Artificial General Intelligence and Classical Neural Network

Pei Wang
Department of Computer and Information Sciences, Temple University
Room 1000X, Wachman Hall, 1805 N. Broad Street, Philadelphia, PA 19122
Web: http://www.cis.temple.edu/∼pwang/
Email: pei.wang@temple.edu

***Abstract*— The research goal of Artificial General Intelligence (AGI) and the notion of Classical Neural Network (CNN) are specified. With respect to the requirements of AGI, the strength and weakness of CNN are discussed, in the aspects of knowledge representation, learning process, and overall objective of the system. To resolve the issues in CNN in a general and efficient way remains a challenge to future neural network research.**

I. Artificial General Intelligence

It is widely recognized that the general research goal of Artificial Intelligence (AI) is twofold:

- As a *science*, it attempts to provide an explanation of the mechanism in the human mind-brain complex that is usually called "intelligence" (or "cognition", "thinking", etc.).
- As a *technology*, it attempts to reproduce the same mechanism in a computer system.

Therefore, a complete AI work should consist of results on three levels:[1]

1) A *theory* of intelligence, as a collection of statements in a natural language;
2) A *model* of the theory, as a set of expressions in a formal language;
3) An *implementation* of the model, as a software in a programming language, or a special-purpose hardware.

On the theoretical level, the central questions to be answered are "What is intelligence?" and "How to achieve it?". For the first question, everyone looks for the answer in the human mind-brain complex, though different people focus on its different aspects:

- **Structure**. Since the human brain is the only place where intelligence is surely observed, some people believe that the best way to achieve AI is to study and duplicate the brain structure. Examples: various brain models.
- **Behavior**. Since we often decide whether we are dealing with an intelligent system by observing its behavior, some people believe that the best way to achieve AI is to reproduce human behavior. Examples: Turing Test and various cognitive models.
- **Capability**. Since we often judge the level of intelligence of other people by evaluating their problem-solving capability, some people believe that the best way to achieve AI is to build systems that can solve hard practical problems. Examples: various expert systems.
- **Function**. Since the human mind has various cognitive functions, such as perceiving, learning, reasoning, acting, and so on, some people believe that the best way to achieve AI is to study each of these functions one by one, as certain input-output mapping. Example: various intelligent tools.
- **Principle**. Since the human mind seems to follow certain principles of information processing, some people believe that the best way to achieve AI is to let computer systems follow the same principles. Example: various definitions of rationality.

Though each of the above beliefs leads to sound scientific research, they aim at different goals, and therefore belong to different research paradigms [3], [4]. The second question ("How to achieve intelligence?") can only be answered with respect to an answer of the first question ("What is intelligence?"). Many confusions in the discussions on AI are caused by evaluating an approach within one paradigm according to the requirements of a different paradigm.

On the modeling level, there are three major frameworks for formalizing a cognitive system, coming from different traditions:

- **dynamical system**. In this framework, the states of the system are described as points in a multidimensional space, and state changes are described as trajectories in the space. It mainly comes from the tradition of physics.
- **inferential system**. In this framework, the states of the system are described as sets of beliefs the system has, and state changes are described as belief derivations and revisions according to inference rules. It mainly comes from the tradition of logic.
- **computational system**. In this framework, the states of the system are described as data stored in the internal data structures of the system, and state changes are described as data processing following algorithms. It mainly comes from the tradition of computer science.

[1] Different from similar level distinctions [1], [2], here the distinction is made according to the *language* in which the result is presented, so as to avoid the dependency on notions like "computation", "knowledge", or "algorithm".

In principle, these three frameworks are equivalent in their expressive and processing power, in the sense that a virtual machine defined in one framework can be implemented by another virtual machine defined in another framework. Even so, for a given problem, it may be easier to find solutions in one framework than in the other frameworks. Therefore, the frameworks are not always equivalent in practical applications.

On the implementational level, the central issues include efficiency and naturalness.

These three levels have strong influence to each other, though the decisions on one level usually do not completely depend on the decisions on another level. Also, the problems on one level usually cannot be solved on a different level. For example, a weakness in a theory usually cannot be made up by a clever formalization.

