Dimensions of Neural-symbolic Integration — A Structured Survey

Sebastian Bader[1] and Pascal Hitzler[2]

1 Introduction

Research on integrated neural-symbolic systems has made significant progress in the recent past. In particular the understanding of ways to deal with symbolic knowledge within connectionist systems (also called artificial neural networks) has reached a critical mass which enables the community to strive for applicable implementations and use cases. Recent work has covered a great variety of logics used in artificial intelligence and provides a multitude of techniques for dealing with them within the context of artificial neural networks.

Already in the pioneering days of computational models of neural cognition, the question was raised how symbolic knowledge can be represented and dealt with within neural networks. The landmark paper [McCulloch and Pitts, 1943] provides fundamental insights how propositional logic can be processed using simple artificial neural networks. Within the following decades, however, the topic did not receive much attention as research in artificial intelligence initially focused on purely symbolic approaches. The power of machine learning using artificial neural networking was not recognized until the 80s, when in particular the backpropagation algorithm [Rumelhart *et al.*, 1986] made connectionist learning feasible and applicable in practice.

These advances indicated a breakthrough in machine learning which quickly led to industrial-strength applications in areas such as image analysis, speech and pattern recognition, investment analysis, engine monitoring, fault diagnosis, etc. During a training process from raw data, artificial neural networks acquire expert knowledge about the problem domain, and the ability to generalize this knowledge to similar but previously unencountered situations in a way which often surpasses the abilities of human experts. The knowledge obtained during the training process, however, is hidden within

[1]Supported by the German Research Foundation (DFG) under GK334.
[2]Supported by the German Federal Ministry of Education and Research (BMBF) under the SwartWeb project.

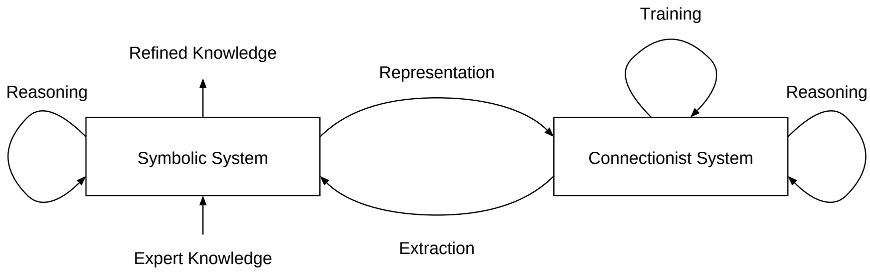


Figure 1. Neural-symbolic learning cycle

the acquired network architecture and connection weights, and not directly accessible for analysis, reuse, or improvement, thus limiting the range of applicability of the neural networks technology. For these purposes, the knowledge would be required to be available in structured symbolic form, most preferably expressed using some logical framework.

Likewise, in situations where partial knowledge about an application domain is available before the training, it would be desirable to have the means to guide connectionist learning algorithms using this knowledge. This is the case in particular for learning tasks which traditionally fall into the realm of symbolic artificial intelligence, and which are characterized by complex and often recursive interdependencies between symbolically represented pieces of knowledge.

The arguments just given indicate that an integration of connectionist and symbolic approaches in artificial intelligence provides the means to address machine learning bottlenecks encountered when the paradigms are used in isolation. Research relating the paradigms came into focus when the limitations of purely connectionist approaches became apparent. The corresponding research turned out to be very challenging and produced a multitude of very diverse approaches to the problem. Integrated systems in the sense of this survey are those where symbolic processing functionalities emerge from neural structures and processes.

Most of the work in integrated neural-symbolic systems addresses the neural-symbolic learning cycle depicted in Figure 1. A front-end (symbolic system) is used to feed symbolic (partial) expert knowledge to a neural or connectionist system which can be trained on raw data, possibly taking the internally represented symbolic knowledge into account. Knowledge acquired through the learning process can then be extracted back to the symbolic system (which now also acts as a back-end), and made available for further processing in symbolic form. Studies often address only parts of the neural-symbolic learning cycle (like the representation or extraction of knowledge), but can be considered to be part of the overall investigations concerning the cycle.

We assume that the reader has a basic familiarity with artificial neural networks and symbolic artificial intelligence, as conveyed by any introductory courses or textbooks on the topic, e.g. in [Russell and Norvig, 2003]. However, we will refrain from going into technical detail at any point, but rather provide ample references which can be followed up at ease. The selection of research results which we will discuss in the process is naturally subjective and driven by our own specific research interests. Nevertheless, we hope that this survey also provides a helpful and comprehensive albeit unusual literature overview to neural-symbolic integration.

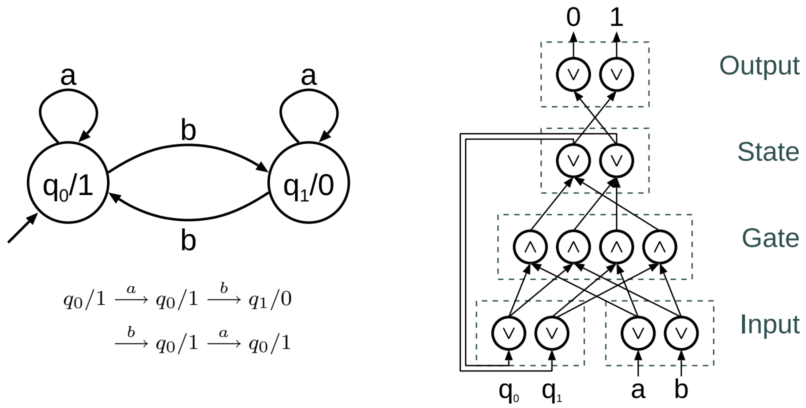
This chapter is structured as follows. In Section 2, we introduce some of those integrated neural-symbolic systems, which we consider to be foundational for the majority of the work undertaken within the last decade. In Section 3, we will explain our proposal for a classification scheme. In Section 4, we will survey recent literature by means of our classification. Finally, in Section 5, we will give an outlook on possible further developments.

2 Neural-Symbolic Systems

As a reference for later sections, we will review some well-known systems here. We will start with the landmark results by McCulloch and Pitts, which relate finite automata and neural networks [McCulloch and Pitts, 1943]. Then we will discuss a method for representing structured terms in a connectionist systems, namely the recursive autoassociative memories (RAAM) [Pollack, 1990]. The SHRUTI System, proposed in [Shastri and Ajjanagadde, 1993], is discussed next. Finally, *Connectionist Model Generation using the Core Method* is introduced as proposed in [Hölldobler and Kalinke, 1994]. These approaches lay the foundations for most of the more recent work on neural-symbolic integration which we will discuss in this chapter.

2.1 Neural Networks and Finite Automata

The advent of automata theory and of artificial neural networks, marked also the advent of neural-symbolic integration. In their seminal paper [McCulloch and Pitts, 1943] Warren Sturgis McCulloch and Walter Pitts showed that there is a strong relation between symbolic systems and artificial neural networks. In particular, they showed that for each finite state machine there is a network constructed from binary threshold units – and vice versa – such that the input-output behaviour of both systems coincide. This is due to the fact that simple logical connectives such as conjunction, disjunction and negation can easily be encoded using binary threshold units,



$$q_0/1 \xrightarrow{a} q_0/1 \xrightarrow{b} q_1/0$$
$$\xrightarrow{b} q_0/1 \xrightarrow{a} q_0/1$$

Figure 2. A simple Moore-machine, the processing of the input word *abba* (left) and a corresponding McCulloch-Pitts-network (right).

with weights and thresholds set appropriately. To illustrate the ideas, we will discuss a simple example in the sequel.

EXAMPLE 1. Figure 2 on the left shows a simple Moore-machine, which is a finite state machine with outputs attached to the states [Hopcroft and Ullman, 1989]. The corresponding network is shown on the right. The network consists of four layers. For each output-symbol $(0, 1)$ there is a unit in the output-layer, and for each input-symbol (a, b) a unit in the right part of the input-layer. Furthermore, for each state (q_0, q_1) of the automaton, there is a unit in the state-layer and in the left part of the input layer. In our example, there are two ways to reach the state q_1, namely by being in state q_1 and receiving an 'a', or by being in state q_0 and receiving a 'b'. This is implemented by using a disjunctive neuron in the state-layer receiving inputs from two conjunctive units in the gate layer, which are connected to the corresponding conditions, as e.g. being in state q_0 and reading a 'b'.

A network of n binary threshold units can be in 2^n different states only, and the change of state depends on the current input to the network only. These states and transitions can easily be encoded as a finite automaton, using a straightforward translation [McCulloch and Pitts, 1943, Kleene, 1956]. An extension to the class of weighted automata is given in [Bader *et al.*, 2004a].

2.2 Connectionist Term Representation

The representation of possibly infinite structures in a finite network is one of the major obstacles on the way to neural-symbolic integration [Bader *et al.*, 2004b].

