THE SEARCHING MECHANISM OF ARTIFICIAL NEURAL NETWORKS in CASE-BASED Building Design

Ju-Hung Lan , Mao-Lin Chiu , Chien-Hung Wei

Dept. of Architecture,Kao Yuan Institute of Technology

E-MAIL: ***jhlan@cc.kyit.edu.tw***

Dept. of Architecture,National Cheng Kung University
E-MAIL: mc2p@mail.ncku.edu.tw
Dept. Of Transportation & Communication Management,National Cheng Kung University
E-MAIL: louiswei@mail.ncku.edu.tw

Abstract: A Case-based reasoning (CBR) system generally includes three parts: case recall, case adaptation, and a case library. The objective of this research is to develop a searching mechanism for the case recall of a CBR system in the context of building design. Our work focuses on developing the similarity relationship among building design cases. The similarity relationship serves as a searching mechanism in retrieving relevant design cases. In this paper, we present an approach to applying artificial neural networks (ANN) to serve as a searching mechanism of a CBR system. The results of this research include: 1) to present suitable knowledge representations of various design cases; and 2) to develop a searching mechanism of a CBR system.

Keywords: Artificial Neural Network, Case-based Reasoning, Building Design

1. INTRODUCTION

1.1 Background and motivation

Case-based reasoning (CBR) is an AI method to solve problems based on analogical reasoning with precedents. CBR has been applied to various applications, such as planning, design, and diagnosis [1][2]. A CBR system generally includes three parts: case recall, case adaptation, and a case library. A framework of a CBR system has been presented [3].

In real world situations, there are many building design problems that could not be solved by standard logic reasoning methods. They are usually solved by referring past similar building design experiences. This reflects the way that human beings solve problems. The way of biological neural system to solve problems is different from traditional computational approaches. Artificial neural network (ANN) is a methodology of problem solving based on simulating the process of biological brain to solve problems [4].

Most CBR systems use relational databases to build their case libraries. Cases are stored based on attributes with index for further retriving. The result of searching a relational database is either some cases matching query or nothing. This kind of searching mechanism could not be applied in problems with incomplete or fuzzy information. In contrast, an ANN system allows ambiguous, uncertain factors of problems to be solved. This feature of ANN provides a possible solution to the searching limitation that general CBR systems currently face.

1.2 Objective and problem

The objective of this research is to develop a searching mechanism for the case recall of a CBR system in the context of building design. Our work focuses on developing the similarity relationship among building design cases. The similarity relationship serves as a searching mechanism in retrieving relevant design cases for a specific building design problem. The searching mechanisms generally used in CBR are based on attributes query, which can only find the cases exactly matching the query attributes. However, in the building design field, design problems usually are hard to specify with their attributes. Furthermore, there are no identical building design cases in the world. Our argument here is, even though users cannot find exactly 100% matching cases, it is very likely to have some design cases providing relevant information for the new design problem that users currently face. Thus, we suggest a method to retrieve cases by a fuzzy way with the notion of case approximation or similarity.

1.3 Approach

In this paper, we present an approach to applying ANN as a searching mechanism of a CBR system. The ANN allows fuzzy and uncertain factors in input data. Our intension is that the similarity of existing design cases to a specific design problem could be classified as differential weights by percentages. To fulfill the requirement, we first implement a research prototype named CBA (Case Base for Architecture) based on the theory of case-based reasoning. The goal of CBA is to develop an electronic tool for assisting building design. Currently, the prototype of CBA includes a database with office building design cases. In this paper, we attemp to incorporate an artificial neural system as a searching mechanism in CBA. We use NeuralWorks Professional II Plus for system development and adopt the back propagation network (BPN) to construct our neural system with a three-layer structure. The office building cases in CBA are used as the sources for the input layer and the output layer for training and testing our neural system.

2. DESIGN KNOWLEDGE OF BUILDING DESIGN CASES

2.1 The design knowledge representation

Currently, the main domain of CBA is office building design cases. In this paper, we use the office building design cases in CBA as sample data to construct our neural systems. The major challenge here is how to get the design knowledge of office building design cases. According to our research in CBA, a case can be represented by 1) basic information; 2) analysis information; and 3) user recommentation. Since it is difficult to get the original design knowledge for each office building design case, we analyze current available databases and literatures for summarizing the attributes of an office building design case. The result is shown in figure 1. The way of recording building attributes provides a means to extract the design knowledge of a builing case. In this paper, we use these attributes as the starting point to represent the design knowledge of office building design cases.

