



(19) **United States**

(12) **Patent Application Publication**  
**MIYAHARA**

(10) **Pub. No.: US 2021/0081772 A1**

(43) **Pub. Date: Mar. 18, 2021**

(54) **RESERVOIR COMPUTER, RESERVOIR DESIGNING METHOD, AND NON-TRANSITORY COMPUTER-READABLE STORAGE MEDIUM FOR STORING RESERVOIR DESIGNING PROGRAM**

(52) **U.S. CI.**  
CPC ..... *G06N 3/063* (2013.01); *G06F 2015/768* (2013.01); *G06F 15/76* (2013.01); *G06N 3/08* (2013.01)

(71) Applicant: **FUJITSU LIMITED**, Kawasaki-shi (JP)

(57) **ABSTRACT**

(72) Inventor: **Shoichi MIYAHARA**, Kawasaki (JP)

(73) Assignee: **FUJITSU LIMITED**, Kawasaki-shi (JP)

A reservoir designing method executed by a computer configured to control a neural network including a reservoir and an output layer, the reservoir including a plurality of nodes and having a coupling structure randomly determined between the plurality of nodes, the output layer having a weight set on each node of the plurality of nodes. In an example, the method includes: changing the coupling structure between the plurality of nodes included in the reservoir; computing an output for an input to the neural network; updating the weight of the output layer based on the output for each of the coupling structures changed by the changing; evaluating the output according to a predetermined criterion; and selecting a predetermined coupling structure from the coupling structures changed by the changing based on an evaluation result obtained by the evaluating.

(21) Appl. No.: **16/987,457**

(22) Filed: **Aug. 7, 2020**

(30) **Foreign Application Priority Data**

Sep. 12, 2019 (JP) ..... 2019-166315

**Publication Classification**

(51) **Int. Cl.**  
*G06N 3/063* (2006.01)  
*G06N 3/08* (2006.01)  
*G06F 15/76* (2006.01)