Input		**Hidden**		**Output**
$(A\ B)$	$\rightarrow$	$R_1(t)$	$\rightarrow$	$(A'(t)\ B'(t))$
$(C\ D)$	$\rightarrow$	$R_2(t)$	$\rightarrow$	$(C'(t)\ D'(t))$
$(R_1(t)\ R_2(t))$	$\rightarrow$	$R_3(t)$	$\rightarrow$	$(R_1'(t)\ R_2'(t))$

Table 1. Extracted training samples from the tree shown in Figure 3.

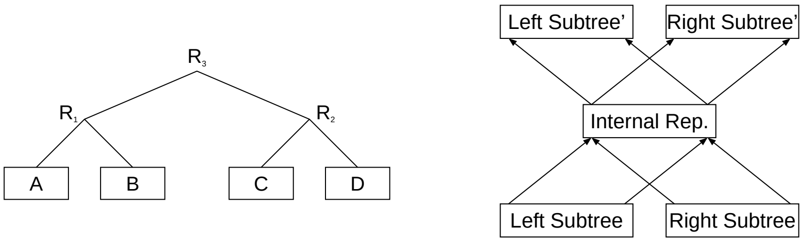


Figure 3. Example tree and a RAAM for binary trees.

One attempt to solve this will be discussed in this section, namely the idea of *recursive autoassociative memories (RAAMs)* as introduced in [Pollack, 1990], where a fixed length representation of variable sized data is obtained by training an artificial neural network using backpropagation. Again, we will try to illustrate the ideas by discussing a simple example.

EXAMPLE 2. Figure 3 shows a small binary tree which shall be encoded in a fixed-length real vector. The resulting RAAM-network is depicted in Figure 3, where each box depicts a layer of 4 units. The network is trained as an encoder-decoder network, i.e. it reproduces the input activations in the output layer [Bishop, 1995]. In order to do this, it must create a compressed representation in the hidden layer. Table 1 shows the activations of the layers during the training of the network. As the training converges we shall have $A = A'$, $B = B'$, etc. To encode the terminal symbols A, B, C and D we use the vectors $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, $(0, 0, 1, 0)$ and $(0, 0, 0, 1)$ respectively. The representations of R_1, R_2 and R_3 are obtained during training. After training the network, it is sufficient to keep the internal representation R_3, since it contains all necessary information for recreating the full tree. This is done by plugging it into the hidden layer and recursively using the output activations, until binary vectors, hence terminal symbols, are reached.

While recreating the tree from its compressed representation, it is necessary to distinguish terminal and non-terminal vectors, i.e. those which represent leafs of the trees from those representing nodes. Due to noise or inac-

Rules	**Facts**
$Owns(y,z) \leftarrow Gives(x,y,z)$	$Gives(john, josephine, book)$
$Owns(x,y) \leftarrow Buys(x,y)$	$Buys(carl, x)$
$Can\text{–}sell(x,y) \leftarrow Owns(x,y)$	$Owns(josephine, ball)$

Table 2. A knowledge base for *Own* and *Can–sell*

curacy, it can be very hard to recognise the "1-of-n"-vectors representing terminal symbols. In order to circumvent this problem different solutions were proposed, which can be found in [Stolcke and Wu, 1992, Sperduti, 1994a, Sperduti, 1994b]. The ideas described above for binary vectors apply also for trees with larger, but fixed, branching factors, by simply using bigger input and output layers. In order to store sequences of data, a version called S-RAAM (for sequential RAAM) can be used [Pollack, 1990]. In [Blair, 1997] modifications were proposed to allow the storage of deeper and more complex data structures than before, but their applicability remains to be shown [Kalinke, 1997]. Other recent approaches for enhancement have been studied e.g. in [Sperduti *et al.*, 1995, Kwasny and Kalman, 1995, Sperduti *et al.*, 1997, Hammerton, 1998, Adamson and Damper, 1999], which also include some applications. A recent survey which includes RAAM architectures and addresses structured processing can be found in [Frasconi *et al.*, 2001]. The related approach on *Holographic reduced representations (HRRs)* [Plate, 1991, Plate, 1995] also uses fixed-length representations of variable-sized data, but using different methods.

2.3 Reflexive Connectionist Reasoning

A wide variety of tasks can be solved by humans very fast and efficiently. This type of reasoning is sometimes referred to as *reflexive reasoning*. The SHRUTI system [Shastri and Ajjanagadde, 1993] provides a connectionist architecture performing this type of reasoning. Relational knowledge is encoded by clusters of cells and inferences by means of rhythmic activity over the cell clusters. It allows to encode a (function-free) fragment of first-order predicate logic analyzed in [Hölldobler *et al.*, 1999b]. Binding of variables – a particularly difficult aspect of neural-symbolic integration – is obtained by time-synchronization of activities of neurons.

EXAMPLE 3. Table 2 shows a knowledge base describing what it means to own something and to be able to sell it. Furthermore it states some facts. The resulting SHRUTI network is shown in Figure 4.

Recent enhancements, as reported in [Shastri, 1999] and [Shastri and Wendelken, 1999], allow e.g. the support of negation and inconsistency. [Wendelken and Shastri, 2003]

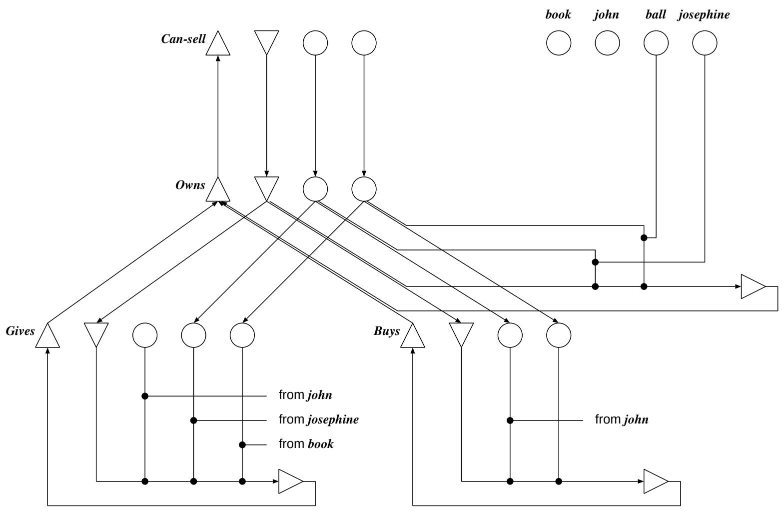


Figure 4. A SHRUTI network for the knowledge base from Table 2. Each predicate is represented by two *relais units* ($\triangle, \bigtriangledown$) and a set of *argument units* ($\bigcirc$). Constants are represented as $\bigcirc$-units in the upper right. Facts are implemented using $\triangleright$-units.

adds very basic learning capabilities to the system, while [Wendelken and Shastri, 2004] addresses the problem of multiple reuse of knowledge rules, an aspect which limits the capabilities of SHRUTI.

2.4 Connectionist Model Generation using the Core Method

In 1994, Hölldobler and Kalinke proposed a method to translate a propositional logic program into a neural network [Hölldobler and Kalinke, 1994] (a revised treatment is contained in [Hitzler *et al.*, 2004]), such that the network will settle down in a state corresponding to a model of the program. To achieve this goal, not the program itself, but rather the associated consequence operator was implemented using a connectionist system. The realization is close in spirit to [McCulloch and Pitts, 1943], and Figure 5 shows a propositional logic program and the corresponding network.

EXAMPLE 4. The simple logic program in Figure 5 states that a is a fact, b follows from a, etc. This "follows-from" is usually captured by the associated consequence operator T_P [Lloyd, 1988]. The figure shows also the corresponding network, obtained by the algorithm given in [Hölldobler and Kalinke, 1994]. For each atom (a, b, c, d, e) there is a unit in the input- and output layer, whose activation represents the truth value of the corresponding atom. Fur-

$C_1 : a.$
$C_2 : b \leftarrow a.$
$C_3 : c \leftarrow a, b$
$C_4 : d \leftarrow e$
$C_5 : e \leftarrow d$

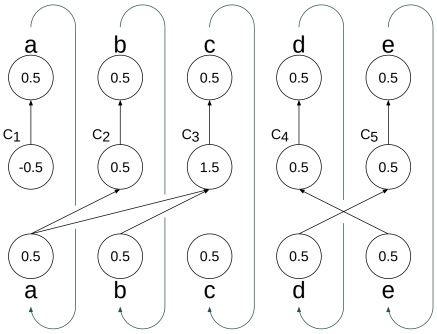


Figure 5. A simple propositional logic program and the corresponding network. Numbers within the units denote the thresholds. All weights are set to 1.

thermore, for each rule in the program there is a unit in the hidden layer, acting as a conjunction. If all requirements are met, this unit becomes active and propagates its activation to the consequence-unit in the output layer.