At the very beginning, AI researchers took "intelligence" as a whole. Though for such a complicated problem, research has to be carried out step by step, people still associated their work with the ultimate goal of the field, that is, the building of a "thinking machine". This attitude was exemplified by the theoretical discussion of Turing Test [5] and the system-building practice of General Problem Solver [6].

However, the previous attempts to build general-purpose intelligent systems all failed. No matter what the reason in each case actually was, the lesson learned by the AI community is that general-purpose intelligent systems cannot be built, at least at the current time. As a consequence, people turn to domain-specific or narrowly-defined problems, which are obviously more manageable.

Even so, the ultimate problem of AI remains there, and there are still a very small number of projects aiming at "general-purpose thinking machines", though each based on a different understanding and interpretation of the goal. In recent years, the term "Artificial General Intelligence" (AGI) was used by some researchers to distinguish their work from the mainstream "narrow/special AI" research [7]. Intuitively, this term is similar to terms like "Thinking Machine" [5], "Strong AI" [8], or "Human-level AI" [9], [10], though the stress is on the general-purpose nature of the design.[2]

In this paper, the concept of "AGI" is defined by certain information processing *principles* that are abstracted from the way the human mind works. Concretely, it means that

1) The system is *general purpose*, that is, not only works on a set of predetermined problems. Instead, the system often faces novel problems that anticipated neither by its designer nor by the system itself.
2) The system has to work in *realistic situations*, meaning that its knowledge and resources are often insufficient with respect to the problems to be solved, and that it has to process many tasks in real time, with a constant processing power.
3) The above restrictions means that the system usually cannot provide optimal solutions to problems. However, it can still provides *rational* solutions that are the best it can get under the current knowledge-resources restriction. It follows that the system must learn, from its experience, about how to use its knowledge and resources in the most efficient way.

More detained discussions about this kind of system can be found in [3], [4], [11].

II. Classical Neural Network

In principle, there are many possible paths toward AGI. This paper is a preliminary step to analyze the approaches that are based on neural networks.

As many other terms in the field, "Neural Networks" (as well as "Artificial Neural Networks" and "Connectionist Models") is used to refer to many different models, and there is no accurate definition of the term. Even so, the models do have common features, which distinguish them from other models in AI and Cognitive Science (CogSci). Instead of comparing each neural network model with the need of AGI (which is impossible for this paper), this paper will first introduce the notion of "Classical Neural Network" (CNN), as a "minimum core" with typical properties shared by many (but not all) neural networks. This notion will inevitably exclude the more complicated designs, but we have no choice but to start at a simple case.

According to the notions introduced in the previous section, the features of CNN are summarized on different levels (theory, model, and implementation).

As a theory of intelligence, CNN focuses on the reproduction of certain cognitive *functions*, defined as input-output mappings as performed by the human mind, such as in pattern recognition, classification, or association.[3] To achieve this goal, the major theoretical ideas behind CNN include:

- The system consists of many relatively simple processing units, which can run in parallel.
- The system solves problems by the cooperation of the units, without any centralized algorithm.
- The system's beliefs and skills are distributively represented by the units and their relations.
- The system is domain-independent by design, and it learns domain-specific beliefs and skills by training.
- Through learning, the system can generalize its training cases to similar problems.
- The system is tolerant to the uncertainty in the (training, testing, and working) data.

The representative literature of these ideas include manifestos of connectionism like [12], [13]. In the following, the above ideas will be called collectively as the "CNN ideas".

[2]Personally, I don't like the term "AGI", because to me, the concept of "intelligence" should be treated as a domain-independent capability [3], [11], therefore "general intelligence" is redundant. However, since the term was coined to stress the difference between this type of research and what is called "AI" by most people at the current time, I don't mind to use it in relevant discussions for this purpose. Hopefully one day in the future people will only use "AI" for general-purpose intelligent systems, and find other terms for the domain-specific works. At that time, the term "AGI" can retire.

[3]Please note that though CNN gets its inspirations from the human brain, it is not an attempt to duplicate the brain structure as faithfully as possible.

