' - A_{min}W_l - A_{min}W_s + \theta) < -A_{min}$ has to be satisfied, which gives $W_s > \frac{2(tanh^{-1}(-A_{min}))+(A_{min}-1)W_l}{(n+1)A_{min}-n+1}$ and the constraint on θ_j, as shown in Algorithm 1, step 5. This completes the proof.

Proposition 7 *For each ADF $\mathcal{A}_a$ there exists a single-hidden layer neural network $\mathcal{N}$ such that $\mathcal{N}$ computes $\mathcal{A}_a$.*
Proof: *The standard ADF interpretation is that $v(\alpha_i, \alpha_j) = 1$ when α_i attacks α_j. This interpretation has been covered in the proof of Proposition 6. In weighted ADFs, however, one can distinguish different combinations of attack and support [5], in particular, a supporting link can be stronger than an attacking link so that the attacked argument*

prevails. Since neural networks with as few as a single hidden layer are universal approximators, it follows that in the neural argumentation approach, any Boolean combination of attacks and support can be computed. In the specific case of support, we need to show that if α_i *supports* α_j *then if* $C_{\alpha_i}(t) = in$ *then* $A_{\alpha_j}(t+1) > A_{min}$*. As before, we associate inputs in the interval* $(A_{min}, 1)$ *to* in *and inputs in the interval* $(-1, -A_{min})$ *to out. In the worst case, we need* $tanh(nW' - W_l + A_{min}W_s + \theta) > A_{min}$*. Thus,* $W_s > \frac{2(tanh^{-1}(-A_{min}))+(A_{min}-1)W_l}{(n+1)A_{min}-n+1}$*, as guaranteed by the algorithm. This completes the proof. Notice that, in practice, we convert inputs in the interval* $(A_{min}, 1)$ *to* 1*, and inputs in the interval* $(-1, -A_{min})$ *to* -1*, which relaxes further the above constraint on* W_s*.*

We can also show that the neural-network approach is very general by proving that it computes the many different argumentation semantics, as defined earlier [6, 13]. Before that is done, however, we should note that an argument that is not attacked by any other argument cannot be reinstated in the neural network (an argument that is attacked but not defeated will be reinstated as usual). For example, argument E in Fig. 1 will be in if its input value is 1, and will be out if its input value is -1. It will continue to be out over time if it is out initially, and will be in otherwise. Argument B, on the other hand, will be reinstated whenever argument A is not present, and vice-versa.

Lemma 8 *For each argumentation framework* $\mathcal{A}$ *there exists a single-hidden-layer recurrent neural network* $\mathcal{N}$ *such that the complete extensions of* $\mathcal{A}$ *are stable states of* $\mathcal{N}$*.*
Proof*: Recall that a set of arguments* $Args$ *is said to be a* complete extension *of* $\mathcal{A}$ *iff* $Args = F(Args)$*. From Proposition 6, one pass through* $\mathcal{N}$ *computes* $F(Args)$*. Recurrently connected,* $\mathcal{N}$ *iterates* $F(Args)$ *until a stable state is reached, which is a fixed point of* F*, that is* $Args = F(Args)$*.*

Corollary 9 *For each argumentation framework* $\mathcal{A}$ *there exists a recurrent neural network* $\mathcal{N}$ *such that the grounded, preferred, stable and semi-stable extensions of* $\mathcal{A}$ *are stable states of* $\mathcal{N}$*.*

As an example, consider again the argumentation framework of Fig. 1. Starting from $\{E, F\} \in in$, without a terminal state, the associated neural network can converge to the following stable states: {B,C,E,F} and {A,E,F}, corresponding to its preferred extensions. With a terminal state, the network will converge to either {E,F},{B,C,E,F} or {A,E,F}, corresponding to its complete extensions.

Finally, consider the argumentation network of Fig. 4, for which no stable state exists. With the use of a terminal state, the corresponding neural network will implement a semi-stable semantics, providing the empty set as the only extension. Without a terminal state, the neural network will loop (until it is either halted or its weights are changed), implementing a stable semantics.

The neural networks of Figs. 2 and 3 should be seen as computational models rather than abstract argumentation frameworks. As discussed, the networks can be used in the parallel computation of prevailing arguments: given input vector $[1, 1, ..., 1]$ at the start, the networks should always converge to a stable state corresponding to a set

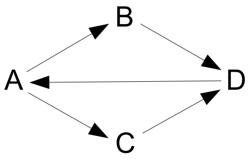

Figure 4: Semantic circularity for which no stable state exists

of prevailing arguments, according to a (value-based, preferred, etc.) argumentation semantics.