It can be shown that every logic program can be implemented using a 3-layer network of binary threshold units, and that 2-layer networks do not suffice. It was also shown that under some syntactic restrictions on the programs, their semantics could be recovered by recurrently connecting the output- and the input layer of the network (as indicated in Figure 5) and propagating activation exhaustively through the resulting recurrent network. Key idea to [Hölldobler and Kalinke, 1994] was to represent logic programs by means of their associated *semantic operators*, i.e. by connectionist encoding of an operator which captures the meaning of the program, instead of encoding the program directly. More precisely, the functional input-output behaviour of a semantic operator T_P associated with a given program P is encoded by means of a feedforward neural network N_P which, when presented an encoding of some I to its input nodes, produces $T_P(I)$ at its output nodes. Output nodes can also be connected recurrently back to the input nodes, resulting in a connectionist computation of iterates of I under T_P, as used e.g. in the computation of the semantics or meaning of P [Lloyd, 1988]. I, in this case, is a (Herbrand-)interpretation for P, and T_P is a mapping on the set I_P of all (Herbrand-)interpretations for P.

This idea for the representation of logic programs spawned several investigations in different directions. As [Hölldobler and Kalinke, 1994] employed binary threshold units as activation functions of the network nodes, the results were lifted to sigmoidal and hence differentiable activation functions in [Garcez *et al.*, 1997, Garcez and Zaverucha, 1999]. This way, the connectionist representation of logic programs resulted in a network architecture which could be trained using standard backpropagation algorithms. The resulting connectionist inductive learning and reasoning system CILP was completed by providing corresponding knowledge extraction algorithms [Garcez *et al.*, 2001]. Further extensions to this include modal [Garcez *et al.*, 2002b] and intuitionistic logics [Garcez *et al.*, 2003]. Metalevel priories between rules were introduced in [Garcez *et al.*, 2000]. An in-depth treatment of the whole approach can be found in [Garcez *et al.*, 2002a]. The *knowledge based artificial neural networks (KBANN)* [Towell and Shavlik, 1994] are closely related to this approach, by using similar techniques to implement propositional logic formulae within neural networks, but with a focus on learning.

Another work following up on [Hölldobler and Kalinke, 1994] concerns the connectionist treatment of first-order logic programming. [Seda, 2005] and [Seda and Lane, 2005] approach this by approximating given first-order programs P by finite subprograms of the grounding of P. These subprograms can be viewed as propositional ones and encoded using the original algorithm from [Hölldobler and Kalinke, 1994]. [Seda, 2005] and [Seda and Lane, 2005] show that arbitrarily accurate encodings are possible for certain programs including definite ones (i.e. programs not containing negation as failure). They also lift their results to logic programming under certain multi-valued logics.

A more direct approach to the representation of first-order logic programs based on [Hölldobler and Kalinke, 1994] was pursued in [Hölldobler *et al.*, 1999a, Hitzler and Seda, 2000, Hitzler *et al.*, 2004, Hitzler, 2004, Bader *et al.*, 2005a, Bader *et al.*, 2005b]. The basic idea again is to represent semantic operators $T_P : I_P \to I_P$ instead of the program P directly. In [Hölldobler and Kalinke, 1994] this was achieved by assigning propositional variables to nodes, whose activations indicate whether the nodes are true or false within the currently represented interpretation. In the propositional setting this is possible because for any given program only a finite number of truth values of propositional variables plays a role – and hence the finite network can encode finitely many propositional variables in the way indicated. For first-order programs, infinite interpretations have to be taken into account, thus an encoding of ground atoms by one neuron each is impossible as it would result in an infinite network, which is not computationally feasible to work with.

The solution put forward in [Hölldobler *et al.*, 1999a] is to employ the capability of standard feedforward networks to propagate real numbers. The problem is thus reduced to encoding I_P as a set of real numbers in a computationally feasible way, and to provide means to actually construct the networks starting from their input-output behaviour. Since sigmoidal units can be used, the resulting networks are trainable by backpropagation. [Hölldobler *et al.*, 1999a] spelled out these ideas in a limited setting for a small class of programs, and was lifted in [Hitzler and Seda, 2000, Hitzler *et al.*, 2004] to a more general setting, including the treatment of multi-valued logics. [Hitzler, 2004] related the results to logic programming under non-monotonic semantics. In these reports, it was shown that approximation of logic programs by means of standard feedforward networks is possible up to any desired degree of accuracy, and for fairly general classes of programs. However, no algorithms for practical generation of approximating networks from given programs could be presented. This was finally done in [Bader *et al.*, 2005b], and implementations of the approach are currently under way, and shall yield a first-order integrated neural-symbolic system with similar capabilities as the propositional system CILP.

There exist two alternative approaches to the representation of first-order logic programs via their semantic operators, which have not been studied in more detail yet. The first approach, reported in [Bader and Hitzler, 2004], uses insights from fractal geometry as in [Barnsley, 1993] to construct iterated function systems whose attractors correspond to fixed points of the semantic operators. The second approach builds on Gabbay's *Fibring logics* [Gabbay, 1999], and the corresponding Fibring Neural Networks [Garcez and Gabbay, 2004]. The resulting system, presented in [Bader *et al.*, 2005a], employs the fibring idea to control the firing of nodes such that it corresponds to term matching within a logic programming system. It is shown that certain limited kinds of first-order logic programs can be encoded this way, such that their models can be computed using the network.

3 A New Classification Scheme

In this section we will introduce a classification scheme for neural-symbolic systems. This way, we intend to bring some order to the heterogeneous field of research, whose individual approaches are often largely incomparable. We suggest to use a scheme consisting of three main axes as depicted in Figure 6, namely *Interrelation*, *Language* and *Usage*.

For the interrelation-axis, depicted in Figure 7, we roughly follow the scheme introduced and discussed in [Hilario, 1995, Hatzilygeroudis and Prentzas, 2004], but adapted to the particular focus which we will put forward. In particular, the classifications presented in [Hilario, 1995, Hatzilygeroudis and Prentzas, 2004] strive to depict each system at exactly one point in a taxonomic tree. From our perspective, certain properties or design decisions of systems are rather independent, and should be understood as different *dimensions*. From this perspective approaches can first be divided into two main classes,

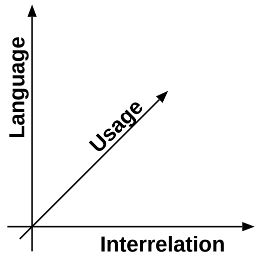


Figure 6. Main Axes

Figure 7. Interrelation

namely into *integrated* (called *unified* or *translational* in [Hilario, 1995, Hatzilygeroudis and Prentzas, 2004]) and *hybrid* systems. Integrated are those, where full symbolic processing functionalities emerge from neural structures and processes – further details will be discussed in Section 4.1. Integrated systems can be further subdivided into neuronal and connectionist approaches, as discussed in Section 4.1. Neuronal indicates the usage of neurons which are very closely related to biological neurons. In connectionist approaches there is no claim to neurobiological plausibility, instead general artificial neural network architectures are used. Depending on their architecture, they can be split into standard and non-standard networks. Furthermore, we can distinguish local and distributed representation of the knowledge which will also be discussed in more detail in Section 4.1.

Note that the subdivisions belonging to the interrelation axis are again independent of each other. They should be understood as independent subdimensions, and could also be depicted this way by using further coordinate axes. We hope that our simplified visualisation makes it easier to maintain an overview. But to be pedantic, for our presentation we actually understand the neural-connectionist dimension as a subdivision of integrated systems, and the distributed-local and standard-nonstandard dimensions as independent subdivisions of connectionist systems – simply because this currently suffices for classification.

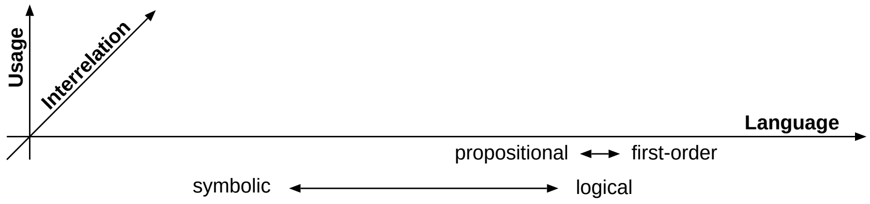


Figure 8. Language

Figure 8 depicts the second axis in our scheme. Here, the systems are divided according to the language used in their symbolic part. We distinguish between *symbolic* and *logical* languages. Symbolic approaches include the relation to automata as in [McCulloch and Pitts, 1943], to grammars [Elman, 1990, Fletcher, 2001] or to the storage and retrieval of terms [Pollack, 1990], whereas the logical approaches require either propositional or first order logic systems, as e.g. in [Hölldobler and Kalinke, 1994] and discussed in Section 2.4. The language axis will be discussed in more detail in Section 4.2.

