データベースシステム　94－12
（１９９３．７．２１）

自己組織化ニューラルネットによるデータベースアクセス

阿江　忠、　近松　良知、川上　勝康、河田　学

広島大学　工学部

ニューラルネットをデータベースシステムに導入する試みについて紹介する。システムは、１）データベースのカテゴリ化、２）（カテゴリ化された）データベース間のオペレーション、という２つのフェーズからなる。それぞれに、自己組織化ニューラルネットを導入することにより得られるメリットについて述べる。目的はリアルタイムシステムにおけるデータベースへのアクセス方法にあるが、個々にはいろいろな問題も含まれており、本報告ではそれらの概説を述べる。

DATABASE ACCESS USING SELF-ORGANIZING NEURAL NETS

Tadashi Ae, Yoshitomo Chikamatsu, Katsuyasu Kawakami, Manabu Kawata

Faculty of Engineering, Hiroshima University

Higashi-Hiroshima, 724 Japan

In this report, we propose to introduce the neural nets to the database system. For the real-time applications, the rapid data processing using neural nets plays an important role, but several problems must be solved to achieve it. We state the outline of essential points, and will have a good solution each.

Outline of Our Research

We introduce the neural nets to the database system. This may be regarded as the introduction of an AI (artificial intelligence) technique as in Fig.1, but it differs explicitly as follows; The conventional AI technique uses the explicit knowledge representation, where deduction and/or induction are applied by rules[1,2]. The neural net, however, has *no explicit knowledge representation*, and therefore, can avoid a cumbersome problem, the *knowledge acquisition*.

The **MBR** (Memory-Based Reasoning) system[3] is a typical example with the same research direction, when regarding the MBR model is an extremal case of neural nets. The MBR system, however, requires a massively-parallel processor such as the Connection Machine, which is not appropriate for a client machine. It may fit a server machine, but we focus on a smaller machine, because the client wants to have each own machine with his/her philosophy, and the massively-parallel processor is too huge.

Our system is divided into two phases as in Fig.2, i.e., the *classification* and the *operation*. The classification is made by categorizing the database as in Fig.3 , and this phase through the neural net differs in *not* using explicit knowledge representations.

The second phase, the *operation*, includes a wide variety. A simple example is " *join* " in the relational databases, but it is not exact join as in Fig.4. We call the join between categorized databases the approximate join (in short, A-join), which is not exact but executed very rapidly. (Our purpose is in the real-time applications[4].)

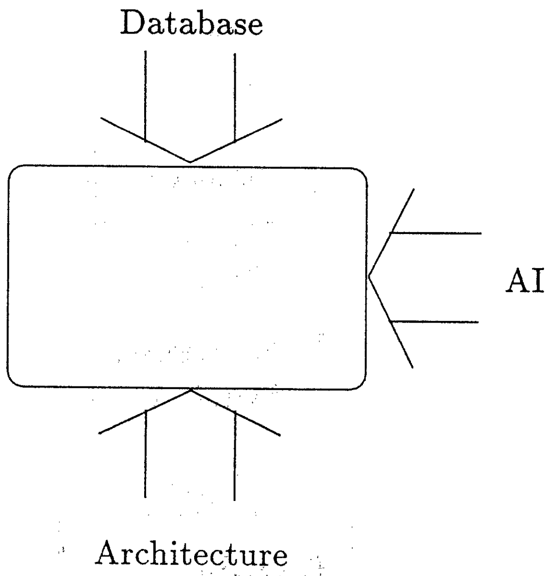
More interesting operations are the cooperative and/or competitive behaviors as in Fig.5 and Fig.6, respectively, which are realized by the multiple neural nets. The interactive behaviors among neural nets are regarded as those among machines (e.g, automata), and the AND/OR-parallel behavior is an elemental factor. When applying the system to the real-time applications, however, not every behavior is parallel, and the time-dependent behaviors are also indispensable. In this sense, the *neural net with states* as in Fig.7 becomes an appropriate model to the database access. The neural net with states are also constructed by *learning* as in the conventional neural net.

The implementing issue (that is our final purpose!) is not described here, but show several references[5, 13-20].

