A Connectionist Approach to Shape Recognition and Transformation

Yu-Tung Liu

Graduate School of Design
Harvard University
Cambridge, MA 02138 USA

In human design processes, many drawings of shapes remain incomplete or are executed inaccurately. Cognitively, a designer is able to discern these anomalous shapes, whereas current CAAD systems fail to recognize them properly so that CAAD systems are unable to match left-hand-side conditions of shape rules. More unfortunately, as a result, current CAAD systems fail to retrieve right-hand-side actions. In this paper, multi-layered neural networks are constructed to solve the recognition and transformation of ill-processed shapes in the light of recent advances of connectionism in cognitive psychology and artificial intelligence.

Keywords: shape recognition, shape transformation, connectionist models, PDP models, content-addressable memory, neural networks.

1 Introduction

According to investigations of Gestalt psychology on visual form perception (Koffka, 1935), it is clear that people have no difficulty recognizing an incomplete form as a closed, regular square as shown in Figure 1a. Sometimes called *the law of closure*, "closure is a tendency, other things being equal, to group into unified structures those components that together constitute a closed entity rather than an open one" (Rock, 1984:118). In design processes, human designers recognize incomplete and inaccurate shapes properly and effortlessly. Nevertheless, when considering the issue in the context of grammatical computer-aided architectural design (CAAD) systems, two questions are raised. First, how can the system recognize an ill-processed shape as the condition, left-hand side, of a given rule as shown in Figure 1b. Second, how can the system transform that condition to the action, the right-hand side of the given rule, when the ill-processed shape is seen.

In the field of CAAD, problems of design shapes and building functions are always solved by techniques of symbolic approach, especially the production systems in the form of shape grammars (Stiny and Mitchell, 1978; Stiny, 1980), graph systems (Levin, 1964; Flemming et al., 1988) and knowledge-based systems with predicate logic (Gero et al., 1985; Akiner, 1986; Liu, 1991). At present, when transforming the current state into next one by applying rules, production systems simply perform exact and complete pattern-matching. However, incomplete and inaccurate shapes are ubiquitous in the preliminary

*U. Flemming and S. Van Wyk (eds.), **CAAD Futures '93**, pp. 19-36.*
© 1993 Elsevier Science Publishers B.V. Printed in the Netherlands.

stages of design but, until recently, are difficult to approach symbolically. Those early stages are critical because important concepts and decision-making will be done.

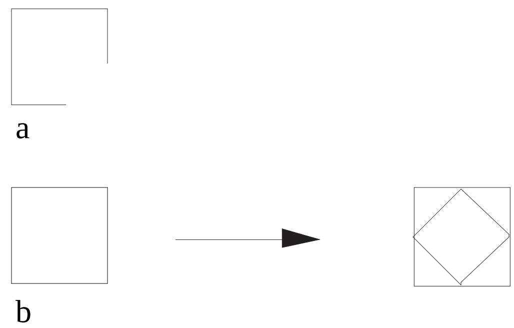


Figure 1. An incomplete shape (a) and a given shape rule (b).

By virtue of neurobiology's contribution in cognitive psychology and artificial intelligence (AI), the swift development of *connectionist models* (also referred to as *parallel distributed processing models*, PDP models, or *neural networks*) have proven that such cognitive processes such as pattern recognition, classification, and memory, can be modelled (Rumelhart et al., 1986b). Consequently, it should be logical to assume that connectionist models would also be adequate problem-solving tools in CAAD. This paper focuses upon shape recognition and transformation in design.

2 Connectionist Models as Cognitive Models

There exists a question with respect to human cognition: “how could higher-level function be achieved by connecting basic elements like neurons” (Anderson, 1990). Connectionist models explore how to connect neurons together in a manner to account for higher-level cognition, such as classification, pattern recognition, and memory retrieval. To simulate how a brain functions, many different connectionist models, as brain models, have been developed, for instance, perceptron (Minsky and Papert, 1969), pattern associator (PA) model (McClelland and Rumelhart, 1989), Boltzmann machine (Hinton and Sejnowski, 1986), competitive learning model (Rumelhart and Zipser, 1986), and back propagation (BackProp) model (Rumelhart et al., 1986a). All of these models share the fundamental characteristics of connectionism. As Rich and Knight (1991:487) summarize, these models have in common:

- a large number of very simple neuron-like processing elements called units;
- a large number of weighted connections between the elements, with the weights on the connections encoding the knowledge of a network;
- highly parallel and distributed control;
- an emphasis on learning internal representations automatically.

The most basic connectionist models are two-layered networks (Figure 2), such as the PA models, which are just comprised of input and output units. They are the simplest neural-like learning machines able to learn associations between input and output patterns. When the input patterns are presented to the network, through initially random weights of connections, the activations of the obtained output can be computed following a two-phase processing cycle. First, the net input to each output unit is computed: "the sum of the activations of the input units times the corresponding weights, plus an optional bias term associated with the output units" (McClelland and Rumelhart, 1989, p. 97) as shown in the following equation:

$$Net_j = \sum_i W_{ji} I_i + Bias_j \tag{1}$$

Second, the activation of the output unit is then determined by one of the four activation functions (McClelland and Rumelhart, 1989:97):

- *Linear*. The activation of output unit j is equal to the net input.
- *Linear threshold*. Each output is a linear threshold unit; namely its activation is set to 1 if its net input exceeds a threshold and is set to 0 otherwise.
- *Stochastic*. The output is set to 1 with a probability p determined by:

$$p(o_j = 1) = \frac{1}{1 + e^{-net_j/T}} \tag{2}$$

  where T is a global parameter analogous to temperature in physical systems.
- *Continuous sigmoid*. Each output unit takes on an activation that is non-linearly related to its input according to the logistic function:

$$o_j = \frac{1}{1 + e^{-net_j/T}} \tag{3}$$

The idea of learning is to adjust the weights of the connections according to activations of input, output, and target patterns so that the difference between obtained and target output, that is error measure, can be reduced. As the following formula indicates:

$$\Delta W_{ji} = \mu(T_j - O_j) I_i \tag{4}$$

Note that μ is the learning rate parameter. In other words, the training process is a procedure of minimizing the resulting error measure which is known as *gradient descent*; it shares the same concept as the hill-climbing search strategy used in symbolic AI systems (Rich and Knight, 1991). Accordingly, the PA model's learning process is a search. In connectionist models, training which includes a great number of two-phrase cycles, namely training "epochs," never terminates until the target output is reached. Sometimes the PA model is sophisticated enough to solve the encountered input/output mapping problems. However, the limitation of a PA model occurs when the input patterns are uncorrelated or "orthogonal," and as a result, the PA network is unable to perform adequate input/output mappings (Minsky and Papert, 1969; McClelland and Rumelhart, 1989). For the entire formal analysis and more details on the PA model, see McClelland and Rumelhart (1989).

