鉄鋼業における板取り問題への ニューラルネットワークの適用

高田智治　佐能克明　福村聡

takada@system.kawasaki-steel.co.jp

川崎製鉄　情報システム部　システム技術室

〒261-01　千葉市美浜区中瀬１丁目３番地　幕張テクノガーデン　Ｂ棟１１Ｆ

鉄鋼業における板取り問題を解くためのニューラルネットワークについて報告する。この板取り問題は、ある決まった幅と長さを持つ鉄の板から、需要家の要求する様々なサイズの矩形を無駄なく取り出す問題である。適用したニューラルネットワークのモデルはホップフィイールドモデルの範疇に入るが、非等式制約も扱えるように拡張してある。このニューラルネットワークの性能は、現在稼働しているエキスパートシステムと比較して評価した。結果としては、熟練者の知識をもとに探索を行うエキスパートシステムに比べ、ニューラルネットワークは同程度の求解時間で、より最適な解を求めることが可能となった。

A Neural Network System for Solving an Assortment Problem in the Steel Industry

Tomoharu Takada, Katuaki Sanou, Satoshi Fukumura

Information Systems Tehnology Section
Information Systems Department

Kawasaki Steeel Corporation

Makuhari Techno Garden 1-3 Nakase, Mihama-ku, Chiba 261-01, JAPAN

A Neural network model for solving an assortment problem found in the iron and steel indstry is discussed in this paper. The problem arises in the yard where steel plate is cut into rectangular pieces. The neural network model can be categorized as a Hopfield model, but the model is expanded to handle inequality constraints. The performance of the neural network was evaluated by comparison with an existing expert system. The results showed that the neural network has the potential to identify in a short time near-optimal solutions to the assortment problem.

1 Introduction

In Kawasaki Steel Corporation, neural networks have been studied and evaluated since 1988. Neural networks have mainly been applied to character recognition problems. On the other hand, few systems use neural networks, although much attention has been paid to recurrent neural networks which can solve combinatorial optimization problems. Scheduling and planning tasks in a steelworks involve many kinds of combinatorial optimization problems. Some have been solved by the technology of operations research, and others by expert systems. As disadvantages of these methods, it takes considerable time to calculate theoretical optima using the methods provided by operations research, while the optimality of solutions by expert systems is questionable because heuristic search based on the knowledge of experts is used.

Recurrent neural networks may be able to overcome the difficulties of both operations research and expert systems, because they have the ability to identify approximate optima quickly. However, one problem with recurrent neural networks is that they tend to be "caught" by local optima, and it is difficult to set the parameters. Therefore, efforts were made here to elucidate network capabilities and limitations and to understand the related issues by applying a recurrent neural network to a practical assortment problem. This paper presents the conclusions reached based on the results of experiments, and evaluates the potential of neural networks for practical use.

2 Neural Networks for Solving Combinatorial Optimization Problems

The neural network model [1] proposed by Hopfield is the most popular of the feedback neural networks. In the Hopfield model, the behavior of a neuron is defined by the following differential equation:

$$\frac{du_i}{dt} = -\frac{u_i}{\eta} + \sum_{j=1}^{n} w_{ij} x_j + v_i \tag{2.1}$$

where $-\frac{u_i}{\eta}$ is a passive decay term, w_{ij} is the strength of the interconnection between neuron i and neuron j, x_i is the output of the activation function for neuron i shown below, u_i is the input of the activation function for neuron i, and v_i is the external input to neuron i. The activation function is typically a smooth sigmoid function:

$$x_i = \sigma(u_i) = \frac{1}{2}\left\{1 + tanh\left(\frac{u_i}{u_0}\right)\right\} \tag{2.2}$$

The activation function $\sigma(u_i)$ meets the Cohen-Grossberg requirements for stability. Thus, if the external inputs are maintained at a constant value, a network of neurons modeled by equation (2.1) will eventually equilibrate, regardless of the starting state.