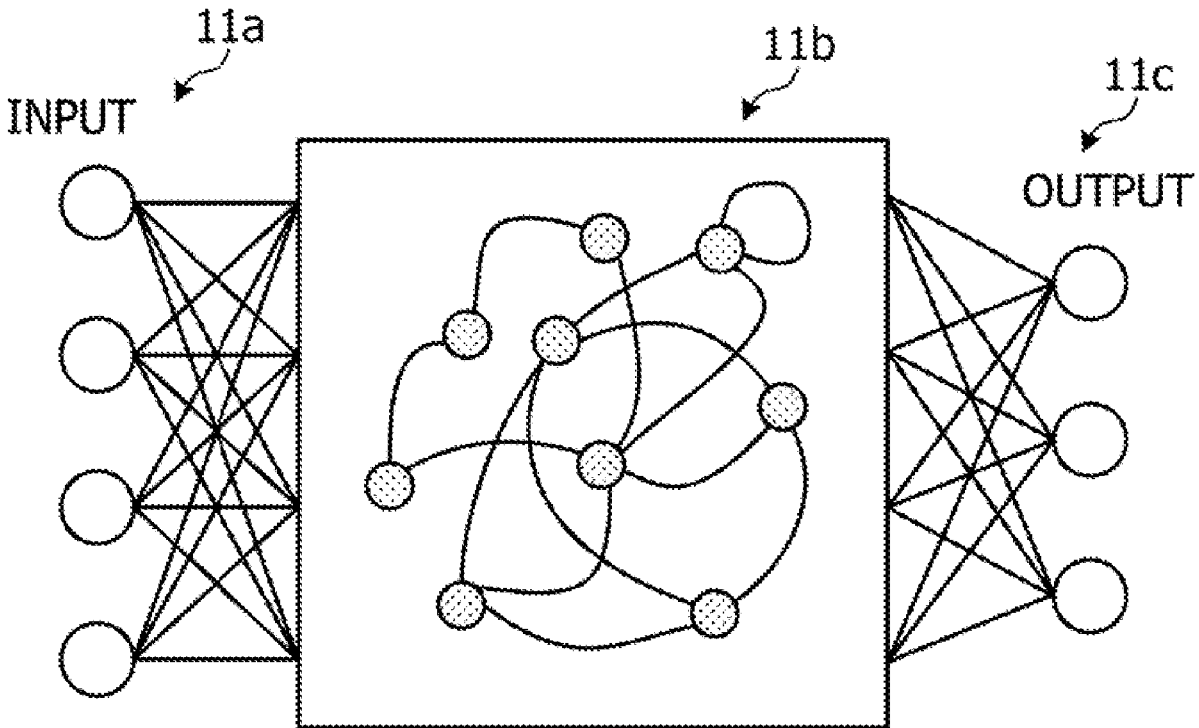


FIG. 1

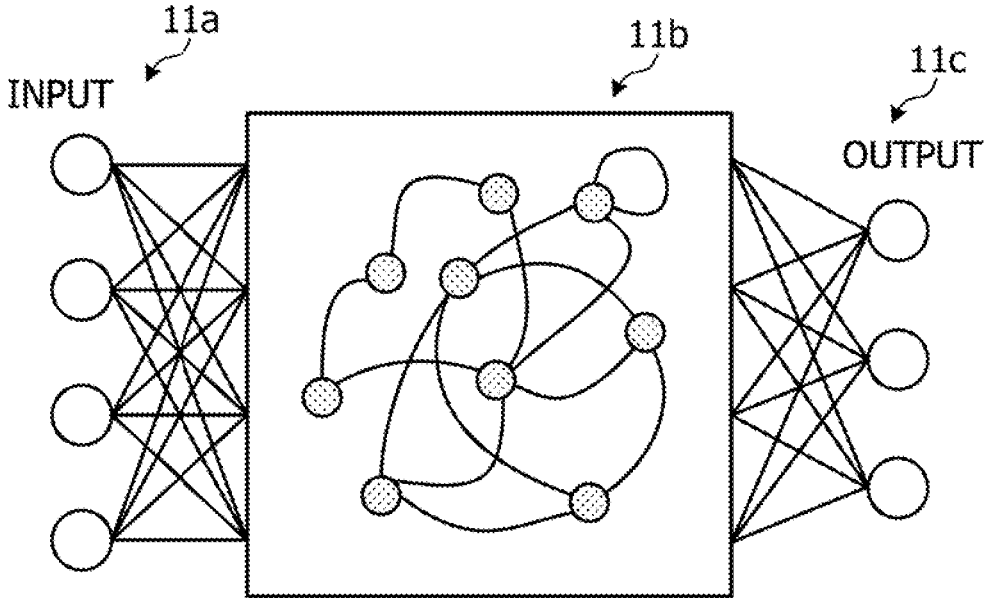


FIG. 2

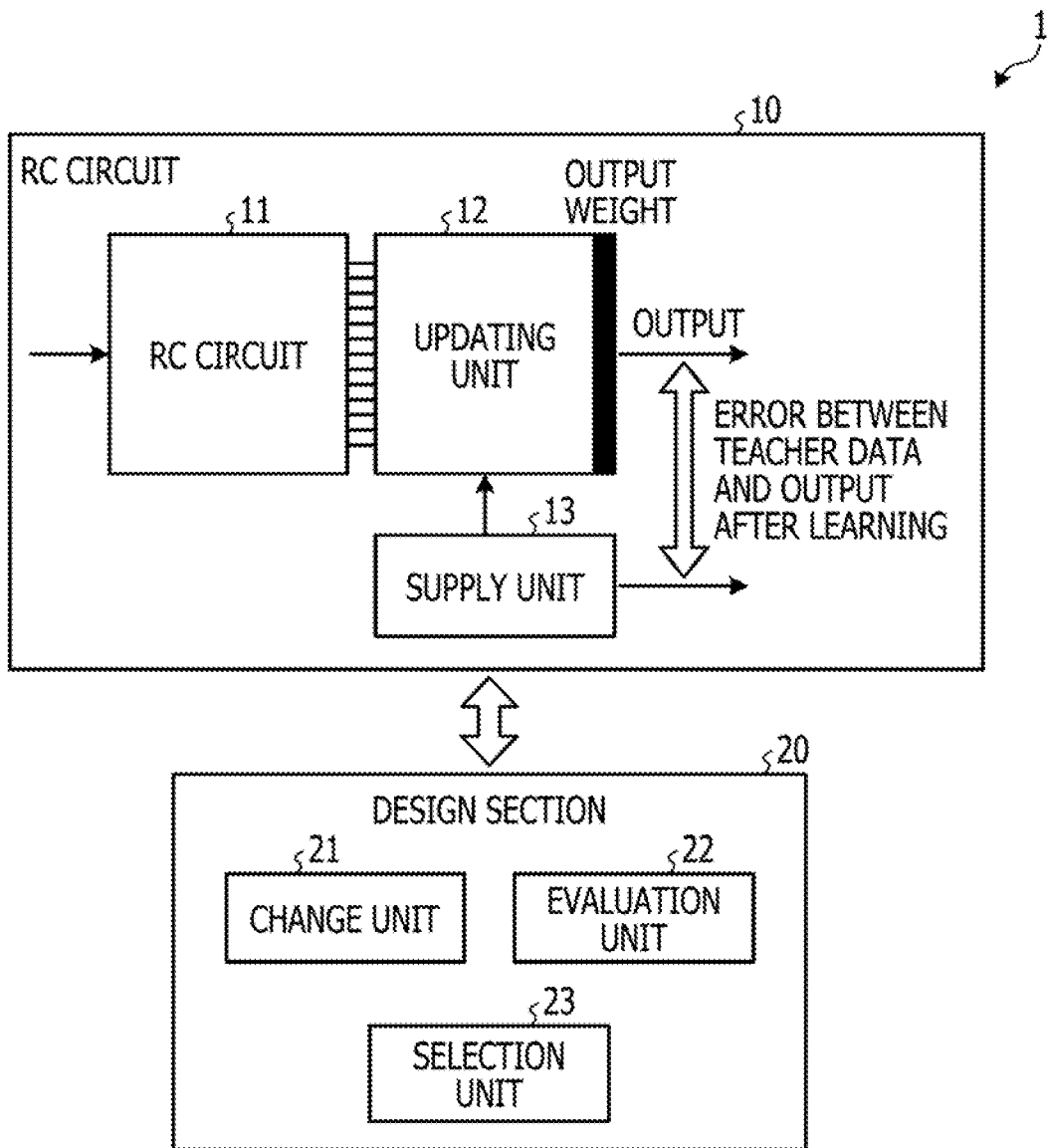


FIG. 3

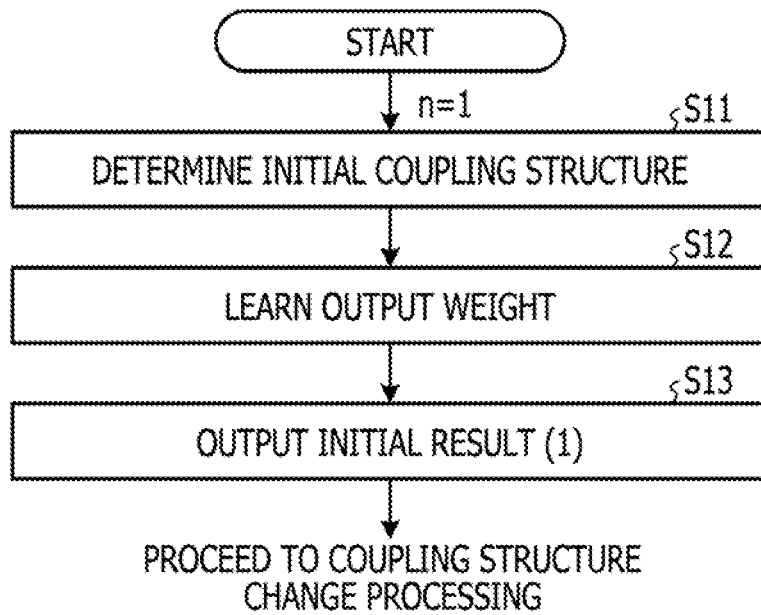


FIG. 4

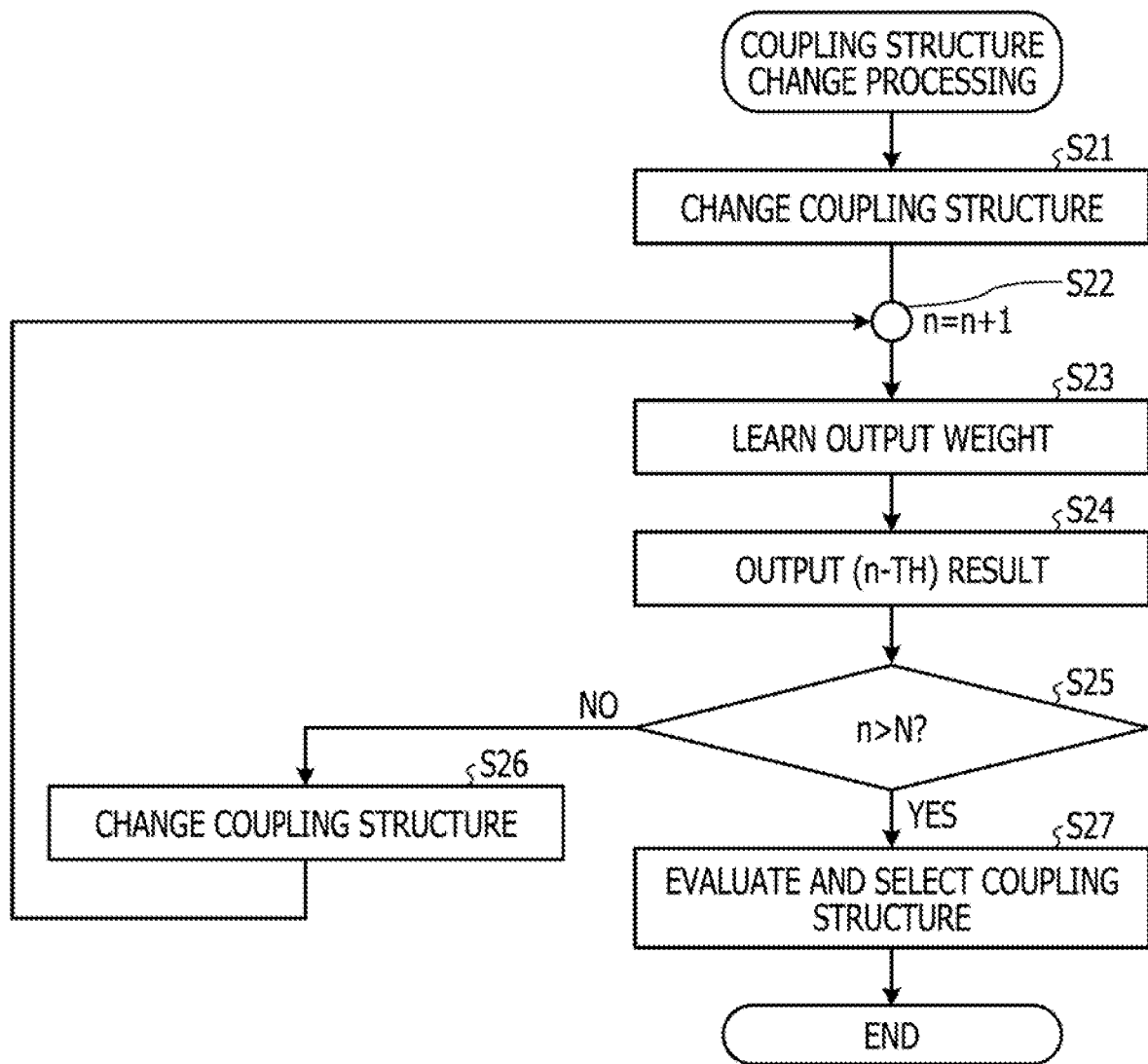


FIG. 5

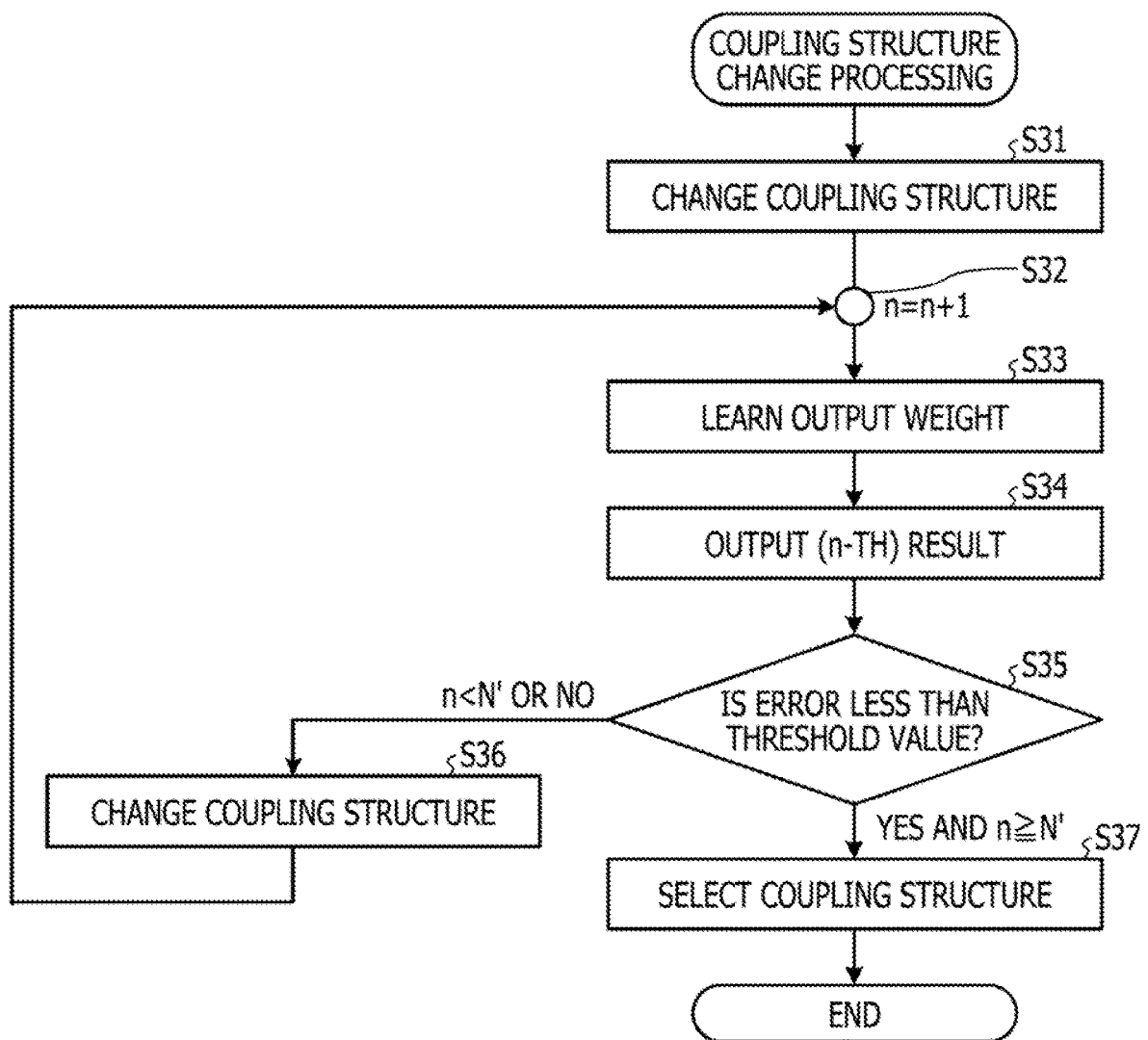


FIG. 6

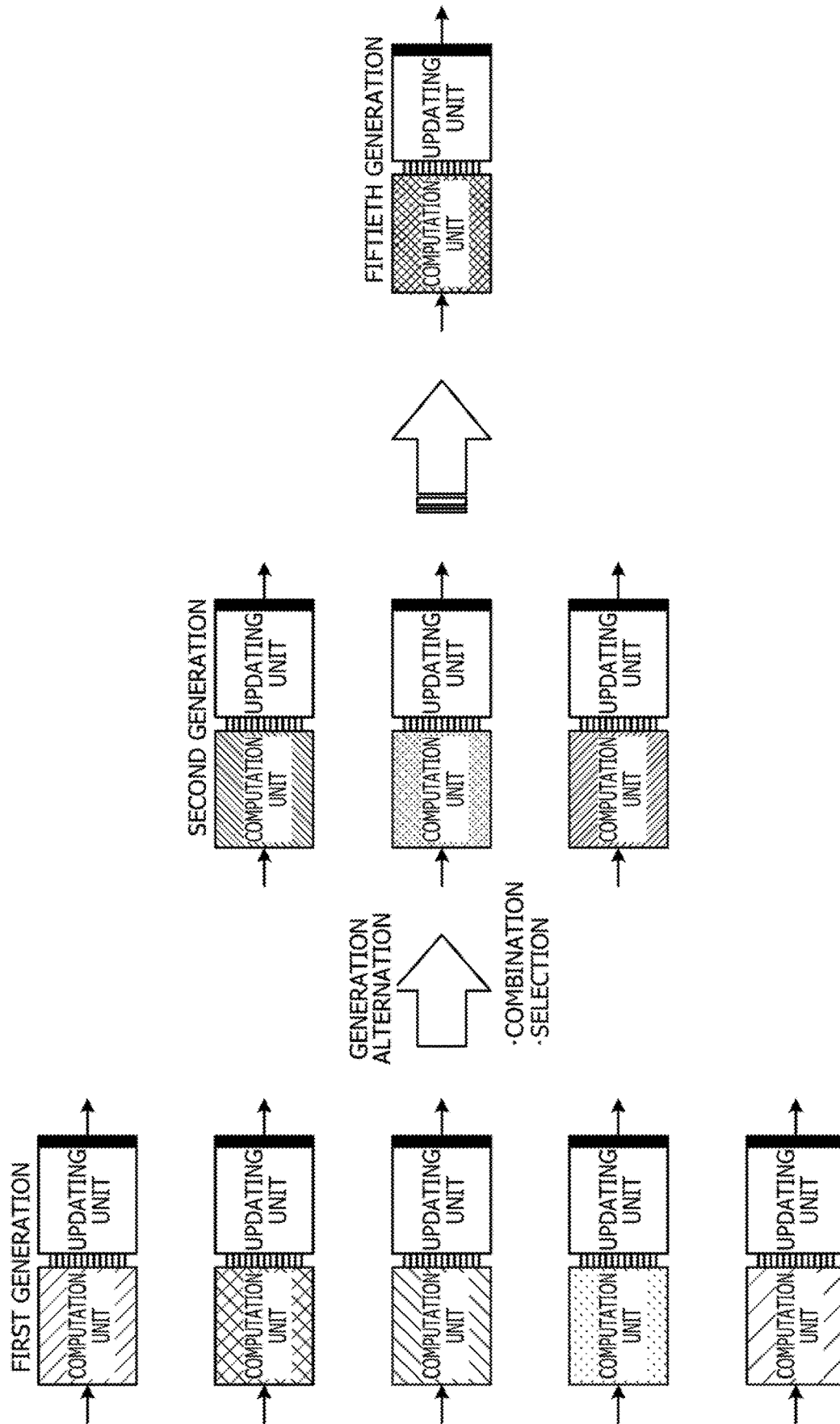


FIG. 7

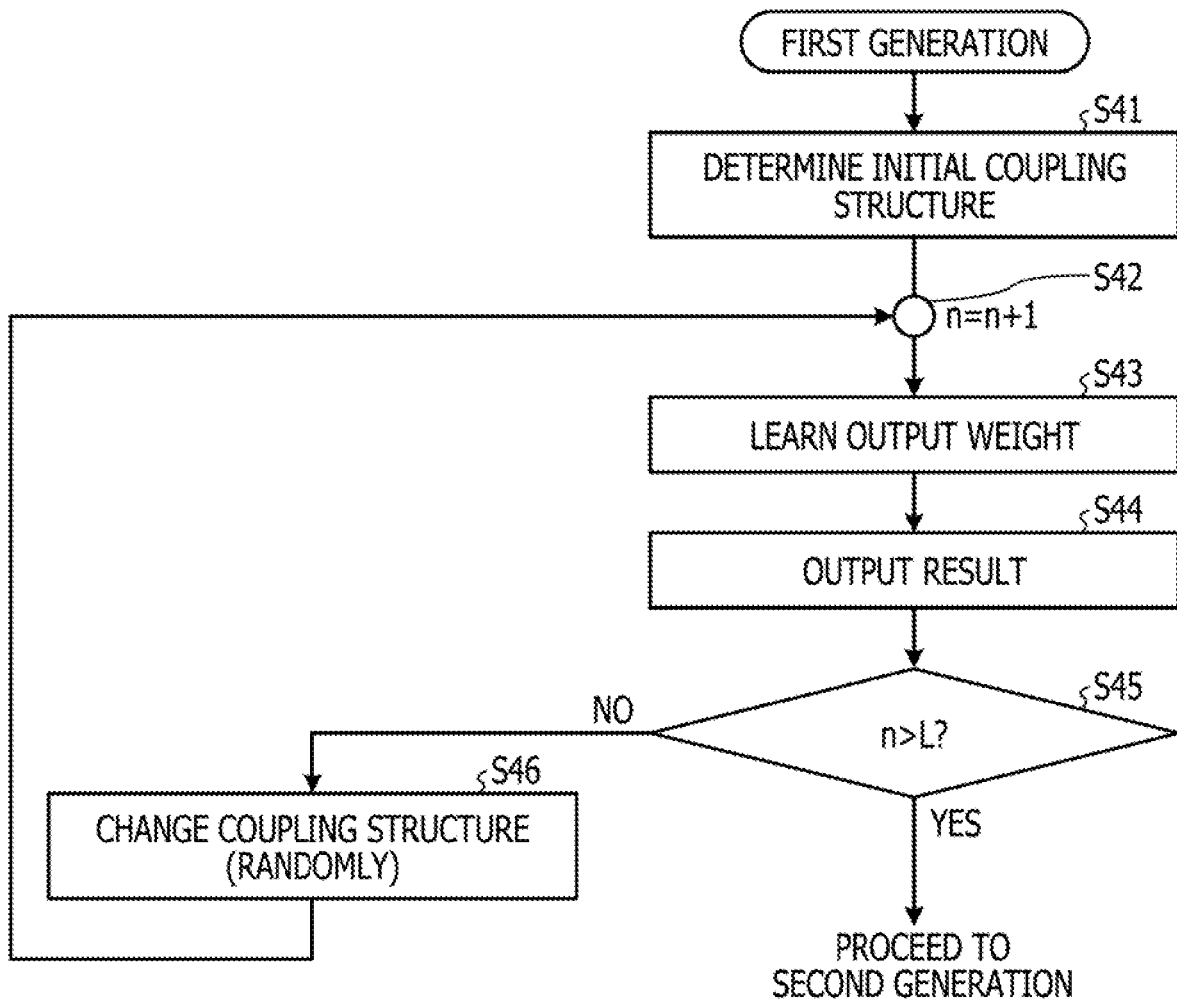




FIG. 8

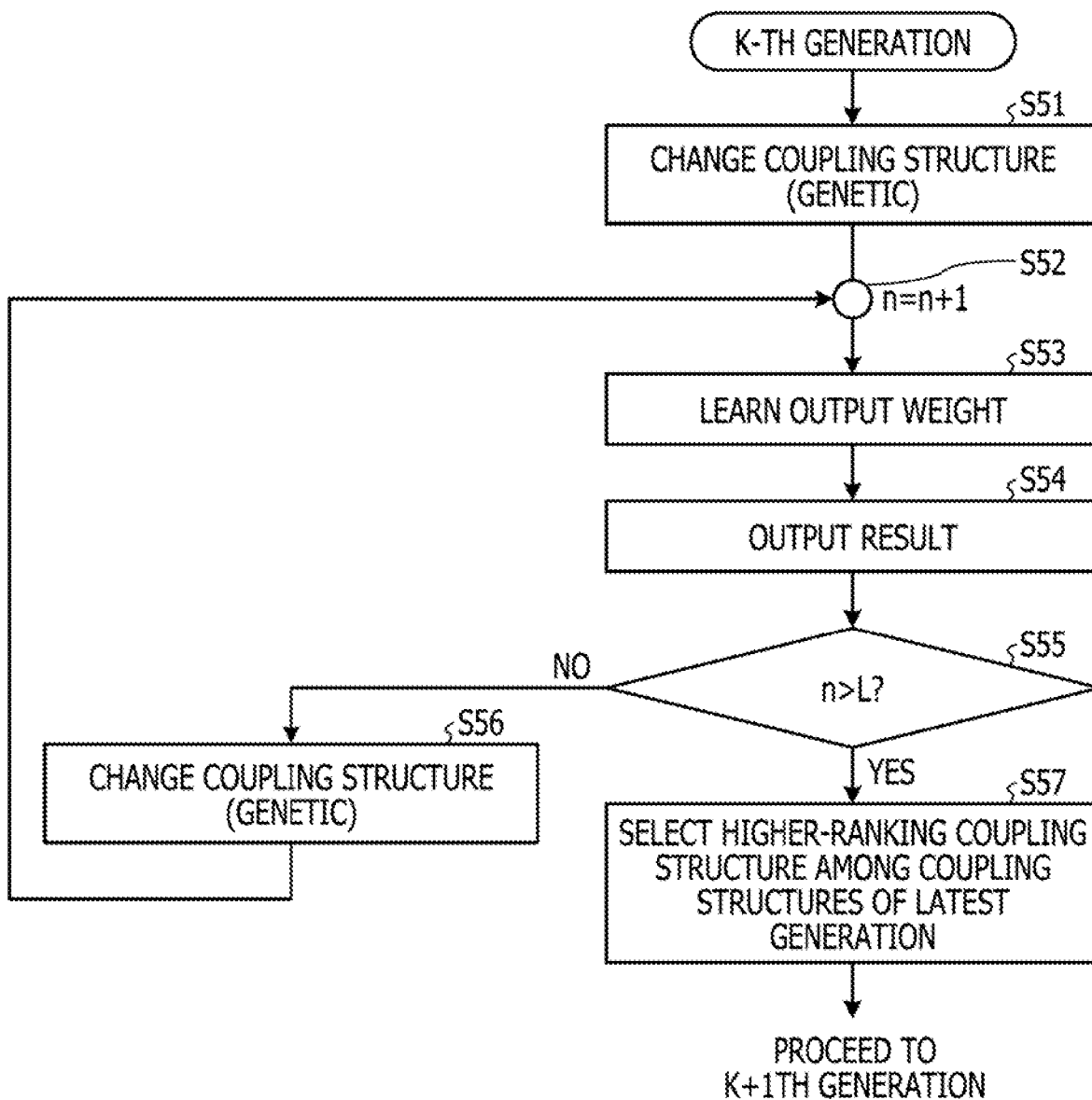


FIG. 9

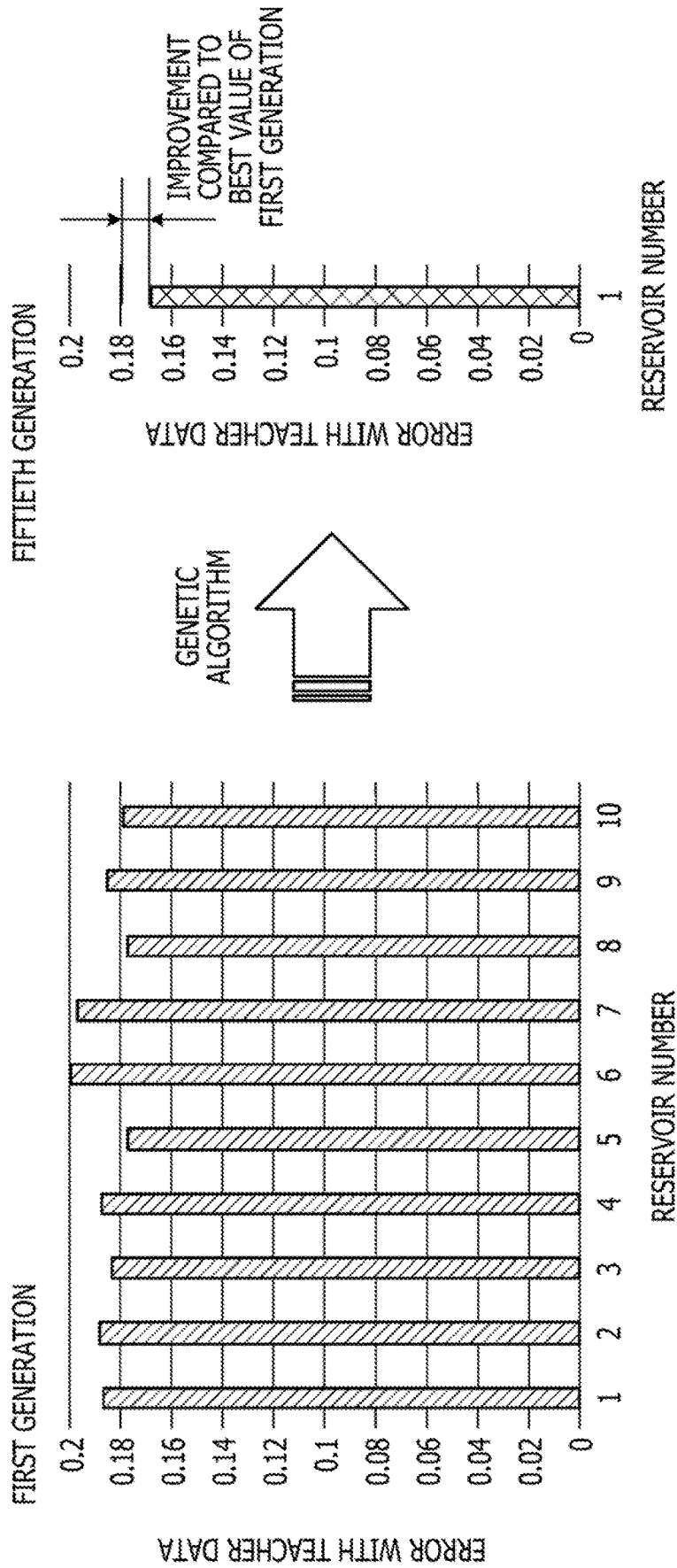
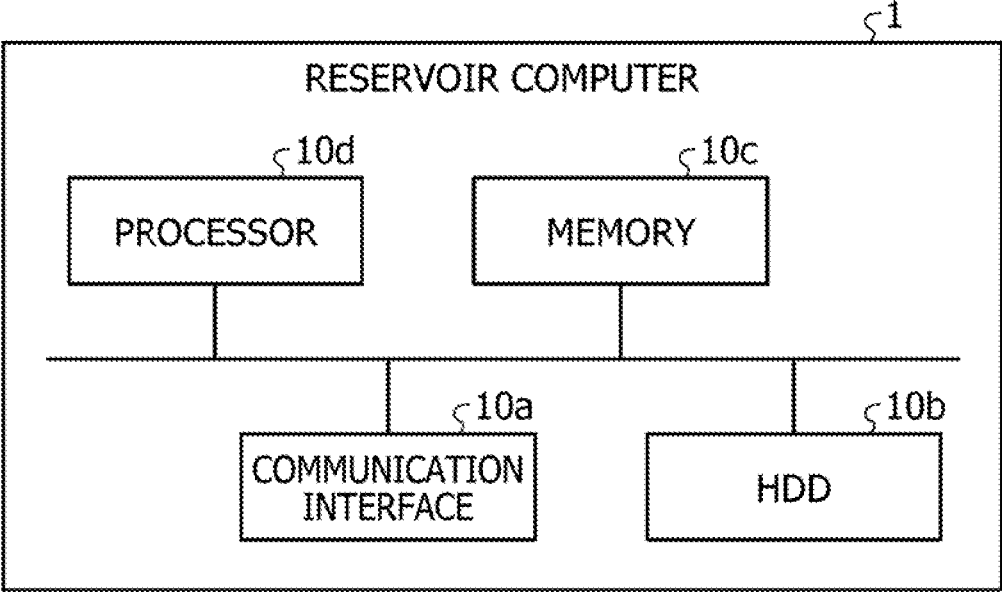


FIG. 10



**RESERVOIR COMPUTER, RESERVOIR  
DESIGNING METHOD, AND  
NON-TRANSITORY COMPUTER-READABLE  
STORAGE MEDIUM FOR STORING  
RESERVOIR DESIGNING PROGRAM**

CROSS-REFERENCE TO RELATED  
APPLICATION

[0001] This application is based upon and claims the benefit of priority of the prior Japanese Patent Application No. 2019-166315, filed on Sep. 12, 2019, the entire contents of which are incorporated herein by reference.

FIELD

[0002] The embodiments discussed herein are related to a reservoir computer, a reservoir designing method, and a non-transitory computer-readable storage medium for storing a reservoir designing program.

BACKGROUND

[0003] Recently, it is known that it is very difficult to update an interlayer weight by backpropagation, particularly in a multilayer recurrent neural network (RNN), and various improvement methods have been studied. Therefore, attention has been focused on an RNN algorithm called reservoir computing for updating a weight of only an output layer.

[0004] Here, the RNN is known as machine learning suitable for handling time-series data. Since the RNN has a structure including a loop inside the network, the RNN may have a correlation between past data and current data as a weighting. The RNN is expected to be applied to dynamic judgment in video processing, natural language processing, and the like.