Definition 10 *We say that neural network $\mathcal{N}$ computes an (extension-based semantics of) argumentation framework $\mathcal{A}$ if every stable state of $\mathcal{N}$ is an extension of $\mathcal{A}$.*

Proposition 11 *If $\mathcal{N}$ computes $\mathcal{A}$ and $\mathcal{A}$ admits a single extension then starting from input vector [$1, 1, ..., 1$], $\mathcal{N}$ always finds a stable state corresponding to the prevailing arguments of $\mathcal{A}$.*
Proof*: The arguments in $\mathcal{N}$ can be viewed as logic programming clauses of the form $\alpha_i \rightarrow \alpha_i$. Starting from [$1, 1, ..., 1$], $\mathcal{N}$ will treat each α_i as a fact unless α_i is attacked and defeated by another argument α_j, which corresponds to adding a clause $\neg\alpha_j \rightarrow \alpha_i$ to the program (where $\neg$ stands for negation by failure). If $\mathcal{A}$ admits a single extension then the corresponding program will have a single model. The stable models of any logic program can be computed by a single-hidden layer neural network; with a single model, the network is known to always settle down in a stable state corresponding to this model, as proved in [9]. Such a result can be applied directly here to complete the proof.*

As exemplified earlier, it should be possible to extend Proposition 11 to certain very general classes of argumentation networks that admit multiple extensions. Due to the numerical nature of the networks, they are unlikely to oscillate unless integer weights with the same absolute values are used. For example, we have seen that, starting from $[1, 1, ..., 1]$, the network of Fig. 3 converges to a terminal state. Otherwise, it loops. As argued in [10], a network that loops should be seen as an opportunity for learning, whereby *what-if* questions can be considered and the network's weights can be changed slightly. Similarly, when a network reaches a terminal state, this should trigger a search for new evidence about the relative strength of the arguments. For example, consider the case where arguments A and B attack each other in a cycle. The corresponding neural network has two stable states, namely $[1, -1]$ and $[-1, 1]$, corresponding to the two stable extensions of the argumentation network. Starting from $[1, 1]$, the neural network produces output $[-1, -1]$, which is a terminal state. At this point, the computation stops. However, a loop exists in the network computation, since input $[-1, -1]$ would produce output $[1, 1]$. A terminal state does not provide much information by itself (a terminal state could be seen as producing maximum uncertainty). However, if the reaching of a terminal state is seen as a trigger for a search for more evidence, perhaps this search could shed new light on the relative strengths of arguments

A and B, defining a preference for either argument by changing the weights of the network. If a neural network is such that the weights do not have the same absolute values then it is unlikely that arguments will cancel each other, as seen above, so that the network loops. Breaking such a symmetry in the set of weights by changing their values slightly is the norm of network learning, and could be seen as an alternative to the use of terminal states and a solution to the problem of having loops in the computation of such neural networks [10].

4 Neural Cognitive Nonclassical Argumentation

It is natural for a neural network to combine the weights representing multiple support or multiple attacks in order to compute the activation state of a neuron/argument. This is interesting in relation to the question of the accrual of arguments [38]. So far, our standard interpretation has been that any attack suffices to defeat an argument, unless a value-based function says otherwise. Another interpretation, however, is that arguments may get together to attack an argument (that is, only the conjunction of the arguments enables the attack; this is called a joint-attack [1]). Fig. 5(a) shows two arguments A and C attacking argument B. When either $v(A, B) = 1$ or $v(C, B) = 1$ for the standard interpretation, the attacks can be implemented by Algorithm 1 above. However, when arguments are allowed to accrue and either $v(A, B) = 0$ or $v(C, B) = 0$, a decision has to be made as to whether or not A and C together can defeat B [10]. The same is true for support. Fig. 5(b) shows an argument A supporting argument B as done in ADFs. It may be that without A's support, B would be defeated, say, by an attack from another argument C, but B is not defeated with A's support. Finally, Fig. 5(c) shows a situation where, only if A and B prevail, can they attack C. Hidden neuron h_1 implements, in the usual way, a logical-AND. This is the situation where arguments A and B "get together" to attack C.

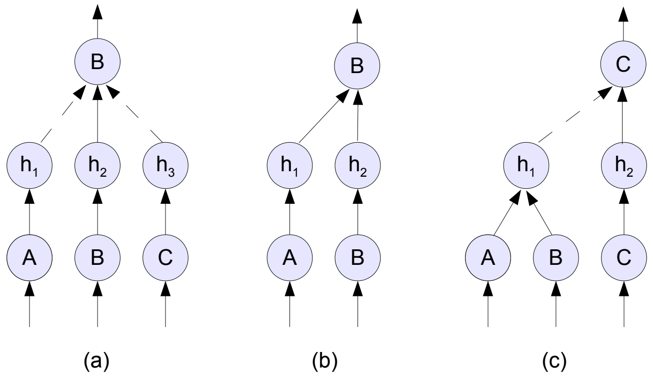

Figure 5: Alternative modes of attack and support: (a) multiple attacks, (b) support, and (c) joint-attacks