Building name	Construction period	Story height	Primary exterior finishing
Location	Construction company	Ceiling height	Simple finishing table
Owner	Structural body	Site area	Plane drawings
Primary Purpose	Electric	Building area	Façade drawings
People Capacity	Water supply system	Total floor area	Section drawings
Ownership	Drainage system	Each floor area	Site location drawing
Architect	Landscape	Purpose of each floor	Detail drawings
Design period	Structural body construction	Building area ratio	Product photos
Structure	Foundation	Floor area ratio	Other drawings
Mechanical	Scale	Purpose zoning	Concept drawings
Air conditioning	Upper ground story	Primary equipment	
Interior design	Lower ground story	Air conditioning	
Furniture	Roof height	Sanitation	
Landscape	Total height	Hot water	
Supervision	Main span	Disaster mitigation	

Figure 1: The attributes of office building cases

2.2 The design knowledge extraction

Our intension in this research is to explore the application of ANN in case-based building design. We narrow down our scope to extract four key attributes with respect to general office building design knowledge. The four attributes are:

1) the type of service core（CORE）.
2) the road relationship of building site（ROAD）.
3) the story of an office building（STORY）.
4) the total floor area（AREA）.

To solve a layout design problem of an office building design case, we usually start from the site analysis, for instance, the road relationship of the building site implies the building orientation and allocation of entrance. Another empirical approach to solve a layout design problem is the design and allocation of the service core of an office building design case. Thus, we use the CORE and ROAD attributes to serve as the layout design knowledge of an office building design case in this research. On the other hand, the building story reflects the scale and technology of a low-rise, middle-rise, or high-rise building. Also, the total floor area usually is relating with the design knowledge of building scale and building code. Therefore, we use the STORY and AREA attributes to serve as the scale design knowledge of an office building design case in this paper.

3. THE SEARCHING MECHANISM

Our research intension is to build the layout similarity relationship （LAYOUT） and the scale similarity relationship （SCALE） between existing office building cases and new office building design problems. The shortcomings of existing searching mechanisms based on query attributes are mentioned above. The notion of a comprehensive similarity relationship among building design cases that this research intends to achieve is shown in figure 2. We argue that there should have a way to level the similarity relationship by percentage for each existing building case to the new design problem. To fulfill the intension, we use the CORE and ROAD attributes of office building cases to construct the LAYOUT neural system. The LAYOUT neural system serves as retrieving office building design cases with similar layout design knowledge. On the other hand, we use the STORY and AREA attributes to construct the SCALE neural system. The SCALE neural system serves as retrieving office building design cases with similar scale design knowledge. The frameworks of two searching mechanisms that this research intends to achieve are shown in figure 3.

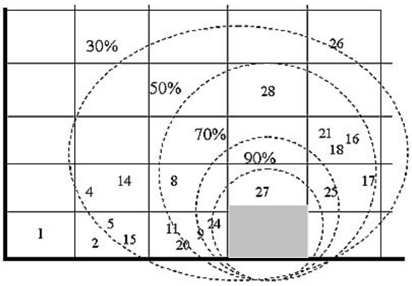


Figure 2: The notion of similarity relationships

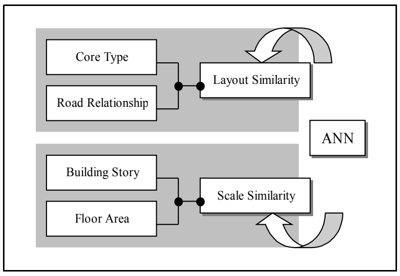


Figure 3: The frameworks of searching mechanisms

4. CASES ANALYSIS

The number of real office building design cases collected in this paper is 27. Most cases are located in Taiwan. Some cases are located in China. To conduct our research, we first record these cases according to the four key attributes (CORE, ROAD, STORY and AREA) that we mentioned above. The sample records of office building cases are shown in table 1.