To overcome limitations of two-layered neural networks as described above, what is needed are advanced models which contain at least one hidden layer between input and

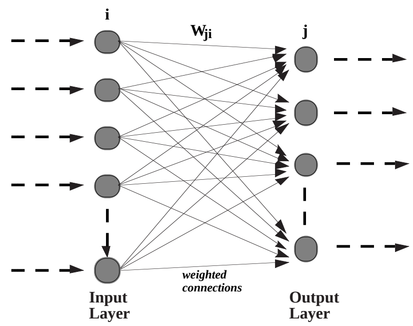


Figure 2. A typical two-layered Pattern Associator network.

output units. With hidden layers, such advanced models can embed internal representations of knowledge for problem-solving. Similar to the PA model's execution, the typical multi-layered BackProp models are fully connected, feedforward networks, as in Figure 3, where each unit in one layer is connected forward to every unit in the next layer. Activations flow from the input layer, through hidden layer(s) to the output layer. The difference between the obtained and target output, or error measure, can be computed to adjust the weights of connections between output and hidden layers. Similar to the PA model's learning rule, "the weight on each line should be changed by an amount proportional to an error signal, δ, available to the unit receiving input along that line and the output of the unit sending activation along that line," (Rumelhart et al., 1986a:326) as is shown below:

$$\Delta_p W_{ji} = \mu \delta_{pj} O_i \tag{5}$$

The term $\Delta_p W_{ji}$ is the adjustment to the weight from the *i*th to the *j*th unit following presentation of input/output pair p. The crucial point is to determine what the error signal should be for the units in output layer and hidden layer(s). If a unit is in the output layer, its error signal is very similar to the formula of the PA model:

$$\delta_{pj} = (T_{pj} - O_{pj}) f'_j (Net_{pj}) \tag{6}$$

Whenever the unit is in the hidden layer(s), because there is no specified target value, its error signal is:

$$\delta_{pj} = f'_j (Net_{pj}) \sum_k \delta_{pk} W_{kj} \tag{7}$$

"This propagates the error back one layer and the same process can be repeated for every layer" (Rumelhart et al., 1986a:327). Finally, errors are propagated back to the connections between the hidden and input layers. In short, the BackProp's behavior is a more complex self-learning process designed to minimize errors and to perform gradient descent search.

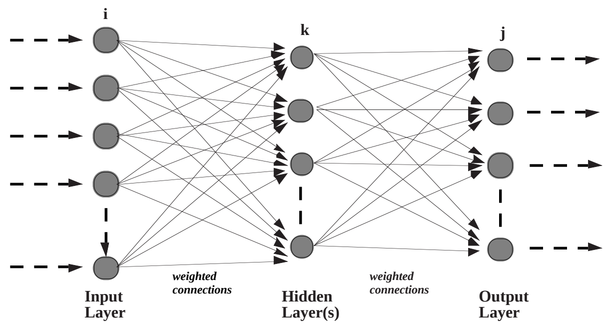


Figure 3. A typical multi-layered back propagation network.

The exact behavior of the hidden units remains unclear. But, because of the internal knowledge representation provided by the hidden units, BackProp networks are capable of overcoming the PA model's problem-solving limitations, namely generalizing all possible input/output mappings. Nevertheless, multi-layered connectionist networks continue to have two serious problems. First, training a large BackProp network is terribly time-consuming. Second, theoretically, during the gradient descent learning systems sometimes slip into local minima. In the literature, it has been generally shown that when such networks have many hidden units, local minima rarely happen; whereas, when there are few hidden units, local minima are more common (Rumelhart et al., 1986a; McClelland and Rumelhart, 1989). For the entire formal analysis and more details on the BackProp model, see Rumelhart et al. (1986a).

3 Connectionist Models in Architectural Design

In the area of CAAD, a two-layered PA model used to be adopted for solving design problem, namely building attribute/form mapping applications (Coyne and Newton, 1990; Coyne, 1991). In an attempt to come up with a typology for the preliminary stages in architectural design, Coyne constructed a two-layered PA network to perform the necessary mappings between building plan forms and their corresponding performances. The knowledge of those attributes refers to “a combination of high-level plan descriptors and the opinions of one designer as to the suitability of the various forms for certain uses and siting conditions,” (Coyne et al., 1990:497). Specifically, the mappings can be seen as a peculiar version of an expert system based upon someone's design preference or expertise. The form-attributes solution pairs can, consequently, express a designer's decision-making style if the stylistic preference or expertise had ever been well-trained.

In Coyne's research on PA networks, the fundamental features of connectionist models are illustrated again—the network can perform appropriate mappings whether input patterns are complete or not. Such generalization capabilities for pattern completion will serve an important role in grappling with design problems because design drawings are often processed roughly and design attributes are not always well-defined during the earlier architectural program phase.

Lacking the hidden layer's internal representation, a two-layer PA model's problem-solving capability is limited: it is unable to perform adequate mappings when structures of input patterns are very different, with few “overlaps.” In their book *Perceptrons*, Minsky and Papert (1969) mention a classic example of this case, the exclusive-or (XOR) problem:

0 - 0 —> 0
1 - 1 —> 0
0 - 1 —> 1
1 - 0 —> 1

Here the input patterns which overlap least are intended to generate identical outputs. In the first and second input/output pairs above, the structures of inputs which overlaps least are 0-0 and 1-1 but their outputs are identical, that is 0. The situation is exactly the same as the third and fourth pairs. No two-layer network can properly solve problems like XOR.