Let us consider Fig. 5(a) in more detail. Let W_{B_i} denote the weight from hidden neuron h_i to output neuron B. As said, the natural computation of a neural network will combine the weights into the input potential of B. Suppose that $W_{B_2} = 3$ and $W_{B_1} = W_{B_3} = -2$. In this situation, neither A nor C defeat B, but, together, A and C may defeat B as an unintended consequence of the combination of the weights. To avoid this problem, we use the following convention: when arguments create a joint attack on another argument, the set-up of Fig. 5(c) is used. If n arguments are joined through hidden neuron h_j, then the bias θ_{h_j} of h_j should satisfy the following inequality, which implements the intended logical-AND.

$$-nWA_{\min} < \theta_{h_j} < W - (n-1)WA_{\min}$$

Let us now consider in more detail the situation where an argument receives multiple support (Fig. 5(b)). This situation is similar to that considered earlier, that is, whether some attack is strong enough to defeat all of the support received by B or whether the support for A overrides the attack, making it unsuccessful. The first case is straightforward and can be implemented by making:

$$W' < \frac{1}{A_{min}}(tanh^{-1}(-A_{min}) - \theta_j - \sum_{i=1}^{k} W_s),$$

where k is the number of supporting arguments, each with weight $W_s > 0$.

The second case, in a more general set-up, takes a linearly ordered set $\alpha_n \succ ... \succ \alpha_2 \succ \alpha_1$ such that argument α_1 prevails if it is not attacked, but α_1 is defeated when attacked by α_2. However, α_1 prevails again if it is supported by another argument α_3, but is defeated again if attacked by α_4, and so on. We need to assign values to weights $W_{\alpha_1}, W_{\alpha_2}, ..., W_{\alpha_n}$, as follows:

$$\begin{aligned}
&W_{\alpha_1} = W, \\
&W_{\alpha_2} = -W + \varepsilon, \\
&W_{\alpha_j} = W_{\alpha_{j-2}} - W_{\alpha_{j-1}} + \varepsilon, j \in \{3, 5, 7, ...\}, j \leq n, \\
&W_{\alpha_j} = W_{\alpha_{j-2}} - W_{\alpha_{j-1}} - \varepsilon, j \in \{4, 6, 8, ...\}, j \leq n.
\end{aligned}$$

where $W > 0$ and ε is a small positive number such that $W >> \varepsilon$ (typically $\varepsilon = 0.1$).

A further possibility would be to allow certain combinations of support to prevail and certain others to be defeated. This is, in fact, a likely outcome if a learning algorithm is to be applied, with the network's weights changing as a result of learning from data. Any combination of attack and support can be encoded in a neural network, as follows. The linear ordering above can be extended to multisets in the usual way, so that each α_j, $2 \leq j \leq n$, denotes a set of arguments. The value of W_{α_j} will then correspond to the sum of the weights either attacking or supporting α_1, each weight being equal to W_{α_j}/k, where k is the cardinality of the set of arguments in question. Since we would like the combination of k arguments, and not of $k-1$ or fewer arguments, to have the above effect, the following inequality should be satisfied:

$$\mid W_{\alpha_{j-1}} \mid \; > \; \mid W_{\alpha_j}/(k-1) \mid$$

The dual of the conjunctive attacks exemplified above are the *disjunctive attacks* shown in Fig. 6(a). In the figure, the activation of hidden neuron $\vee$ should attack either argument B or C according to some probability distribution. We say that neuron $\vee$ behaves stochastically. Having a standard hidden neuron in Fig. 6(a) would denote that A attacks both B and C. With a stochastic hidden neuron, either B or C is attacked with a probability. This offers a way of implementing the idea of a disjunctive attack, i.e. one that attacks either argument, but it does not matter which argument. If, for example, the probability of attacking argument B should be 50%, a random number is generated in the interval [0,1] and, if this number is greater than 0.5 then argument B is attacked; otherwise, argument C is attacked.

Let $\bigvee_i \alpha_1, ..., \alpha_n$ denote the arguments (neurons) attacked stochastically through hidden neuron $\vee_i$. From the point of view of the network computation, for each $\vee_i$, we select a neuron α_j, $1 \leq j \leq n$, at a time (at each round, a single α_j is chosen to receive activation from $\vee_i$. A stable state denoting a set S_{ij} of potential prevailing arguments is then obtained in the usual way (with the same α_j being selected if the network recurs through $\vee_i$). We take $\bigcap_{i \times j} S_{ij}$ as our final set of prevailing arguments (i.e. following a skeptical semantics).