[0005] The reservoir computing is a special type of RNN that has a structure called a “reservoir” by a random and fixed coupling rather than a multi-layered coupling as in deep learning. The coupling structure (network) between nodes in the reservoir is determined prior to learning. For example, when the reservoir is mounted in a circuit such as a field-programmable gate array (FPGA), the coupling structure is uniformly and randomly determined at the time of design. Also, the learning of the RNN having the reservoir proceeds while the coupling between the nodes of the reservoir is fixed, and only the weight of the output layer is updated. When the learning is completed, the output weight is fixed and output data is obtained.

[0006] Examples of the related art include Japanese Laid-open Patent Publication No. 2018-180701 and International Publication Pamphlet No. WO 2016/194248.

SUMMARY

[0007] According to an aspect of the embodiments, provided is a reservoir designing method executed by a computer configured to control a neural network including a reservoir and an output layer, the reservoir including a plurality of nodes and having a coupling structure randomly determined between the plurality of nodes, the output layer having a weight set on each node of the plurality of nodes. In an example, the method includes: changing the coupling structure between the plurality of nodes included in the reservoir; computing an output for an input to the neural network; updating the weight of the output layer based on the output for each of the coupling structures changed by the

changing; evaluating the output according to a predetermined criterion; and selecting a predetermined coupling structure from the coupling structures changed by the changing based on an evaluation result obtained by the evaluating.

[0008] The object and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the claims.

[0009] It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory and are not restrictive of the invention.

BRIEF DESCRIPTION OF DRAWINGS

[0010] FIG. 1 is a diagram for explaining reservoir computing;

[0011] FIG. 2 illustrates a configuration example of the reservoir computer;

[0012] FIG. 3 is a flowchart illustrating a flow of reservoir design processing according to Embodiment 1;

[0013] FIG. 4 is a flowchart illustrating a flow of coupling structure change processing according to Embodiment 1;

[0014] FIG. 5 is a flowchart illustrating a flow of the coupling structure change processing according to Embodiment 2;

[0015] FIG. 6 is a diagram for explaining a genetic algorithm;

[0016] FIG. 7 is a flowchart illustrating a flow of coupling structure change processing of a first generation according to Embodiment 3;

[0017] FIG. 8 is a flowchart illustrating a flow of coupling structure change processing of a K-th generation according to Embodiment 3;

[0018] FIG. 9 illustrates results of an experiment; and

[0019] FIG. 10 is a diagram for explaining a hardware configuration example.

DESCRIPTION OF EMBODIMENT(S)

[0020] However, there may be circumstances that reservoir computing of related art may have difficulties efficiently obtaining a useful reservoir.

[0021] For example, in the reservoir computing of the related art, the problem may be solved even when there is no bias in a coupling structure between nodes of the reservoir and the coupling structure is uniformly random. On the other hand, in such cases, accuracy may be poor, a large number of learning steps may be desired, and a large number of nodes may be desired to improve the problem. In this disclosure, the term “a learning step” may be referred to as “a training step”, “training”, and the like.