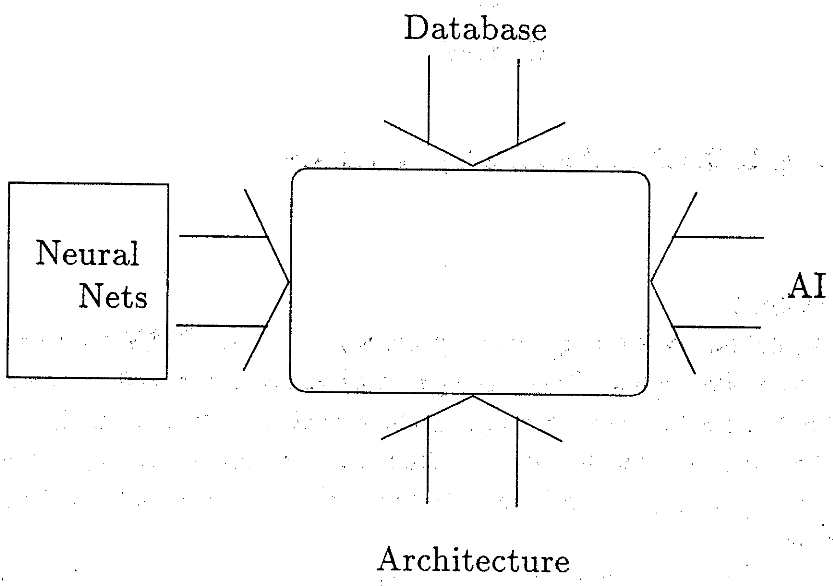
In this report we sketch briefly each point, because each section consists of deep contents, although some of them are not completed yet.

Information Filtering through Neural Net

Any kind of data is applicable for our system as in Fig.2, but the preprocess for some kind of data (e.g., the sentence written in a natural language) may be required when the data set is not a form of database. This preprocess is similar to that in the MBR system[3,5]. To focus on the main discussion, suppose that the database be a typical one



(a) Introduction of AI Technique .



(b) Introduction of Neural Nets .

Fig. 1. Introduction of AI Technique to Database System .