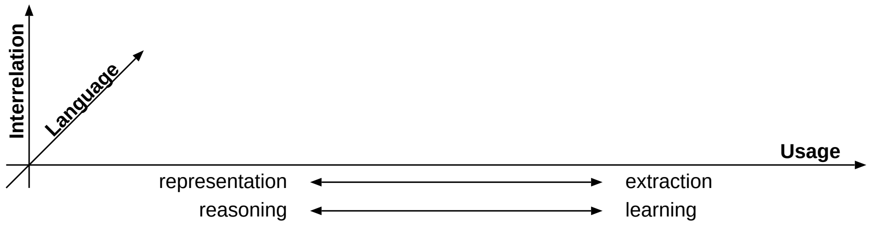


Figure 9. Usage

Most systems focus on one or only a few aspects of the neural-symbolic learning cycle depicted in Figure 1, i.e. either the representation of symbolic knowledge within a connectionist setting, or the training of preinitialized networks, or the extraction of symbolic systems from a network. Depending on this main focus we can distinguish the systems as shown in Figure 9. The issues of *extraction* vs. *representation* on the one hand and *learning* vs. *reasoning* on the other hand, are discussed in Section 4.3. Systems may certainly cover several or all of these aspects, i.e. they may span whole subdimensions.

4 Dimensions of Neural Symbolic Integration

In this section, we will survey main research results in this area by classifying them according to eight dimensions, marked by the arrows in Figures 7-9.

- Interrelation
  1. Integrated versus hybrid
  2. Neuronal versus connectionist
  3. Local versus distributed
  4. Standard versus nonstandard
- Language
  5. Symbolic versus logical
  6. Propositional versus first-order
- Usage
  7. Extraction versus representation
  8. Learning versus reasoning

As discussed above, we believe that these dimensions mark the main points of distinction between different integrated neural-symbolic systems. The chapter is structured accordingly, examining each of the dimensions in turn.

4.1 Interrelation

Integrated versus Hybrid

This section serves to further clarify what we understand by *neural-symbolic integration.* Following the rationale laid out in the introduction, we understand why it is desirable to combine symbolic and connectionist approaches, and there are obviously several ways how this can be done. From a bird's eye view, we can distinguish two main paradigms, which we call *hybrid* and *integrated* (or following [Hilario, 1995], *unified*) systems, and this survey is concerned with the latter.

Hybrid systems are characterized by the fact that they combine two or more problem-solving techniques in order to address a problem, which run in parallel, as depicted in Figure 10.

An *integrated* neural-symbolic system differs from a hybrid one in that it consists of one connectionist main component in which symbolic knowledge is processed, see Figure 10 (right). Integrated systems are sometimes

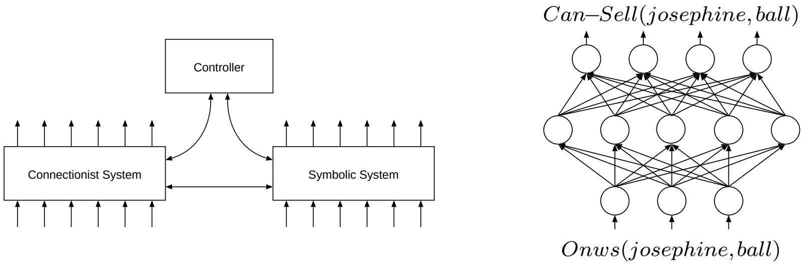


Figure 10. Hybrid (left) versus integrated (right) architecture.

also referred to as *embedded* or *monolithic* hybrid systems, cf. [Sun, 2001]. Examples for integrated systems are e.g. those presented in Sections 2.2-2.4.

For either architecture, one of the central issues is the representation of symbolic data in connectionist form [Bader *et al.*, 2004b]. For the hybrid system, these transformations are required for passing information between the components. The integrated architecture must implicitly or explicitly deal with symbolic data by connectionist means, i.e. must also be capable of similar transformations.

This survey covers integrated systems only, the study of which appears to be particularly challenging. For recent selective overview literature see e.g. [Browne and Sun, 2001, Garcez *et al.*, 2002a, Bader *et al.*, 2004b]. The first, [Browne and Sun, 2001], focuses on reasoning systems. The field of propositional logic is thoroughly covered in [Garcez *et al.*, 2002a], where the authors revisit the approach of [Hölldobler and Kalinke, 1994] and explain their extensions including applications to real world problems, like fault diagnosis. In [Bader *et al.*, 2004b] the emphasis is on the challenge problems arising from first-order neural-symbolic integration.

Neuronal versus Connectionist

There are two driving forces behind the field of neural-symbolic integration: On the one hand it is the striving for an understanding of human cognition, and on the other it is the vision of combining connectionist and symbolic artificial intelligence technology in order to arrive at more powerful reasoning and learning systems for computer science applications.

In [McCulloch and Pitts, 1943] the motivation for the study was to understand human cognition, i.e. to pursue the question how higher cognitive – logical – processes can be performed by artificial neural networks. In this line of research, the question of *biological feasability* of a network architecture is prominent, and inspiration is often taken from biological counterparts.

The SHRUTI system [Shastri and Ajjanagadde, 1993] as described in Section 2.3, for example, addresses the question how it is possible that biological networks perform certain reasoning tasks very quickly. Indeed, for some complex recognition tasks which involve reasoning capabilities, human responses occur sometimes at *reflexive* speed, particularly within a time span which allows processing through very few neuron layers only. As mentioned above, time-synchronization was used for the encoding of variable binding in SHRUTI.

The recently developed spiking neurons networks [Maass, 2002] take an even more realistic approach to the modelling of temporal aspects of neural activity. Neurons, in this context, are considered to be firing so-called *spike trains*, which consist of patterns of firing impulses over certain time intervals. The complex propagation patterns within a network are usually analysed by statistical methods. The encoding of symbolic knowledge using such temporal aspects has hardly been studied so far, an exception being [Sougne, 2001]. We perceive it as an important research challenge to relate the neurally plausible spiking neurons approach to neural-symbolic integration research. To date, however, only a few preliminary results on computational aspects of spiking neurons have been obtained [Natschläger and Maass, 2002, Maass and Markram, 2004, Maass *et al.*, 2005].

Another recent publication, [van der Velde and de Kamps, 2005], shows how natural language could be encoded using biologically plausible models of neural networks. The results appear to be suitable for the study of neural-symbolic integration, but it remains to be investigated to which extent the provided approach can be transfered to symbolic reasoning. Similarly inspiring might be the recent book [Hawkins and Blakeslee, 2004] and accompanying work, though it discusses neural-symbolic relationships on a very abstract level only.

The lines of research just reviewed take their major motivation from the goal to achieve biologically plausible behaviour or architectures. As already mentioned, neural-symbolic integration can also be pursued from a more technically motivated perspective, driven by the goal to combine the advantages of symbolic and connectionist approaches by studying their interrelationships. The work on the Core Method, discussed in Section 2.4, can be subsumed under this technologically inspired perspective.

Local versus Distributed Representation of Knowledge

For integrated neural-symbolic systems, the question is crucial how symbolic knowledge is represented within the connectionist system. If standard networks are being trained using backpropagation, the knowledge acquired during the learning process is spread over the network in diffuse ways, i.e. it is in general not easy or even possible to identify one or a small number of nodes whose activations contain and process a certain symbolic piece of knowledge.

The RAAM architecture and their variants as discussed in Section 2.2 are clearly based on distributed representations. Technically, this stems from the fact that the representation is initially learned, and no explicit algorithm for translating symbolic knowledge into the connectionist setting is being used.

Most other approaches to neural-symbolic integration, however, represent data locally. SHRUTI (Section 2.3) associates a defined node assembly to each logical predicate, and the architecure does not allow for distributed representation. The approaches for propositional connectionist model generation using the Core Method (Section 2.4) encode propositional variables as single nodes in the input resp. output layer, and logical formulae (rules) by single nodes in the hidden layer of the network.

The design of distributed encodings of symbolic data appears to be particular challenging. It also appears to be one of the major bottlenecks in producing applicable integrated neural-symbolic systems with learning and reasoning abilities [Bader *et al.*, 2004b]. This becomes apparent e.g. in the difficulties faced by the first-order logic programming approaches discussed in Section 2.4. Therein, symbolic entities are not represented directly. Instead, interpretations (i.e. valuations) of the logic are being represented, which contain truth value assignments to language constructs. Concrete representations, as developed in [Bader *et al.*, 2005b], distribute the encoding of the interpretations over several nodes, but in a diffuse way. The encoding thus results in a distributed representation. Similar considerations apply to the recent proposal [Gust and Kühnberger, 2005], where first-order logic is first converted into variable-free form (using topoi from category theory), and then fed to a neural network for training.

Standard versus Non/standard Network Architecture

Even though neural networks are a widely accepted paradigm in AI it is hard to make out a standard architecture. But, all so called standard-architecture systems agree at least on the following:

- only real numbers are propagated along the connections
- units compute very simple functions only
- all units behave similarly (i.e. they use similar simple functions and the activation values are always within a small range)
- only simple recursive structures are used (e.g. connecting only the output back to the input layer, or use selfrecursive units only)

When adhering to these standard design principles, powerful learning techniques as e.g. backpropagation [Rumelhart *et al.*, 1986] or Hebbian Learning [Hebb, 1949] can be used to train the networks, which makes them applicable to real world problems.

