# On Gabbay's Fibring Methodology for Bayesian and Neural Networks

A. S. d'Avila Garcez
City University London
Northampton Square
London EC1V 0HB UK

January 22, 2004

**Abstract**

This paper discusses how Gabbay's fibring methodology - originally aimed at combining logics - can be applied also to combine sub-symbolic structures such as Bayesian networks or neural networks. I start by commenting on the paper *Recursive Causality in Bayesian Networks and Self-Fibring Networks* by J. Williamson and D. Gabbay, which shows how Bayesian networks can be fibred. I then discuss how neural networks can be fibred in the same spirit, compare the Bayesian and neural approaches, and illustrate the fibring of Bayesian and neural networks by applying it to value-based argumentation in legal reasoning. I conclude by offering a first account of how symbolic and sub-symbolic systems such as logics and networks can be fibred together.

## 1 Introduction

Fibring is a methodology for combining logics by breaking, manipulating and rearranging them into simple components. Briefly, the languages and inference rules of each logic may appear in the combined system, while the semantics of the fibring needs to be carefully compiled as a combination of the classes of models of the logics. For example, in the case of the fibring of a propositional temporal logic with a propositional linear space logic, each model of the fibring will contain a time line and a space line [14].

In addition to its appeal from a purely logical point of view, fibring has a number of applications in mathematics and computer science. It has important implications to knowledge representation and artificial intelligence -

where, for example, the combination of temporal and deontic logics may be necessary - and to formal methods and software engineering, where one may need to work with declarative specifications and procedural ones. Putting together artificial intelligence and software engineering, in robotics, for instance, a robot's visual system may require one logical representation, while its planning system may require another [8]. Moving even further, some of the robot's systems may not be a logical (symbolic) system, but a connectionist (sub-symbolic) one, as in the case of neural networks systems very successfully used for visual information processing, or Bayesian networks also successfully used to model uncertainty. The concept of fibring, therefore, needs to be extended to cater for hybrid systems. It may, however, continue to have logic as its underlying mechanism with the help of neural-symbolic learning systems [3], as we shall exemplify in the sequel.

In what follows, we discuss how Bayesian networks may be self-fibred. We then discuss how Neural Networks are fibred, and exemplify the fibring of Bayesian and neural networks using an argumentation framework example in law. We conclude by discussing how neural-symbolic learning systems may serve as the underlying framework for the fibring of logics and networks.

## 2 Fibring Bayesian Networks

The idea of fibring Bayesian networks emerged from the observation that causal relations may themselves take part in causal relations. This has been simply and effectively exemplified in the very first section of Williamson and Gabbay's *Recursive Causality in Bayesian Networks and Self-Fibring Networks* [15]. The example states that the fact that smoking causes cancer, for instance, causes the government to restrict tobacco advertising. Such a recursive definition of causal relations was then used to define what Williamson and Gabbay call a *recursive Bayesian network*, a Bayesian network in which certain nodes may be Bayesian networks in their own right. This is defined with the help of the concept of *network variables,* which are variables that may take Bayesian networks as values. Thus, a network variable $SC$ may be used to represent the fact that *smoking causes cancer*, and then $SC$ causes $A$, where $A$ stands for *restricting advertising*. This may be written as $SC \to A$ where $SC = S \to C$, or simply as $(S \to C) \to A$.

A recursive network can be interpreted as a non-recursive network if one treats network variables $SC$ as simple variables $SC'$. In addition, a recursive network can be *flattened* into its non-recursive counterpart, given a consistent assignment of values $v$ to the domain $V$ of the recursive network. Consistency of assignments is, therefore, a requirement for the use of standard Bayesian networks algorithms by recursive Bayesian networks. This is why Williamson and Gabbay devote an entire section of their paper to the issue of consistency. They identify three forms of consistency between networks: causal consistency (w.r.t. causal relations), Markov consistency (w.r.t. implied probabilistic independencies), and probabilistic consistency (w.r.t. probability functions).

