On relations between neural networks, symbols and logic in machine learning, philosophy, and artificial intelligence

TERM PAPER, LOGIC AND COGNITIVE SCIENCE II

LECTURER *Karen Kwast*

Faculty of Science

University of Amsterdam

by Mick de Neeve

EMAIL mneeve@science.uva.nl

STUDENT NR 9305017

October 31, 2000

Keywords

Machine learning, philosophy, artificial intelligence, neural networks, symbolic logic, subsymbolic computation, dynamic logic, nonmonotonic logic, interdisciplinary research

Abstract

Three papers from the literature of machine learning, philosophy, and artificial intelligence (respectively) are reviewed as primary literature - each of which links neural networks to symbolic logic in a way considered typical of the respective fields. The merits and shortcomings of the methods in eliciting relations between symbolic systems and connectionist systems are discussed as well as the their success in shedding light on the notion of subsymbol (some general criticisms are also given). Indications are given as to what kind of cross-fertilization between the approaches might be useful. Also discussed are the logical properties of dynamicism and nonmonotony which are (more or less explicitly) suggested by the methods and may be required in general by symbolic formalism relating to neural networks, for the purpose of highlighting that nonclassical logical properties appear to be required of a neural logic. Dynamic logic is argued for as a general 'neuro-logical' formalism, as well as strong(er) interdiciplinary research links between the mentioned fields, such that the gap between subsymbolic and symbolic computation may be bridged from multiple sides.

1 Introduction

This work is concerned with the problem of characterizing the relationship between symbolic and connectionist computation. Early artificial intelligence (AI) systems tended to be implemented as rule-based reasoning systems, with the symbols being reasoned with explicit in the programs. With the advent of subsymbolic techniques such as neural networks [1] this transparency is lost - we see merely neurons and activations in the implementation. Or in the words of (Mozer and Smolensky [8]):

> ... one thing that connectionist networks have in common with brains is that if you open them up and peer inside, all you can see is a big pile of goo.

The idea of connectionist networks was indeed to have something in common with brains, but this is a bit of self-irony on behalf of the authors who are in fact proponents of connectionism. However, the problem is a serious one, as witnessed by the extreme position of (McClosky [7]) who argues that, in the absense of a way to interpret connectionist networks,

> ... connectionist networks should not be viewed as theories of human cognitive functions, or as simulations of theories or even as demonstrations of specific theoretical points.

Interpretation means to assign semantic content, but the only semantic content apperantly assignable at the level of neuronal actions relates to the simple arithmetic of those actions. However, it is important to be able to assign a semantic content at a higher level, to actions perceived above the level of individual neurons - ie. patterns of activation among groups of neurons, which we would like to call symbols - and be able to use these symbols to describe the rationales of these actions, to be able to reason about their suitability in similar contexts, to critisize them, or to compare them to other (eg. symbolic) systems.

This paper reviews three methods for relating the neuronal to the symbolic, the first of which has semantic content as relating to the (higher-level) problem domain assigned at the neural level; the other two attempt to elicit higher-level symbolic properties from this neural base-level. All three approaches study neural networks working in domains that are symbolic in nature, this being a good starting point for relating the neural actions back to a well-understood logic domain [2]. Naturally, a review of a mere three papers is very incomplete if this were to serve as an overview of research into the nature of the relationship between neurons and symbols, but in my opinion, the first approach is typical of the research spirit in the field of machine learning, the second is typical of theoretical philosophy, and the third of artificial intelligence, which somewhat justifies such meagre primary literature.

The objective here shall firstly be to give an overview of the three approaches and criticize them, and give suggestions concerning how the one approach might learn from the other (and why this is necessary). The approaches are decribed in Section 2 and discussed in Section 3, which serve to fuel the second objective, in the form of further discussion in Section 4 concerning the necessity of incorporating nonclassical properties into a symbolic system that is to properly model neural networks. This suggest the necessity of strong(er) links between researchers in the respective fields.

[1] The terms neural and connectionist networks shall be used interchangably.

[2] The second example is sort of an exception since it consists in analyzing the activities of (certain) neural networks as sharing characteristics with a not terribly well-understood logic - ie. nonmonotonic logic.

2 Approaches to neural networks and symbols

This section serves the purpose of describing the research in the primary literature of this paper.