However, these standard architectures do not easily lend themselves to neural-symbolic integration. In general, it is easier to use non-standard architectures in order to represent and work with structured knowledge, with the drawback that powerful learning abilities are often lost.

Neural-symbolic approaches using standard networks are e.g. the CILP system [Garcez and Zaverucha, 1999], KBANN [Towell and Shavlik, 1994], RAAM (Section 2.2) and [Seda and Lane, 2005] (Section 2.4). Usually, they consist of a layered network, consisting of three or in case of KBANN more layers, and sigmoidal units are being used. For these systems experimental results are available showing their learning capabilities. As discussed above, these systems are able to handle propositional knowledge (or first order with a finite domain). Similar observations can be made about the standard architectures used in [Hölldobler *et al.*, 1999a, Hitzler *et al.*, 2004, Bader *et al.*, 2005b] for first-order neural-symbolic integration.

Non-standard networks were used e.g. in the SHRUTI system [Shastri and Ajjanagadde, 1993], and in the approaches described in [Bader and Hitzler, 2004] and [Bader *et al.*, 2005a]. In all these implementations non-standard units and non-standard architectures were used, and hence none of the usual learning techniques are applicable. However, for the SHRUTI system limited learning techniques based on Hebbian Learning [Hebb, 1949] were developed [Shastri, 2002, Shastri and Wendelken, 2003, Wendelken and Shastri, 2003].

4.2 Language

Symbolic versus Logical

One of the motivations for studying neural-symbolic integration is to combine connectionist learning capabilities with symbolic knowledge processing, as already mentioned. While our main interest is in pursuing logical aspects of symbolic knowledge, this is not necessarily always the main focus of investigations.

Work on representing automata or weighted automata [Kleene, 1956, McCulloch and Pitts, 1943, Bader *et al.*, 2004a] (Section 2.1) using artificial neural networks, for example focuses on computationally relevant structures, such as automata, and not directly on logically encoded knowledge. Nevertheless, such investigations show how to deal with structural knowledge within a connectionist setting, and can serve as inspiration for corresponding research on logical knowledge.

Recursive autoassociative memory, RAAM, and their variants as discussed in Section 2.2, deals with terms only, and not directly with logical content. RAAM allows connectionist encodings of first-order terms, where the underlying idea is to present terms or term trees sequentially to a connectionist system which is trained to produce a compressed encoding characterized by the activation pattern of a small collection of nodes. To date, storage capacity is very limited, and connectionist processing of the stored knowledge has not yet been investigated in detail.

A considerable body of work exists on the connectionist processing and learning of structured data using recurrent networks [Sperduti *et al.*, 1995, Sperduti *et al.*, 1997, Frasconi *et al.*, 2001, Hammer, 2002, Hammer, 2003, Hammer *et al.*, 2004a, Hammer *et al.*, 2004b]. The focus is on tree representations and manipulation of the data.

[Hölldobler *et al.*, 1997, Kalinke and Lehmann, 1998] study the representation of counters using recurrent networks, and connectionist unification algorithms as studied in [Hölldobler, 1990, Hölldobler and Kurfess, 1992, Hölldobler, 1993] are designed for manipulating terms, but already in a clearly logical context. The representation of grammars [Giles *et al.*, 1991] or more generally of natural language constructs [van der Velde and de Kamps, 2005] also has a clearly symbolic (as opposed to logical) focus.

It remains to be seen, however, to what extent the work on connectionist processing of structured data can be reused in logical contexts for creating integrated neural-symbolic systems with reasoning capabilities. Integrated reasoning systems like the ones presented in Sections 2.3 and 2.4 currently lack the capabilities of the term-based systems, so that a merging of these efforts appears to be a promising albeit challenging goal.

Propositional versus First-Order

Logic-based integrated neural-symbolic systems differ as to the knowledge representation language they are able to represent. Concerning the capabilities of the systems, a major distinction needs to be made between those which deal with propositional logics, and those based on first-order predicate (and related) logics.

What we mean by propositional logics in this context includes propositional modal, temporal, non-monotonic, and other non-classical logics. One of their characteristic feature which distinguishes them from first-order logics for neural-symbolic integration is the fact that they are of a finitary nature: propositional theories in practice involve only a finite number of propositional variables, and corresponding models are also finite. Also, sophisticated symbol processing as needed for nested terms in the form of substitutions or unification is not required.

Due to their finiteness it is thus fairly easy to implement propositional logic programs using neural networks [Hölldobler and Kalinke, 1994] (Section 2.4). A considerable body of work deals with the extension of this approach to non-classical logics [Garcez *et al.*, 2005, Garcez *et al.*, 2000, Garcez *et al.*, 2002b, Garcez *et al.*, 2003, Garcez *et al.*, 2004a, Garcez *et al.*, 2004b, Garcez and Lamb, 200x]. This includes modal, intuitionistic, and argumentation-theoretic approaches, amongst others. Earlier work on representing propositional logics is based on Hopfield networks [Pinkas, 1991b, Pinkas, 1991a] but has not been followed up on recently.

In contrast to this, predicate logics – which for us also include modal, non-monotonic, etc. extensions – in general allow to use function symbols as language primitives. Consequently, it is possible to use terms of arbitrary depth, and models necessarily assign truth values to an infinite number of ground atoms. The difficulty in dealing with this in a connectionist setting lies in the finiteness of neural networks, necessitating to capture the infinitary aspects of predicate logics by finite means. The first-order approaches presented in [Hölldobler *et al.*, 1999a, Hitzler and Seda, 2000, Bader and Hitzler, 2004, Hitzler *et al.*, 2004, Bader *et al.*, 2005a, Bader *et al.*, 2005b] (Section 2.4) solve this problem by using encodings of infinite sets by real numbers, and representing them in an approximate manner. They can also be carried over to non-monotonic logics [Hitzler, 2004].

[Bader *et al.*, 2005a], which builds on [Garcez and Gabbay, 2004] and [Gabbay, 1999] uses an alternative mechanism in which unification of terms is controlled via fibrings. More precisely, certain network constructs encode the matching of terms and act as gates to the firing of neurons whenever corresponding symbolic matching is achieved.

A prominent subproblem in first-order neural-symbolic integration is that of variable binding. It refers to the fact that the same variable may occur in several places in a formula, or that during a reasoning process variables may be bound to instantiate certain terms. In a connectionist setting, different parts of formulae and different individuals or terms are usually represented independently of each other within the system. The neural network paradigm, however, forces subnets to be blind with respect to detailed activation patterns in other subnets, and thus does not lend itself easily to the processing of variable bindings.

Research on first-order neural-symbolic integration has led to different means of dealing with the variable binding problem. One of them is to use temporal synchrony to achieve the binding. This is encoded in the SHRUTI system (Section 2.3), where the synchronous firing of variable nodes with constant nodes encodes a corresponding binding. Other approaches, as discussed in [Browne and Sun, 1999], encode binding by relating the propagated activations, i.e. real numbers.

Other systems avoid the variable binding problem by converting predicate logical formulae into variable-free representations. The approaches in [Hölldobler *et al.*, 1999a, Hitzler and Seda, 2000, Hitzler *et al.*, 2004, Hitzler, 2004, Seda, 2005, Seda and Lane, 2005, Bader *et al.*, 2005a, Bader *et al.*, 2005b] (Section 2.4) make conversions to (infinite) propositional theories, which are then approximated. [Gust and Kühnberger, 2005] use topos theory instead.

It shall be noted here that SHRUTI (Section 2.3) addresses the variable binding problem, but allows to encode only a very limited fragment of first-order predicate logic [Hölldobler *et al.*, 1999b]. In particular, it does not allow to deal with function symbols, and thus could still be understood as a finitary fragment of predicate logic.

4.3 Usage

Extraction versus Representation

The representation of symbolic knowledge is necessary even for classical applications of connectionist learning. As an example, consider the neural-networks-based Backgammon playing program TD-Gammon [Tesauro, 1995], which achieves professional players' strength by temporal difference learning on data created by playing against itself. TD-Gammon represents the Backgammon board in a straightforward way, by encoding the squares and placement of pieces via assemblies of nodes, thus representing the structured knowledge of a board situation directly by a certain activation pattern of the input nodes.

In this and other classical application cases the represented symbolic knowledge is not of a complex logical nature. Neural-symbolic integration, however, attempts to achieve connectionist processing of complex logical knowledge, learning, and inferences, and thus the question how to represent logical knowledge bases in suitable form becomes dominant. Different forms of representation have already been discussed in the context of local versus distributed representations.