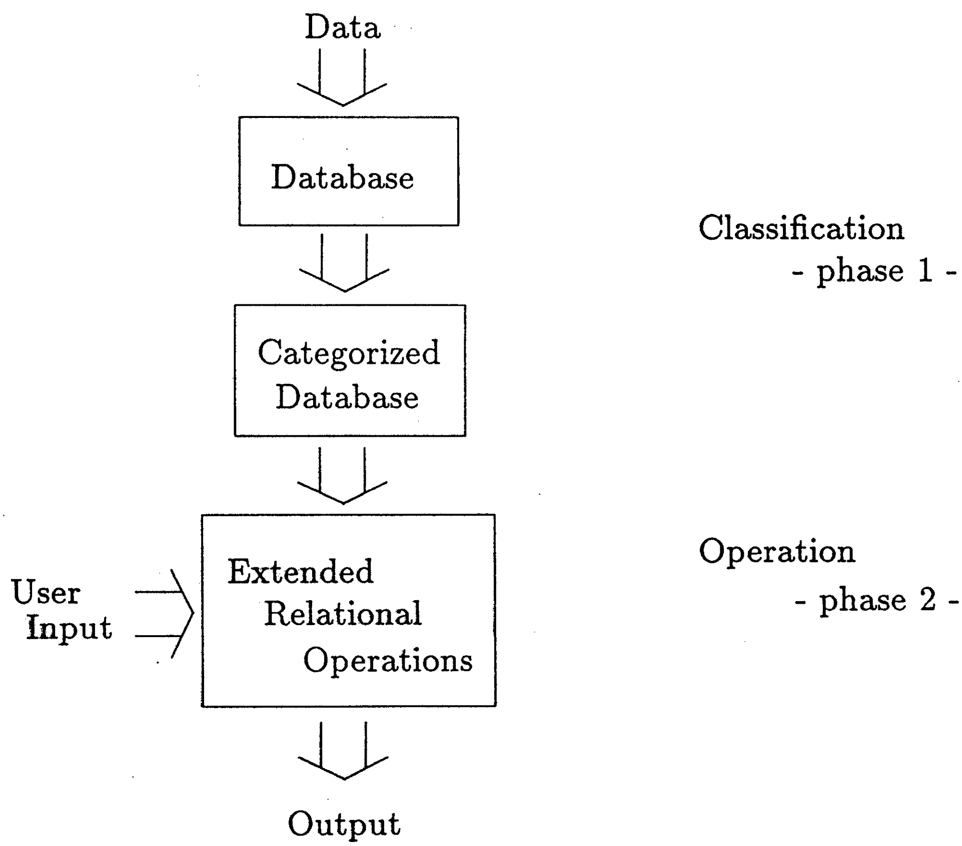


Fig. 2. Database Access using Neural Nets .

(e.g., the table).

As the neural net, the multilayer perceptron type (in short, **MLP**) is well-known with the error Back Propagation learning algorithm(in short, **BP**)[6]. The MLP is also applied to the sematic applications (e.g., the natural language processing), but the BP seems not to be appropriate for such applications. On the other hand, the self-organizing type is recently coming up, and the Kohonen net (in short, **K-net**) is the most popular among them.

The AI technique using an explicit knowledge representation is based on the inference (deduction and/or induction). For the problem to obtain easily the knowledge the AI technique is enough powerful, but for the cumbersome problem it is not. We propose to use the neural net (which has no explicit knowledge representation) instead of getting the explicit knowledge representation (e.g., rules). More precisely, we propose to use the K-net as a self-organizing neural net for the database access[5]. The K-net is known to be powerful for the pattern recognition and the signal processing[7], but we insist that the K-net is powerful also for the deep semantic application such as the database. When applying the LVQ (Learning Vector Quantization) algorithm to the database, it behaves just like the case of the pattern or the signal, if the database consists of the quantitative expression. (Otherwise, we need the preprocess, but the K-net can be applied also for the qualitative expression.)

As a result, the database is categorized as in Fig.3. When representing this function by f as in Fig.3, f is understood to be a set of rules in the AI technique. In the classification using the neural net, however, f is not clear, but enough powerful. Mathematically, f can be represented by a set of rules, if we get a categorized partition, but it is meaningless for the database with complex factors. The mathematical characterization of f using the neural net is partially reported in [8,9].

This is the first feature of our system. The MBR is the extremal case that f is the identity mapping, and no partition occurs. Then, the MBR requires a tremendously large space, while our system is realized by a small system.

Similarity-Based Reasoning by Neural Nets

We can realize a relational operation in our system adding Ichikawa's database[10] to the K-net. The "join" between categorized databases differs from the exact *join*, and therefore, we call it the approximate join (in short, **A-join**) to distinguish it with the exact join. Moreover, we call the reasoning on the approximate operations the similarity-based reasoning (in short, **SBR**).

Example.

we show a list of used cars;

USED CARS of five elements : (name, year, price, purpose, imported or not).

(Rover,	'90,	$10,000,	leisure,	imported)
(Audi,	'88,	$20,000,	leisure,	imported)
(Peugeot,	'90,	$15,000,	business,	imported)
(Civic,	'90,	$10,000,	business,	domestic).

For filtering of the database, we need the category for each reference data (vector) to apply Kohonen's LVQ algorithm. Then, we add the category to the reference vector (if it is not attached yet). For the case of easy categorization the learning time steps are extremely reduced.

The join is classified into many types, but we focus on only Ichikawa's join[10]. The join operation between two relational databases of n elements each takes O(n^2) steps, and therefore, the hardware acceleration is needed for the large-scale databases in real-

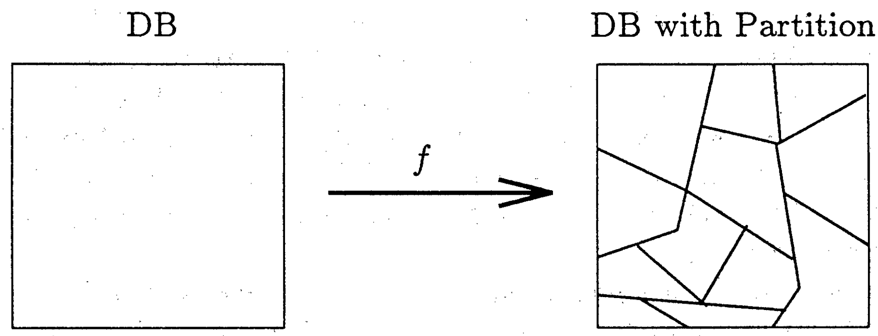


Fig. 3. Classification Phase .