CNN is obtained by formalizing these ideas in the framework of *dynamic system*. Concretely, the major aspects of "CNN model" are specified as the following:

- **Static description: state vector and weight matrix**. A network consists of a constant number of neurons, with connections among them. Each neuron has an activation value, and each connection has a weight value. A neuron can be input, output, or hidden, depending on its relationship with the environment. Therefore, the state of a network with N neurons at a given time is represented by a vector of length N, which may be divided into an input (sub)vector, an output (sub)vector, and a hidden (sub)vector. All the weight values in the network at a given time can be represented by an N by N matrix.
- **Short-term changes: activation spreading**. In each step of state change, a neuron gets its input from the other neurons (and/or from the outside environment, for input neurons), then its activation function transforms this input into its (updated) activation value, which is also its output to be sent to the other neurons (and/or to the outside environment, for output neurons). As a result, the state vector changes its value. This process is repeated until the state vector converges to a stable value.
- **Long-term changes: parameter adjusting**. During the training phase of the network, the weights of connections are adjusted by a learning algorithm, according to the current state of the system. Training cases usually need to be repeatedly presented to the system, until the weight matrix converges to a stable value.
- **Overall objective: function learning**. As the consequence of training, the network becomes a function that maps input vectors to output vectors or state vectors. That is, starting from an initial state determined by an input vector,[4] the activation spreading process will lead the system to a stable output vector (such as when the network is used as an approximate function) or state vector (such as when the network is used as an associative memory).

The representative literature of these specifications include many textbooks on neural networks and AI, such as [14], [15], [16], [17].

Different types of CNN correspond to different topological structures, activation functions, learning algorithms, and so on [15], [18]. Since the following analysis and discussion are based on the above general ideas and model, they will be independent of the details of the model, as well as to the software/hardware implementation of the model. Therefore, this paper only addresses the "CNN ideas" and "CNN model", as defined above.

III. Strength and Weakness

Given the previous introduction, this section will evaluate CNN as a candidate technique for AGI.

[4]There are recurrent neural networks where an input vector does not correspond to the initial state of a running process of the network, therefore they are not included in the notion of CNN.

There is no doubt that neural networks has successfully solved many theoretical and practical problems in various domains. However, it remains unclear whether it is the preferred way to explain human cognition or to reproduce human intelligence as a whole, though that is a belief the connectionists have been promoting. From the very beginning, neural networks was not proposed merely as one more technique in the AI toolbox, but as an alternative to the "classical", symbolic tradition of AI and CogSci [19].

There is already a huge literature on the "symbolic vs. connectionist" debate [20], and this paper is not an attempt to address all aspects of it. Instead, there are some special features of the following discussion that distinguish this paper from the previous ones:

- It will not talk about neural networks in general, but focus on a special form, CNN, as defined previously.
- It will evaluates CNN against the goal of AGI, as defined above, rather than against other (also valid) goals.
- It will not talk about what "can" or "cannot" be achieved by CNN, because if we do not care about complexity and efficiency, what can be achieved by one AI technique can probably also be achieved by another one. Instead, it will discuss whether CNN is the *preferred* technique, compared to the other known candidates, when complexity and efficiency are taken into consideration.

The discussion will be organized around three major aspects of CNN: knowledge representation, learning process, and overall objective.

A. Knowledge representation

The knowledge in CNN is represented *distributively*, in the sense that there is no one-to-one mapping between the conceptual units and the storage unit of the system. Instead, a piece of knowledge usually corresponds to a value of the state vector, or is embedded in the weight matrix as a whole.

In general, it is usually impossible to say what a hidden neuron or a connection stands for.

For input and output neurons, there are two possibilities. In some CNNs, they still have individual denotations in the environment. For example, an input neuron may correspond to one attribute of the objects to be classified, and an output neuron may correspond to a class that the objects are classified into. In the following, such CNNs will be called "semi-distributed". On the other hand, there are CNNs where each input/output neuron has no individual denotation, and it is the input/output vectors that are the minimum meaningful units in the system. Such CNNs will be called "fully-distributed".