Table 1: The sample records of office building cases

Case name	CORE	ROAD	STORY	AREA (m^2)
Wang-lae	Single edge (3)	Single edge (1)	6	851
Sin-kung	Central belt (2)	Peninsula (3)	50	118229.8
San-hi	Twin edges (4)	Block (4)	41	101235
Shen-yang	Four corners (5)	Corner (2)	10	22771.09
Yuan-chi	Twin towers (6)	Block (4)	41	159206.3

After recording the 27 office building cases, we use one matrix to locate these 27 cases according to the CORE and ROAD attributes, and another matrix to locate 27 cases according to the STORY and AREA attributes. The distribution results of these 27 cases on both matrices are shown in table 2. By analyzing both matrices, we can find these 27 cases are not equally distributed. Some cells in the matrices is gathered with many cases, however, many cells are empty. Since both matrices actually imply the LAYOUT and SCALE neural systems that we intend to develop, the unequal distribution results of 27 cases are not competent to design a neural system with well performance from statistic viewpoint.

The office building cases are the resources to construct our neural systems. A good resource should be: 1) the number of samples is sufficient; 2) the distribution of samples is equal; and 3) the data of samples is correct. A good resource is a key to construct a successful neural system. To avoid the situation mentioned above, we need more office building cases to get equal distribution results on both matrices. Thus, we refer the records in table 1 and simulate another 63 office building design cases. Consequently, the total number of office building cases in this paper is 90. We further divide these 90 cases into two parts. One part includes 70 cases for training our neural systems. The other part includes 20 cases for testing the trained neural systems. The distribution results on both matrices of the 70 cases are shown in table 3. These 70 cases are distributed more equally on both matrices. Later we will use the 90 cases to construct our neural systems.

Table 2: The distribution results of 27 cases

Matrix	Distribution Result						
LAYOUT CORE & ROAD							
	兩面臨路 (5)			6 10 22			
	街廓 (4)	14		11 13 26	3 5 23 25		9
	半島 (3)	27	24	2 7 8 12 17		21	
	角地 (2)			4 15 18 19		16	
	單邊臨路 (1)		20	1			
	服務核	中央 (1)	中間帶狀 (2)	單邊核 (3)	兩邊核 (4)	四角核 (5)	雙塔式 (6)
SCALE STORY & AREA	建築面積						
	50000以上			5 17	3 9 21 24		
	40000~50000			4 12			
	30000~40000			11 18 23			
	20000~30000		13 14 16 25				
	10000~20000		2 6 7 26				
	10000以下	1	8 10 15 19 20 22 27				
	樓層數	6F以下	7F~14F	15F~30F	30F以上		

Table 3: The distribution results of 70 cases

Matrix	Distribution Result						
LAYOUT CORE & ROAD	道路關係						
	兩面臨路 (5)	40 65	41 66	6 10 22	42 67	43 68	44 69
	街廓 (4)	14 61	38 62	11 13 26	3 5 23 25	39 63	9 64
	半島 (3)	27 59	24 48	2 7 8 12 17	36 47 70	21 49	37 60
	角地 (2)	32 56	33 46	4 15 18 19	34 50	16 57	35 58
	單邊臨路 (1)	28 45	20 51	1 52	29 53	30 54	31 55
	服務核	中央 (1)	中間帶狀 (2)	單邊核 (3)	兩邊核 (4)	四角核 (5)	雙塔式 (6)
SCALE STORY & AREA	建築面積						
	50000以上	43 59	44 60	5 17 69	3 9 21 24		
	40000~50000	40 57	41 48 67	4 12 68	42 58		
	30000~40000	37 55	38 46 66	11 18 23 65	39 56		
	20000~30000	34 53	13 14 16 25	35 47 64	36 54		
	10000~20000	31 45 61	2 6 7 26 62	32 49 63	33 52		
	10000以下	1 28 70	8 10 15 19 20 22 27	29 50	30 51		
	樓層數	6F以下	7F~14F	15F~30F	31F以上		

5. VARIABLES ENCODING

A basic neuron (processing element, PE) in an ANN system transfers an input value into an output value. To use the 70 office building cases to construct our neural systems, we need to identify the input variables and output variables and encode these variables as necessary. The following section describe the steps of encoding these variables for developing our neural systems.