[0022] Furthermore, depending on the problem to be solved, it may be better to use a network with a biased structure as a reservoir. How much the bias in the coupling network between the nodes is desired depends on the problem.

[0023] In one aspect, the purpose is to efficiently obtain a useful reservoir.

[0024] Embodiments of a reservoir computer, a reservoir designing method, and a reservoir designing program according to the present invention will be described below in detail with reference to drawings. The embodiment does not limit the present invention. In addition, respective embodiments may be combined with each other as appropriate without contradiction.

## Embodiment 1

## Reservoir Computing

[0025] First, reservoir computing will be described with reference to FIG. 1. FIG. 1 is a diagram for explaining the reservoir computing. As illustrated in FIG. 1, a neural network in the reservoir computing has an input layer 11a, a reservoir 11b, and an output layer 11c.

[0026] In the example of FIG. 1, each circle with a pattern is a node of the reservoir 11b. As described above, in a method of related art, the weight between the nodes of the reservoir is randomly determined and fixed in advance, and then learning of the weight of the output layer is performed.

## Functional Configuration

[0027] A configuration of a reservoir computer according to an embodiment will be described with reference to FIG. 2. FIG. 2 illustrates a configuration example of the reservoir computer. As illustrated in FIG. 2, the reservoir computer 1 has a reservoir computing (RC) circuit 10 and a design section 20.