Hopfield discovered a Lyapunov function for a network of n neurons characterized by equation (2.1), which can be expressed as:

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j - \sum_{i=1}^{n} v_i x_i + \frac{1}{\eta}\sum_{i=1}^{n}\int^{x_i} \sigma^{-1}(x)dx. \tag{2.3}$$

This is the expression which Hopfield refers to as the "energy function" of the network. The term "energy function" stems from an analogy between the behavior of a neural network and that of certain physical systems. Just as a physical system may evolve toward an equilibrium state, a network of neurons will always evolve toward the minimum value of the energy function. The stable states of a network of neurons therefore correspond to the local minima of the energy function.

Hopfield and Tank had a key insight when they recognized that it is possible to use the energy function to perform computations. Because a network of neurons will seek to minimize the energy function, one may design a neural network for function minimization by associating variables in an optimization problem with neurons. Actually, "design" means the task of selecting appropriate values for the connection strength w_{ij} and the external inputs v_i so that the desired network behavior can result. Hopfield and Tank illustrated the use of the energy function to configure networks for several optimization applications including the traveling salesman problem.

3 A Neural Network for Solving an Assortment Problem

3.1 An Assortment Problem in the Steel Industry

The problem described in this section is a kind of assortment problem, which arises in steelworks when large steel plates are cut into smaller pieces as required by product orders. It may be described as follows:

> In this assortment problem, the following three items are given; a set of actual product orders, each of which specifies width and length; a steel plate, whose width and length are also specified; and the number of instructions. Each order can be divided into several lengthwise pieces and is thus "placed" on the plate. In practice, the plate is first divided into the several pieces in the length direction and each piece is then cut into smaller segments to fill actual orders. Instructions specify the length and combination of order widths. Naturally, orders must be arranged on the steel plate without overlap. The goal of the problem is to specify the contents of every instruction. The key consideration may be summarized as follows:
>
> - The area of segments which are not covered by orders must be minimized.
> - Orders must be finished (The maximum length of any order must be no longer than the ordered length plus a specified allowance.)
> - The number of instructions must be minimized.

Table 1 shows an example of a set of orders.

Figure 1 shows two examples of order placement on a plate 990mm wide, using two and three instructions respectively. In placement 1, three instructions are used to finish four orders. The 16800m of ORDER 01 and 5040m of ORDER 02 are placed in the leftmost instruction, whose length is 1680m. Hence, the necessary length

Table 1: Example of a set of orders

Order Number	Width[mm]	Length[m]	Allowance[m]
ORDER 01	50	16115	848
ORDER 02	160	5036	265
ORDER 03	200	6043	318
ORDER 04	220	3809	289

for ORDER 01 is secured in this area by merging ten rectangles, each of which has a length of 1680m. Placement 2 is better than placement 1 because fewer instructions are required for placement 2, and the area of segments uncovered in placement 2 is smaller than that in placement 1.

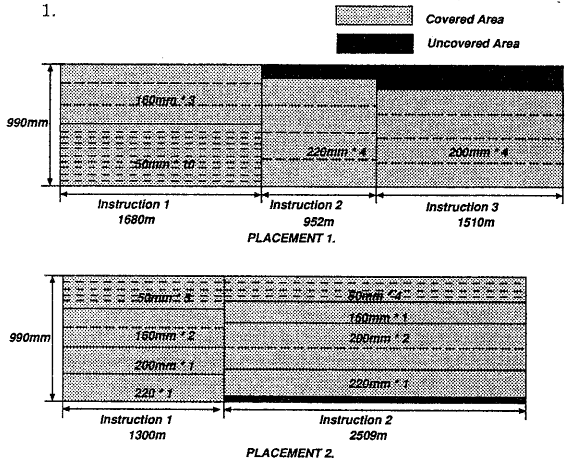

Figure 1: Examples of order placements

The problem may be formulated as: limiting the total width of orders, limiting the number of points to be cut, maximizing the area of segments covered by orders(maximizing yield), and minimizing the area of segments uncovered by orders(minimizing waste). Reducing the number of instructions is one consideration which the authors deemed important, but it must enlarge the size of the neural network to formulate this requirement as an objective function. The number of instructions exists whithin narrow bounds in practical use. Therefore the authers created some partial problems which had fixed the number of instructions for one problem. After we solved some partial problems, the best solution was chosen.