5.1 The input variables

The four attributes (CORE, ROAD, STORY, and SCALE) of office building cases are used as the input variables to constuct our neural systems. The CORE and ROAD are the input variables to construct the LAYOUT neural system. The STORY and AREA are the input variables to construct the SCALE neural system. Before using these four variables, we need to encode them first. The CORE and ROAD variables are non-sequencial classification variables. In contrast, the STORY and AREA variables are sequencial classification variables. Both non-sequencial and sequencial variables are encoded with single PE or multi-PEs for further performance comparison. The encoding principles of the four input variables are shown in table 4. To illustrate it in more detail, for instance, the value of a CORE variable is an integer between 1 and 6 while it is encoded as single PE. However, while it is encoded as multi-PEs, it will need 6 PEs to represent the CORE variable, and the value of each PE is either 1 or 0.

Table 4: The encoding principles of input variables

Encoding	CORE	ROAD	STORY	AREA
a31 building				
Single PE	6	1	1	2
Multi-PEs	0 0 0 0 0 1	1 0 0 0 0	1 0 0 0	0 1 0 0 0 0
a33 building				
Single PE	2	2	4	2
Multi-PEs	0 1 0 0 0 0	0 1 0 0 0	0 0 0 1	0 1 0 0 0 0
Encoding Principles	CORE	ROAD	STORY	AREA (m^2)
	1) Central	1) Single edge	1) <=6F	1) <=10000
	2) Central belt	2) Corner	2) 7F~14F	2) 10000~20000
	3) Single edge	3) Peninsula	3) 15F~30F	3) 20000~30000
	4) Twin corners	4) Block	4) >=31F	4) 30000~40000
	5) Twin edges	5) Two sides		5) 40000~50000
	6) Twin towers			6) >=50000

5.2 The output variables

What are the output variables of our neural systems? The research intension, we mentioned before, is to get the similarity relationship between current building design problem and existing office building design cases. Thus, the similarity relationship should be the expected output. However, how to represent it by corresponding output variables to achieve our goal? Our solution is to use multi-PEs to represent and encode the corresponding output variables. Each PE carries the corresponding similarity value of existing cases to current building design problem. The value of each PE is a real number between 0 and 1. The overall concept of similarity relationship is shown as table 5. In table 5, for instance, the output value of a29 building via a vector representation is (0.43, 0.53, 0.43, 0.33, 0.68, 0.83, 0.68, 0.38, 0.83, 0.98, 0.83, 0.53, 0.68, 0.83, 0.68, 0.43, 0.43, 0.53, 0.43, 0.33, 0.18, 0.18, 0.18, 0.18). The order of the values in the vector is corresponding to the values in the matrix in table 5, from left to right and bottom to up. The cell with gray mark in table 5 means the location of current building design problem. The value in each cell means the existing cases at that cell with the same similarity value to current design problem. One thing important here is the similarity value in table 5 is customizable. We could decide the similarity value of each cell by our wish. Currently, we use a simple formula to automate calculate the value of each cell based on the distance relationship among cells. To represent the output variables of a29 building, we need 24 PEs to encode them. Clearly, we will need 30 PEs to encode the output variables of a28 building to develop the LAYOUT neural system.

Once the neural systems are developed, from a scenario viewpoint, if we get an output result like a29 building in table 5 after inputting the variables of a new design problem, then we should be able to know the cases located at the cell (2,3) with 98% similarity, the cases located at cell (3,2) with 68% similarity, and the cases located at cell (4,5) with 33% similarity to current building design problem.

Table 5: The concept of similarity relationship and encoding principles of output variables