[0028] The RC circuit 10 has a computation unit 11, an updating unit 12, and a supply unit 13. The computation unit 11 computes an output for an input to a neural network including a reservoir that includes a plurality of nodes and has a coupling structure randomly determined between the plurality of nodes, and an output layer having a weight set on each node of the plurality of nodes. At least a reservoir portion of the computation unit 11 may be a field-programmable gate array (FPGA).

[0029] The updating unit 12 updates the weight of the output layer. Further, the coupling structure of the reservoir is randomly determined in an initial state, and then changed by the change unit 21 described later. The updating unit 12 updates the weight of the output layer based on the output computed by the computation unit 11 for each of the coupling structures changed by the change unit 21. At this time, the updating unit 12 updates the weight of the output layer so that the error between teacher data (may be referred to as "training data") and the output becomes small. The teacher data is supplied to the updating unit 12 by the supply unit 13.

[0030] The design section 20 has a change unit 21, an evaluation unit 22, and a selection unit 23. The change unit 21 changes the coupling structure between the nodes of the reservoir. For example, the change unit 21 changes the coupling structure by rewriting the FPGA that functions as the reservoir. For example, the change unit 21 changes the coupling structure by a predetermined number of times.

[0031] Here, the coupling structure of the reservoir is logically represented by using an adjacency matrix having the same number of rows and columns as the nodes that may be included in the reservoir. At this time, since each component of the adjacency matrix represents the weight between the plurality of nodes included in the reservoir, the change unit 21 may determine the weight by changing each component. Further, since the nodes of the reservoir do not have to be fully coupled, the component of the adjacency matrix may be 0.

[0032] For example, the change unit 21 may randomly determine each component of the adjacency matrix in the range of 0 or more and 1 or less, and rewrite the FPGA that functions as the reservoir so as to match the adjacency

matrix. At this time, an eigenvalue of the adjacency matrix, a ratio of components having a value of 0, and the like may be set in advance as constraint conditions in accordance with a use of the reservoir and the like. Further, as the ratio of 0 components increases, the reservoir becomes sparse, and as a ratio of real number components increases, the reservoir becomes dense.

[0033] The evaluation unit 22 evaluates the output computed by the computation unit 11 according to a predetermined criterion. For example, the evaluation unit 22 computes an error of the output computed by the computation unit 11 with respect to the teacher data prepared in advance, as the evaluation result.

[0034] The selection unit 23 selects a predetermined coupling structure from the coupling structures changed by the change unit 21 based on the evaluation result obtained by the evaluation unit 22. For example, the selection unit 23 selects a coupling structure such that an output error is minimized from the coupling structures changed by the change unit 21. Further, an FPGA having the coupling structure selected by the selection unit 23 may be treated as a single device that functions as an optimized reservoir. For example, the FPGA having the coupling structure selected by the selection unit 23 may be a target of transfer or the like.

## Processing Flow

[0035] A flow of reservoir design processing will be described with reference to FIG. 3. FIG. 3 is a flowchart illustrating the flow of the reservoir design processing according to Embodiment 1. First, as illustrated in FIG. 3, the design section 20 of the reservoir computer 1 determines an initial coupling structure (step S11). An n represents the number of times of change of the coupling structure. Here, it is assumed that n=1. Further, the design section 20 may determine the initial coupling structure by the same method as the change of the coupling structure.

[0036] Next, the RC circuit 10 learns (may be referred to as "performs training") an output weight of the output layer (step S12). Specifically, the updating unit 12 updates the output weight so that the error becomes small. Then, the computation unit 11 computes the output using the output layer where learning is completed as an initial result (step S13). After that, the reservoir computer 1 proceeds to coupling structure change processing.

[0037] A flow of the coupling structure change processing will be described with reference to FIG. 4. FIG. 4 is a flowchart illustrating a flow of the coupling structure change processing according to Embodiment 1. As illustrated in FIG. 4, the design section 20 changes the coupling structure (step S21). For example, the design section 20 may randomly change the coupling structure under predetermined constraint conditions.

[0038] The design section 20 increases n by 1 (step S22). Then, the RC circuit 10 learns (performs training) an output weight of the output layer (step S23). Furthermore, the computation unit 11 computes the output using the output layer where learning is completed as an n-th result (step S24).

[0039] Here, N is a predetermined upper limit number of times of repetition, for example, 100. When n>N is not satisfied (step S25, No), the design section 20 further changes the coupling structure (step S26). Then, the design section 20 further increases n by 1 (step S22). Consequently,

the reservoir computer **1** repeatedly executes the processing of steps **S26**, **S22**, **S23**, and **S24** until  $n$  exceeds  $N$ .

**[0040]** When  $n > N$  is satisfied, the design section **20** evaluates and selects the coupling structure (step **S27**). For example, the design section **20** evaluates the error between the final output and the teacher data (i.e., the training data) for each coupling structure in step **S24**, and selects the coupling structure corresponding to the output with the minimum error.

#### Advantages

**[0041]** As described above, the computation unit **11** computes the output for the input to the neural network including the reservoir that includes the plurality of nodes and has the coupling structure randomly determined between the plurality of nodes, and the output layer having the weight set on each node of the plurality of nodes. The change unit **21** changes the coupling structure between the plurality of nodes included in the reservoir. The updating unit **12** updates the weight of the output layer based on the output computed by the computation unit **11** for each of the coupling structures changed by the change unit **21**. The evaluation unit **22** evaluates the output computed by the computation unit **11** according to a predetermined criterion. The selection unit **23** selects a predetermined coupling structure from the coupling structures changed by the change unit **21** based on the evaluation result obtained by the evaluation unit **22**. As described above, the reservoir computer **1** may optimize the reservoir by using a meta-heuristic method. Therefore, a useful reservoir may be efficiently obtained according to Embodiment 1.

**[0042]** Also, the reservoir may be the field-programmable gate array (FPGA). In this case, the change unit **21** changes the coupling structure by rewriting the FPGA that functions as the reservoir. Consequently, the FPGA that functions as the optimized reservoir may be taken out and used.

**[0043]** The change unit **21** changes the coupling structure by the predetermined number of times. The evaluation unit **22** computes the error of the output computed by the computation unit **11** with respect to the teacher data prepared in advance, as the evaluation result. Further, the selection unit **23** selects the coupling structure with the minimum error from the coupling structures changed by the change unit **21**. By predetermining the number of times of change as described above, the reservoir may be optimized using limited time and computation resources.

**[0044]** The present invention solves the problem of the reservoir computing. As described above, the reservoir computing is a special type of RNN. Here, a background in which RNN is used will be described.

**[0045]** In order to break through the limit of Moore's Law, the entire IT industry is searching for a new computer architecture that is not a von Neumann type of the related art. Non-Neumann type machines are often proposed, ranging from machines that will take years to develop such as quantum computing to machines that are relatively easy to start such as FPGA, GPU, and effective use of GPU. All of these non-Neumann type machines efficiently solve a problem in a specific field that desires computational resources by avoiding a von Neumann bottleneck.

**[0046]** Machine learning may be considered as the problem combined with the non-Neumann type machines. Particularly in today's multi-layered deep learning, many companies are conducting research and development because

many computational resources are desired. In addition, with success of the deep learning, new models of the machine learning are being developed and improved daily at research institutes and companies. From such the background, the RNN is used as the machine learning, particularly suitable for handling time-series data.

#### Embodiment 2

**[0047]** In Embodiment 1, coupling structure change processing is performed until the condition of  $n > N$  is satisfied as illustrated in FIG. 4. On the other hand, a reservoir design system of Embodiment 2 terminates the coupling structure change processing at the point of time when a coupling structure such that the output error is reduced to a certain degree is obtained.

**[0048]** In Embodiment 2, for example, an evaluation unit **22** computes an error of output computed by a computation unit **11** with respect to teacher data prepared in advance, as an evaluation result, every time an update of a weight of an output layer is completed for each of coupling structures changed by a change unit **21**. Further, when the error is equal to or less than the threshold value, a selection unit **23** selects the coupling structure corresponding to the error.

**[0049]** A flow of the coupling structure change processing will be described with reference to FIG. 5. FIG. 5 is a flowchart illustrating a flow of the coupling structure change processing according to Embodiment 2. As illustrated in FIG. 5, a design section **20** changes the coupling structure (step **S31**). The design section **20** increases  $n$  by 1 (step **S32**). Then, an RC circuit **10** learns an output weight of the output layer (step **S33**). The computation unit **11** computes the output using the output layer where learning (training) is completed as an  $n$ -th result (step **S34**).