The problem is formulated as follows.

[$F1$]: Limiting the summation of widths of orders

The summation of widths of orders in instruction i must not be greater than the plate width. This constraint is expressed as follows:

$$\sum_{j=1}^{J} U_j \times x_{ij} \leq W \tag{3.1}$$

where J is the number of orders, U_j is the width of an order j, W is the width of the plate, and x_{ij} is the number of rectangles in instruction i whose widths are equal to U_j.

[$F2$]: Limiting the number of points to be cut

The number of rectangles in instruction i must be equal to or less than some positive integer due to the limitation of facilities. This constraint is expressed as follows:

$$\sum_{j=1}^{J} x_{ij} \leq C \tag{3.2}$$

where C is that integer.

[$F3$]: Maximizing the area of segments covering orders

One of the objective functions of this problem is to maximize the area of segments covering orders. This objective function is expressed as follows:

$$\text{minimize} \quad \sum_{j=1}^{J} U_j \times S_c(V_j - \sum_{i=1}^{I} y_i \times x_{ij}) \tag{3.3}$$

where V_j is the length of order j, I is the number of instructions, y_i is the length of instruction i, and S_c is the function shown below:

$$S_c(Z) = \begin{cases} Z & (Z \geq 0) \\ 0 & (Z < 0) \end{cases}$$

[$F4$]: Minimizing the area of segments uncovered by orders

Another objective function is to minimize the area of segments uncovered by orders. Uncovered segments are categorized into two types, uncovered segments in the width direction, and uncovered segments in the length direction. If the summation of the widths of orders in instruction i is less than the width of the plate, an uncovered segment of the first type results. If the summation of the length of the rectangles for order j is greater than the length of order j, one or more uncovered segments of the second type are produced. Hence, this objective is expressed as follows:

$$\text{minimize} \quad \sum_{i=1}^{I} y_i \times S_c(W - \sum_{j=1}^{J} U_j \times x_{ij}) + \sum_{j=1}^{J} U_j \times S_c \left\{ \sum_{i=1}^{I} y_i \times x_{ij} - (V_j + T_j) \right\} \tag{3.4}$$

where T_j is the allowance of the length of the order j.

3.2 A Neural Network Model for the Assortment Problem

3.2.1 Assumptions and Notation

The formulas shown in Section 3.1 must be expressed in terms of $0-1$ *variables* to construct a neural network model for the assortment problem which they define. The following two notations are used for that purpose.

1. x_{ijk} to express the number of rectangles for order j in instruction i

   When there are p rectangles for order j in instruction i, x_{ijk} which is a $0-1$ *variable* satisfies the following equation:

$$\sum_{k=1}^{N_j} x_{ijk} = p$$

where N_j is the maximum number of rectangles for order j in instruction i calculated by the following formula:

$$N_j = \lfloor W/U_j \rfloor$$

2. y_i to express the length of instruction i
Because the length of an instruction is a continuous value, the length of instruction i y_i is expressed as follows:

$$y_i = L \times (z_{i1} + z_{i2} \cdots z_{iH})$$

where z_{ih} is a $0-1$ *variable*, L is some constant value, and H is a sufficiently large integer.