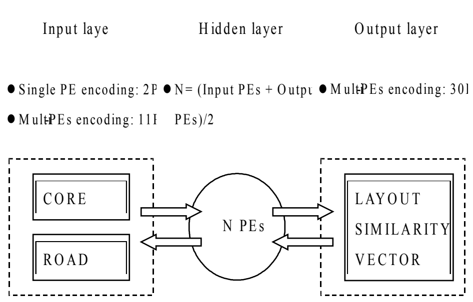
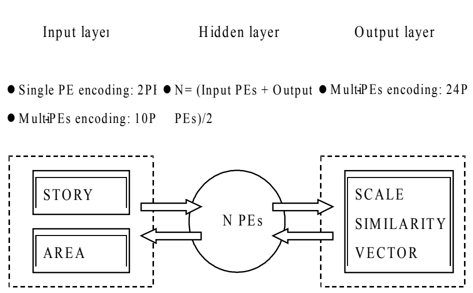
ROAD							
5	*0.09*	*0.29*	*0.39*	*0.49*	*0.39*	*0.29*	
4	*0.09*	*0.39*	*0.64*	*0.79*	*0.64*	*0.39*	
3	*0.09*	*0.49*	*0.79*	*0.99*	*0.79*	*0.49*	
2	*0.09*	*0.39*	*0.64*	*0.79*	*0.64*	*0.39*	
1	*0.09*	*0.29*	*0.39*	*0.49*	*0.39*	*0.29*	CORE
a28	**1**	**2**	**3**	**4**	**5**	**6**	
AREA							
6	*0.18*	*0.18*	*0.18*	*0.18*			
5	*0.43*	*0.53*	*0.43*	*0.33*			
4	*0.68*	*0.83*	*0.68*	*0.43*			
3	*0.83*	*0.98*	*0.83*	*0.53*			
2	*0.68*	*0.83*	*0.68*	*0.38*			
1	*0.43*	*0.53*	*0.43*	*0.33*	STORY		
a29	**1**	**2**	**3**	**4**			

6. SYSTEM IMPLEMENTATION

6.1 System structure and network parameters

The system structures of two neural systems that this research intends to develop are shown in table 6. We adopt back-propagation network (BPN) to construct our neural systems since it is the most popular approach to build a supervised learning network.

Table 6: The system structures of two neural systems

<table>
<tr><th>System</th><th>ANN structure</th></tr>
<tr><td>LAYOUT</td><td>
</td></tr>
<tr><td>SCALE</td><td>
</td></tr>
</table>

We use a general three-layer structure to construct our neural systems. The input layer of LAYOUT includes 2 PEs (single PE encoding) or 11 PEs (multi-PEs encoding). The output layer of LAYOUT includes 30 PEs (multi-PEs encoding) based on the similarity concept in table 5. On the other hand, the input layer of SCALE includes 2 PEs (single PE encoding) or 10 PEs (multi-PEs encoding). The output layer of SCALE includes 24 PEs (multi-PEs encoding). The number of PEs in the hidden layer of both neural systems is an emperical value: (Input PEs + Output PEs)/2. Other network parameters relating with constructing our neural systems all use empirical settings or default values in our development tool. Part important network parameters are listed in table 7.

Table 7: The settings of network parameters

Network type	Network parameters
BPN	Learning rule: Delta-Rule Transfer function: Sigmoid Learning rate: 0.1~1.0 Momentum: 0.0~0.9 Instrument: RMS Error, Network Weights, Classification Rate

6.2 The neural system training process

Based on the system structures described in table 6, we start to construct the LAYOUT and SCALE neural systems. The number of PEs in the three-layer structure is described above. Two sample diagrams about the training process of our neural systems are shown in figure 4 and figure 5.

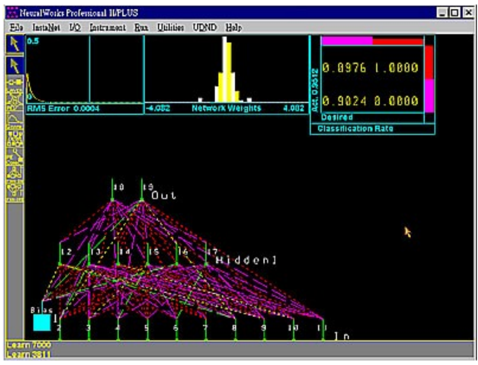


Figure 4: LAYOUT sample training diagram（2PEs+16PEs+30PEs）

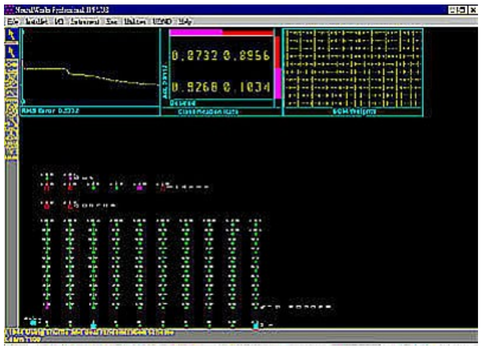


Figure 5: SCALE sample training diagram（10PEs+17PEs+24PEs）

7. RESULTS AND DISCUSSIONS

7.1 Results

The number of network structures to construct the LAYOUT and SCALE neural systems is two for each. The training and testing results of the two neural systems are listed in table 8.