Note that variables are allowed to occur more than once in a recursive network, e.g., $(A \rightarrow B) \rightarrow A$. If we are not careful, its flattened counterpart network may have cycles, what would prevent the use of standard Bayesian inference algorithms. Williamson and Gabbay concentrate, as a result, on consistent acyclic assignments. It is worth noting also that there exists no straightforward flattening of a recursive Bayesian network into a Bayesian network when no values $v$ are given, i.e. flatenings are relative to an assignment $v$ on the domain $V$.

Once the flattening of recursive Bayesian networks is fully defined (allowing for the use of a standard Bayesian network to determine the probability distributions), Williamson and Gabbay investigate the more general concept of allowing graphs inside graphs, referred to as *self-fibring of networks*. They present a generalisation of recursive Bayesian networks in the form of an important definition of fibred networks in terms of a fibring function, which is general enough to comprise the fibring of neural networks as well [10], as discussed in the following section. This definition is obtained by looking at the arrows in directed graphs as different types of implications in logic such as substructural or intuitionistic implication, Dempster-Shafer or causal Bayesian implication. In this general setting, the fibring function is responsible for providing the different interpretations; in the case of Bayesian networks, the fibring function depends on a table of conditional probabilities.

The general case problem of self-fibring of networks can be defined as follows: Let $\mathbf{B}$(X) be a network containing a node X. Let $\mathbf{A}$ be a network. Find $\mathbf{C}$=$\mathbf{B}$(X/$\mathbf{A}$), a new network which is the result of substituting $\mathbf{A}$ for X in $\mathbf{B}$. The power of fibring, however, lies in the fact that this substitution is not only performed on the syntactical level, but takes the form of a *semantic insertion*, which makes use of the fibring function as follows. The substitution of $\mathbf{A}$ takes into account the meaning of $\mathbf{B}$. For example, if $\mathbf{B}$ is a Bayesian network in which X may take two values, say {0,1}, then if X is 0 we substitute $\mathbf{A}_0$, and if X is 1 we substitute $\mathbf{A}_1$. We write $\mathbf{C}= f_x(\mathbf{A},\mathbf{B})$, where $f$ is our fibring function.

It is the concept of semantic insertion that makes fibring a powerful mechanism. For example, consider the network of Figure 1. The output of fibring function $f$ is allowed to modify the fibring function $g$ itself. In the case of Bayesian networks, where $f$ and $g$ are probability matrices, one could decide to modify $g$ by, say, multiplying it by $f$. The resulting network would be: $a \rightarrow_f (b \rightarrow_{f.g} c)$.

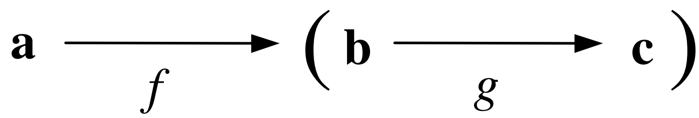


Figure 1: A recursive network $a \rightarrow X$ where $X = b \rightarrow c$.

In the case of neural networks, where $f$ and $g$ are weight vectors, the idea of multiplying $f$ by $g$ is quite natural since this is already what non-recursive neural networks do. We shall see in the sequel that applying the concept of semantic insertion to neural networks results in recursive networks being strictly more expressive than non-recursive ones, in the sense that they cannot be flattened into equivalent non-recursive networks, even when the straightforward *multiplication rule* is used for fibring.

## 3 Fibring Neural Networks

The goal of Neural-Symbolic integration [2, 3] is to benefit from the combination of the symbolic and the connectionist paradigms of Artificial Intelligence. To this end, we know that a fundamental aspect of symbolic computation lies in the ability to perform recursion. Recursion in neural networks, then, is the idea of allowing networks to be composed not only of interconnected neurons but also of other networks (called embedded networks). But the idea is not simply to organise networks as a number of sub-networks. Borrowing the concept of semantic insertion from fibring, the function computed by an embedded network may depend on the function computed by the embedding network.