Distributed knowledge representation in CNN provides the following major advantages:

- **associative memory**. After a training pattern is stored in the weight matrix, it becomes *content-addressable*, in the sense that it can be recalled from a partial or distorted version of itself. The same functionality is difficult to be achieved with the traditional *local representation*, where a piece of knowledge must be accessed by its address, which is independent of its content.

- **fault tolerance**. If part of a CNN is destroyed, it does not lose its memory. Usually patterns stored in it can still be recalled, though with a lower accuracy — this is called "graceful degradation". On the contrary, in a system with local representation, the information in the damaged part will be completely lost (unless the system keeps duplications of the information somewhere else).

On the other hand, when a CNN is evaluated according to the requirement of AGI, there are issues in knowledge representation.

First, a CNN with semi-distributed representation has the following problems:

- **knowledge in multiple spaces**. Though for a given problem, it is often possible to describe the input and output as a set of attributes, it is hard to do so for a general-purpose system, unless we assume that for all the problems the system needs to deal with, the inputs and outputs still fall into the same multidimensional space. Usually this assumption leads to a space whose number of dimensions is too large to be actually implemented.
- **missing values**. Even when all input and output values can be assumed to be in a single space, a general-purpose system cannot assume further that all the input attributes have meaningful values in every input pattern — some attributes will be undefined for some input patterns. These missing values should not be replaced by average or "normal" values, as what usually happens in a CNN [21]. Though for a concrete situation it may be possible to find an *ad hoc* way to handle missing values, it is unknown whether there is a general and justified way to solve this problem.

A CNN with fully-distributed representation avoids the above problems by coding a piece of knowledge as a pattern in the whole input/output vector, where an individual input/output neuron no longer has separate meaning. However, the following problems still exist:

- **structured knowledge**. Though in theory a piece of knowledge can always be coded by a vector, it is not easy to represent its *structure*, when it is composited by smaller pieces in certain way.[5] Many solutions have been proposed for this problem (such as [23]), but each of them is typically designed for a special situation. The concept of "structure" seems to be foreign to vector-based representation, which is "flat" by nature.
- **selective processing**. Since all pieces of knowledge are mixed in the weight matrix, it is not easy (though not impossible) to process some of them selectively, without touching the others. For a general-purpose system, such selective processing is often absolutely necessary, since in different situations, only the *relevant* knowledge should be used.
- **explanation**. It is well known that to get a *meaningful* explanation of the internal process in a CNN is very difficult. Though it is possible to "explain" why a network produces a certain result in terms of state vector calculations, it is not what people can understand intuitively. "Meaningful explanation" usually means to describe the process in a concise way with familiar concepts, which seems incompatible with the very idea of distributed representation.[6]

[5]This issue is similar to the issue of "compositionality" raised by Fodor and Pylyshyn [22], though I do not fully agree with their analysis and conclusion.

B. Learning process

A CNN learns from training patterns by adjusting its weights to optimize a certain function of the system state, such as an *error function* or an *energy function*.

Each training pattern provides either an input-output pair to be remembered (supervised learning) or an input to be remembered or classified (unsupervised learning). All training patterns are usually repeatedly presented to the system during the training phase, before the network can be used as a function, memory, of classifier.

Compared to the other major AI techniques, the learning process in a CNN has the following advantages:

- **general-purpose algorithm**. The learning algorithms in CNNs are designed in a domain-independent manner (though for a concrete problem the designer usually needs to choose from several such algorithms, according to domain-specific considerations). The systems do not require built-in domain knowledge, unlike expert systems developed in the symbolic AI tradition, where the "intelligence" a system shows mainly come from its special design.
- **generalization and similarity**. A CNN does not merely remember a training pattern, but automatically generalizes it to cover similar patterns. Each training pattern not only influences the system's behavior in a point within the state space, but in a neighborhood. Consequently, the system can handle patterns it has never seen before, as far as they bear some similarity to certain training patterns. On the contrary, symbolic systems are notorious for their "brittleness" [24].
- **noise tolerance**. A learned regularity will not be destroyed by a small number of counterexamples. Instead, the counterexamples are considered as "noise" in the data, and therefore should be ignored. Similarly, if a pattern contains certain values that do not agree with the overall tendency among the patterns, they are considered as caused by noise, and the "normal" values will be used or remembered, instead. The same thing is hard to achieve using binary logic, where a single counterexample can "falsify" a general conclusion.