In addition to disjunctive attacks, the recent literature on argumentation has discussed extensively the modelling of self-defeating arguments as well as the concept of meta-argumentation [15, 25, 26]. Fig. 6(b) exemplifies the implementation of a self-defeating argument in a neural network. In the figure, argument A attacks argument B, but A is self-defeating, that is, the weight of the connection to output neuron A is negative. Hence, B prevails, as expected. Fig. 6(c) exemplifies a metalevel attack, i.e. an attack not on an argument, but on another attack. In Fig. 6(c), argument A does not attack argument B, but it attacks B's attack on C. As a result, C prevails. In this setting, it is possible for B to succeed in attacking another argument, say D, through a different hidden neuron. Notice the similarity between Fig. 6(c) and Fig. 5(c). Not surprisingly, the semantics of metalevel attacks can be characterised in terms of joint-attacks, by adding the metalevel attack itself as a node in the argumentation network [15].

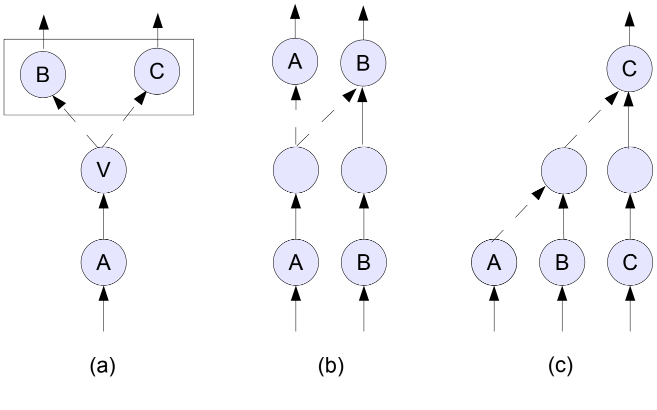

Figure 6: Disjunctive attacks (a), self-defeating attacks (b), and attacks on attacks (c)

Definition 12 *Let $a \mapsto b$ denote an attack from argument a on argument b. Suppose* *that argument c attacks this attack; we write it as $c \mapsto (a \mapsto b)$. A meta-argumentation network is an argumentation network extended with such attacks on attacks.*

Lemma 13 *([15]) Let $\mathcal{A}_m$ be a meta-argumentation network where $\alpha_i \mapsto (\alpha_j \mapsto \alpha_k)$. $\mathcal{A}_m$ can be reduced to an argumentation network containing an extra node $(\alpha_j \mapsto \alpha_k)$ such that α_j and $(\alpha_j \mapsto \alpha_k)$ jointly attack α_k, and α_i attacks $(\alpha_j \mapsto \alpha_k)$.*

Proposition 14 *For any meta-argumentation network $\mathcal{A}_m$ there exists a neural network $\mathcal{N}$ such that $\mathcal{N}$ computes $\mathcal{A}_m$.*
***Proof**: We are concerned with the situation $\alpha_i \mapsto (\alpha_j \mapsto \alpha_k)$ without recursion. In the reduced network, node i attacks node jk, and nodes jk and j jointly attack node k. In the neural network (with the same structure as in Fig. 6(c)), we have $\neg i \wedge j$ attacking k (recall that we have defined joint-attacks as conjunctions). If i is in then jk is out and k is in; if i is out then k is out iff j is in. In the neural network, the hidden neuron representing jk will be activated, attacking and defeating k iff $i < -A_{min}$ (or out) and $j > A_{min}$ (or in), which clearly produces the same intended outcome.*

5 Case Study: Public Prosecution Charging Decision

In this section, we apply the neural cognitive argumentation framework to the modelling of a public prosecution charging decision, which was part of a real legal case. The modelling of the charging decision includes many of the argumentation aspects addressed by our proposed framework, notably uncertainty, joint-attacks, argument support and ADF-style reasoning. The results show that the neural model can be a useful tool in the analysis of legal decision making, including the analysis of what-if questions, as detailed below.

The following is an extract from a charging decision statement made 29 May 2012 by Alison Levitt, Queen's Counsel, Principal Legal Advisor to the Director of Public Prosecutions in relation to allegations that a police officer passed confidential information to a journalist about *Operation Weeting*, a police investigation into allegations of phone hacking by newspapers in the United Kingdom.

In what follows, we will represent the main aspects of the arguments discussed in the charging decision statement as a neural cognitive model, and analyse the possible model set-ups and computations, and their alternative conclusions in relation to the actual outcomes of this legal case study. We are interested in exemplifying the use of the proposed model in practice, and analysing its usefulness as a tool for modelling the different aspects of argumentation, including the analysis of what-if questions, as detailed below.

> "On 2 April 2012 the Crown Prosecution Service received a file of evidence from the Metropolitan Police Service requesting charging advice in relation to two suspects. The first is a serving Metropolitan Police Officer in the Operation Weeting team whose name is not in the public domain. He is currently suspended. The second suspect is Amelia Hill, a journalist who writes for The Guardian newspaper.

The allegation is that the police officer passed confidential information about phone hacking cases to the journalist.

All the evidence has now carefully been considered and I have decided that neither the police officer nor the journalist should face a prosecution. The following paragraphs explain the reasons for my decision.