2.1 KBANN nets

(Shavlik and Towell [12]) describe a system called that works by translating a theory of propositional rules into a neural network referred to as a KBANN, for Knowledge-Based Artificial Neural Network. For example, the PROLOG-like rule set of Table 1 would be translated into the network of Figure 1. The example is due to (D. Opitz [9]).

```
a :- b, not c.
b :- d, f, g.
b :- d, not f, i.
c :- h, j, k.
```

Table 1: PROLOG domain theory for membership in category a

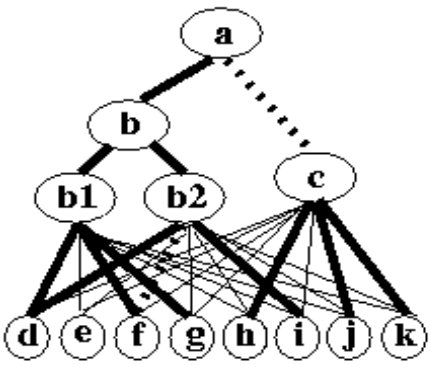

Figure 1: Network built from PROLOG-rules in Table 1

The nodes in Figure 1 that represent disjuncts get an activation near 1 just in case an antecedent with a high weight is true, conjuncts must have all such antecedents true. The b_1 and b_2 nodes represent the 2 disjunctive rules defining b. Thick lines represent near 1 activations, thin lines near 0 ones, which represent rules to which new antecedents may be added during training. Dotted lines represent negative values for falsity. The rules from Table 1 are initially translated individually into subnetworks that reproduce the behaviour of the single rule, after which these are assembled to get the behaviour of the whole rule set. See (Shavlik and Towell [12]) for details.

Symbolic rule extraction methods (such as SUBSET or NOFM) that are put to work after a KBANN net has been trained, refine the original rule and effectively reduce the network to a binary network in that activation values are interpreted as being either active or inactive (cf. true or false). They do so by searching for input values to a unit that result in a near 1 value. This can be achieved by simply searching for ways in which the weighted sum of the inputs exceeds the bias. For example, the SUBSET method searches for a subset of the links into a unit such that the sum of the weights

of the links in that set guarantees that the total input exceed the bias. The goal of the method is to return a new set of symbolic rules that are refinements of the original rules and do a better classification job.

The SUBSET method would extract the rules in Table 2 from the network in Figure 2.

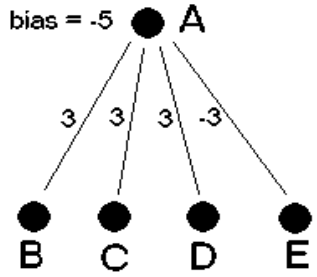

Figure 2: Network for SUBSET rule extraction

```
a :- b, c, not e.
a :- b, d, not e.
a :- c, d, not e.
a :- b, c, d.
```

Table 2: Rules extracted by SUBSET

The idea of rule extraction from KBANN-nets is to circumvent the problem of interpreting arbitrarily configured neural networks (ie. the real issue is simply avoided altogether here) by configuring it according to a symbolic description of the problem at hand. The KBANN-networks can be seen to have *ultra-local* representations, ie. the nodes stand for propositions. Given the fact that such networks are far from typical among connectionist models and certainly not to be taken as definitive in connectionism (Smolensky [13], p. 170), this may not be very interesting, since most neural networks might be said to have *distributed* representations. Nevertheless, it is a start in the linking of the approaches of symbolic and neural computation. Note that this is not meant contructively, as local representations are quite irrelevant to distributed ones (Smolensky [14], p. 234) [3].

2.2 Nonmonotonic inferences

(Balkenius and Gärdenfors [1]) interpret the activities of neural networks as nonmonotonic inferences between the input and the output in a high level description of neural network properties - abstracting from activation values of individual neurons - and show that a certain class of neural networks (self-organizing Hopfield networks) generate cumulative nonmonotonic inferences, see (Gärdenfors and Makinson [4]).

[3] It is possible to conceive of local representations as being relevant to distributed ones with reference to the recursivity of (backpropagation) networks, but that would assume that neurons which partake in a concept's representation to be grouped together by synaptic links, disabling the possibility of a concept being represented 'all over' a network, thus seriously restricting possible meanings of 'subsymbol'.

In developing a higher-level description of a neural network, the approach differs from the other two discussed here. Instead of giving a neural network a logic-related problem and analysing its output, the starting point here is the development of the higher-level description. I shall outline the construction here, see (Balkenius and Gärdenfors [1]) for further details.

A neural network is described as a 4-tuple $< S, F, C, G >$, where S is the state space of the network, F is a set of activation functions relative to a given configuration, C is the set of configurations in terms of the connections between the neurons and G gives the set of learning functions that describe how configurations alter upon various inputs (however, C and G are assumed constant in the discussion). $S = [a, b]^n$, with $[a, b]$ the activation range of a neuron and n the number of neurons. A state in S is a vector in $[a, b]^n$.