A neural network consists of interconnected neurons (or processing units) that compute a simple function according to the weights (real numbers) associated to the connections. Learning in this setting is the incremental adaptation of the weights. The interesting characteristics of neural networks do not arise from the simple functionality of each neuron, but from their collective behaviour. In the case of recursive neural networks, introduced in [4], we see fibring as learning, where the change of the weights of a neural network implies the change of the function computed by it. So, we simply use the weights of the embedding network to change the weights of the embedded network. When running the network $a \rightarrow_f (b \rightarrow_g c)$, we obtain the network $a \rightarrow_f (b \rightarrow_{f.g} c)$, where $a$, $b$ and $c$ are neurons, and $f$ and $g$ are weights. In other words, one network is being used to train the other, as the following example illustrates.

**Example 1** *Consider the networks* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *of Figure 2, each with inputs* $i_1$ *and* $i_2$*, and output* $o_1$*. Let network* $\boldsymbol{B}$ *be embedded into the output neuron of network* $\boldsymbol{A}$*, as shown in the figure. This indicates that the state of* $\boldsymbol{A}$*'s output neuron* ($\mathbf{s}$) *will influence the weights of* $\boldsymbol{B}$ ($W_1^{\mathbf{B}}, W_2^{\mathbf{B}}, W_3^{\mathbf{B}}$)*, according to a fibring function* ($\varphi$)*. Using the simple multiplication rule for* $\varphi$*, let us take* $\mathbf{W}_{new}^{\mathbf{B}} = \mathbf{s} \cdot \mathbf{W}_{old}^{\mathbf{B}}$*. Notice how network* $\boldsymbol{B}$ *is being trained (when* $\varphi$ *changes its weights) at the same time that network* $\boldsymbol{A}$ *is running.*

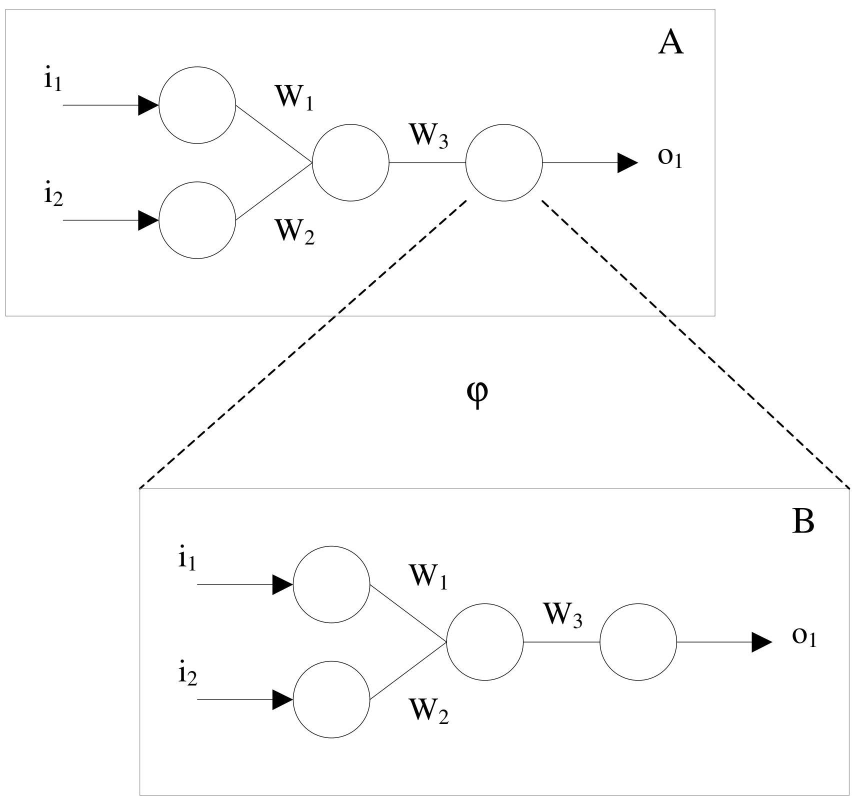


Figure 2: Fibring two simple networks

Fibred neural networks can be trained from examples in the same way that standard feedforward networks are (for example, with the use of Backpropagation [13]). Networks **A** and **B** of Figure 2, for example, could have been trained separately before being fibred. Network **A** could have been trained, e.g., with a robot's visual system, while network **B** would have been trained with its planning system. For now, we assume that, once defined, the fibring function itself remains unchanged. Future extensions of fibring neural networks could, however, consider the task of learning fibring functions as well.