The suspects have been considered separately, as different considerations arise in relation to each of them.

Between 4 April 2011 and 18 August 2011, Ms. Hill wrote ten articles which were published in The Guardian. I am satisfied that *there is sufficient evidence to establish that these articles contained confidential information* derived from Operation Weeting, including the names of those who had been arrested. I am also satisfied that there is sufficient evidence to establish that *the police officer disclosed that information* to Ms Hill.

I have concluded that there is insufficient evidence against either suspect to provide a realistic prospect of conviction for the common law offence of misconduct in a public office or conspiracy to commit misconduct in a public office.

In this case, *there is no evidence that the police officer was paid any money* for the information he provided.

Moreover, the *information disclosed by the police officer, although confidential, was not highly sensitive*. It did not expose anyone to a risk of injury or death. It did not compromise the investigation. And the information in question would probably have made it into the public domain by some other means, albeit at some later stage.

In those circumstances, I have concluded that there is no realistic prospect of a conviction in the police officer's case because his alleged conduct is not capable of reaching the *high threshold necessary to make out the criminal offence of misconduct in public office*. It follows that *there is equally no realistic prospect of a conviction against Ms. Hill* for aiding and abetting the police officer's conduct.

However, *the information disclosed was personal data* within the meaning of the Data Protection Act 1998 and I am satisfied that there is arguably sufficient evidence to charge both the police officer and Ms. Hill with offences under section 55 of that Act, even when the available defences are taken into account.

I have therefore gone on to consider whether a prosecution is required in the public interest. There are finely balanced arguments tending both in favour of and against prosecution.

Journalists and those who interact with them have no special status under the law and thus the public interest factors have to be considered on a case by case basis in the same way as any other. However, in cases affecting the media, the DPP's Interim Guidelines require prosecutors to consider

> *whether the public interest served by the conduct in question outweighs the overall criminality alleged.*
>
> So far as Ms Hill is concerned, the public interest served by her alleged conduct was that she was working with other journalists on a series of articles which, taken together, were capable of *disclosing the commission of criminal offences, were intended to hold others to account*, including the Metropolitan Police Service and the Crown Prosecution Service, and were capable of *raising and contributing to an important matter of public debate, namely the nature and extent of the influence of the media*. The alleged overall criminality is the breach of the Data Protection Act, but, as already noted, any damage caused by Ms. Hill's alleged disclosure was minimal. In the circumstances, *I have decided that in her case, the public interest outweighs the overall criminality alleged*.
>
> Different considerations apply to the police officer. As a serving police officer, *any claim that there is a public interest in his alleged conduct carries considerably less weight* than that of Ms Hill. However, there are other important factors tending against prosecution, including as already noted, the fact that no payment was sought or received, and that the disclosure did not compromise the investigation. Moreover, *disclosing the identity of those who are arrested is not, of itself, a criminal offence. It is only unlawful in this case because the disclosure also breached the Data Protection Act*.
>
> In the circumstances, *I have decided that a criminal prosecution is not needed* against either Ms. Hill or the police officer.
>
> However, in light of my conclusion that *there is sufficient evidence to provide a realistic prospect of convicting the police officer for an offence under the Data Protection Act*, I have written to the Metropolitan Police Service and to the IPCC *recommending that they consider bringing disciplinary proceedings against him*." Alison Levitt QC

Let us start by considering the arguments for and against prosecuting the journalist (pj) and the police officer (pp).

Arguments for prosecution:

A: *The articles contained confidential information*;

B: *The police officer disclosed the information*;

C: *A prosecution is required in the public interest*.

Arguments against prosecution:

D: *There is no evidence that the police officer was paid any money*;

E: *Information disclosed by the police officer, although confidential, was not highly sensitive*;

F: *It did not expose anyone to a risk of injury or death*;

G: *It did not compromise the investigation*;

H: *It would probably have made it into the public domain by some other means*;

I: *The public interest outweighs the overall criminality alleged*;

J: *Together, the articles would expose the commission of criminal offences*;

K: *Together, the articles would hold others to account*;

L: *The articles contributed to an important matter of public debate, namely the nature and extent of the influence of the media*.

Let us also analyse more closely the arguments relating to whether a prosecution is required in the public interest. The assumption is that both the journalist and the police officer have violated the Data Protection Act.