Returning to the TD-Gammon example, we would also be interested in the complex knowledge as acquired by TD-gammon during the learning process, encoding the strategies with which this program beats human players. If such knowledge could be extracted in symbolic form, it could be used for further symbolic processing using inference engines or other knowledge based systems.

It is apparent, that both the representation and the extraction of knowledge are of importance for integrated neural-symbolic systems. They are needed for closing the neural-symbolic learning cycle (Figure 1). However, they are also of independent interest, and are often studied separately.

As for the representation of knowledge, this component is present in all systems presented so far. The choice how representation is done often determines whether standard architectures are used, if a local or distributed approach is taken, and whether standard learning algorithms can be employed.

A large body of work exists on extracting knowledge from trained networks, usually focusing on the extraction of rules. [Jacobsson, 2005] gives a recent overview over extraction methods. A method from 1992 [Giles *et al.*, 1991] is still up to date, where a method is given to extract a grammar represented as a finite state machine from a trained recurrent neural network. [McGarry *et al.*, 1999] show how to extract rules from radial basis function networks by identifying minimal and maximal activation values. Some of the other efforts are reported in [Towell and Shavlik, 1993, Andrews *et al.*, 1995, Bologna, 2000, Garcez *et al.*, 2001, Lehmann *et al.*, 2005]

It shall be noted that only a few systems have been proposed to date which include representation, learning, and extraction capabilities in a meaningful way, one of them being CILP [Garcez *et al.*, 1997, Garcez and Zaverucha, 1999, Garcez *et al.*, 2001]. It is to date a difficult research challenge to provide similar functionalities in a first-order setting.

Learning versus Reasoning

Ultimately, our goal should be to produce an effective AI system with added reasoning and learning capabilities, as recently pointed out by Valiant [Valiant, 2003] as a key challenge for computer science. It turns out that most current systems have either learning capabilities or reasoning capabilities, but rarely both. SHRUTI (Section 2.3), for example, is a reasoning system with very limited learning support.

In order to advance the state of the art in the sense of Valiant's vision mentioned above, it will be necessary to install systems with combined capabilities. In particular, learning should not be independent of reasoning, i.e. initial knowledge and logical consequences thereof should help guiding the learning process. There is no system to-date which realizes this in any way, and new ideas will be needed to attack this problem.

5 Conclusions and Further Work

Intelligent systems based on symbolic knowledge processing, on the one hand, and on artificial neural networks, on the other, differ substantially. Nevertheless, these are both standard approaches to artificial intelligence and it would be very desirable to combine the robustness of neural networks with the expressivity of symbolic knowledge representation. This is the reason why the importance of the efforts to bridge the gap between the connectionist and symbolic paradigms of Artificial Intelligence has been widely recognised. As the amount of hybrid data containing symbolic and statistical elements as well as noise increases in diverse areas such as bioinformatics or text and web mining, neural-symbolic learning and reasoning becomes of particular practical importance. Notwithstanding, this is not an easy task, as illustrated in the survey.

The merging of theory (background knowledge) and data learning (learning from examples) in neural networks has been indicated to provide learning systems that are more effective than e.g. purely symbolic and purely connectionist systems, especially when data are noisy [Garcez and Zaverucha, 1999]. This has contributed decisively to the growing interest in developing neural-symbolic systems, i.e. hybrid systems based on neural networks that are capable of learning from examples and background knowledge, and of performing reasoning tasks in a massively parallel fashion.

However, while symbolic knowledge representation is highly recursive and well understood from a declarative point of view, neural networks encode knowledge implicitly in their weights as a result of learning and generalisation from raw data, which are usually characterized by simple feature vectors. While significant theoretical progress has recently been made on knowledge representation and reasoning using neural networks, and on direct processing of symbolic and structured data using neural methods, the integration of neural computation and expressive logics such as first order logic is still in its early stages of methodological development.

Concerning knowledge extraction, we know that neural networks have been applied to a variety of real-world problems (e.g. in bioinformatics, engineering, robotics), and they were particularly successful when data are noisy. But entirely satisfactory methods for extracting symbolic knowledge from such trained networks in terms of accuracy, efficiency, rule comprehensibility, and soundness are still to be found. And problems on the stability and learnability of recursive models currently impose further restrictions on connectionist systems.

In order to advance the state of the art, we believe that it is necessary to look at the biological inspiration for neural-symbolic integration, to use more formal approaches for translating between the connectionist and symbolic paradigms, and to pay more attention to potential application scenarios.

The general motivation for research in the field of neural-symbolic integration (just given) arises from conceptual observations on the complementary nature of symbolic and neural network based artificial intelligence described above. This conceptual perspective is sufficient for justifying the mainly foundations-driven lines of research being undertaken in this area so far. However, it appears that this conceptual approach to the study of neural-symbolic integration has now reached an impasse which requires the identification of use cases and application scenarios in order to drive future research.

Indeed, the theory of integrated neural-symbolic systems has reached a quite mature state but has not been tested extensively so far on real application data. The current systems have been developed for the study of general principles, and are in general not suitable for real data or application scenarios that go beyond propositional logic. Nevertheless, these studies provide methods which can be exploited for the development of tools for use cases, and significant progress can now only be expected as a continuation of the fundamental research undertaken in the past.

In particular, first-order neural-symbolic integration still remains a widely open issue, where advances are very difficult, and it is very hard to judge to date to what extent the theoretical approaches can work in practice. We believe that the development of use cases with varying levels of expressive complexity is, as a result, needed to drive the development of methods for neural-symbolic integration beyond propositional logic [Hitzler *et al.*, 2005].

BIBLIOGRAPHY

[Adamson and Damper, 1999] M. J. Adamson and R. I. Damper. B-RAAM: A connectionist model which develops holistic internal representations of symbolic structures. *Connection Science*, 11(1):41–71, 1999.

[Andrews *et al.*, 1995] R. Andrews, J. Diederich and A. Tickle. A survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge–Based Systems*, 8(6), 1995.

[Bader and Hitzler, 2004] S. Bader and P. Hitzler. Logic programs, iterated function systems, and recurrent radial basis function networks. *Journal of Applied Logic*, 2(3):273–300, 2004.

[Bader *et al.*, 2004a] S. Bader, S. Hölldobler and A. Scalzitti. Semiring artificial neural networks and weighted automata. In G. Palm S. Biundo, T. Frühwirth, editor, *KI 2004: Advances in Artificial Intelligence. Proceedings of the 27th Annual German Conference on Artificial Intelligence, Ulm, Germany, September 2004*, volume 3238 of *Lecture Notes in Artificial Intelligence*, pages 281–294. Springer, 2004.

[Bader *et al.*, 2004b] S. Bader, P. Hitzler and S. Hölldobler. The integration of connectionism and knowledge representation and reasoning as a challenge for artificial intelligence. In L. Li and K. K. Yen, editors, *Proceedings of the Third International Conference on Information, Tokyo, Japan*, pages 22–33. International Information Institute, 2004. ISBN 4-901329-02-2.

[Bader *et al.*, 2005a] S. Bader, A. S. d'Avila Garcez and P. Hitzler. Computing first-order logic programs by fibring artificial neural network. In I. Russell and Z. Markov, editors, *Proceedings of the 18th International Florida Artificial Intelligence Research Symposium Conference, FLAIRS05, Clearwater Beach, Florida, May 2005*, pages 314–319. AAAI Press, 2005.

[Bader *et al.*, 2005b] S. Bader, P. Hitzler and A. Witzel. Integrating first-order logic programs and connectionist systems — a constructive approach. In A. S. d'Avila Garcez, J. Elman and P. Hitzler, editors, *Proceedings of the IJCAI-05 Workshop on Neural-Symbolic Learning and Reasoning, NeSy'05, Edinburgh, UK*, 2005.

[Barnden, 1995] J. A. Barnden. High-level reasoning, computational challenges for connectionism, and the conposit solution. *Applied Intelligence*, 5(2):103–135, 1995.

[Barnsley, 1993] M. Barnsley. *Fractals Everywhere.* Academic Press, San Diego, CA, USA, 1993.

[Bishop, 1995] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[Blair, 1997] A. Blair. Scaling-up RAAMs. Technical Report CS-97-192, University of Brandeis, 1997.

[Bologna, 2000] G. Bologna. Rule extraction from a multi layer perceptron with staircase activation functions. In *IJCNN (3)*, pages 419–424, 2000.

[Browne and Sun, 1999] A. Browne and R. Sun. Connectionist variable binding. *Expert Systems*, 16(3):189–207, 1999.

[Browne and Sun, 2001] A. Browne and R. Sun. Connectionist inference models. *Neural Networks*, 14(10):1331–1355, 2001.

[Elman, 1990] J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[Fletcher, 2001] P. Fletcher. Connectionist learning of regular graph grammars. *Connection Science*, 13(2):127–188, 2001.

[Frasconi *et al.*, 2001] P. Frasconi, M. Gori, A. Kuchler and A. Sperduti. From sequences to data structures: Theory and applications. In J. Kolen and S. C. Kremer, editors, *Dynamical Recurrent Networks*, pages 351–374. IEEE Press, 2001.