3.2.2 Constraints and Objective Functions Expressed in Terms of 0-1 Variables

[$F1$]: Constraint for limiting the summation of widths of orders

$$\text{minimize} \quad E_A = \sum_i^I S_A(\sum_j^J \sum_k^{N_j} U_j \times x_{ijk} - W) \qquad (3.5)$$

$$S_A(Z) = \begin{cases} Z^2 & (Z \geq 0) \\ 0 & (Z < 0) \end{cases}$$

[$F2$]: Constraint for limiting the number of points to be cut

$$\text{minimize} \quad E_B = \sum_i^I S_A(\sum_j^J \sum_k^{N_j} x_{ijk} - C) \qquad (3.6)$$

[$F3$]: Objective function for maximizing the area of segments covered by orders

$$\text{minimize} \quad E_C = \sum_j^J U_j \times S_B(V_j - \sum_i^I \sum_k^{N_j} y_i \times x_{ijk}) \quad (3.7)$$

$$S_B(Z) = \begin{cases} Z & (Z \geq 0) \\ 0 & (Z < 0) \end{cases}$$

[$F4_a$]: Objective function for minimizing the area of segments uncovered by orders in the width direction

$$\text{minimize} \quad E_{D_a} = \sum_i^I y_i \times S_B(W - \sum_j^J \sum_k^{N_j} U_j \times x_{ijk}) \quad (3.8)$$

[$F4_b$]: Objective function for minimizing the area of segments uncovered by orders in the length direction

$$\text{minimize} \quad E_{D_b} = \sum_j^J U_j \times S_B \left\{ \sum_i^I \sum_k^{N_j} y_i \times x_{ij} - (V_j + T_j) \right\} \quad (3.9)$$

3.2.3 Energy Function

The energy function is the sum of all the functions described above, multiplied by some coefficient, and is expressed as follows:

$$E = \alpha E_A + \beta E_B + \gamma E_C + \delta E_{D_a} + \varepsilon E_{D_b}. \qquad (3.10)$$

3.2.4 A Neural Network for Handling Inequality Constraints

The assortment problem discussed here includes the inequality constraints represented by equations (3.1) and (3.2). Therefore, the Hopfield neural network has to be expanded to handle them. The basic idea is explained below using (3.1) as an example.

Equation (3.1) is expressed as equation (3.5) in terms of $0-1$ *variables*. Equation (3.5) is a kind of penalty function which applies a large penalty value when (3.1) is not satisfied, and is 0 otherwise. The procedure is described below.

1. Calculate the output value of each neuron.
2. Decide by the outputs of all neurons whether each inequality constraint is satisfied or not.
   (a) If the inequality constraint for instruction i is not satisfied, the interconnective weights between neurons are determined by the quadratic function $(\sum_j^J \sum_k^{N_j} U_j \times x_{ijk} - W)^2$.
   (b) If the inequality constraint for instruction i is satisfied, the interconnective weights between neurons are set to zero.
3. Return to 1.

When the inequality constraint is implemented in the neural network explained above, the convergence of the network is the most difficult problem. To overcome this difficulty, a special term is added to the energy function shown by equation (3.10). The new energy function is shown below:

$$E = \alpha E_A + \beta E_B + \gamma E_C + \delta E_{D_a} + \varepsilon E_{D_b} + \zeta E_{B}. \quad (3.11)$$

$$E_B = \sum_i^I \sum_j^J \sum_k^{N_j} x_{ijk}(1 - x_{ijk}) + \sum_i^I y_i(1 - y_i) \quad (3.12)$$

where ζ is a coefficient.

The coefficient ζ is increased gradually during the iteration. As ζ increases, the network approaches a stable state and finally converges.

4 Performance of the Neural Network

4.1 Purpose and Means of the Experiment

The purpose of this experiment was to evaluate the fundamental ability of the neural network discussed in the previous section in practical use. Therefore, the results of the neural network were compared with those of an expert system which is used practically. This expert system can solve the assortment problem by inference based on rules provided by experts. Moreover, the solution by the expert system can be changed by a skilled operator of his own accord. Therefore, the final solution by the expert system is equal to the best solution of the system corrected heuristically by a skilled operator.