#### Processing Flow

**[0050]** Here, the design section **20** evaluates whether or not the error between the output and the teacher data is less than the threshold value (step **S35**). When the error is not less than the threshold value or when  $n$  is less than  $N'$  (step **S35**, No or  $n < N'$  ( $N'$  is 10, for example)), the design section **20** further changes the coupling structure (step **S36**). Then, the design section **20** further increases  $n$  by 1 (step **S32**). Consequently, a reservoir computer **1** repeatedly executes the processing of steps **S36**, **S32**, **S33**, and **S34**.

**[0051]** When the error is less than the threshold value and  $n$  is equal to or more than  $N'$  (step **S35**, Yes and  $n \geq N'$ ), the design section **20** selects the coupling structure (step **S37**). For example, the design section **20** evaluates the error between the final output and the teacher data for each coupling structure in step **S24**, and selects the coupling structure corresponding to the output with the minimum error.

#### Advantages

**[0052]** The evaluation unit **22** computes the error of the output computed by the computation unit **11** with respect to the teacher data prepared in advance, as the evaluation result, every time the update of the weight of the output layer by an updating unit **12** is completed for each of coupling structures changed by the change unit **21**. Further, when the error is equal to or less than the threshold value, a selection unit **23** selects the coupling structure corresponding to the error. As described above, since the processing is able to be

completed at the time when the reservoir whose error is equal to or less than the threshold value is obtained, the time desired for an optimization of the reservoir may be shortened.

### Embodiment 3

**[0053]** In Embodiment 3, a reservoir design system performs coupling structure change processing by using a genetic algorithm. FIG. 6 is a diagram for explaining a genetic algorithm. As illustrated in FIG. 6, first, a design section 20 prepares a plurality of coupling structures of a first generation. Then, the design section 20 evaluates each coupling structure, and generates mating and mutation of a combination by using a coupling structure having a higher evaluation result. When reaching a determined generation, the design section 20 selects the coupling structure with the minimum error from the coupling structures of the generation.

**[0054]** In Embodiment 3, for example, a change unit 21 generates the coupling structure of each generation by combination and selection of the coupling structures according to the genetic algorithm so that an evaluation result obtained by an evaluation unit 22 is improved by setting a plurality of coupling structures prepared in advance as the first generation. Further, a selection unit 23 selects a coupling structure from the coupling structures of the predetermined generation generated by the change unit 21 based on the evaluation result obtained by the evaluation unit 22.

**[0055]** For example, a reservoir computer 1 optimizes a reservoir by the genetic algorithm in a following procedure. First, the reservoir computer 1 generates a plurality of reservoirs by random coupling to set them as the first generation. The reservoir computer 1 then learns each reservoir of the first generation, derives an objective function from the results, and evaluates an adjacency matrix of each reservoir. Here, the objective function may be one that minimizes errors or one that maximizes learning efficiency. In the present embodiment, the reservoir computer 1 has the fixed number of learning steps and derives the objective function that minimizes errors.

**[0056]** Next, the reservoir computer 1 partially combines the adjacency matrices having high evaluation to generate a reservoir of a second generation. Furthermore, the reservoir computer 1 evaluates the reservoir of the second generation (evaluation of the adjacency matrix), and selects the reservoir having high evaluation from the results.

**[0057]** Hereafter, the reservoir computer 1 repeats above-mentioned steps of generation, combination, and selection to stochastically give noise to the adjacency matrix as the mutation. For example, the reservoir computer 1 gives the noise by changing a real number component of the adjacency matrix to 0 or changing a 0 component to a real number.

### Processing Flow

**[0058]** A flow of the coupling structure change processing of the first generation will be described with reference to FIG. 7. FIG. 7 is a flowchart illustrating a flow of coupling structure change processing of a first generation according to Embodiment 3. First, as illustrated in FIG. 7, the design section 20 of the reservoir computer 1 determines an initial coupling structure (step S41). The design section 20 increases  $n$  by 1 (step S42). Next, the RC circuit 10 learns

an output weight of the output layer (step S43). Then, the computation unit 11 computes the output using the output layer where learning is completed as an initial result (step S44).

**[0059]** Here, when  $n > L$  is not satisfied (step S45, No), the design section 20 further randomly changes the coupling structure (step S46). Then, the design section 20 further increases  $n$  by 1 (step S42). Consequently, the reservoir computer 1 repeatedly executes the processing of steps S46, S43, and S44 until  $n$  exceeds  $L$ .  $L$  is a predetermined upper limit number of the coupling structures for each generation. In addition, an initial value of  $n$  is set to 0. When  $n > L$  is satisfied (step S45, Yes), the reservoir computer 1 proceeds to the coupling structure change processing of the second generation.

**[0060]** A flow of the coupling structure change processing of a  $K$ -th generation will be described with reference to FIG. 8. FIG. 8 is a flowchart illustrating a flow of coupling structure change processing of the  $K$ -th generation according to Embodiment 3.  $K$  is an integer of 2 or more. First, as illustrated in FIG. 8, the design section 20 of the reservoir computer 1 changes a coupling structure according to the genetic algorithm (step S51). Here, the design section 20 changes by combining and mating the coupling structures of the  $K-1$ th generation.

**[0061]** The design section 20 increases  $n$  by 1 (step S52). Next, the RC circuit 10 learns the output weight of the output layer (step S53). Then, the computation unit 11 computes the output using the output layer where learning is completed as the initial result (step S54).

**[0062]** Here, when  $n > L$  is not satisfied (step S55, No), the design section 20 changes the coupling structure according to the genetic algorithm. (step S56). Here, the design section 20 changes by generating mutation in addition to mating combining the coupling structures of the  $K-1$ th generation.

**[0063]** Then, the design section 20 further increases  $n$  by 1 (step S52). Consequently, the reservoir computer 1 repeatedly executes the processing of steps S56, S53, and S54 until  $n$  exceeds  $L$ . When  $n > L$  is satisfied (step S55, Yes), the design section 20 selects a higher-ranking coupling structure among the coupling structures of the latest generation (step S57). Furthermore, the reservoir computer 1 proceeds to the coupling structure change processing of a  $K+1$ th generation.

**[0064]** In addition,  $L$  may become smaller as the generation proceeds. Consequently, the number of obtained coupling structures decreases as the generation proceeds, and the best coupling structure may be finally selected from the small number of coupling structures.

### Advantages

**[0065]** The change unit 21 generates the coupling structure of each generation by combination and selection of the coupling structures according to the genetic algorithm so that the evaluation result obtained by the evaluation unit 22 is improved by setting the plurality of coupling structures prepared in advance as the first generation. Further, a selection unit 23 selects a coupling structure from the coupling structures of the predetermined generation generated by the change unit 21 based on the evaluation result obtained by the evaluation unit 22. By using the genetic algorithm as described above, a better reservoir may be obtained than when the coupling structure is randomly changed.

**[0066]** FIG. 9 illustrates results of an experiment. In an example of FIG. 9, the reservoir computer 1 generated 10 reservoirs as the first generation ( $L=10$ ). The objective function (error) for evaluating the reservoir of the first generation has a minimum value of 0.1773 at an average of 0.1862. As a result of repeating for 50 generations by the genetic algorithm, the reservoir computer 1 was able to generate a reservoir in which an error value of 0.1673 was obtained.

#### System

**[0067]** Processing procedures, control procedures, specific names, and information including various kinds of data and parameters indicated in the above-mentioned specification and the drawings may be changed in any manner unless otherwise specified. Specific examples, distributions, numerical values, and so on described in the embodiments are merely examples and may be changed in any manner.

**[0068]** Further, constituent elements of the respective devices illustrated in the drawings are functional conceptual ones and not desired to be configured physically as illustrated in the drawings. That is, specific forms of distribution and integration of the respective devices are not limited to those illustrated in the drawings. In other words, all or some of the devices may be configured to be distributed or integrated functionally or physically in any units depending on various loads, usage conditions, and so on. Furthermore, all or any part of processing functions performed by the respective devices may be realized by a central processing unit (CPU) and a program to be analyzed and executed by the CPU, or may be realized as hardware by wired logic.

**[0069]** A meta-heuristic method for solving a combination optimization problem as well as the genetic algorithm may be used for optimizing the reservoir. For example, the reservoir may be optimized by an annealing method or the like.

#### Hardware

**[0070]** FIG. 10 is a diagram for describing a hardware configuration example. As illustrated in FIG. 10, the reservoir computer 1 includes a communication interface 10a, a hard disk drive (HDD) 10b, a memory 10c, and a processor 10d. Further, the respective units illustrated in FIG. 10 are coupled to each other by a bus or the like.