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

[Gallant, 1993] S. I. Gallant. *Neural network learning and expert systems*. MIT Press, Cambridge, MA, 1993.

[Garcez and Gabbay, 2004] A. S. d'Avila Garcez and D. M. Gabbay. Fibring neural networks. In *In Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 04). San Jose, California, USA, July 2004*. AAAI Press, 2004.

[Garcez and Lamb, 200x] D. M. Gabbay, A. S. d'Avila Garcez and L. C. Lamb. Value-based argumentation frameworks as neural-symbolic learning systems. *Journal of Logic and Computation*, 200x. To appear.

[Garcez and Zaverucha, 1999] A. S. d'Avila Garcez and G. Zaverucha. The connectionist inductive lerarning and logic programming system. *Applied Intelligence, Special Issue on Neural networks and Structured Knowledge*, 11(1):59–77, 1999.

[Garcez *et al.*, 1997] A. S. d'Avila Garcez, G. Zaverucha and L. A. V. de Carvalho. Logical inference and inductive learning in artificial neural networks. In C. Hermann, F. Reine and A. Strohmaier, editors, *Knowledge Representation in Neural networks*, pages 33–46. Logos Verlag, Berlin, 1997.

[Garcez *et al.*, 2000] A. S. d'Avila Garcez, K. Broda and D. M. Gabbay. Metalevel priorities and neural networks. In *Proceedings of the Workshop on the Foundations of Connectionist-Symbolic Integration, ECAI'2000, Berlin*, August 2000.

[Garcez *et al.*, 2001] A. S. d'Avila Garcez, K. Broda and D. M. Gabbay. Symbolic knowledge extraction from trained neural networks: A sound approach. *Artificial Intelligence*, 125:155–207, 2001.

[Garcez *et al.*, 2002a] A. S. d'Avila Garcez, K. B. Broda and D. M. Gabbay. *Neural-Symbolic Learning Systems — Foundations and Applications*. Perspectives in Neural Computing. Springer, Berlin, 2002.

[Garcez *et al.*, 2002b] A. S. d'Avila Garcez, L. C. Lamb and D. M. Gabbay. A connectionist inductive learning system for modal logic programming. In *Proceedings of the IEEE International Conference on Neural Information Processing ICONIP'02, Singapore*, 2002.

[Garcez *et al.*, 2003] A. S. d'Avila Garcez, L. C. Lamb and D. M. Gabbay. Neural-symbolic intuitionistic reasoning. In M. Koppen A. Abraham and K. Franke, editors, *Frontiers in Artificial Intelligence and Applications*, Melbourne, Australia, December 2003. IOS Press. Proceedings of the Third International Conference on Hybrid Intelligent Systems (HIS'03).

[Garcez *et al.*, 2004a] A. S. d'Avila Garcez, D. M. Gabbay and L. C. Lamb. Argumentation neural networks. In *Proceedings of 11th International Conference on Neural Information Processing (ICONIP'04)*, Lecture Notes in Computer Science LNCS, Calcutta, November 2004. Springer-Verlag.

[Garcez *et al.*, 2004b] A. S. d'Avila Garcez, L. C. Lamb, K. Broda and D. M. Gabbay. Applying connectionist modal logics to distributed knowledge representation problems. *International Journal of Artificial Intelligence Tools*, 2004.

[Garcez *et al.*, 2005] A. S. d'Avila Garcez, D. M. Gabbay and L. C. Lamb. *Connectionist Non-Classical Logics*. Springer-Verlag, 2005. To appear.

[Giles *et al.*, 1991] C. Giles, D. Chen, C. Miller, H. Chen, G. Sun and Y. Lee. Second-order recurrent neural networks for grammatical inference. In *Proceedings of the International Joint Conference on Neural Networks 1991*, volume 2, pages 273–281, New York, 1991. IEEE.

[Gust and Kühnberger, 2005] H. Gust and K.-U. Kühnberger. Learning symbolic inferences with neural networks. In *CogSci 2005*, 2005. to appear.

[Hammer *et al.*, 2004a] B. Hammer, A. Micheli, A. Sperduti and M. Strickert. Recursive self-organizing network models. *Neural Networks*, 17(8–9):1061–1085, 2004. Special issue on New Developments in Self-Organizing Systems.

[Hammer *et al.*, 2004b] B. Hammer, A. Micheli, M. Strickert and A. Sperduti. A general framework for unsupervised processing of structured data. *Neurocomputing*, 57:3–35, 2004.

[Hammer, 2002] B. Hammer. Recurrent networks for structured data — a unifying approach and its properties. *Cognitive Systems Research*, 3(2):145–165, 2002.

[Hammer, 2003] B. Hammer. Perspectives on learning symbolic data with connectionistic systems. In R. Kühn, R. Menzel, W. Menzel, U. Ratsch, M. M. Richter and I.-O. Stamatescu, editors, *Adaptivity and Learning*, pages 141–160. Springer, 2003.

[Hammerton, 1998] J. A. Hammerton. *Exploiting Holistic Computation: An evaluation of the Sequential RAAM*. PhD thesis, University of Birmingham, 1998.

[Hatzilygeroudis and Prentzas, 2000] I. Hatzilygeroudis and J. Prentzas. Neurules: Integrating symbolic rules and neurocomputing. In D. Fotiades and S. Nikolopoulos, editors, *Advances in Informatics*, pages 122–133. World Scientific, 2000.

[Hatzilygeroudis and Prentzas, 2004] I. Hatzilygeroudis and J. Prentzas. Neuro-symbolic approaches for knowledge representation in expert systems. *International Journal of Hybrid Intelligent Systems*, 1(3-4):111–126, 2004.

[Hawkins and Blakeslee, 2004] J. Hawkins and S. Blakeslee. *On Intelligence*. Henry Holt and Company, LLC, New York, 2004.

[Healy, 1999] M. J. Healy. A topological semantics for rule extraction with neural networks. *Connection Science*, 11(1):91–113, 1999.

[Hebb, 1949] D. O. Hebb. *The Organization of Behavior*. Wiley, New York, 1949.

[Hilario, 1995] M. Hilario. An overview of strategies for neurosymbolic integration. In R. Sun and F. Alexandre, editors, *Proceedings of the Workshop on Connectionist-Symbolic Integration: From Unied to Hybrid Approaches*, Montreal, 1995.

[Hitzler and Seda, 2000] P. Hitzler and A. K. Seda. A note on relationships between logic programs and neural networks. In Paul Gibson and David Sinclair, editors, *Proceedings of the Fourth Irish Workshop on Formal Methods, IWFM'00*, Electronic Workshops in Comupting (eWiC). British Computer Society, 2000.

[Hitzler *et al.*, 2004] P. Hitzler, S. Hölldobler and A. K. Seda. Logic programs and connectionist networks. *Journal of Applied Logic*, 3(2):245–272, 2004.

[Hitzler *et al.*, 2005] P. Hitzler, S. Bader and A. S. d'Avila Garcez. Ontology learning as a use case for artificial intelligence. In A. S. d'Avila Garcez, J. Elman and P. Hitzler, editors, *Proceedings of the IJCAI-05 workshop on Neural-Symbolic Learning and Reasonin, NeSy'05, Edinburgh, UK, August 2005*, 2005.

[Hitzler, 2004] P. Hitzler. Corollaries on the fixpoint completion: studying the stable semantics by means of the clark completion. In D. Seipel, M. Hanus, U. Geske and O. Bartenstein, editors, *Proceedings of the 15th International Conference on Applications of Declarative Programming and Knowledge Management and the 18th Workshop on Logic Programming, Potsdam, Germany, March 4-6, 2004*, volume 327 of *Technichal Report*, pages 13–27. Bayerische Julius-Maximilians-Universität Würzburg, Institut für Informatik, 2004.

[Hölldobler and Kalinke, 1994] S. Hölldobler and Y. Kalinke. Towards a massively parallel computational model for logic programming. In *Proceedings ECAI94 Workshop on Combining Symbolic and Connectionist Processing*, pages 68–77. ECCAI, 1994.

[Hölldobler and Kurfess, 1992] S. Hölldobler and F. Kurfess. CHCL – A connectionist inference system. In B. Fronhöfer and G. Wrightson, editors, *Parallelization in Inference Systems*, pages 318 – 342. Springer, LNAI *590*, 1992.

[Hölldobler *et al.*, 1997] S. Hölldobler, Y. Kalinke and H. Lehmann. Designing a counter: Another case study of dynamics and activation landscapes in recurrent networks. In *Proceedings of the KI97: Advances in Artificial Intelligence*, volume 1303 of *LNAI*, pages 313–324. Springer, 1997.

[Hölldobler *et al.*, 1999a] S. Hölldobler, Y. Kalinke and H.-P. Störr. Approximating the semantics of logic programs by recurrent neural networks. *Applied Intelligence*, 11:45–58, 1999.