Table 2: Result of the experiment

		Expert System	Neural Network	Number of Neurons
Case 1	A. %	94.1	95.5	351
	B.	4	4	
	C.	3	3	
Case 2	A. %	96.2	93.4	294
	B.	5	5	
	C.	3	3	
Case 3	A. %	91.3	97.1	165
	B.	4	4	
	C.	3	2	
Case 4	A. %	96.4	93.4	420
	B.	10	10	
	C.	4	4	
Case 5	A. %	93.3	95.7	256
	B.	7	8	
	C.	3	4	

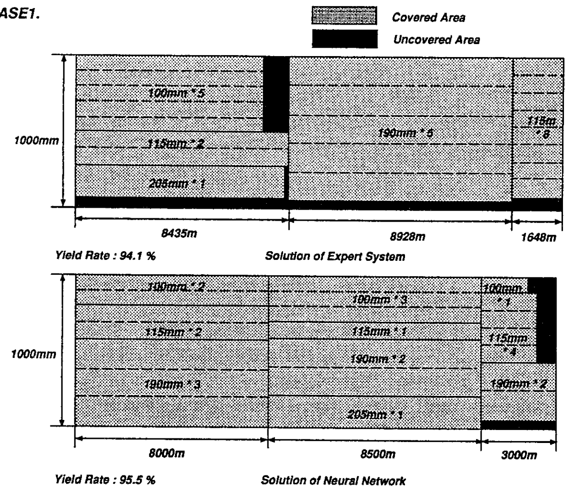

Figure 2: Comparative example of solutions

4.2 Conditions of the Experiment

Five cases were chosen for the experiment, based on practical data from the past. In other words, this neural network was asked to solve a real problem of a certain scale.

Several suitable numbers of instructions around the number set by the expert system were used. The neural network ran ten times for each case, and the best of the ten solutions was selected.

When handling a case including many orders, the case is divided into several subcases, each of which includes fewer orders, because the neural network has difficulty in rapidly searching for the optimal solution when the number of orders is great.

4.3 Results of the Experiment

The results of the experiment are represented by the following three indexes;

A. Yield rate : The yield rate is calculated by dividing the area of covered segments by the area of the steel plate. A high yield rate is desirable because it means a small amount of scrap.

B. Number of finished orders: Indicates how many orders are finished. It is desirable to finish as many orders as possible among those given.

C. Number of instructions: Indicates how many instructions are needed to finish the orders given. A small number is desirable for improving plant efficiency.

The results of the experiment are shown in Table 2. As an example, a comparison of the solutions for Case 1 is shown in Figure 2.

4.4 Evaluation of the Experiment

As a primary consideration, the number of orders finished with the neural network should always be at least as great as the number of finished with the expert system. After this requirement is met, it is desirable that the yield rate be higher and/or the number of instructions be fewer.

In Case 1 and Case 3, the solutions of the neural network were superior to the solutions of the expert system in terms of the yield rate. Particularly, in Case 3, the solution of the neural network was excellent in terms of both the yield rate and the number of instructions. In Case 5, the expert system finished seven orders, but the neural network finished eight orders. Moreover if the expert system, had finished eight orders, the yield rate would necessarily have been lower. Therefore, the solution provided by the neural network was not merely adequate, but was actually superior to that of the expert system.

These results mean that the neural network has a higher ability to solve this assortment problem than the expert system in terms of finding approximate optima.

However, in Case 2 and Case 4, the solutions offered by the expert system were superior to those of the neural network as a result of the difficulty of setting appropriate initial values of neurons and parameter values of the neural network. Hence, although the neural network has the potential to provide better solutions than the expert system, actual performance may differ.

5 A Neural Network System for Solving the Assortment Problem

5.1 A Neural Network System for the Assortment Problem

The ultimate aim is to develop a system for solving the assortment problem using as the system core the neural network discussed in the previous section. To make the neural network system suitable for practical use, the main issues are the following;

1. Number of trials

   The point where the neural network converges depends on the initial state of the neural network, that is, on the initial values of the neurons. Hence, it is necessary to run the neural network system more than one time, using different sets of initial values. The number of trials is a crucial issue from the viewpoint of both the time required for finding solutions and the quality of solutions. In the system, the number is set at twenty five based on experiments.