Arguments for bringing charges under the Data Protection Act:

M: *The information disclosed was personal data*;

Arguments against bringing charges under the Data Protection Act:

N: *Disclosing the identity of those who are arrested is not, of itself, a criminal offence*;

O: *It is only unlawful in this case because the disclosure also breached the Data Protection Act.*

Remark 15 *Notice how argument O was used rhetorically as part of an argument against bringing charges under the Data Protection Act. Argument O is, in fact, simply stating that the disclosure was unlawful because it breached the Data Protection Act. Consider also how the sentence below is used as part of the argumentation: "The alleged overall criminality is the breach of the Data Protection Act, but, as already noted, any damage caused by Ms. Hill's alleged disclosure was minimal". Our model will help quantify such* damage *and will require a definition of* minimal*, as will become clear. Similarly, the following sentences provide clues as to the weights to be assigned to the neural network, in relation to the police officer: "Any claim that there is a public interest in his alleged conduct carries considerably less weight" and "There is a high threshold to make out the criminal offence of misconduct in public office". We shall return to these once we have created the network model.*

The QC's conclusion can be summarized as follows:

(a) *It was decided that a criminal prosecution is not needed*;

(b) *It was decided that in the case of the journalist, the public interest outweighs the overall criminality alleged*;

(c) *There is sufficient evidence to provide a realistic prospect of convicting the police officer for an offence under the Data Protection Act*;

(d) *A recommendation was made to the police to consider bringing disciplinary proceedings against the police officer*.

Our model's conclusion: Our model is concerned with making explicit the following relations (items 1 to 4 below).

On the issue of the police officer's misconduct:

1. Do the weights of the arguments exceed the high threshold for the offence of misconduct?

Arguments A, B and C should support the prosecution of the police officer (pp). Argument C should do so with a low weight (w) since the prosecution would be for misconduct in public office. Arguments D, F, G and H attack pp collectively (argument D with a high weight (W) for obvious reasons, and argument H with a low weight due to its speculative nature). Argument E attacks argument B. These are all the relevant arguments in relation to pp, as shown in Figure 7, where dashed lines indicate attacks. As a result, neuron B fails to activate, and the weight of the arguments that collectively attack neuron pp should overcome the weight of the arguments that support pp. Hence, neuron pp should fail to activate. We discuss neuron/argument pj next.

2. If the police officer should not be prosecuted for misconduct then the journalist should not be prosecuted for aiding his conduct.

Item 2 above can be modelled in Figure 7 using ADFs simply by stating that if argument pp is out (represented by a negative weight from input neuron pp to the hidden layer) then argument ¬pj should be in; hence, the journalist should not be prosecuted (in the neural network, if input neuron pp is not activated then output neuron ¬pj will be activated; see dashed line representing a negative weight from input neuron pp in Figure 7). This separation between arguments pj and ¬pj allows one to ignore how arguments would influence neuron pj during the modelling of neuron ¬pj, and is referred to *explicit negation* in logic programming [19]. Thus, in case neuron pp fails to activate, which is the case here, neuron ¬pj will be activated. This completes the prosecution's analysis on the basis on misconduct.

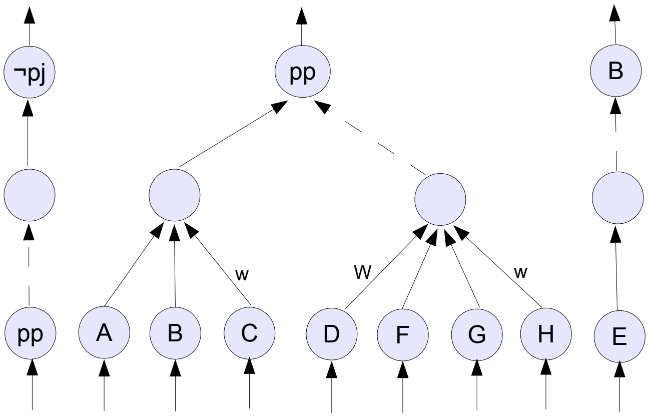

Figure 7: Neural implementation of legal case: prosecution decision

On the issue of the violation of the Data Protection Act:

3. Do the weights of the arguments show that the public interest outweighs the violation of the Data Protection Act?

It is clear that arguments J, K and L support argument I, while argument M attacks I. In addition, argument N attacks M, and O attacks N, as shown in Figure 8. The conclusion is that indeed argument I should prevail.

4. Different weights apply to the journalist and the police officer.

Argument M supports the argument that the journalist should be prosecuted for violation of the Data Protection Act (let us call this pj_{DP}). It also supports the argument for prosecuting the police officer for a violation of the Data Protection Act (call it pj_{DP}). The attacks from argument I on pj_{DP} and pp_{DP} should have different weights: a high (negative) weight W for pj_{DP} and a low (negative) weight (w) for pp_{DP}. Our model's conclusion, therefore, is that pj_{DP} should not prevail (i.e. the journalist should not be prosecuted), but differently from the QC's conclusion, if the value of the weight ω connecting argument M to pp_{DP} should be greater than the absolute value of w then argument pp_{DP} should prevail, i.e. the police officer would be prosecuted for violating the Data Protection Act. The actual values of weights ω and w may be a matter for debate, but perhaps the QC's arguments should have focused more on providing a justification for such values.