To link the subsymbolic and symbolic levels, *schemata* representation is introduced. This slot-and-filler structure is very general, cropping up in many (classical) theories of cognition. A schema α corresponds to a vector $< \alpha_1, \ldots, \alpha_n > \in S$ with $0 \leq i \leq 1$ for all α_i, thus in line with (Smolensky [13]). Schemas can be partially ordered along an axis of informational content: $\alpha \geq \beta$ iff $\alpha_i \geq \beta_i$ for all $1 \leq i \leq n$. The idea behind schema representation is that the informational content of the activity vector of the neural network in which a schema is represented is at least that of the schema itself, ie. to require that the represented schema is more general than that activity vector - which means that the activity vector is an *instantiation* of the schema that it represents. A schema is said to be represented in a network if $x_i \geq \alpha_i$ for all $1 \geq n$.

A network is said to perform an inference when it goes into a *resonant* state upon some input. Given a state $x \in S$, $f^0(x)$ or $f(x)$ denotes the resulting state after a single cycle; $f^{n+1} = f(f^n(x))$. The limit $\lim_{n\to\infty} f^n(x)$ (relating to the asymptotic stability below) is written as $[x]$ and is called the resonance function. The resonant state $[\alpha]$ is interpreted as containing information on what a neural network expects to hold when given α as input, i.e. the resonance function fills in the default assumptions about the environment. A network goes into a resonant state when the following holds:

1. Equilibrium: The state is unaltered by the state transition function;
2. Stability: The output is the same every time within reasonable limits when the same input is repeatedly presented an equal number of cycles;
3. Asymptotic stability: The network remains stable.

For a schema α to be 'clamped' in a network α must be included in a resonant state, meaning the activity levels of neurons must be above α_i for all neurons i. The state transition function is modified to include α: $f_\alpha(x) = f(x) \bullet \alpha$ for all $x \in S$. The '$\bullet$' operator is the conjunction of two schemas. For two schemas α and β, $\alpha \bullet \beta =< max(\alpha_1, \beta_1), \ldots, max(\alpha_n, \beta_n) >$. The resonance function with respect to a schema α is given as $x_c^\alpha = \lim_{n\to\infty} f^n(x)$. The inference relation between schemata looks as follows:

Definition 2.2.1 *Let α and β be schemata.*

$$\alpha \mid\sim \beta \textit{ iff } [\alpha]^\alpha \geq \beta$$

The network of Figure 3 has excitory links (arrows) and inhibitory ones (dots). Using binary activity values, let $\alpha =< 1, 1, 0, 0 >$, $\beta =< 0, 0, 0, 1 >$ and $\gamma =< 0, 1, 1, 0 >$. It

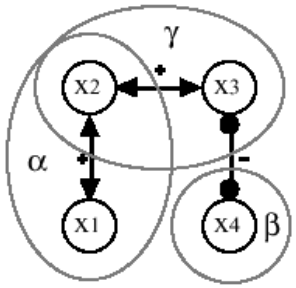

Figure 3: Nonmonotonic inference $\alpha \cdot \beta \mid\sim \gamma$

is assumed x_4 inhibits x_3 more than x_2 excites x_3. With α as input, γ becomes active; $\alpha \mid\sim \gamma$. With $\alpha \bullet \beta$ as input, x_3 inhibited by x_4 causes γ to be withdrawn; $\alpha \bullet \beta \not\mid\sim \gamma$.

The authors then claim the relation of Definition 2.2.1 to satisfy 'certain' postulates of cumulative reasoning, namely Cut (if $\alpha \mid\sim \beta$ and $\alpha \bullet \beta \mid\sim \gamma$ then $\alpha \mid\sim \gamma$), Cautious Monotony (if $\alpha \mid\sim \beta$ and $\alpha \mid\sim \gamma$ then $\alpha \bullet \beta \mid\sim \gamma$), and Cumulativity (if $\alpha \mid\sim \beta$ and $\beta \vdash \alpha$ then $\alpha \mid\sim \gamma$ iff $\beta \mid\sim \gamma$); besides two of the basic inference relation postulates (Supraclassicality and And - see Section 3.1.2 for more on this).

2.3 Wire-tapping hidden units

The neural network analysis method of (Berkely *et al.* [11]) analyses the activation levels of the hidden units of a network, seeking the groups of units that cooperate in a computation to make distributed representations explicit. It does not build a network from a symbolic version of the problem at hand; the network can be built and trained by trial and error 'as usual'. The approach starts out with a neural net without an underlying symbolic theory, translates from unit activations to a symbolic description of their function in solving a problem, by using statistical analysis of an arbitrarily configured network. These networks are the most 'usual' out of the three types discussed here, the only difference is that they work with Gaussian activation functions rather than the more usual sigmoidal one. The merits of these so-called value unit networks include faster learning of linearly inseperable classes and better generalization, but more importantly using the Gaussian function allowed a method of interpretation which did not work with a sigmoid.