In addition to using different fibring functions, networks can be fibred in a number of different ways as far as their architectures are concerned. The networks of Figure 2, for example, could have been fibred by embedding Network **B** into an input neuron of Network **A** (say, the one with input $i_1$), thus changing the value of $\mathbf{s}$ (to the state of this input neuron) used to calculate the new set of weights of **B**, and thus also changing the function computed by **B**.

The recursive network of Figure 2 is capable of computing, with the use of the multiplication rule for fibring, an output $n^2$ for $n \in \Re$ given as input. In other words, recursive networks are capable of performing the exact computation of the square of their input for any input in $\Re$. This indicates that recursive networks approximate functions in an unbounded domain [11, 12], as opposed to non-recursive feedforward networks. In fact, in addition to being universal approximators, recursive networks can approximate any polynomial function, and thus are more expressive than non-recursive feedforward neural networks. The proofs are given in [4].

Finally, note that many networks can be embedded into a single network, and that networks can be nested so that network B is embedded into network A, network C is embedded into network B, and so on. In addition to the existing multitude of fibring functions, there is a multitude of network architectures for fibring that might be interesting investigating. The choice of fibring function and architecture is domain dependent, and an empirical evaluation of recursive networks in comparison with standard neural networks would also be required. Another interesting open question is that of which logics could be represented by recursive networks. The extra expressiveness of such networks contributes to the development of the research on Neural-Symbolic integration, where neural networks need to be used to perform complex symbolic computation.

**Remark 1** *In summary, both recursive Bayesian networks and recursive neural networks share the more general principles and definitions of Gabbay's fibring methodology. However, each technique has very specific char-*

*acteristics and algorithms to deal with uncertainty (in the case of Bayesian networks) and learning and generalisation capabilities (in the case of neural networks). Both recursive Bayesian and neural networks are strictly more expressive than their standard versions, not only because of the idea of allowing networks inside networks, but also because of the general concept of semantic insertion from fibring. As we have seen, recursive Bayesian networks when flattened may produce networks with cycles instead of standard Bayesian networks, while recursive neural networks are capable of computing unbounded functions exactly, as opposed to standard networks. Standard Bayesian networks algorithms can be applied directly to the component networks of a recursive Bayesian network, and under certain conditions to the recursive network itself. Similarly, standard neural networks learning algorithms may be applied directly to each of the component networks of a recursive neural network. Finally, both types of networks may have counterpart symbolic representations with the help of Labelled Deductive Systems [7] and Neural-Symbolic Learning Systems [3], and these logical representations may be helpful in guiding the development of the research on fibring networks.*

## 4 Fibring Applied to Argumentation

An interesting application of fibring is in the area of legal reasoning and argumentation [5, 9]. In [9], for example, Gabbay and Woods argue for the combined use of Labelled Deductive Systems and Bayesian networks to support legal evidence reasoning under uncertainty. In addition, they argue that neural networks, as learning systems, could play a role in this process by being used to update/revise degrees of belief and the rules of the system whenever a new evidence is presented. The three different representations all expand a value-based argumentation framework outlined in [1], in which argumentation networks are used to model arguments and counter-arguments. A typical example in the area is the following *moral debate* example.

> Hal, a diabetic, loses his insulin in an accident through no fault of his own. Before collapsing into a coma, he rushes to the house of Carla, another diabetic. She is not at home, but Hal breaks into her house and uses some of her insulin.. Was Hal justified? Does Carla have a right to compensation?

The following are some of the arguments involved as presented in [1].