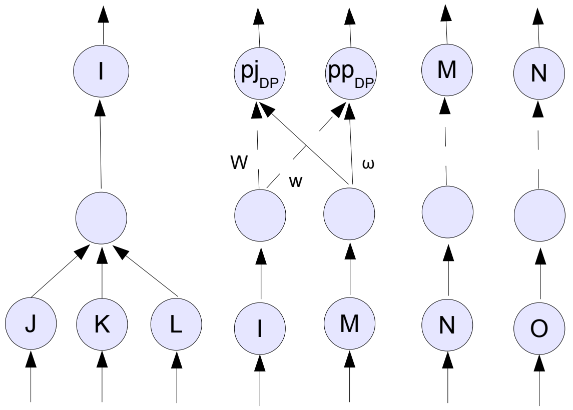

Figure 8: Neural implementation of legal case: decision based on data protection act

The above modelling exercise with the use of a neural cognitive model, may also help in the separation of concerns and systematic questioning of some of the assumptions made. For example, in this case study, some questions that emerge include: should different weights really apply to the journalist and the police officer, assuming they had to work as a team in the public interest? Aside from possible issues of remit, why should prosecution not be recommended straight away for the violation of the data protection act, and disciplinary proceedings should be evoked and recommended instead? We believe that our model should help prompt the user to ask such questions, organise the relationships among the different arguments under consideration, and investigate the impact of different weight assignments to the network model. For example, what if the weights W and w are assigned the same value? What if the weights ω and w are assigned the same absolute value? The user would then be able to run the model and consider the possible outcomes by analysing the different sets of prevailing arguments obtained as stable states of the neural network.

6 Conclusion and Future Work

We have presented a neural cognitive model of argumentation that is capable of capturing a range of argumentation semantics and situations including joint-attacks, argument support, ordered attacks, disjunctive attacks, metalevel attacks and self-defeating attacks. All these different modes of argumentation can be modelled, learned and computed by means of a connectionist representation. In its most general form, arguments are weighted according to their strength, can support or attack other arguments directly, but can also combine conjunctively or disjunctively, sequentially or in parallel, at object- or meta-level, as exemplified throughout the paper. We have shown that all these different modes of argumentation can be represented and computed in a natural way by a connectionist network. This also indicates that the connectionist approach can offer an adequate tool for argument computation.

When dealing with uncertainty and metalevel preferences, in [14], the question of where the weights would come from is raised. In [2], voting by an audience is evoked as a solution that depends on metalevel considerations, as in [26]. With the framework proposed in this paper, the question gains a new dimension in that, as with any neural network, the weights can be learned from examples (i.e. instances from previous cases). As future work, we plan to explore the framework's learning capacity as part of a larger case study.

Uncertainty is intrinsic in human argumentation, yet most logic-based models of argumentation do not deal with uncertainty explicitly. Argumentation can be seen as a method for reducing one's uncertainty with the prevailing arguments being precisely those that are less-uncertain. In line with [21, 23, 35], the neural cognitive model introduced here lends itself well to this idea due to the use of weights and activation intervals as part of a neural network. However, we believe that the neural cognitive approach may also be advantageous from a purely computational perspective due to the networks' ability to adapt through learning and to compute prevailing arguments in parallel. All of the above is important given the objective of developing models of human argumentation. Future work includes, in addition to the evaluation of the framework's learning capacities, further experimentation on legal reasoning, a comparison of the framework's knowledge representation and learning capacity, e.g. in contrast with [23, 32], and the evaluation of the framework's parallel computation gains. A graphical interface is being developed to facilitate the interactive drawing and running of the networks as shown in the figures above, with all the features introduced here. We believe that such an interface can be useful as a tool for modelling, running and the elaboration of argumentation frameworks in a range of application areas.

References

[1] H. Barringer, D.M. Gabbay, and J. Woods. Temporal dynamics of support and attack networks: From argumentation to zoology. In D. Hutter and W. Stephan, editors, *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, volume 2605 of *Lecture Notes in Computer Science*, pages 59–98. Springer, 2005.

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

[3] A. Bondarenko, P. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.

[4] G. Brewka. Dynamic argument systems: A formal model of argumentation processes based on situation calculus. *Journal of Logic and Computation*, 11(2):257–282, 2001.

[5] G. Brewka and S. Woltran. Abstract dialectical frameworks. In *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning KR 2010*. AAAI Press, 2010.

[6] M. Caminada. On the issue of reinstatement in argumentation. In *Prof. 10th European Conference on Logics in Artificial Intelligence JELIA'06*, pages 111–123. Springer, LNAI 4160, 2006.

[7] M. Caminada and L. Agmoud. On the evaluation of argumentation for- malisms. *Artificial Intelligence*, 171(5-6):286–310, 2007.

[8] C.I. Chesñevar, A.G. Maguitman, and R. P. Loui. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383, December 2000.