As mentioned, there is no symbolic domain theory to start with, but a trained neural network that can perform some classification task - a standard job for neural networks. The object in the present discussion of hidden unit analysis is to determine the rules in symbolic form that the trained neural network uses and to analyze the roles of the hidden units in the task. To accomplish this, Berkely *et al.* trained a network on a logic problem of determining validity and problem type of inferences, the rationale being that to understand the relationship between neurons and symbols, it is a good idea to train a neural network to do some typically symbolic task. See (Berkely *et al.* [11]) for the details of the logic domain.

The interpretation method comprises the following steps:

1. Configurate and train a network on the problem at hand until it converges.
2. Present the training set again without backpropagation (ie. fixed weights), recording the activation level of each hidden unit for each example.

3. Make density plots for each hidden unit that represent the level of network activity for each example.

The density plots are jittered plots, with a horizontal scale representing the range of possible activations of a given unit, representing each example for a unit by a single dot. A random jittering is added so the dots will not overlap. See Figure 4.

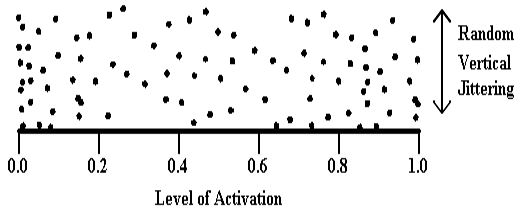

Figure 4: Example of jittered density plot

Berkely *et al.* found the density plots to be highly structured, revealing distinct bands of activation, as seen in Figure 5. The letters are names for the states the unit may be in.

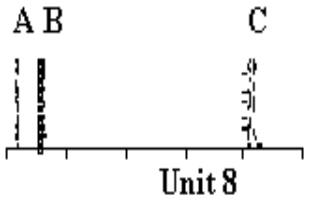

Figure 5: Density plot revealing activation bands

The banding can be used to interpret the kinds of features to which a given hidden unit tends to respond. Each example within such a band is characterized by some feature or set of features which Berkely *et al.* could identify using simple statistics (such as calculating mean, standard deviations and correlations) and define them as *definite features*, to be associated with the bands in the interpretation process. For example, a unary definite feature is an input value that is constant for all examples within the band, and a binary definite feature is a positive or negative correlation between pairs of binary features, ie. two input values are always equal or always opposite. Berkely *et al.*'s analysis of the three distinct bands in the unit of Figure 5 had them conclude that its role in the logic task was to detect logical connectives, since depending on the connective presented in the input (three posibilities) it would go into the corresponding state (A, B or C) of Figure 5, leading the authors to consider these *subsymbolic* states in the sense of (again) (Smolenksy [15]). The authors list the network's algorithm for validity determination in symbolic form in their paper, and note the network's rules and the traditional rules are close to identical to one another.

Another classification task that was analyzed in this manner is described in (Dawson and Medler [2]). This paper describes how a neural network is taught a default hierarchy for deciding whether or not a mushroom is edible based on its properties. It was

found that mushrooms that fell into a certain band had passed the first few stages of the decission procedure. The concept to be learned was 'edible'; mushrooms falling into a band had in a sense been gathered into a hidden unit that was seen to generate strong activity, and were awaiting further testing, so to speak. The corresponding subsymbol could be paraphrased as 'these mushrooms are not yet known to be inedible'; leaving it to another subsymbol to figure out more about the mushroom's edibility (or make a final decission to culminate the subsymbols into the 'edible' symbol or its negation). Other units generated strong signals in the presence of mushrooms that were certainly poisonous. The analysis again enabled the authors to list the complete decission algorithm of the network in their paper, which consist of a number of test steps at which a decission 'edible' or 'inedible' can be made, or the decission can be postponed until after further testing in a later step, with the steps corresponding to the activities of subsymbols.

3 Discussion of the approaches

This section serves to discuss the merits and shortcomings of the above approaches with respect to the understanding of the relationship between neurons and symbols, as well as some more general criticisms concerning the research. I then discuss in which ways the approaches might influence one another.