**A**: Hal is justified, he was trying to save his life;

**B**: It is wrong to infringe the property rights of another;
**C**: Hal compensates Carla;
**D**: Hal is endangering Carla's life; and
**E**: Carla has abundant insulin..

In [1], arguments and counter-arguments are arranged in an argumentation network, as in Figure 3, where an arrow from argument X to argument Y indicates that X attacks Y. For example, the fact that it is wrong to infringe Carla's right of property (**B**) attacks Hal's justification (**A**).

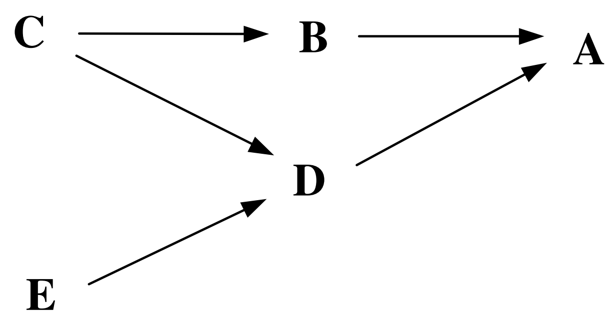


Figure 3: Part of the argumentation network for the moral debate example

Some aspects of the argumentation network of Figure 3 are probabilistic. For example, the question of whether Carla has abundant insulin (**E**) depends on the time and is a matter of probability. The question of whether Hal will be able to compensate Carla with replacement insulin in time (**C**) is also a matter of probability. If Carla has abundant insulin, the chances that Hal will be able to compensate her are higher. The probability matrices of this Bayesian network (**E**→**C**) influence whether Hal is endangering Carla's life by stealing some of her insulin (**D**). In the same argumentation network, some other aspects may change as the debate progresses and actions are taken; the strength of one argument in attacking another may change in time. This is a learning process that can be implemented using a neural network in which the weights record the strength of the arguments. The neural network for the set of arguments {**A**, **B**, **D**} is depicted in Figure 4.

The neural network of Figure 4 is an auto-associative single hidden layer network with input (A,B,D), output (A,B,D) and hidden layer (h1,h2,h3). Solid arrows represent positive weights and dotted arrows represent negative weights. Arguments are supported by positive weights and attacked by negative ones. Argument **A** (input neuron **A**), for example, supports itself (output neuron **A**) with the use of hidden neuron h1. Similarly, **B** supports

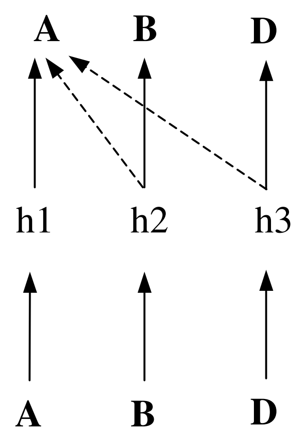


Figure 4: A neural network for arguments {A,B,D}

itself (via h2), and so does **C** (via h3). From the argumentation network, **B** attacks **A**, and **D** attacks **A**. The attacks are implemented in the neural network by the negative weights (see dotted lines in Figure 4) with the use of h2 and h3.[1] The network of Figure 4 is a standard feedforward neural network that can be trained, e.g., with the use of the standard Backpropagation learning algorithm [13]. Training would change the initial weights of the network (the initial belief on the strength of arguments and counter-arguments, which could be random), according to examples of input/output patterns, i.e. examples of the relationship between arguments **A**, **B** and **D**. Roughly speaking, if then the absolute value of the weight from neuron h1 to output neuron A is greater than the sum of the absolute values of the weights from neurons h2 and h3 to **A**, one can say that argument **A** prevails (in which case output neuron **A** should be *activated* in the neural network).

The key to running the network properly is to connect output neurons to their corresponding input neurons using weights fixed at 1, so that the activation of output neuron **A**, for example, is fed into the activation of input neuron **A** the next time round. This implements chains such as A attacks B, B attacks C, C attacks D, and so on, by propagating activations around the network. The following example illustrates the dynamics of argumentation neural networks (see [5] for details).