[Hölldobler *et al.*, 1999b] S. Hölldobler, Y. Kalinke and J. Wunderlich. A recursive neural network for reflexive reasoning. In S. Wermter and R. Sun, editors, *Hybrid Neural Systems*. Springer, Berlin, 1999.

[Hölldobler, 1990] S. Hölldobler. A structured connectionist unification algorithm. In *Proceedings of AAAI*, pages 587–593, 1990.

[Hölldobler, 1993] S. Hölldobler. *Automated Inferencing and Connectionist Models*. Fakultät Informatik, Technische Hochschule Darmstadt, 1993. Habilitationsschrift.

[Hopcroft and Ullman, 1989] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison Wesley, 1989.

[Jacobsson, 2005] H. Jacobsson. Rule extraction from recurrent neural networks: A taxonomy and review. *Neural Computation*, 17(6):1223–1263, 2005.

[Kalinke and Lehmann, 1998] Y. Kalinke and H. Lehmann. Computations in recurrent neural networks: From counters to iterated function systems. In G. Antoniou and J. Slaney, editors, *Advanced Topics in Artificial Intelligence*, volume 1502 of *LNAI*, Berlin/Heidelberg, 1998. Proceedings of the 11th Australian Joint Conference on Artificial Intelligence (AI'98), Springer–Verlag.

[Kalinke, 1997] Y. Kalinke. Using connectionist term representation for first–order deduction – a critical view. In F. Maire, R. Hayward and J. Diederich, editors, *Connectionist Systems for Knowledge Representation Deduction*. Queensland University of Technology, 1997. CADE–14 Workshop, Townsville, Australia.

[Kleene, 1956] S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, volume 34 of *Annals of Mathematics Studies*, pages 3–41. Princeton University Press, Princeton, NJ, 1956.

[Kwasny and Kalman, 1995] S. Kwasny and B. Kalman. Tail-recursive distributed representations and simple recurrent networks. *Connection Science*, 7:61–80, 1995.

[Lehmann *et al.*, 2005] J. Lehmann, S. Bader and P. Hitzler. Extracting reduced logic programs from artificial neural networks. In A. S. d'Avila Garcez, J. Elman and P. Hitzler, editors, *Proceedings of the IJCAI-05 Workshop on Neural-Symbolic Learning and Reasoning, NeSy'05, Edinburgh, UK*, 2005.

[Lloyd, 1988] J. W. Lloyd. *Foundations of Logic Programming*. Springer, Berlin, 1988.

[Maass and Markram, 2004] W. Maass and H. Markram. On the computational power of recurrent circuits of spiking neurons. *Journal of Computer and System Sciences*, 69(4):593–616, 2004.

[Maass *et al.*, 2005] W. Maass, T. Natschläger and H. Markram. On the computational power of circuits of spiking neurons. *J. of Physiology (Paris)*, 2005. in press.

[Maass, 2002] W. Maass. Paradigms for computing with spiking neurons. In J. L. van Hemmen, J. D. Cowan and E. Domany, editors, *Models of Neural Networks*, volume 4 of *Early Vision and Attention*, chapter 9, pages 373–402. Springer, 2002.

[McCulloch and Pitts, 1943] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[McGarry *et al.*, 1999] K. J. McGarry, J. Tait, S. Wermter and J. MacIntyre. Rule-extraction from radial basis function networks. In *Ninth International Conference on Artificial Neural Networks (ICANN'99)*, volume 2, pages 613–618, Edinburgh, UK, 1999.

[Natschläger and Maass, 2002] T. Natschläger and W. Maass. Spiking neurons and the induction of finite state machines. *Theoretical Computer Science: Special Issue on Natural Computing*, 287:251–265, 2002.

[Niklasson and Linåker, 2000] Lars Niklasson and Fredrik Linåker. Distributed representations for extended syntactic transformation. *Connection Science*, 12(3–4):299–314, 2000.

[Pinkas, 1991a] G. Pinkas. Propositional non-monotonic reasoning and inconsistency in symmetrical neural networks. In *IJCAI*, pages 525–530, 1991.

[Pinkas, 1991b] G. Pinkas. Symmetric neural networks and logic satisfiability. *Neural Computation*, 3:282–291, 1991.

[Plate, 1991] T. A. Plate. Holographic Reduced Representations: Convolution algebra for compositional distributed representations. In J. Mylopoulos and R. Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sydney, Australia, August 1991*, pages 30–35, San Mateo, CA, 1991. Morgan Kauffman.

[Plate, 1995] T. A. Plate. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641, May 1995.

[Pollack, 1990] J. B. Pollack. Recursive distributed representations. *AIJ*, 46:77–105, 1990.

[Rodriguez, 1999] P. Rodriguez. A recurrent neural network that learns to count. *Connection Science*, 11(1):5–40, 1999.

[Rumelhart *et al.*, 1986] D. E. Rumelhart, G. E. Hinton and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland and the PDP Research Group, editors, *Parallel Distributed Processing, vol. 1: Foundations*, pages 318–362. MIT Press, 1986.

[Russell and Norvig, 2003] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2 edition, 2003.

[Seda and Lane, 2005] A. K. Seda and M. Lane. On approximation in the integration of connectionist and logic-based systems. In *Proceedings of the Third International Conference on Information (Information'04)*, pages 297–300, Tokyo, November 2005. International Information Institute.

[Seda, 2005] A. K. Seda. On the integration of connectionist and logic-based systems. In T. Hurley, M. Mac an Airchinnigh, M. Schellekens, A. K. Seda and G. Strong, editors, *Proceedings of MFCSIT2004, Trinity College Dublin, July 2004*, Electronic Notes in Theoretical Computer Science, pages 1–24. Elsevier, 2005.

[Shastri and Ajjanagadde, 1993] L. Shastri and V. Ajjanagadde. From associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioural and Brain Sciences*, 16(3):417–494, September 1993.

[Shastri and Wendelken, 1999] L. Shastri and C. Wendelken. Soft computing in SHRUTI: — A neurally plausible model of reflexive reasoning and relational information processing. In *Proceedings of the Third International Symposium on Soft Computing, Genova, Italy*, pages 741–747, June 1999.

[Shastri and Wendelken, 2003] L. Shastri and C. Wendelken. Learning structured representations. *Neurocomputing*, 52–54:363–370, 2003.

[Shastri, 1999] L. Shastri. Advances in Shruti — A neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony. *Applied Intelligence*, 11:78–108, 1999.

[Shastri, 2002] L. Shastri. Episodic memory and cortico-hippocampal interactions. *Trends in Cognitive Sciences*, 6:162–168, 2002.

[Sougne, 2001] J. P. Sougne. Binding and multiple instantiation in a distributed network of spiking nodes. *Connection Science*, 13(1):99–126, 2001.

[Sperduti *et al.*, 1995] A. Sperduti, A. Starita and C. Goller. Learning distributed representations for the classifications of terms. In *Proceedings of the 14th International Joint Conference on AI, IIJCAI-95*, pages 509–517. Morgan Kaufmann, 1995.

[Sperduti *et al.*, 1997] A. Sperduti, A. Starita and C. Goller. Distributed representations for terms in hybrid reasoning systems. In Ron Sun and Frédéric Alexandre, editors, *Connectionist Symbolic Integration*, chapter 18, pages 329–344. Lawrence Erlbaum Associates, 1997.

[Sperduti, 1994a] A. Sperduti. Labeling RAAM. *Connection Science*, 6(4):429–459, 1994.

[Sperduti, 1994b] A. Sperduti. Encoding labeled graphs by labeling RAAM. In Jack D. Cowan, G. Tesauro and J. Alspector, editors, *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pages 1125–1132. Morgan Kaufmann, 1994.

[Stolcke and Wu, 1992] A. Stolcke and D. Wu. Tree matching with recursive distributed representations. Technical Report tr-92-025, ICSI, Berkeley, 1992.

[Sun, 2001] R. Sun. Hybrid systems and connectionist implementationalism. In *Encyclopedia of Cognitive Science*. MacMillan Publishing Company, 2001.

[Tesauro, 1995] G. Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3), March 1995.

[Towell and Shavlik, 1993] G. G. Towell and J. W. Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13:71–101, 1993.

[Towell and Shavlik, 1994] G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1–2):119–165, 1994.

[Valiant, 2003] L. G. Valiant. Three problems in computer science. *Journal of the ACM*, 50(1):96–99, 2003.

[van der Velde and de Kamps, 2005] F. van der Velde and M. de Kamps. Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences*, 2005. to appear.

[Wendelken and Shastri, 2003] C. Wendelken and L. Shastri. Acquisition of concepts and causal rules in shruti. In *Proceedings of Cognitive Science, Boston, MA, August 2003*, 2003.

[Wendelken and Shastri, 2004] C. Wendelken and L. Shastri. Multiple instantiation and rule mediation in shruti. *Connection Science*, 16:211–217, 2004.