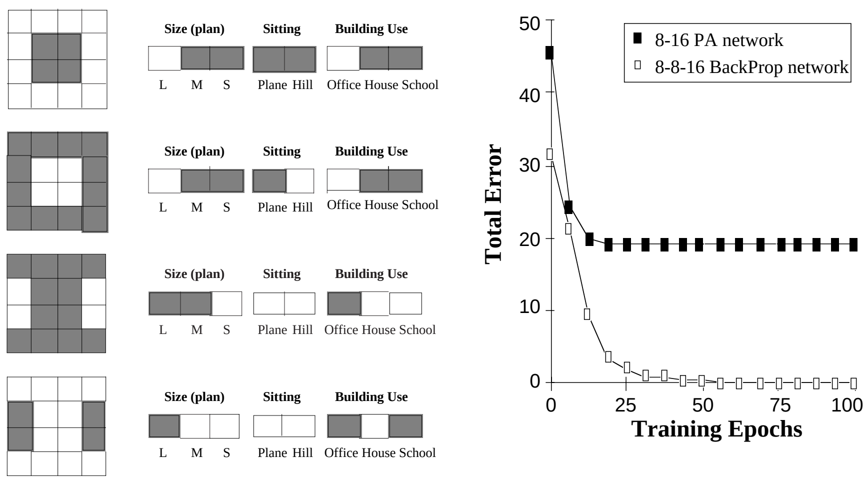


Figure 4. A set of form/attributes mappings which cannot be solved by the PA network but is solved by the BackProp (left) and the total error for the two networks to solve the problem as a function of training epochs (right).

In the form/attributes mapping problems of design, it cannot be guaranteed that mapping problems similar to the XOR problem never exist. For instance, if an experienced designer acquires a particular, simple set of form/attributes mappings based on his design preference or expertise as shown in Figure 4, the 16-8 PA network (with 16 input and 8 output units) fails to perform the required mappings—the gradient descent learning process is always stuck at the error value of 19 (Figure 4). However, the same problem is easily solved by a 16-8-8 BackProp network (coupled with 8 hidden units in addition) as shown in Figure 4. This is because, as Minsky and Papert have mentioned, there is a recoding (internal representation) of the input patterns in the hidden units in which the similarity of patterns among the hidden units can support any required mapping from the inputs to the output units.

As demonstrated in the above historical XOR and design typology examples, multilayer neural-like networks, I will argue, are critical to solve design form/attributes mapping problems as well as shape recognition and transformation. Therefore, in this study, multi-layered BackProp models are used to bear on ill-processed shape problems.

4 Objectives and Methodology

In human design processes, many drawings of shapes remain incomplete or are executed roughly and inaccurately, namely irregular and "noisy" shapes as shown in the left-hand sides of Figure 5. In Gestalt and cognitive psychologists' research on human visualization, as discussed previously, human recognition capability is so natural and potent (Koffka, 1935; Anderson, 1990). Cognitively, the human designer is able to discern these anomalous shapes, whereas current CAAD systems fail to recognize them properly so that CAAD systems are unable to match left-hand-side conditions of shape rules. More unfortunately, as a result, current CAAD systems also fail to retrieve right-hand-side actions from their long-term memory (LTM), namely the knowledge base. The objective of this research is to construct connectionist networks that simulate a human designer's shape recognition and transformation (Figure 5). Such networks will be able to solve the recognition and transformation problems of ill-processed shapes in design which try to come up with associative approaches for design shape reasoning as well. This, in turn, may lead to

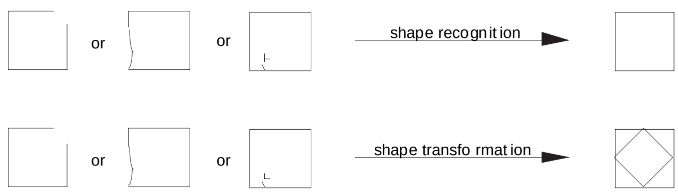


Figure 5. Recognition and transformation for ill-processed shapes.

improvements in shape grammars and knowledge-based CAAD systems with predicate logic

To achieve these objectives, a set of shape rules (see Figure 6) is arbitrarily selected as an example for simulation. The design shapes are drawn in 7x7 grids and, thus, represented by the 49-element vectors. Three BackProp networks: 49-12-49, 49-25-49, and 49-49-49 (the first, middle, and last digits represent the numbers of units in input, hidden and output layers respectively), are constructed to solve problems of shape recognition and transformation. The appendix compares the learning performances of these networks with different hidden units in depth.

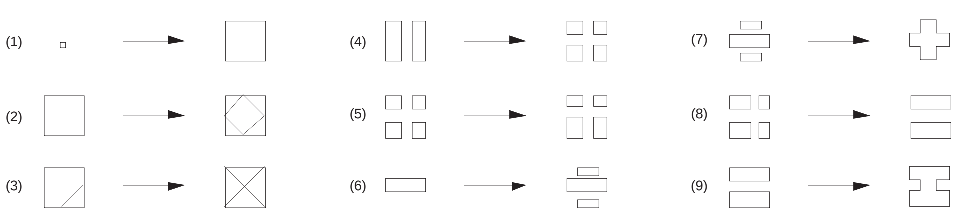


Figure 6. Selected shape rules.

5 Application One: Shape Recognition

Recognition has long been a critical topic in cognitive psychology. Different models have been created to explain how people recognize patterns, such as template-matching, feature analysis and so forth (Lindsay and Norman, 1977; Anderson, 1990). In these models, information is processed serially and symbolically. In contrast to serial and symbolic models of recognition, connectionist models represent sophisticated ways of connecting neurons and processing information in parallel. Therefore they are valuable to understand the mechanism of human recognition in mind. In design literature, issues of shape recognition used to be investigated following symbolic and algorithmic approaches (Krishnamurti, 1981; Krishnamurti and Giraud, 1986); here, the application for shape recognition is based upon connectionism in the light of the recent advances in neuropsychology and connectionist AI.