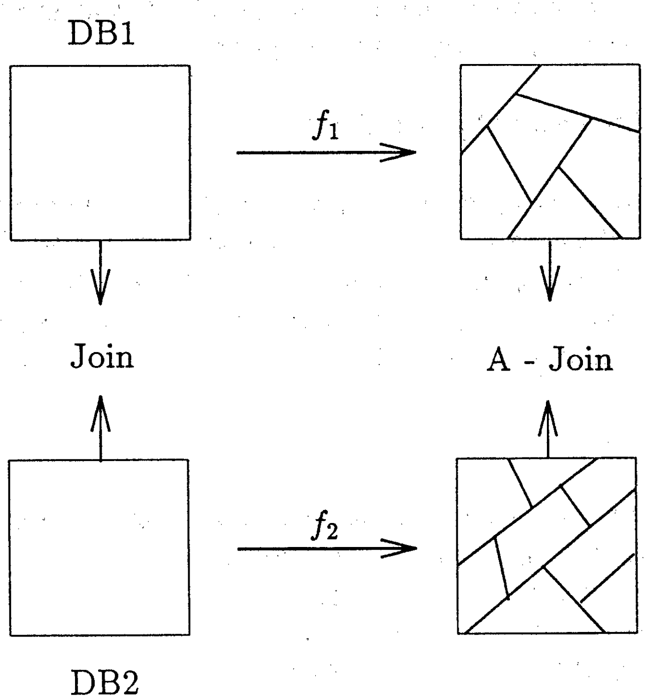


Fig. 4. Join Operation .

time applications. The A-join in Fig.4 is the join between the sets of reference data (vectors), and takes only $O(k^2)$ steps, where k is the maximum number of reference data (vectors), supposed to be 64 - 256 in our system, and unchanged for n. Mathematically, k is the number of equivalence class (based on "similarity") and is extremely reduced from n, the number of data (vectors) of the original set.

To demonstrate the A-join, we introduce a new list as follows;

CUSTOMERS

(Customer1, >'90, <$10,000, business, domestic)

(Customer2, >'87, <$18,000, leisure, imported).

Clearly, *Customer1 is just recommended to buy Civic* in USED CARS, since the join (then, also the A-join) holds between two lists. No other exact join holds between two lists, but the A-join may hold. Really, *Audi and Customer2* is the case of **A-join** between USED CARS and CUSTOMERS. Each list described as the above should be viewed as the list of reference data (vectors).

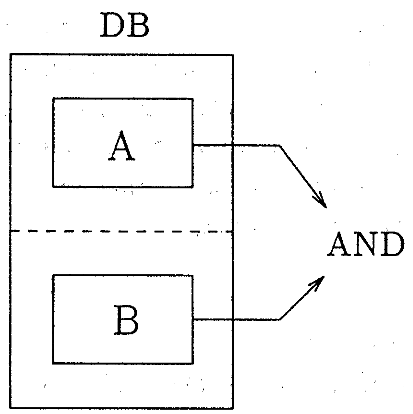
The operations between two databases are similarly defined as the above, and we have the **SBR** (similarity-based reasoning) as a simplified version of MBR. The SBR is quite natural for a set of neural networks. The MBR is an extremal case of neural networks, but requires the huge system. On the other hand, the SBR is realized in the small system using several chips[5]. The SBR in the neural networks is born by reasoning with the combination of Kohonen net and Ichikawa's database.