**[0071]** The communication interface 10a is a network interface card or the like and performs communication with other servers. The HDD 10b stores a program or a database (DB) for operating functions illustrated in FIG. 2.

**[0072]** The processor 10d is a hardware circuit that reads, from the HDD 10b or the like, a program for executing the same processing as each processing unit illustrated in FIG. 2 and loads the program to the memory 10c to operate a process of executing each function described in FIG. 2 or the like. That is, this process executes the same function as each processing unit included in the reservoir computer 1. Specifically, the processor 10d reads a program having the same function as the computation unit 11, an updating unit 12, the change unit 21, the evaluation unit 22, and the selection unit 23 from the HDD 10b or the like. Then, the processor 10d executes a process of executing the same processing as the computation unit 11, the updating unit 12, the change unit 21, the evaluation unit 22, the selection unit 23, and the like.

**[0073]** As described above, the reservoir computer 1 operates as an information processing device that performs a learning method as a result of reading and executing the program. Further, the reservoir computer 1 may also realize the same function as the embodiments described above by reading the program from a recording medium by a medium reading device and executing the read program. In addition, the program described in other embodiments is not limited to a program that is executed by the reservoir computer 1. For example, the present invention may also be similarly applied to cases where another computer or a server executes the program and where the other computer and the server execute the program in cooperation with each other.

**[0074]** This program may be distributed via a network such as the Internet. In addition, this program may be recorded on a computer-readable recording medium such as a hard disk, a flexible disk (FD), a compact disc read-only memory (CD-ROM), a magneto-optical disk (MO), or a digital versatile disc (DVD) and may be executed after being read from the recording medium by a computer.

**[0075]** All examples and conditional language provided herein are intended for the pedagogical purposes of aiding the reader in understanding the invention and the concepts contributed by the inventor to further the art, and are not to be construed as limitations to such specifically recited examples and conditions, nor does the organization of such examples in the specification relate to a showing of the superiority and inferiority of the invention. Although one or more embodiments of the present invention have been described in detail, it should be understood that the various changes, substitutions, and alterations could be made hereto without departing from the spirit and scope of the invention.

What is claimed is:

1. A reservoir computer comprising:

processor circuitry configured to

execute a computation processing configured to compute an output for an input to a neural network, the neural network including a reservoir and output layer, the reservoir including a plurality of nodes and having a coupling structure randomly determined between the plurality of nodes, the output layer having a weight set on each node of the plurality of nodes;

execute a change processing configured to change the coupling structure between the plurality of nodes included in the reservoir;

execute an updating processing configured to update the weight of the output layer based on the output computed by the computation processing for each of the coupling structures changed by the change processing;

execute an evaluation processing configured to evaluate the output computed by the computation processing according to a predetermined criterion; and

execute a selection processing configured to select a predetermined coupling structure from the coupling structures changed by the change processing based on an evaluation result obtained by the evaluation processing.

2. The reservoir computer according to claim 1, wherein the change processing is configured to change the coupling structure by a predetermined number of times, the evaluation processing is configured to compute, as the evaluation result, an error of the output computed by



- the computation processing, the error of the output being a difference between the output and training data prepared in advance, and
- the selection processing is configured to select a coupling structure with the minimum error from the coupling structures changed by the change processing.
3. The reservoir computer according to claim 1, wherein the evaluation processing is configured to compute, as an evaluation result, an error of the output every time an update of the weight of the output layer by the updating processing is completed for each of the coupling structures changed by the change processing, the error of the output being a difference between the output and training data prepared in advance,
- the selection processing is configured to select a coupling structure corresponding to the error when the error is equal to or less than a threshold value.
4. The reservoir computer according to claim 1, wherein the change processing is configured to generate a coupling structure of each generation by combination and selection of coupling structures according to a genetic algorithm so that an evaluation result obtained by the evaluation processing is improved by setting a plurality of coupling structures prepared in advance as a first generation, and
- the selection processing is configured to select a coupling structure from coupling structures of a predetermined generation generated by the change processing based on the evaluation result obtained by the evaluation processing.
5. The reservoir computer according to claim 1, wherein the reservoir is a field-programmable gate array (FPGA), the change processing is configured to change the coupling structure by rewriting the FPGA functioning as the reservoir.
6. A reservoir designing method executed by a computer configured to control a neural network including a reservoir and an output layer, the reservoir including a plurality of nodes and having a coupling structure randomly determined between the plurality of nodes, the output layer having a weight set on each node of the plurality of nodes, the method comprising:
- changing the coupling structure between the plurality of nodes included in the reservoir;
  - computing an output for an input to the neural network;
  - updating the weight of the output layer based on the output for each of the coupling structures changed by the changing;
  - evaluating the output according to a predetermined criterion; and
  - selecting a predetermined coupling structure from the coupling structures changed by the changing based on an evaluation result obtained by the evaluating.
7. The reservoir designing method according to claim 6, wherein
- the change processing is configured to change the coupling structure by a predetermined number of times,
  - the evaluation processing is configured to compute, as the evaluation result, an error of the output computed by the computation processing, the error of the output being a difference between the output and training data prepared in advance, and
  - the selection processing is configured to select a coupling structure with the minimum error from the coupling structures changed by the change processing.
8. The reservoir designing method according to claim 6, wherein
- the evaluation processing is configured to compute, as an evaluation result, an error of the output every time an update of the weight of the output layer by the updating processing is completed for each of the coupling structures changed by the change processing, the error of the output being a difference between the output and training data prepared in advance, the selection processing is configured to select a coupling structure corresponding to the error when the error is equal to or less than a threshold value.
9. The reservoir designing method according to claim 6, wherein
- the change processing is configured to generate a coupling structure of each generation by combination and selection of coupling structures according to a genetic algorithm so that an evaluation result obtained by the evaluation processing is improved by setting a plurality of coupling structures prepared in advance as a first generation, and
  - the selection processing is configured to select a coupling structure from coupling structures of a predetermined generation generated by the change processing based on the evaluation result obtained by the evaluation processing.
10. The reservoir designing method according to claim 6, wherein
- the reservoir is a field-programmable gate array (FPGA),
  - the change processing is configured to change the coupling structure by rewriting the FPGA functioning as the reservoir.
11. A non-transitory computer-readable storage medium for storing a reservoir designing program which causes a processor of a computer to perform processing, the computer configured to control a neural network including a reservoir and an output layer, the reservoir including a plurality of nodes and having a coupling structure randomly determined between the plurality of nodes, the output layer having a weight set on each node of the plurality of nodes, the processing comprising:
- changing the coupling structure between the plurality of nodes included in the reservoir;
  - updating the weight of the output layer for each of the coupling structures changed by the changing based on an output for an input to the neural network;
  - evaluating the output according to a predetermined criterion; and
  - selecting a predetermined coupling structure from the coupling structures changed by the changing based on an evaluation result obtained by the evaluating.
12. The non-transitory computer-readable storage medium according to claim 11, wherein
- the change processing is configured to change the coupling structure by a predetermined number of times,
  - the evaluation processing is configured to compute, as the evaluation result, an error of the output computed by the computation processing, the error of the output being a difference between the output and training data prepared in advance, and
  - the selection processing is configured to select a coupling structure with the minimum error from the coupling structures changed by the change processing.

the selection processing is configured to select a coupling structure with the minimum error from the coupling structures changed by the change processing.

13. The non-transitory computer-readable storage medium according to claim 11, wherein

the evaluation processing is configured to compute, as an evaluation result, an error of the output every time an update of the weight of the output layer by the updating processing is completed for each of the coupling structures changed by the change processing, the error of the output being a difference between the output and training data prepared in advance,

the selection processing is configured to select a coupling structure corresponding to the error when the error is equal to or less than a threshold value.

14. The non-transitory computer-readable storage medium according to claim 11, wherein

the change processing is configured to generate a coupling structure of each generation by combination and selec-

tion of coupling structures according to a genetic algorithm so that an evaluation result obtained by the evaluation processing is improved by setting a plurality of coupling structures prepared in advance as a first generation, and

the selection processing is configured to select a coupling structure from coupling structures of a predetermined generation generated by the change processing based on the evaluation result obtained by the evaluation processing.

15. The non-transitory computer-readable storage medium according to claim 11, wherein

the reservoir is a field-programmable gate array (FPGA),

the change processing is configured to change the coupling structure by rewriting the FPGA functioning as the reservoir.

\* \* \* \* \*