[9] A.S. d'Avila Garcez, K. Broda, and D.M. Gabbay. *Neural-Symbolic Learning Systems: Foundations and Applications*. Perspectives in Neural Computing. Springer, 2002.

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

[11] A.S. d'Avila Garcez, L.C Lamb, and D.M. Gabbay. *Neural-Symbolic Cognitive Reasoning*. Cognitive Technologies. Springer, 2009.

[12] P. M. Dung and P. M. Thang. Modular argumentation for modelling legal doctrines in common law of contract. *Artif. Intell. Law*, 17(3):167–182, 2009.

[13] P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[14] P. E. Dunne, A. Hunter, P. McBurney, S. Parsons, and M. Wooldridge. Inconsistency tolerance in weighted argument systems. In *Proc. 8th Int. Conf. on Autononous Agents and Multiagent Systems, AAMAS 2009*, pages 851–858, 2009.

[15] D. M. Gabbay. Fibring argumentation frames. *Studia Logica*, 93:231–295, 2009.

[16] D. M. Gabbay and A. S. d'Avila Garcez. Logical modes of attack in argumentation networks. *Studia Logica*, 93(2-3):199–230, 2009.

[17] A.J. García and G.R. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.

[18] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of the Fifth Logic Programming Symposium*, pages 1070–1080. MIT Press, 1988.

[19] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[20] S. A. Gomez and C. I. Chesnevar. Integrating Defeasible Argumentation with fuzzy art neural networks for pattern classification. *Journal of Computer Science & Technology*, 4(1):45–57, 2004.

[21] R. Haenni, J. Kohlas, and N. Lehmann. Probabilistic argumentation systems. In D. Gabbay and Ph Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 5, page 221288. Kluwer, 2000.

[22] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[23] A. Hunter. A probabilistic approach to modelling uncertain logical arguments. *International Journal of Approximate Reasoning*, 54(1):47–81, 2013.

[24] R.A. Kowalski and F. Toni. Abstract argumentation. *Artificial Intelligence and Law*, 4(3-4):275–296, 1996.

[25] W. Makiguchi and H. Sawamura. A hybrid argumentation of symbolic and neural net argumentation (part 1). In *Argumentation in Multi-Agent Systems ArgMAS 2007*, pages 197–215, Honolulu, USA, 2007. Springer, LNCS.

[26] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, 173(9-10):901–934, 2009.

[27] M. Mozina, J. Zabkar, and I. Bratko. Argument based machine learning. *Artificial Intelligence*, 171(10-15):922–937, 2007.

[28] S. H. Muggleton and D. Lin. Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. In *IJCAI*, 2013.

[29] F. S. Nawwab, T. Bench-Capon, and P. E. Dunne. Exploring the role of emotions in rational decision making. In *Third International Conference on Computational Models of Argument COMMA*, Desenzano del Garda, Italy, 2010.

[30] J.L. Pollock. Self-defeating arguments. *Minds and Machines*, 1(4):367–392, 1991.

[31] J.L. Pollock. How to reason defeasibly. *Artificial Intelligence*, 57:1–42, 1992.

[32] D. Poole. The independent choice logic and beyond. In Luc De Raedt, Paolo Frasconi, Kristian Kersting, and Stephen Muggleton, editors, *Probabilistic Inductive Logic Programming: Theory and Application*. Springer, LNAI 4911, 2008.

[33] H. Prakken and G.A.W. Vreeswijk. Logical systems for defeasible argumentation. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*. Kluwer, 2nd edition, 2000.

[34] I. Rahwan, M.I. Madakkatel, J-F Bonnefon, R.N. Awan, and S. Abdallah. Behavioral experiments for assessing the abstract argumentation semantics of reinstatement. *Cognitive Science*, 34(8):1483–1502, 2010.

[35] R. Riveret, A. Rotolo, and G. Sartor. Probabilistic rule-based argumentation for norm-governed learning agents. *Artif. Intell. Law*, 20(4):383–420, 2012.

[36] L Su, P. Barnard, and H. Bowman. On the fringe of awareness: The glance-look model of attention-emotion interactions. In K. Diamantaras, W. Duch, and L.S. Iliadis, editors, *International Conference on Artificial Neural Networks*, volume 6354 of *Lecture Notes in Computer Science*, page 504509. Springer-Verlag, July 2010.

[37] Y. Tang, T. J. Norman, and S. Parsons. A model for integrating dialogue and the execution of joint plans. In *AAMAS*, pages 883–890, 2009.

[38] B. Verheij. Accrual of arguments in defeasible argumentation. In *Proceedings of Second Dutch/German Workshop on Nonmonotonic Reasoning*, pages 217–224. Utrecht, 1995.

[39] B. Verheij. *Rules, Reasons, Arguments: formal studies of argumentation and defeat*. PhD thesis, Maastricht University, The Netherlands, 1996.