Interaction among Neural Nets with States

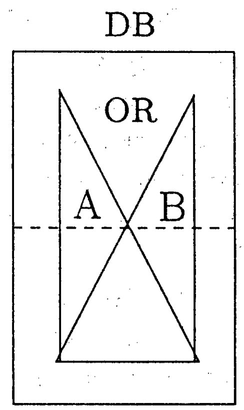
The behaviors among more than one neural nets include the *cooperation* and/or the *competition* as in the distributed AI system. From the viewpoint of parallelism the fundamental behaviors of machine are the AND-parallel and the OR-parallel. The cooperative behavior of neural nets includes both AND-parallel and OR parallel as in Fig.5. The explicit representation of states in the neural net is unnecessary for the time-independent system.

For the time-dependent system the explicit representation of states becomes essential, because the access to the database is often *exclusive* with each other as in Fig.6. (This is the case of competition, but the states are important also for the case of cooperation if the system is time-dependent.) From the viewpoint of machine the case of Fig.6 occurs when more than one machines want to write each own data to the same address. To represent such a complex system the *neural net with states* as in Fig.7 is an appropriate model, and the competitive behavior is represented by the interaction among more than one *neural nets with states*.

We omit the details of the interaction among neural nets with states for the sake of page limit.

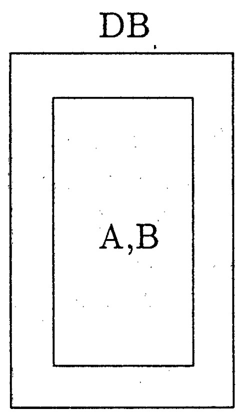


(a) AND - parallel .

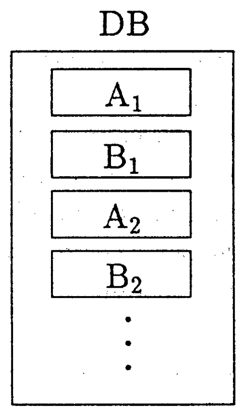


(b) OR - parallel .

Fig. 5. Cooperative Behavior with AND/OR - Parallel .



(a) Read together .



(b) Take one each .

Fig. 6. Competitive Behavior .

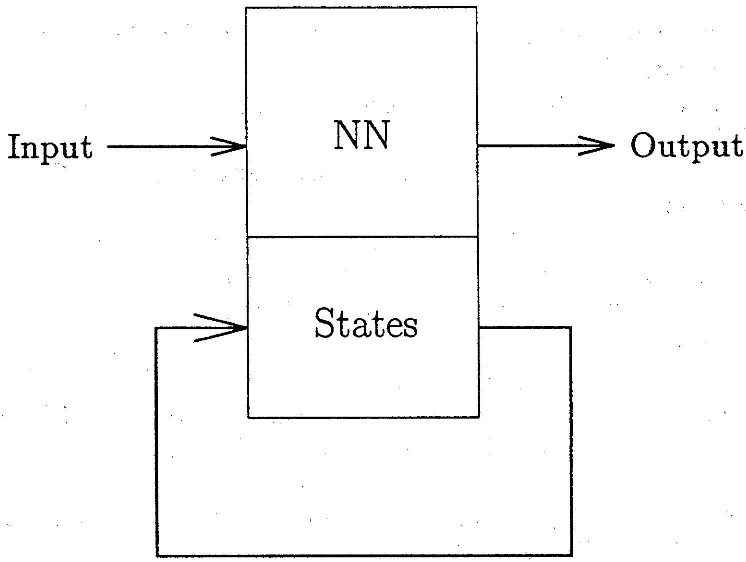


Fig. 7. Neural Net with States .

Remarks

The most important point using the neural nets is *how to make it learn*. In our system we need two kind of learning, i.e.,
1) Learning at classification, and
2) Learning at interaction,
which correspond to the phase 1 and the phase 2 in Fig.2, respectively. For each we prepare them as follows;
1) LVQ algorithm, and
2) New algorithm using neural nets with states.
About 2) we will state the details in near future.

Our research is originally based on *architecture*, and the early machines[11,12] are **inference-based** as the same as in a lot of other researches. Since 1987, however, we start the reseaches of machines with no explicit inference, i.e., the neural net machine[5, 13-20]. This report describes an application of such machines.