3.1 Critique

3.1.1 KBANN

The KBANN method is an example of a hybrid system. It has a symbolic component by virtue of the presence of a symbolic domain theory, which is then refined by a neural component. KBANN's neural component could in principle be replaced by a non-neural refinement component. So the KBANN approach mimics the behaviour of a simple (propositional) symbolic domain theory by a neural network. The method is not so much a method for understanding the relationship between neural networks and symbols in general but more for improving and simplifying bodies of propositional knowledge. It contributes, in the words of the authors, "to the understanding of how symbolic and connectionist approaches to artificial intelligence can profitably be integrated" - which is (noting the word 'profitably') almost more of a practical instead of a theoretical interest - rather than to the understanding of the relationship between the two approaches. KBANNs are, in a sense, not 'genuine' neural networks - so for the purposes of this paper, the implications of the KBANN research does not carry very far, except that it is an example of networks with ultra-local representations that implement classical logic.

3.1.2 Nonmonotonic inferences

The Balkenius and Gärdenfors approach takes the development a high level description of connectionist nets as its starting point and relates it to nonmonotonic logic. I merit the approach for taking this higher-order logic approach but feel it has some shortcomings in the presentation of the research. For instance, even though the paper explicitly states to exemplify the relationship between subsymbolic and symbolic computation, the concept of subsymbol is not defined anywhere. It appears as if a subsymbol could be a unit, ie. a schema could be distributed over multiple units, but this only holds in the case of binary activations. With real-valued activations of units, the approach effectively collapses to ultra-local representations; schemata can no longer be sets of active neurons. In other words, in studying 'real' neural networks

as usually encountered, one would be little further than with KBANN - as KBANN too, uses binary values too with its 'near 1' and 'near 0' activations. One might object and say that schemas are more than the simple KBANN propositions, but although Balkenius and Gärdenfors claim schemata are "officially not propositions" but treated as such in the definition of $\mid\sim$, that would be of little help in the analysis of a more common neural network.

Also bothersome is the failure of Balkenius and Gärdenfors to show that $\mid\sim$ is a proper cumulative inference relation. Granted, Cut and Cautious Monotony are shown to hold; which imply Cumulativity, but only two of the basic postulates for making a relation between propositions an inference relation at all are shown; Supraclassicality and And, missing are Left Logical Equivalence and Right Weakening. In other words, we may not even be dealing with an inference relation proper here. It is not clear whether the authors were not able to define implication, which is used in the forgotten postulates but not in the treated ones - all postulates treated make use of $\vdash$, $\mid\sim$ and $\bullet$ [4]. I shall not attempt to show the missing postulates hold (that is not my intention here) but remark that possibly, the authors have over-attempted to make one theory (connectionsm) fit another (cumulative logic) and leave the issue for what it is, returning to more general issues of nonmonotony in Section 4.2.

3.1.3 Wire-tapping hidden units

I take interest in this particular method because of its empirical approach to the connectionist interpretation problem and because it is the only method that appears to have anything really sensible to say about distributed representations and subsymbolic computation. The KBANN approach has nothing at all to say about this; Balkenius and Gärdenfors' approach does to some extent, but then only when the network's activation values are binary. Moreover, the wire-tapping research is the most fundamental and urgent, in that further research could seriously benefit from it, since it addresses the problems of interpretation using networks that are the most similar to the common backpropagation networks with logistic activation function.

3.2 Possibilities of cross-fertilization

3.2.1 Wire-tapping and higher-level logic

I have stated that I merit the wire-tapping approach because it is emperical and works with networks that are closest to the most commonly used ones, and the Balkenius-Gärdenfors approach (despite its apperant logical flaws) because it develops a higher-level logic as its starting point [5]. The question now is how the one might benefit from the other; the idea being that in bridging the gap between the connectionist and symbolic approaches the construction of the bridge must be started on both sides of the trench. Let me note here that both these approaches discussed should be viewed as being examples of a certain spirit of doing research, the implication is not necessarily that precisely these two specific approaches should be welded together. Nevertheless, considering the path I have taken here so far, that is a suitable form of discussing this issue.

It seems clear that the relationship between connectionism and symbolic logic requires a thorough understanding of the notion of subsymbol. It is possible to conceive of this notion in a variety of ways, such as a group of connected neurons, or a group of

[4] Disjunction and negation are defined and used in some examples not relating to the postulates but cannot be used for implication because negation is taken to be schema complement.

[5] Balkenius and Gärdenfors are in fact also emperical in that they actually ran real networks.

arbitrary neurons (one of which - it is unclear which one - is suggested by Balkenius and Gärdenfors), or the spread of activation across such an arbitrary group of neurons without any geographical specification in the network (ie. the role of individual neurons may alter), or a set of states through which a network passes in the solving of a problem, as suggested in Berkely *et al.*, or perhaps even the set of differential equations governing the evolution of a set of activation vectors, as might be understood from (Smolensky [13]).