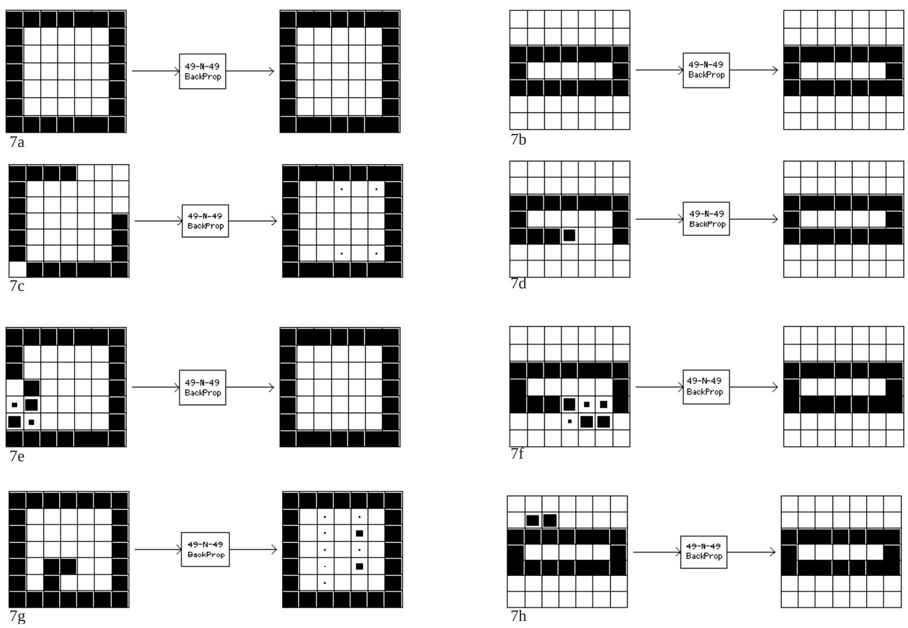


Figure 7. Graphic results of shape recognition (tolerable noises, as represented by the small dots, for example, within output patterns in (e) and (g), are caused by the higher value of the predefined summed squared errors of 0.05).

In order to recognize incomplete, irregular, and noisy design shapes by BackProp models, the left-hand sides of shape rules act as both inputs and targets. The major concerns here are not only how to model a designer's shape recognition, but also how to improve the pattern-matching techniques of production systems. Such a network, which possesses versatile matching capabilities, could be incorporated into *existing* CAAD systems and would thus benefit them.

Once the BackProp networks are well-learned over training epochs, that is, in this project, the *summed squared errors* are less than 0.05, it is time to test the systems' behaviors. Whenever a complete shape is presented to the BackProp network, the system definitely “knows” what shape it is (Figure 7a,b). Matching of this kind is the exact performance of current CAAD system pattern-matching capabilities, namely 100% matching based on a symbolic-AI-oriented approach. When the BackProp network "sees" incomplete shapes (Figure 7c,d), irregular shapes (Figure 7e,f), and noisy shapes (Figure 7g h), it still "recognizes" those inputs as corresponding complete shapes with few tolerable "noises" in Figure 7c and Figure 7g. Those tolerable noises are caused by the higher value of the predefined *summed squared errors* of 0.05. Lowering that value in learning process will significantly eliminate the noises.

As a result, BackProp recognition capabilities are able to improve current CAAD system's deficient pattern-matching techniques. Another important significant factor is that such a connectionist network can cognitively model a human designer's recognition of design shapes.

6 Application Two: Shape Transformation

People can retrieve knowledge in their memories by accessing proper clues or partial description of memories' contents; this is called *content-addressable memory*. In cognitive psychology, the neuron-like networks are able to model human content-addressable memory (Hinton et al., 1986). Further, applying procedural rules in production systems

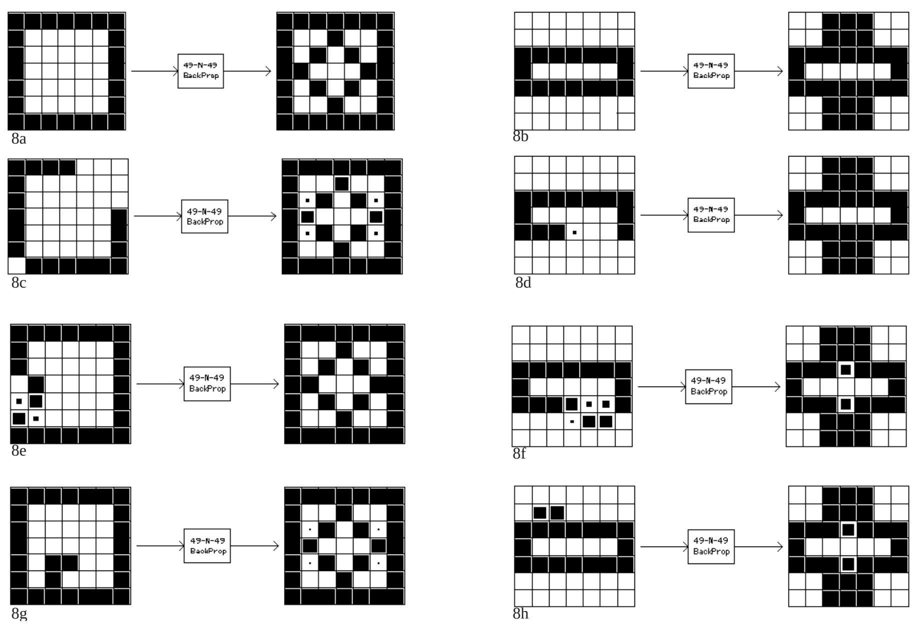


Figure 8. The graphic results of shape transformation.

can be viewed as content-addressable memory: all "the right-hand sides are the content, the encoded knowledge in the memory; and the left-hand sides are the retrieval clues, the accessing structure" (Newell, 1990:165). Therefore, in design, shape transformation by applying shape rules can be seen not only as content-addressable memory of design knowledge, but also be achieved by connectionist models.