**Example 2** *Take the case in which an argument A attacks an argument B, and B attacks an argument C, which in turn attacks A in a cycle. In order*

[1] The hidden neurons are used in the network to provide a greater flexibility as to what can be learned as combinations of the input neurons.

*to implement this in a neural network, we need three hidden neurons (h1, h2, h3), positive weights to explicitly represent the fact that A supports itself (via h1), B supports itself (via h2), and so does C (via h3). In addition, we need negative weights from h1 to B, from h2 to C and from h3 to A to implement attacks. If all the weights are the same in absolute terms, no argument wins, as one would expect, and the network stabilises with none of output neurons {A,B,C} activated. If, however, the value of argument A (i.e. the weight from h1 to A) is stronger than the value of argument C (the weight from h3 to C, which is expected to be the same in absolute terms as the weight from h3 to A), C cannot attack and defeat A. As a result, A is activated. Since A and B have the same value, B is not activated, since the weights from h1 and h2 to B will both have the same absolute value. Finally, if B is not activated then C will be activated, and a stable state {A,C} will be reached in the network. In Bench-Capon's model [1], this is precisely the case in which colour blue is assigned to A and B, and colour red is assigned to C with blue being stronger than red. Note that the order in which we reason does not affect the final result (the stable state reached). For example, if we started from B successfully attacking C, C would not be able to attack A, but then A would successfully attack B, which would this time round not be able to successfully attack C, which in turn would be activated in the final stable state {A,C}. This indicates that a neural implementation of this reasoning process may, in fact, be advantageous from a purely computational point of view due to neural networks' parallel nature.*

Now that we have two more concrete models of the arguments involved in the moral debate example - a probabilistic model and a learning/action model - we can reason about the problem at hand in a more realistic way. We just need to put the two models together with the use of the fibring methodology for networks. The (more abstract) argumentation network of Figure 3 can be used to tell us how the networks (Bayesian and neural) are to be fibred. From Figure 3, one can see that both arguments $C$ and $E$ attack argument $D$ directly. As a result, we would like the probabilities in our Bayesian network $E \rightarrow C$ to influence the activation of neuron $D$ in the neural network. Thus, network $E \rightarrow C$ needs to be embedded into node $D$. Again from the argumentation network (Figure 3), one can see that argument $C$ also attacks argument $B$ directly. As a result, we would like the probabilities associated with $C$ to influence the activation of neuron $B$. As before, this can be done by embedding Bayesian network $C$ into neuron $B$. This produces the recursive network of Figure 5.

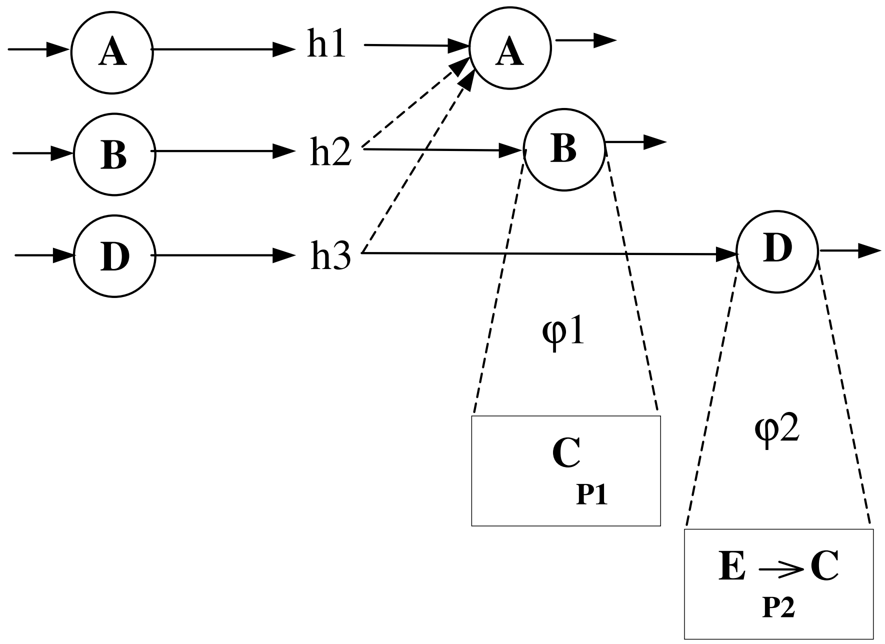


Figure 5: Fibring of Bayesian and neural networks applied to value-based argumentation