As it has proven remarkably difficult to interpret the output of arbitrary neural networks, it is not to be expected that we can figure out the nature of subsymbols by reason alone - and I believe that what one takes to be a subsymbol cannot help but influence a logic one might device of connectionism. This is the reason it is important to bring to bear emperical research of neural networks, as Berkely *et al.* have done. This is not to say that if only Balkenius and Gärdenfors had used Berkely *et al.* their work would have fixed all the postulates of cumulative reasoning, but they might possibly have looked to a different directions as well for the higher-logic that might fit (eg. dynamic logic, see Section 4.1).

Conversely, the existence of logical formalisms that have been devised through reason can assist the search for meaningful interpretations of neural networks since it may restrict the search for possible interpretations of the result of empirical research such as performed by Berkely *et al.*. As it stands, this emperical research has made use of statistical analysis and has managed to say something about the nature of subsymbols. However, it has not attempted to wed these results with the body of knowledge concerning symbolic formalisms, which might cater for a genuine understanding of how what we know about subsymbols relates to what we know concerning symbols.

3.2.2 KBANN

Finally, after downplaying the use of KBANNs like I have done so far, it is difficult to state any useful purpose for them in relation to the other approaches. However, the rule extraction methods associated with KBANNs refine rule sets, and it may be possible to interpret certain networks used for purposes outside the KBANN domain (such as the ones used in the wire-tapping approach) as propositional just to be able to arrive at more concise networks (this would make the rule-searching more computationally expensive though, because these nets are not binary). If a neural network were to serve as a model for the task that it performed, it would after all be preferable to have the dimensions of the model as small as possible - in accordance with Occam's razor - without diminishing its performance. But there are more tailored ways of reducing network size than this (eg. Mozer and Smolensky ([8]) or (Opitz [9])) - which would probably be viewed as unscholarly hacking. Let me leave the issue with the remark that I could not resist the attempt of giving KBANNs a place in this section.

4 Suggested properties of neural logics

Here, I shall discuss two (related) properties that appear to be required of logical systems that are to serve as logics of neural network computation. The inspiration comes from the papers reviewed above, but some more general remarks shall also be made.

4.1 Dynamicism

As indicated in Section 1, it is desirable to have symbols associated with neural networks to be able to reason about their suitability in similar contexts and to compare

them to other types of systems capable of performing similar tasks. Viewing neural networks as (self-organizing) programs, it would be useful to formulate *program logics* of them, which are dynamic logics after (Harel [5]), and model the dynamic aspects of algorithmic states. In a dynamic logic, the truth of a statement depends not only on the present state, but also on its relationship with other states the algorithm might enter into. Statements such as AFTER(P, F) are expressible in such logics, meaning that after executing a (sub) program P, some statement S will be the case.

The AFTER construct might be applied at various levels. Its first argument could be the neural network algorithm as programmed in (say) the context of the UNIX environment. At this level the dynamic logic may be used to reason about the suitability of a neural network as compared to a symbolic induction algorithm for some classification task. But one might also take it to be an input pattern and view the neural network itself as the environment in which this 'program' is embedded. It is then possible to express statements relating to giving input patterns to the network, such as AFTER(I, C), meaning that input I leads to a classification of I in class C. The second argument of AFTER then relates to a state of the entire network. It is also possible to let the second argument relate to state changes in a unit, such that the analysis of Berkely *et al.* as well as Dawson and Medler might be served. Relating to their network analysis, we might formalize, for example, state changes in each hidden unit using constructs such as AFTER(I, A) meaning that input I leads the unit to go into subsymbolic state A, when I contains the logical connective associated with substate A, which may be helpful in the devision of principled methods of symbolic algorithm extraction from neural networks, or one might use constructs from dynamic logic to formulate and prove statements concerning the possibility of a hidden unit being able to enter into a certain subsymbolic state upon certain input, given some network configuration. It could then be the case that one could prove that a certain type of network configuration were necessary to achieve the desired state - ie. dynamic logic may assist the goals of low-level network configuration and high-level aims as formulated symbolically, to go hand in hand.

4.2 Nonmonotony

Nonmonotony, and specifically cumulative reasoning, is explicitly researched as a property of connectionist logics by Balkenius and Gärdenfors. However, nonmonotony is probably a more general property of possible logics of connectionism, the reason being that neural networks are primarily *learning systems*, and learning arguably proceeds in incremental steps, ie. knowledge about some domain or learning of a concept does not happen in one go, but instead the learning agent refines knowledge in a stepwise fashion. It suffices to consider any incremental learning system, such as the symbolic induction algorithms from the literature of machine learning - eg. (Langley [6]) - to see that this type of learning involves a nonmonotonic step.