At the same time, this kind of learning also brings the following issues to CNN:

- **incremental learning**. In general, a CNN assumes that all the training happens before the network achieves its

[6]It is important to realize that explanation is not only required by the human users. A truly intelligent system also needs to explain certain internal processes to itself.

problem-solving capability, and all the training patterns are available at the beginning of the training phase. When the training patterns become available from time to time (life-long learning), the network often has to be re-trained with all the new and old patterns, which lead to unrealistic time-space demands. On the other hand, to incrementally adjust a trained network often leads to the destroy of old memory (catastrophic forgetting [25]).

- **learning in real-time**. It is well known that learning in a CNN often takes a very long time, and it does not support "one-shot" learning, which is often observed in the human mind, and is required for AGI. The long time expense comes from the demands that (1) all training patterns must be processed together (mentioned above), (2) the learning of a pattern is a *global* operation, in the sense that all neurons and weights may get involved, and (3) this process only ends in a stable state of the system.[7]
- **multi-strategy learning**. A CNN learns from concrete examples *only*, while human learning has many other forms, where new knowledge come as general statements, and new conclusions are obtained by inference that is more complicated than simple induction or analogy. Though it is nice for a CNN not to depend on built-in domain knowledge, it becomes a problem when domain knowledge is available but cannot be learned by the network except in the form of training patterns.
- **semantic similarity**. The generalization and similarity nature of CNN mentioned above is directly applicable only when the network uses semi-distributed representation, where each input dimensions has concrete meaning, so similar input patterns have similar meanings. With fully-distributed representation, however, this correspondence is no longer guaranteed anymore. If an input vector as a whole represents an arbitrary concept, then it is possible for similar vectors to represent semantically distant concepts. To handle semantic similarity in such a CNN, additional (i.e., not innate in CNN) learning mechanisms are needed, where vector similarity is irrelevant.
- **exception handling**. It is not alway valid to treat exceptions as noise to be ignored. Very often we hope the system to remember *both* the general tendency among training cases *and* the known exceptions of this tendency, then to select which of the two is applicable when a new case needs to be handled. This issue is related to the previous issue of selective processing.

Some of the above problems, such as incremental learning and exception handling, are addressed by the ART networks [26], [27], [28]. However, since their solutions are based on additional assumptions (such as the availability of uncommitted neurons for novel input patterns), the above problems remain unsolved, specially in a general-purpose system that has to face novel input patterns all the time.

C. Overall objective

After training, a CNN can carry out a function that maps an input vector either to an output vector or to a state vector.

Compare to the other AI techniques, the advantages of CNN on this aspect are:

- **complicated function**. A CNN can learn a very complicated function, which may be very difficult to get by the other AI learning techniques.
- **novel cases**. A CNN responds to novel cases in a reasonable manner, by treating it according to what has been learned from similar cases.

Even so, the implicit assumption that "everything of interest in AI and CogSci can be modeled as some kind of *function*" is problematic, from the viewpoint of AGI. This assumption implies that for any given input state, there is one and only one output state in the system, which provides meaningful result, and this output state correspond to an attractor of the CNN. Related to this implication, there are the following issues:

- **multiple goals**. A general-purpose system typically has multiple goals to pursue or to maintain at the same time, which are related to each other, but also remain mutually independent. It is not clear how to carry out multiple functions in this way within a single CNN.
- **multiple consequences**. Even for a single goal, it is often desired to get multiple consequences, which are produced by the system on the way. It is not a "function" in the sense that only the final value matters. Also, for the same input, there are often many valid outputs for an intelligent system. If a system always gives the same response to the same stimulus, its intelligence is quite doubtful.
- **open-ended processes**. For many intelligent/cognitive processes, there is not even a final state where a function value can be obtained. Instead, it is more like an open-ended anytime algorithm that never stops itself in the lifespan of the system. For example, a "creative thinking" process may never converge to any "attractor state", though the process remains fruitful.