Table 8: The training and testing results

Input PEs+ Hidden PEs+ Output PEs	RMS Error	Classification Rate	Run/Test	Run/Learn
LAYOUT neural system				
2+16+30	0.0343	1.0000	1.0000	7000
11+20+30	0.0362	1.0000	0.4667	9000
SCALE neural system				
2+13+24	0.0349	1.0000	0.4167	10000
10+17+24	0.0294	1.0000	0.4167	12000

7.2 Discussions

Based on the results shown in table 8, we find the four network structures that perform well during training phase. The value of classification rate of each structure is 100% during training phase. However, we also find the classification rate during testing phase is not good except the network structure composed by ‘2+6+30’ PEs. The value of classification rate during testing phase is only getting

42% in other three network structures. To figure it out, we start to investigate the contents of the output files (*.nnr). However, the output values in those output files all seem reasonable. They are quite close to our expecting results, even though they are not exactly the same. Also, we didn't find any information relating with *over learning* during training and testing phases.

So far, to explain the situation mentioned above, we could only make some possible judgments and leave it for further study. One possible explanation might be the insufficient number of building cases that we collect. Although we already simulate another 63 building cases, the number and the quality of these cases might still be not able to construct a comprehensive neural system. Another possible explanation might be the formula of classification rate that we use in our tool. Perhaps we should go through each output file manually to see if there are any misunderstandings about the formula of classification rate.

8. SUMMARY AND FUTURE WORK

8.1 Summary

Retriving the right building design cases at the right time is an challenge to the application of CBR to building design problem. This is more important especially while there are many building design cases in a case library. The use of previous building design cases as a starting point could have significant effects on the quality of the final design [3].

The concept of similarity provides a means to retrieve previous cases without the shortcomings of searching by query attributes. In this paper, we propose a solution to the case retriving problem of a CBR system by applying ANN. Using the neural systems we developed, an office building designer could find relevant design cases according to the building design problem that he/she currently faces. The goal of our neural system is not only provide the most matching cases but also providing all previous design cases with different similarity percepentage to current building design problems.

8.2 Future work

The studies of ANN in construction engineering and cost estimation attracted more research attention in the last decade. This paper provides an ANN approach to serving a searching mechanism in the field of case-based building design. The results provide a reference for researchers with the same inetrests.

Currently, the link between our neural systems and CBA is undertaken. The NeuralWorks Professional II provides a means to transfer a neural system into C code. We attempt to link our neural system in web environment to serve the searching mechanism in CBA. Besides, from our research, different building types will need different neural systems to serve. Consequently, there are many neural systems to serve different building design problems in CBA. Therefore, the optimizion issue will getting more important and need to be considered carefully.

Another possible approach by applying ANN to serve as a searching mechanism might be the abilities of pattern recognition. Literature reviews show there are many research progresses of ANN in pattern recognition [4][5]. In building design field, the contents of building drawings include invaluable building design knowledge. If we could recognize the building design knowledge by building design drawings, we should be able to retrive relevant building design cases by comparing building drawings. Thus, the ability of pattern recognition of ANN provides another research potential of developing a searching mechanism to our work.

Acknowledgments

The research reported here is supported in part by National Science Council in Taiwan under contract NSC89-2211-E006-062. The authors would like to express their gratitude.

REFERENCES

[1] Kolodner, J. L., *Case-Based Reasoning*, Morgan Kaufmann, 1993

[2] Leake, D. B., *Case-Based Reasoning – Experiences, Lessons, & Future Dircetions*, The MIT Press, 1996

[3] Maher, M. L., Balachandran, M. B., Zhang, D. M., *Case-Based Reasoning in Design*, Lawrence Erlbaum Associates, 1995

[4] Nelson M. M., Illingworth W. T., *A practical guide to Neural Nets*, Addison-Wesley , 1991

[5] Bose, N. K., *Neural network fundamentals with graphs, algorithms, and application*, McGraw-Hill, 1996