In attempting to transform shapes using the BackProp networks, the input patterns are the left-hand sides of the rules; and the targets are the right-hand sides. The significance of this simulation is threefold: (1) the shape transformations provide solutions to problems in design; (2) it is capable of cognitively modelling a human designer's content-addressable memory with regard to design shapes; and (3) it can be viewed as an associative form of shape reasoning that will be able to guide knowledge processing between the working memory (short-term memory) and production memory (long-term memory) for building *future* CAAD systems.

After the network is well-trained, when presenting complete retrieval clues (left-hand sides of the rules) to the BackProp network, the corresponding contents (right-hand sides) can be retrieved from memory (Figure 8a and 8b), exactly how current CAAD systems perform. However, when incomplete (Figure 8c and 8d), irregular (Figure 8e and 8f) or noisy (Figure 8g and 8h) retrieval clues are presented to the BackProp, the correct contents with few tolerable “noises” are still able to be acquired. The later phenomenon is what current CAAD systems fail to do. Note that the magnitude of tolerable noises is determined and caused by the value *summed squared errors* of 0.05 set up initially. The smaller the *summed squared errors* are set, the less noise occurs. Using the powerful generalization of the BackProp model, the acting content-addressable memory in this concern is consistent with a human designer's cognitive processes.

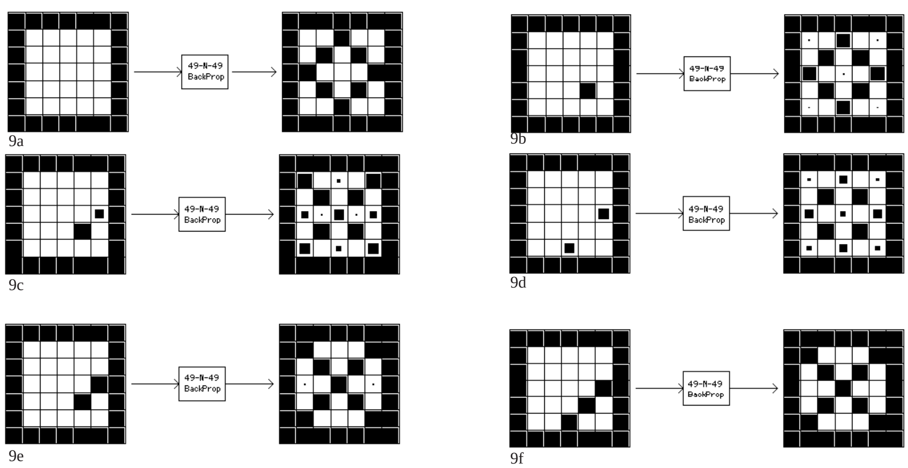


Figure 9. The graphic results of similar shape discrimination.

7 Discussion on Similar Shape Discrimination

As shown in preceding examples, artificial neural-like networks are capable of discerning ill-processed shapes as left-hand sides (premises) of rules and further transforming them into right-hand sides (consequences). This raises a question “how well does such a network discriminate among training shapes similar to one another?” And more profoundly, “what transformation does such a network perform when an ill-processed shape is equally similar to two training shapes.” In the set of training rules shown in Figure 6, the left-hand sides of rule two and three look very similar, whereas the right-hand sides are different. When presenting exact left-hand sides of two rules to the well-learned BackProp network, the system can flawlessly discriminate between them and successfully transform them into the right-hand sides of rule two and three, respectively (Figures 9a and 9f). This means that connectionist models provide fine shape discrimination between rules whose left-hand sides are very similar.

When the system “sees” ill-processed shapes drawn between the two left-hand sides, it has to recognize them from two different perspectives, either adding something to rule two's premise or diminishing something from rule three's promise. In Figure 9b, the system transforms the presenting shape into rule two's right-hand side with tolerable noises because the presenting shape is closer to rule two's left-hand side. In a similar fashion, in Figure 9e, BackProp network transforms the ill-processed shape into rule three's consequence because that presenting shape is more similar to rule three's promise. As demonstrated in Figures 9c and 9d, if the presenting shape is equally similar to both rule two and three's left-hand side, the system “gets confused.” The system cannot transform the presenting shape into one of the exact right-hand sides, but instead, retrieves a combined shape equally characterized by two right-hand sides.

The above BackProp behaviors are consistent with human visual performance. Between the letters A and H, ambiguous letters which are closer to an A (Figure 10b) or an H (Figure 10d) can be perceived as an A or an H, respectively. However, the same ambiguous letter, which is difficult to discriminate without a specific context (Figure 10c), can be easily perceived as an A or an H, depending on the context as shown in Figure 10f (Selfridge, 1955). This implies that ill-processed shapes equally similar to well-learned shapes, as presented in Figure 9c and 9d, can be well-recognized only when viewed within specific contexts.

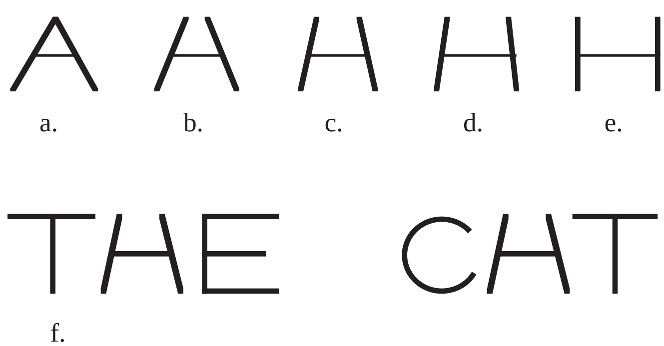
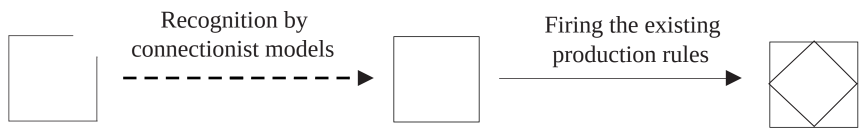


Figure 10. An example of ambiguous letters and context.