Let us consider again the embedding of **E**→**C** into **D**. We have seen that the embedding is guided by the arrow in the original argumentation network. The arrow in an argumentation network indicates an *attack*. As a result, the higher the probability P(**C**/**E**) in **P2** is, the lower the activation value of neuron **D** should be. Similarly, the higher the probability P(**C**) in **P1**, the lower the value of **B** should be. Thus, we take $\varphi 1 : \mathbf{s}^{\mathbf{B}}_{new} = \mathbf{s}^{\mathbf{B}}_{old} -$ P(**C**) and $\varphi 2 : \mathbf{s}^{\mathbf{D}}_{new} = \mathbf{s}^{\mathbf{D}}_{old} -$ P(**C**/**E**), where $\mathrm{P}(\mathbf{X}) \in [0,1]$ and $\mathbf{s} \in (0,1)$. This definition of the fibring functions completes the fibring.

In this case study, we have seen that the fibring of networks can be used to combine different systems, yet maintaining their individual characteristics and using their algorithms for learning and reasoning. In the combined system, the new state of output neuron **D** ($\mathbf{s}^{\mathbf{D}}_{new}$) will then be fed into input neuron **D** and affect the new state of **A** (through hidden neuron h3 and the negative weight from h3 to **A**), such that the higher the value of **D** the lower the value of **A**. The same will happen through **B** according to the dynamics of the embedding and embedded networks, and this will allow the reasoning as to whether Hal is justified or not to proceed.

# 5 Conclusion

We have seen that the general methodology of fibring logical systems can be applied also to sub-symbolic systems such as Bayesian and neural networks. As in symbolic systems, where for example the fibring of two logics without embedded implication may result in a logic with embedded implication, the fibring of networks may result in embedded networks such as $(A \to B) \to C$, which are strictly more expressive than standard networks (i.e.: do not have a flattened counterpart network). This indicates, now from the point of view of neural-symbolic integration, that fibring may be used to produce simple neural network architectures (an important requirement for effective neural networks learning) that represent powerful logics such as modal, temporal, first order and higher order logics.

As an example, consider the network structure of Figure 6. The small networks inside each agent may contain an intricate architecture used to represent the knowledge of the agent at a time point. These networks may relate to each other in a metalevel network where, in the horizontal axis, the knowledge of the different agents at a time point is presented, and in the vertical axis, the evolution of the agents' knowledge through time is presented. This yields a distributed, massively parallel, multi-agent system, created out of recursive networks, in which space and time logics are incorporated (see [6] for details). This illustrates how the general fibring methodology can be applied to the integration of symbolic and sub-symbolic systems.

Neural-Symbolic Learning Systems [3], therefore, by providing translation algorithms and proofs of the correctness of such translations between different neural networks and different logics, may serve as the underlying framework for the progress of the study of fibring between networks and logics, and of the self-fibring of networks. In this setting, an appropriate fibring of two networks A and B, for example, would be one in which the logic extracted from the fibred network is the same as the logic obtained from fibring the logics extracted from networks A and B, respectively. There are many avenues of research on the self-fibring of networks and, more generally, on the integration of logics and networks. In this paper, I hope to have convinced the reader that a way forward to develop the research on Artificial Intelligence is to study hybrid systems in conjunction with Gabbay's fibring methodology.