To see this, consider that induction is ampliative - it typically provides more knowledge than can be deduced from examples. By its very nature it is, of course, not sound - the knowledge provided is not known for certain to be true [6] : Suppose now some incremental classification algorithm - symbolic or neural - has been run over a set of examples Δ, leading it to classify a new instance as belonging to some class C_1. Rerunning the algorithm over a set $\Delta' \supset \Delta$ may very well classify the same instance as belonging to a different class C_2. In other words, the classification into C_1 is defeasible, i.e. nonmonotonic - the set of 'beliefs' has grown, but not the set of conclusions that may be drawn from it; after all, the classification into C_1 has had to be withdrawn. We can also call what is going on here 'default reasoning' in the sense of (Reiter [10]),

[6] 'Information' might therefore philosophically be a more appropriate term

since in the absence of information to the contrary, ie. new information, the default class remains C_1.

Another way of looking at nonmonotony in learning systems is in relation to dynamicism as discussed above, using the second level of applying the AFTER construct, ie. the first argument is an input pattern. For instance, reasoning from identical initial states, AFTER(I, C) may be true, but AFTER$(I; J, C)$ may be false, where ';' stands for the sequencing of inputs. This in fact an alternative way of putting the example of the previous paragraph; (the 'incremental nonmonotony'): we can say AFTER(Δ, C_1) is true and AFTER(Δ', C_1) is false. Balkenius and Gärdenfors suggest a 'non-incremental nonmonotony' (remember their learning functions play no significant role) - this can also be expressed. If we equate ';' with '$\bullet$' we get an alternative way of expressing what happens in Figure 3; AFTER(α, γ) is true, and AFTER$(\alpha; \beta, \gamma)$ is false. From a pragmatic point of view, considering the fact that research into dynamic logics currently outnumbers research into non-monotonic logics, perhaps this form is more suitable for expressing and modeling these phenomena than the one used by Balkenius and Gärdenfors. From a theoretical point of view, dynamic logic may be the more suitable candidate if it is powerful is great enough to express the nonmonotonic phenomena under their scrutiny, and the above loosely suggests this may be so (though such a claim must be separately researched). If so, lower and higher level neural network phenomena might be expressible in the same formalism.

4.3 Nonclassical logic

The above indicates that it is probably not a good idea to search for connections between neural and symbolic computation exclusively in the realm of traditional classical logic, which would invalidate the points made by (Fodor and Pylyshyn [3]) that connectionism is 'mere implementation' of classical cognitive architecture [7] (incidentally, the view of connectionist architecture of these authors is not unlike the KBANN sort of networks), and strengthens the view of (Smolensky [13]) that connectionism is a *refinement* of classical architecture, since connectionism strongly suggests that manifestly nonclassical properties - as given above - appear to emerge from neural computation. This again fuels the argument given in Section 3.2.1 that interdiciplinary research is required to get the best chance at achieving a good understanding of the relationship between neurons and symbols. For instance, to provide the nonclassical logics, theoretical philosophers are required - and to provide the empirical evidence, the artificial intelligence researchers must be brought in [8].

4.3.1 Levels of logical operation

Unfortunately, there still remains some unclarity on my behalf concerning the level at which the nonmonotonic logic but particularly the dynamic logic operates in neural network analysis. I can say the following about this at this moment: The nonmonotonic logic appears to operate on the 'inferential level', ie. the high level at which a neural network makes classifications, but the dynamic logic operates at least on a process-level concerning whether and how the network may form particular (sub)solutions to serve its (larger) purpose, but I believe it also operates at a higher level.

[7] I assume their 'classical cognitive architecture' as being driven by an engine of 'classical logic' here.

[8] I have not even touched upon the question whether the *artificial* neural networks discussed here have sufficient bearing with the *biological* networks in our heads which are responsible for producing the arguments that give logicians such a hard time modelling them - ie. strictly speaking researchers in biology and neuroscience would also have to be part of the game. However, we are concerned here with *artificial* instead of *real* intelligence...

In this light, the attempt in Section 4.1 at viewing dynamic logic at two levels and the one in Section 4.2 at viewing nonmonotonic logic from a dynamic perspective arises from a desire to be able to use similar logical constructs at different levels of analysis, since it might be preferable to eventually have some sort of compositional account of emergent logic in the same formalism. Answering the question whether this can be done, let alone do it, would require much research.