2. Number of neurons

   The neural network is composed of two types of neurons, one expressed by x_{ijk} and the other by y_i. The number of x_{ijk} neurons is uniquely determined by the orders and the characteristics of the steel plate, but the number of y_i neurons depends on the constant L and integer H. When L is small, H must be large. In this situation, y_i is expressed accurately, but the system has difficulty in identifying the optimal solution. In the system, H is set at fifty, also based on experiments, and L is computed from H and the contents of orders.

3. Number of instructions

   The authors found a decision table in which the number of instructions corresponds to the number of orders, based on experiments with many cases. The number of instructions for all cases is set based on the decision table.

5.2 Performance of the Neural Network System

The performance of the neural network system was compared to that of the expert system from the viewpoint of the following four indicators:

A. Yield rate : A high yield rate is desirable.

B. Number of orders finished / number of orders given : Indicates how many orders are finished using one instruction. A high value is desirable.

C. Number of instructions / number of orders finished : Indicates how many instructions are needed to finish one order. A low value is desirable.

D. Time to find a solution : The value of the expert system is approximate.

The results of the experiments are summarized in Table 3 for the four indicators (A$\sim$D) given above. All values are averages of several dozen cases.

Table 3: Results of Experiments

	A.	B.	C.	D.
Expert System	95.18 %	0.70	0.71	10 min
Neural Network	96.89 %	0.91	0.74	12.8 min

The following can be said regarding the performance of the neural network system:

A. The yield rate is higher than that of the expert system. The difference is around 1.7%, a figure which is regarded as meaningful in Kawasaki Steel Corporation.

B. The number of finished orders is greater than with the expert system. These numbers show that the neural network system finishes about nine orders when ten orders should be finished, although the expert system finishes about only seven orders. Hence, the neural network system reduces the number of small-volume orders and consequently improves plant efficiency.

C. In terms of the number of instructions/order, the results of the neural network system are slightly inferior to those of the expert system. However, the difference seems to present no problem for practical use, as the figure shown here indicates that the number of instructions the neural network system needs in order to finish a given number of orders is not significantly different from that required by the expert system.

D. The speed of the neural network system is adequate for practical use.

As shown above, when ten orders are given, the yield rate of the neural network system is higher than that of the expert system, and the neural network system finishes nine orders using six instructions, while the expert system finishes seven orders using five instructions. If the expert system finished the same number of orders as the neural network system, either the yield rate would decrease or the number of instructions would increase. Therefore, the neural network system appears to be more useful than the expert system and adequate for practical use. Moreover, the yield rate should improve if the neural network system is adopted.

6 Conclusion

The capabilities of neural networks for a practical assortment problem were clarified up. It can be concluded that the neural network has the ability to find optimal solutions with adequate speed. However, it is also true that the neural network cannot always find the optimal solution in a short time. When the neural network is used practically, its performance must be stable and it must find useful solutions rather than merely restrict optima. Therefore, several special methods were used to construct the neural network model, and many difficulties in the implementation of the neural network system were solved. The results of experiments with the system show that the neural network is useful in a practical assortment problem in terms of both the optimality of solutions and the time required for finding those solutions. Interconnective neural networks should therefore be useful in solving practical problems.

References

[1] J.J.Hopfield and D.W.Tank (1985). "Neural Computation of Decisions in Optimization Problems," *Biological Cybernetics*, Vol.52, pp.141-152.

[2] G.A.Tagliarini, J.F.Christ, and E.W.Page (1991). "Optimization Using Neural Networks," *IEEE TRANSACTIONS ON COMPUTERS*, Vol.40, No.12, pp.1347-1358.