IV. Conclusions

As a candidate technique to achieve AGI, the theoretical ideas behind CNN are all acceptable and justifiable, but the formal framework of CNN seems to be too rigid, for several reasons:

- The language is poor, because everything has to be represented as a numerical vector.
- Many cognitive facilities cannot be properly represented as functions.
- The system needs to learn in different ways, not just from concrete examples.
- A real-time system must properly respond to various time pressure.

[7] Though most neural networks are implemented as software running on serial processors, it is unrealistic to expect this problem to disappear in parallel-processing hardware implementations. Though more powerful hardware will definitely make a CNN run faster, for a general-purpose system there will always be complicated problems whose solutions cannot be carried out in parallel by the available processors. Consequently the time-cost problem will always be there.

Since this analysis only used the properties of the general "CNN model" defined previously, these problems cannot be easily solved by the changes in technical details within that framework, such as in network topology, activation spreading functions, learning algorithms, and so on.

Of course, the above conclusions about CNN cannot be directly generalized to all kinds of neural networks. After decades of research, there are already many types of neural network models that no longer fit into the CNN framework as defined in this paper. However, it does not mean that they are completely immune from the above issues, because there is still a recognizable "family resemblance" among the various models, including CNN (more or less as the ancestor of the family). Hopefully this work can provide a starting point for future research on the relation between AGI and neural networks, by using CNN as a reference point. For a concrete model, we can analyze its difference from CNN, and check whether it resolves the related issues discussed in this paper.

Many of the issues addressed in this paper have been raised before, and the neural network community has proposed various solutions to them. There is no doubt that there will be progress with respect to each of them. However, for neural network to be recognized as the best approach toward AGI, it not only has to resolve each of the above issues, but also has to resolve them altogether in a domain-independent manner. Furthermore, the solution should be better than what other AI techniques can provide. Therefore, it is not enough to only handle some special cases, then to assume that the result will generalize in a self-evident way, or to only specify certain desired details of the system, then to claim that all other problems will be solved by their "emergent consequences". This is still a challenge to the neural network community, since as of today, we are not aware of any neural network that has resolved all the issues discussed above.[8]

One possible approach believed by many people is to build a "hybrid system" which uses multiple techniques, including neural network. The argument supporting this approach is that since each technique has its strength and weakness, it will be better to use different techniques for different purposes, and therefore to get best overall performance. The problem about this approach is the consistency and efficiency of the system. Since this issue is discussed in more details in [29], it will not be repeated here.

As far as the current discussion is concerned, the author's own approach toward AGI can be roughly described as "formalizing the neural network ideas in a symbolic model". An AGI system named "NARS" realizes almost all of the "CNN ideas" listed previously in the paper, while on the model level, it is designed and implemented as an inference system (please visit the author's website for the related publications and demonstrations). Consequently, it has many (though not all) of the advantages of CNN, without suffering from its problems. A complete comparison between CNN and NARS will be left to a future publication.

[8] I do not mean that because of the above problems, the traditional "Symbolic AI" wins its war against the challenge of neural network. Actually I believe the *theoretical beliefs* behind Symbolic AI are mostly wrong, but that is not an issue to be discussed here (see [3], [4]).

Acknowledgment

Thanks to Shane Legg, Ben Goertzel, and the anonymous reviewers for their helpful comments on previous versions of the paper.

References

[1] D. Marr, "Artificial intelligence: a personal view," *Artificial Intelligence*, vol. 9, pp. 37–48, 1977.

[2] A. Newell, *Unified Theories of Cognition*. Cambridge, Massachusetts: Harvard University Press, 1990.

[3] P. Wang, "Non-axiomatic reasoning system: Exploring the essence of intelligence," Ph.D. dissertation, Indiana University, 1995.

[4] ——, "The logic of intelligence," in *Artificial General Intelligence*, B. Goertzel and C. Pennachin, Eds. New York: Springer, 2006.

[5] A. Turing, "Computing machinery and intelligence," *Mind*, vol. LIX, pp. 433–460, 1950.