References

[1]PANEL: Are Neural Networks a Tool for AI?, Proc. 1991 IEEE Int. Conf. on Tools for AI, San Jose (Nov.1991).

[2] S.Nishio; "Knowledge Discovery in Very Large Databases," Proc. IPSJ, vol.34, no.3, pp343-350 (1993, in Japanese).

[3]C.Stanfill, D.Waltz; "Toward Memory-Based Reasoning," C.ACM, vol.29, no.12, pp.1213-1228 (1986).

[4]T.Ae; "Research Trends on Real-Time Systems in Europe and USA (Invited Talk)," Tech. Report IEICE.J, CPSY91-71 (March 1992, in Japanese).

[5]T.Ae, R.Aibara, K.Kioi; "Design of Neural Self-Organizing Chips for Semantic Applications," Proc.3rd Int. Workshop on VLSI for NNs and AI, Oxford (Sept. 1992).

[6]D.E.Rumelhart, D.E.McCleland, PDP Research Group; *Parallel Distributed Processing: Explorations in Microstructure of Cognition*, Vol 1&2, MIT Press (1986).

[7]T.Kohonen; "The Self-Organizing Map," Proc.IEEE, vol.78, no.9, pp.1464-1480 (1990).

[8]T.Ae, Y.Chikamatsu, K.Kawakami; "Topological Study of Neural Nets (1)," Tech. Report IEICE.J, NC92-159 (March 1993).

[9] i.b.i.d.; "Topological Neural Net," Proc.WCNN'93, Portland (July 1993).

[10]T.Ichikawa, M.Hirakawa; "ARES: A Relational Database with the Capability of Performing Flexible Interpretation of Queries," IEEE Trans. Software Engineering, vol.SE-12, no.5, pp.743-750 (1986).

[11]S.Fujita, R.Aibara, T.Ae; "A Real-Time Production System Architecture using 3-D VLSI Technology," Proc. 5th Int. Workshop on Database Machines, pp.369-380 (1987) and *Database Machine and Knowledge Base Machine*, Kluwer Academic Pub., pp.532-543 (1988).

[12]S.Fujita, M.Yamashita, T.Ae; "Search Level Parallel Processing of Production Systems," PARLE'91(Parallel Architectures and Language Europe), Lecture Notes in Computer Science 506, Springer-Verlag, pp.471-488 (1991).

[13]T.Ae; *VLSI Neurocomputer*, Kyoritsu Shuppan (1991, in Japanese).

[14]T.Ae; "A Hopfield Neural Network with RAM Neurons," Electronics and Communications in Japan, Part 2, vol.73, no.12, pp.26-35 (1990).

[15]T.Ae et al.; ' A RAM-based Neural Chip for Optimization Problem Solver," Proc. IFIP Workshop on Parallel Architectures on Silicon, Grenoble, pp.80-99 (1989).

[16]T.Ae, R.Aibara; "Memory-based Architecture for Artificial Neural Networks," Proc. 2nd Int. Conf. on Microelectronics for Neural Networks, Munich, pp.135-142 (1991).

[17]T.Ae, R.Aibara; "A Neural Network for 3-D VLSI Accelerator," Proc. 1st Int. Workshop on VLSI for AI, Oxford (1988) and *VLSI for Artificial Intelligence*, Kluwer Academic Pub., pp.179-188 (1989).

[18]T.Ae, Y.Mitsui, S.Fujita, R.Aibara; "Binary Neural Network with Delayed Synapses," Proc. 2nd Int. Workshop on VLSI for AI and NNs, Oxford (1990) and *VLSI for Artificial Intelligence and Neural Networks*, Plenum Press, pp.295-304 (1991).

[19]T.Ae, R.Aibara; "Non von Neumann Chip Architecture - Present and Future - (Invited Paper)," Trans. IEICE.J (to appear).

[20]T.Ae, R.Aibara; " The Present and Future of Neural Chip Architecture (Invited Paper)," 5th Int. Sympo. on IC Tech., System & Applications, Singapore (to appear).