8 Concluding Remarks

As mentioned in the beginning of this paper, Gestalt psychologists have pointed out the phenomenon that people are capable of recognizing incomplete forms, but the mechanism for achieving this was not found. From the two applications introduced in this paper, neural-like networks seem to be adequate to model human shape recognition and content-addressable memory because they connect artificial neurons in ways that bear on higher-level cognition. The BackProp's behavior, therefore, provides profound explanations for a human designer's cognition regarding ill-processed shape recognition and transformation. Based upon the BackProp model's abilities to learn and generalize, this project illustrates two implications to CAAD.

First, ill-processed shape recognition solved by connectionist models provides a way to improve existing CAAD systems' pattern-matching techniques without revising the overall structure of those systems:



The condition (right-hand side) of a shape rule is first recognized by a BackProp network; the action (left-hand side) of that shape rule can be then retrieved from the CAAD system knowledge base. In other words, to pursue better performance based on findings for human visual recognition performance, current CAAD systems can synthesize with connectionist networks.

Second, content-addressable memory simulations of design shapes have proposed a new idea for shape transformation, namely knowledge retrieval from the LTM to the working memory:

Such an associative approach to shape reasoning can also benefit CAAD systems.

To use connectionist models in CAAD, which traditionally employs a symbolic approach, is by no means facile, however. The symbolic approach builds on the fundamental assumption that the human mind, which deals with symbols, such as points, lines, words, numbers, etc., is a *symbol system*, also called *information-processing system* (Simon, 1978, 1981; Gardner, 1985). The major obstacles then are that, at present, connectionist and symbolic approaches belong to two opposite *logics*: First, primitives in connectionist models, neurons, are very small in scale and numerous in quantity; whereas ones in symbolic systems, symbols, are comparatively very large but their numbers are much smaller (Figure 11). Second, symbolic systems are processed serially; in contrast, connectionist models are processed in parallel. The distinctions make these two approaches difficult to integrate and to even communicate. Thus, exploring the intermediate area between connectionist and symbolic processing is the inevitable, primary mission for the next decade in AI and cognitive psychology as well as CAAD (Minsky, 1992).

The developments of connectionist models are still in their infancy and are still a new conception in the field of CAAD. In this study, two applications of connectionism in architectural design have been implemented and discussed, however, there are still many limitations and problems to be addressed in future investigations:

Inference by association. When building his recursive auto-associative memory (RAAM) model, Pollack (1988) indicate the possibility of associative inference performed by connectionist models. A human designer's inference capabilities are naturally powerful and thus play a crucial role in design thinking and reasoning. A simple design inference inspired by Newell's *chunking* idea in his cognitive system SOAR (Laird et al., 1986; Newell, 1990) is illustrated in Figure 12. Such a simple design inference should be one of the worthwhile directions applied to further connectionist studies in CAAD.

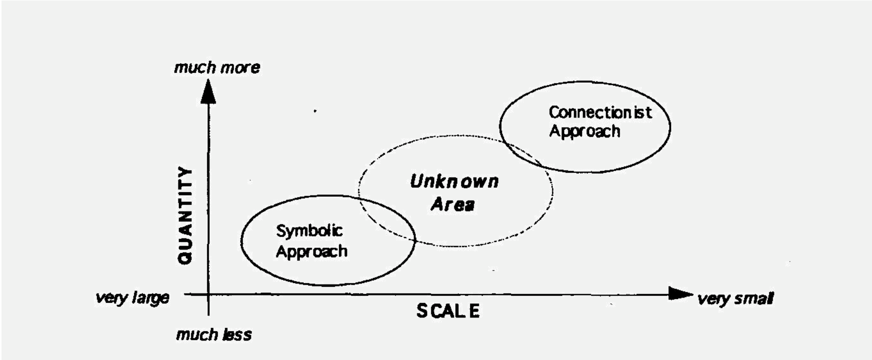


Figure 11. Symbolic versus connectionist approach in scale and quantity (after Minsky, 1992).

Encoding emergent sub-shapes. The ambiguity and flexibility of design shapes has been pointed out by Mitchell (1990a, 1990b, 1992) and Stiny (1990), therefore, encoding emergent subshapes as shown in Figure 13 is of central *interest in* this concern because it is important to explore conception in the processes of human design problem-solving.

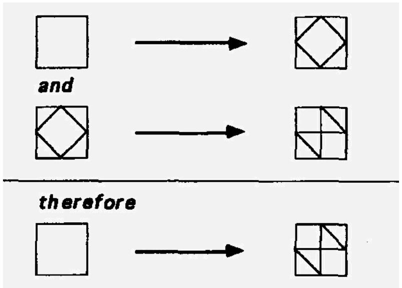


Figure 12. A simple example of design shape inference.

Tan (1990) uses a different data structure to decompose shapes into primitive parts for further encoding. Nagakura (1990) uses the advantage of script representation of knowledge, namely a way of encoding regularities in *categories, to* find the members of a particular shape category. Both provide clues for addressing subshape recognition by

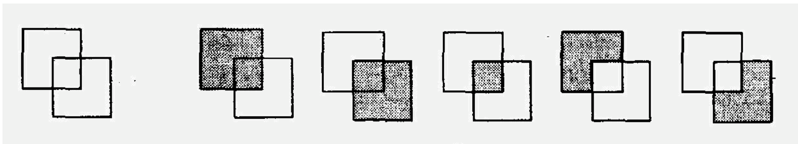


Figure 13. A shape and its emergent subshapes.

computer. Theoretically, people can recognize the emergent subshapes as well as the most obvious shapes because they have learned a huge amount of shapes in different sizes, colors, locations, and even line-types from past experience. With great learning and generalization capability, the connectionist networks should be able to model and solve the problems of encoding emergent subshapes.