[6] A. Newell and H. Simon, "GPS, a program that simulates human thought," in *Computers and Thought*, E. Feigenbaum and J. Feldman, Eds. McGraw-Hill, New York, 1963, pp. 279–293.

[7] B. Goertzel and C. Pennachin, Eds., *Artificial General Intelligence*. New York: Springer, 2006.

[8] J. Searle, "Minds, brains, and programs," *The Behavioral and Brain Sciences*, vol. 3, pp. 417–424, 1980.

[9] J. McCarthy, "From here to human-level AI," 1996, invited talk at KR-96, http://www-formal.stanford.edu/jmc/human/human.html.

[10] N. Nilsson, "Human-level artificial intelligence? Be serious!" *AI Magazine*, vol. 26, no. 4, pp. 68–75, 2005.

[11] P. Wang, "Rigid Flexibility — The Logic of Intelligence," 2006, manuscript under review.

[12] P. Smolensky, "On the proper treatment of connectionism," *Behavioral and Brain Sciences*, vol. 11, pp. 1–74, 1988.

[13] D. E. Rumelhart, "The architecture of mind: A connectionist approach," in *Foundations of Cognitive Science (Second Edition)*, M. I. Posner, Ed. Cambridge, Massachusetts: MIT Press, 1990, pp. 133–159.

[14] D. Rumelhart and J. McClelland, Eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, Foundations*. Cambridge, Massachusetts: MIT Press, 1986.

[15] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 1999.

[16] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 2002.

[17] G. Luger, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, 5th ed. Reading, Massachusetts: Addison-Wesley, 2005.

[18] J. Šíma and P. Orponen, "General-purpose computation with neural networks: a survey of complexity theoretic results," *Neural Computation*, vol. 15, no. 12, pp. 2727–2778, 2003.

[19] P. Smolensky, "Connectionism and the foundations of AI," in *The Foundations of Artificial Intelligence: a Sourcebook*, D. Partridge and Y. Wilkes, Eds. Cambridge: Cambridge University Press, 1990, pp. 306–326.

[20] D. Chalmers, "Contemporary philosophy of mind: An annotated bibliography, Section 4.3: Philosophy of connectionism," 2005, http://consc.net/biblio/4.html#4.3.

[21] C. M. Ennett, M. Frize, and C. R. Walker, "Influence of missing values on artificial neural network performance," in *Proceedings of MEDINFO 2001*, V. Patel, R. Rogers, and R. Haux, Eds.

[22] J. A. Fodor and Z. W. Pylyshyn, "Connectionism and cognitive architecture: a critical analysis," *Cognition*, vol. 28, pp. 3–71, 1988.

[23] J. B. Pollack, "Recursive distributed representations," *Artificial Intelligence*, vol. 46, no. 1-2, pp. 77–105, 1990.

[24] J. Holland, "Escaping brittleness: the possibilities of general purpose learning algorithms applied to parallel rule-based systems," in *Machine Learning: an artificial intelligence approach*, R. Michalski, J. Carbonell, and T. Mitchell, Eds. Los Altos, California: Morgan Kaufmann, 1986, vol. II, ch. 20, pp. 593–624.

[25] R. French, "Catastrophic forgetting in connectionist networks: Causes, consequences and solutions," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[26] S. Grossberg, “Adaptive pattern classification and universal recoding: I. Parallel development and coding in neural feature detectors,” *Biological Cybernetics*, vol. 23, pp. 121–134, 1976.
[27] ——, “Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, and illusions,” *Biological Cybernetics*, vol. 23, pp. 187–202, 1976.
[28] G. Carpenter and S. Grossberg, “Adaptive resonance theory,” in *The Handbook of Brain Theory and Neural Networks*, 2nd ed., M. Arbib, Ed. Cambridge, Massachusetts: MIT Press, 2003, pp. 87–90.
[29] P. Wang, “Artificial intelligence: What it is, and what it should be,” in *The 2006 AAAI Spring Symposium on Between a Rock and a Hard Place: Cognitive Science Principles Meet AI-Hard Problems*, Stanford, California, 2006.