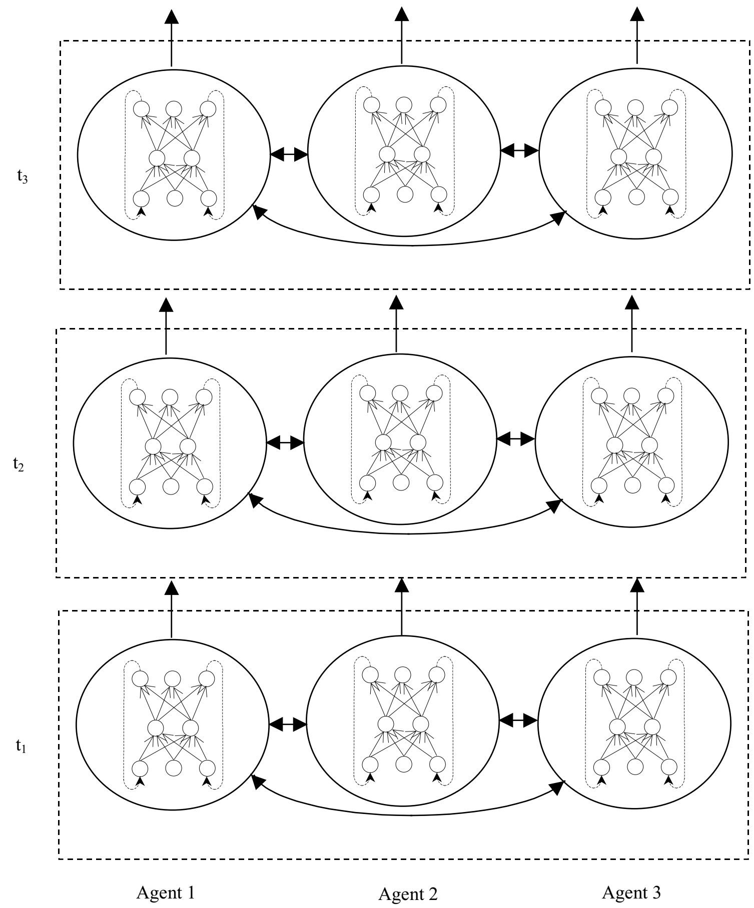


Figure 6: Fibring of knowledge and time in neural networks

# References

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

[2] I. Cloete and J. M. Zurada, editors. *Knowledge-Based Neurocomputing*. The MIT Press, 2000.

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

[4] A. S. d'Avila Garcez and D. Gabbay. Fibring neural networks. Technical report, Department of Computing, City University London, UK, 2003.

[5] A. S. d'Avila Garcez, D. M. Gabbay, and L. C. Lamb. Value-based argumentation as neural networks. *forthcoming*, 2004.

[6] A. S. d'Avila Garcez and L. C. Lamb. Reasoning about time and knowledge in neural-symbolic learning systems. In L. S. S. Thrun and B. Schoelkopf, editors, *Advances in Neural Information Processing Systems 16*, Proceedings of the NIPS 2003 Conference, Vancouver, Canada, To appear 2004. MIT Press.

[7] D. M. Gabbay. *Labelled Deductive Systems*, volume 1. Clarendom Press, Oxford, 1996.

[8] D. M. Gabbay. *Fibring Logics.* Oxford Univesity Press, 1999.

[9] D. M. Gabbay and J. Woods. The law of evidence and labelled deduction: A position paper. *Phi News*, 4, October 2003.

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

[11] J. Henderson. Estimating probabilities of unbounded categorization problems. In *Proceedings of European Symposium on Artificial Neural Networks*, pages 383–388, Bruges, Belgium, April 2002.

[12] J. W. Hines. A logarithmic neural network architecture for unbounded non-linear function approximation. In *Proceedings of IEEE International Conference on Neural Networks*, Washington, USA, June 1996.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. MIT Press, 1986.

[14] A. Sernadas, C. Sernadas, and C. Caleiro. Fibring of logics as a categorical construction. *Journal of Logic and Computation*, 9(2):149–179, 1999.

[15] J. Williamson and D. Gabbay. *Recursive Causality in Bayesian Networks and Self-Fibring Networks.* This volume. 2004.