Acknowledgments

The author expresses appreciation to Professor William Mitchell, Graduate School of Design, Harvard University, not only for his encouragement and insightful comments but particularly for posing problems to the author regarding recognizing design shapes as the initial step to this study. The author also thanks Professor Jay Rueckl, Harvard University Department of Psychology, for suggestions and readings on the performances of the back propagation model; Professor Stephen Ervin Graduate School of Design, Harvard University, for his challenging comments and constructive suggestions about organizing the paper; and special thanks Professor Marvin Minsky, Media Laboratory, MIT, for his lectures on *The Society of Mind* and valuable comments that have broadened and deepened the author's understanding of symbolic and connectionist perspectives and their inter-relationships.

References

Akiner, V. T., 1986. "Topology-1: A Knowledge-based System for Reasoning About Objects and Spaces," *Design Studies* 7, No. 2, pp. 94-105.

Anderson, J. R., 1990. *Cognitive Psychology and its Implications* (3rd edition). New York: W. H. Freeman.

Coyne, R. D., 1991. "Modelling the Emergence of Design Descriptions Across Schemata," *Environment and Planning B: Planning and Design* 18, pp. 427-458.

Coyne, R. D. and Newton, S., 1990. "Design Reasoning by Association," *Environment and Planning B: Planning and Design* 17, pp. 39-56.

Coyne, R. D., Rosenman, M. A., Radford, A. D., Balachandran, M. and Gero, J. S., 1990. *Knowledge-based Design Systems*. Reading, MA: Addison-Wesley.

Hemming, U., Coyne, R., Glavin, T., and Rychener, M., 1988. "A Generative Expert System for the Design of Building Layouts-Version 2," in J. S. Gero (ed.), *Artificial Intelligence in Engineering Design*. Amsterdam: Elsevier, p. 445.

Gardner, H., 1985. *The Mind's New Science*. New York: Basic Books.

Gem, J. S., Radford, B. D., Coyne, R.. and Akiner, V. T., 1985. "Knowledge-based Computer-aided Architectural Design," in J. S. Gero (ad.), *Knowledge Engineering in Computer-Aided Design*. Amsterdam: North-Holland, p. 57.

Hinton, G. E. and Sejnowski, T. J., 1986. "Learning and relearning in Boltzmann machines," in D. Rumeihart, J. McClelland, and the PDP Research Group (eds.), *Parallel Distributed Processing*. Cambridge: MIT Press, p. 282.

Koffka, K., 1935. *Principles of Gestalt Psychology*. New York: Harcourt, Brace and Co.

Krishnamurti, R., 1981. "The Construction of Shapes," *Environment and Planning B: Planning and Design* 8, pp. 5-40.

Krishnamurti, R. and Giraud, C., 1986. "Towards a Shape Editor: The Implementation of a Shape Generation System," *Environment and* Planning B*: Planning and Design* 13, pp. 391-404.

Laird, J. E., Rosenbloom, P. S. and Newell. A., 1986. "Chunking in Soar: The Anatomy of a General Learning Mechanism," *Machine learning 1,* pp. 11-46.

Levin, P. H., 1964. "Use of Graphs to Decide the Optimum Layout of Buildings," *The Architects'Journal, pp*. 809-817.
Lindsay, P. H. and Norman, D. A., 1977. *Human Information Processing*. New York: Academic Press.
Liu, Y. T., 1991. "Schematic-Designer: A Knowledge-based CAD System for Schematic Design in Architecture," *Design Studies* 12(3), pp. 151-167.
McClelland, I. L. and Rumeihart, D. E., 1989. *Explorations in Parallel Distributed Processing*. Cambridge: MIT Press.
Minsky, M., 1992. Lecture on "The Society of Mind," (March 4) at Media Laboratory, MIT, Cambridge, MA.
Minsky, M. and Papert, S., 1969. *Perceptrons*. Cambridge: MIT Press.
Mitchell, W. J., 1990a. The Logic *of Architecture*. Cambridge: MIT Press.
Mitchell, W. J., 1990b. "A New Agenda for Computer-Aided Design," in M. McCullough, W. J. Mitchell, and P. Purcell (eds.), *The Electronic Design Studio*. Cambridge: MIT Press, p. 1.
Mitchell, W. J., 1992. "The Uses of Inconsistency in Design," in Y. E. Kalay (ed.), *Principles of Computer-Aided Design: Evaluating and Predicting Design Performance. New York: John Wiley, p. 1*.
Nagakura, T., 1990. "Shape Recognition and Transformation: A Script-based Approach," in M. McCullough, W. J. Mitchell, and P. Purcell (eds.), *The Electronic Design Studio*. Cambridge: MIT Press, p. 149.
Newell, A., 1990. *Unified Theories of Cognition*. Cambridge: Harvard University Press.
Pollack, J.,1988. "Implications of Recursive Distributed Representations," in D. Touretzky (ed.), *Advances in Neural Information Processing Systems*. San Mateo: Morgan Kaufman.
Rich, E. and Knight, *K.,* 1991. *Artificial Intelligence* (Second edition). New York: McGraw-Hill.
Rock, I., 1984. *Perception*. New York: Scientific American Books.
Rumelhart, D. E. and Zipser, D., 1986. "Feature Discovery by Competitive Learning," in D. Rumelhart, I. McClelland, and the PDP Research Group (eds.), *Parallel Distributed Processing*. Cambridge: MIT Press, p. 151.
Rumelhart, D. E., Hinton, G. E. and Williams, R. J., 1986a. "Learning Internal Representations by Error Propagation," in D. Rumelhart, J. McClelland, and the PDP Research Group (eds.), *Parallel Distributed Processing*. Cambridge: MIT Press, p. 318.
Rumelhart, D. E., McClelland, I. L. and the PDP Research Group (eds.), 1986b. *Parallel Distributed Processing*. Cambridge: MIT Press.
Selfridge, O. G., 1955. "Pattern Recognition and Modern Computers," in *Proceedings,* Western Joint Computer Conference. New York: IEEE.
Simon, H. A., 1978. "Information-Processing Theory of Human Problem Solving," in W. *K*. Estes (ed.), *Handbook of Learning and Cognitive Processes, Vol. 5: Human Information Processing*. Hilisdale, NJ: Lawrence Erlbaum Associates, p. 271.
Simon, H. A., 1981. *The Sciences of the Artificial* (Second edition). Cambridge: MIT Press.
Stiny, G., 1980. "Introduction to Shape and Shape Grammars," *Environment and Planning B: Planning and Design, p*. 343-351.
Stiny, G., 1990. "What Designers do that Computers Should," in M. McCullough, W. J. Mitchell, and P. Purcell (eds.), *The Electronic Design Studio*. Cambridge: MIT Press, p. 17.