4.3.2 Interpretation

As for the success of the enterprise of interpreting neural networks, the following must be said. All methods discussed here work with special types of neural networks; the problem of network interpretation in its full generality remains to be solved. Interpretation, as noted, is a semantic issue. The KBANN research is, as noted, of little explanatory value in this light, since it effectively implements a classical logic, and we have just seen that neural networks have nonclassical properties. The work of Balkenius and Gärdenfors is useful in that it highlights the possibility of beginning the interpretation from the 'logic side', and bring to bear a nonclassical system. Berkely *et al.* start from the side of the networks themselves. They come closest in my opinion to actually interpreting the networks, but the interpretation is with reference to the semantics of statistics. This is a step forward, but what is needed is a semantic interpretation with reference to the semantics related to a (higher-level) symbolic-logical formalism. I have put forward dynamic logic as a possible suitable candidate, although it is extremely difficult to get full correctness results for even parts of dynamic logics. Nevertheless, I believe the formalism required in the end must be some type of dynamic logic - with the enterprise of interpretation starting from the logic as well as from the network side, and hopefully meeting in the middle.

5 Conclusions

Reviewing three approaches to eliciting relations between neural and symbolic computation in the research spirit of three different fields, it can be seen that seen that the issue is tackled in various vains with various success at clarifying issues concerning subsymbolic computation. I have attempted to show that certain nonclassical properties of a 'symbolic logic' of neural computation must arguably hold for a neural logic.

In order to achieve the best possible understanding of the relationship between neural and symbolic computation, it is necessary to bridge the gap from both sides, ie. from the research communities of neural (eg. artificial intelligence) as well as those from symbolic computation (eg. theoretical philosophy). Researchers working in the various fields relating to research into this issue tend to use varying notions of 'logic', ranging from the classically oriented PROLOG-type propositional logic to the more intricate logics dealing with nonmonotonic aspects. Nonclassical logics appear to be related to the workings of neural networks, and since much research has gone into them (eg. in theoretical philosophy and computer science), it is arguably useful on the one hand to pursue lines of research to further clarify these relations, rather than just attempting to interpret neural networks 'from scratch' (the best candidate to pursue researching in my opinion being some type of dynamic logic). On the other hand, empirical neural network research via statistical analysis in the work of Berkely *et al* as discussed above can give some clues concerning the symbolic-logical properties that may in turn be researched from a more theoretical perspective, in other words, research into this issue must continually go both - or rather - multiple ways: Research into the relationship between neural computation and symbolic logic will have to be an interdiciplinary enterprise.

References

[1] C. Balkenius and P. Gärdenfors. Nonmonotonic inferences in neural networks. Technical report, Department of Philosophy, University of Lund, 1991.

[2] M. Dawson and D. Medler. Of mushrooms and machine learning: Identifying algorithms in a PDP network. *Canadian Artificial Intelligence*, 38, 1996.

[3] J. Fodor and Z. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. In C. MacDonald and G. MacDonald, editors, *Connectionism: Debates on Psychological Explanation, Volume Two*. Blackwell, 1995.

[4] P. Gärdenfors and D. Makinson. Nonmonotonic inferences based on expectations. In *J. Allen, R. Fikes and E. Sandewall (eds): Proceedings Second International Conference on Principles of Knowledge Representation and Reasoning*, 1989.

[5] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Volume* II. Reidel Publishing Co., 1984.

[6] P. Langley. *Elements of Machine Learning*. Morgan Kaufman, 1996.

[7] M. McCloskey. Networks and theories: The place of connectionism in cognitive science. *Psychological Science*, 2(6), 1991.

[8] M. Mozer and P. Smolensky. Using relevance to reduce network size automatically. *Connection Science*, 1, 1989.

[9] D. Opitz. Connectionist theory refinement: Genetically searching the space of network topologies. *Journal of Artificial Intelligence Research*, 6, 1997.

[10] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13, 1980.

[11] I. Berkeley M. Dawson D. Medler D. Schopflocher and L. Hornsby. Density plots of hidden unit activations reveal interpretable bands. *Connection Science*, 7(2), 1995.

[12] J. Shavlik and G. Towell. Refining symbolic knowledge using neural networks. In R. Michalski and G. Tecuci, editors, *Machine Learning - a multistrategy approach*. Morgan Kaufmann, 1994.

[13] P. Smolensky. Connectionism, constituency and the language of thought. In G. MacDonald C. MacDonald, editor, *Connectionism: Debates on Psychological Explanation, Volume Two*. Blackwell, 1995.

[14] P. Smolensky. Integrated connectionist/symbolic architecture. In C. MacDonald and G. MacDonald, editors, *Connectionism: Debates on Psychological Explanation, Volume Two*. Blackwell, 1995.

[15] P. Smolensky. On the proper treatment of connectionism. In C. MacDonald and G. MacDonald, editors, *Connectionism: Debates on Psychological Explanation, Volume Two*. Blackwell, 1995.