Stiny, G. and Mitchell, W. J., 1978. "The Palladian Grammars," *Environment and Planning B*, p. 5-18.
Tan, M., 1990. "Closing in on an Open Problem: Reasons and a Strategy to Encode Emergent Subshapes," in J. P. Jordan (ed.) *From Research to Practice, ACADIA* '90 Workshop, Montana State University, Big Sky, MT, p. 5.

Appendix
Performance of the 49-N-49 BackProp Networks

Training a BackProp network is very difficult - in the two applications of ill-processed shapes, training a network well requires 179-706 learning epochs to solve problems and, roughly speaking, every 100 epochs takes at least one hour to minimize errors on a Macintosh LC computer. It is difficult to decide the number of hidden units suitable to the problem at hand. It may be generally true that:

- With respect to problem solving, a BackProp network with less hidden units may not be able to solve the problem; whereas one with more hidden units may over-specialize the problem.
- With respect to the training process, when all other variables and parameters are kept the same, the more hidden units a BackProp network possesses, the more efficient its learning processes will be.

As mentioned previously, the major objective of this research is to solve properly some problems regarding design shapes. However, the problem-solving performance of the BackProp network with various hidden units and the learning rate parameter is also important for future studies. Therefore, in this project, three BackProp networks with different number of hidden units, 49-12-49, 49-25-49 and 49-49-49, were constructed to solve shape recognition and transformation problems so that the above two features of BackProp's performance could be revealed, compared, and discussed.

Table 1: Higher Learning Rate - 0.5

Structure	Run	Epochs	Solved?	Stuck error
49-12-49	Run-1	1000	No	33.366
	Run-2	1000	No	33.314
	Run-3	1000	No	29.503
49-25-49	Run-1	477	Yes	-
	Run-2	1000	No	38.715
	Run-3	1000	No	26.462
49-49-49	Run-1	264	Yes	-
	Run-2	601	Yes	-
	Run-3	1000	No, almost	0.064

A.1 Higher Learning Rate - 0.5

To pursue more efficient learning, the learning rate should be as large as possible without leading to "oscillation." First, the higher learning rate of 0.5 was selected as the starting point since the value of 0.5 is most commonly used in many connectionist models (McClelland and Rumelhart, 1989). Without a doubt, this selection was risky: the valuewas too large for the design shape problem, it could lead to heavy oscillation and the problem was not solved.

Three runs were taken for each network. For simplicity, if the network could not solve the problem (the summed squared error was still higher than 0.05) over 1000 training epochs, the run was ended. The performances in this simulation are shown in Table 1. The results indicate that with a higher learning rate of 0.5, only the 49-49-49 network can perform problem-solving of the proposed design shapes properly. The learning processes of the networks with 25 and 12 hidden units are stuck in higher values of the summed squared errors which probably are the local minima. During training processes, because of the too-high learning rate of 0.5, the gradient descent processes had many oscillations for all the three networks.

Table 2: Higher Learning Rate - 0.25

Structure	Run	Epochs	Solved?	Stuck error
49-12-49	Run-1	1000	No, almost	33.366
	Run-2	706	Yes	-
	Run-3	2000	No	21.143
49-25-49	Run-1	419	Yes	-
	Run-2	362	Yes	-
	Run-3	318	Yes	-
49-49-49	Run-1	179	Yes	-
	Run-2	210	Yes	-
	Run-3	199	Yes	-

A.2 Lower Learning Rate - 0.25

Because heavy oscillation occurred in previous simulations, the learning rate was changed from 0.5 to 0.25. Predictably, gradient descent learning will perform more harmoniously in this case. Other parameters, such as the momentum constant and the error criterion, remain accordingly in this simulation. The performances are as shown in Table 2. In the performances of the 49-12-49 network, the design shapes problem could be solved once (the second run). There were over 1000 training epochs in the third run, with the total error of 3.127. Therefore, it was decided to continue the network for another 1000 epochs to observe results. As indicated in Table 2, however, the situation grew worse. In the performances of the 49-25-49 network, the same problems could be solved properly over 419, 362, and 318 training epochs, respectively. The learning process in this case was more efficient than the 49-12-49 BackProp network. Finally, in the performance of the 49-49-49 network, the problem was well-solved over 179,210, and 199 training epochs, respectively. It is also clear that the 49-49-49 BackProp's learning process is faster than the previous two networks.

A.3 Remarks

As mentioned previously in this appendix, a higher learning rate is more likely to lead to oscillation. Observed from the above two simulations with two different learning rates, this point is clearly proven. It is also noted that there is no absolute "good" learning rate; the learning rate of 0.5 performs well for problems in most of connectionist models, but it is too high for the design shape problems solved above.

The importance of the number of hidden units was briefly described in the begin-fling of this appendix. From the second table (with a learning rate of 0.25), the 49-12-49 network could only solve the problem once; however, the networks with more hidden units solved the problem perfectly. Thus, the problem solving capability of the network with less hidden units is less sufficient than those networks with more hidden units. When focusing on the 49-25-49 and 49-49-49 networks, the behaviors of both gradient descents were good. Therefore, if the 49-25-49 network is able to solve problems properly; perhaps the 49-4949 network over-specialized the problem. The validation is very consistent with the general features of the Backprop network mentioned previously.