US 20190325311A1

(19) **United States**

(12) **Patent Application Publication**

**NAKAMURA et al.**

(10) **Pub. No.: US 2019/0325311 A1**

(43) **Pub. Date: Oct. 24, 2019**

(54) **NEURAL NETWORK CIRCUIT**

(71) Applicant: **Hitachi, Ltd.**, Chiyoda-ku, Tokyo (JP)

(72) Inventors: **Toshiaki NAKAMURA**, Tokyo (JP); **Teppei HIROTSU**, Tokyo (JP); **Tatsuya HORIGUCHI**, Tokyo (JP)

(73) Assignee: **Hitachi, Ltd.**, Chiyoda-ku, Tokyo (JP)

(21) Appl. No.: **16/470,633**

(22) PCT Filed: **Jan. 10, 2017**

(86) PCT No.: **PCT/JP2017/000367**

§ 371 (c)(1),
(2) Date: **Jun. 18, 2019**

**Publication Classification**

(51) **Int. Cl.**

| | |
|---|---|
| ***G06N 3/08*** | (2006.01) |
| ***G06N 3/04*** | (2006.01) |
| ***G06F 17/11*** | (2006.01) |

(52) **U.S. Cl.**

CPC ..... ***G06N 3/082*** (2013.01); ***G06F 17/11*** (2013.01); ***G06N 3/04*** (2013.01)

(57) **ABSTRACT**

The present invention addresses the problem of implementing a neural network using a small-scale circuit by simplifying the multiplication of the input data by weight data. The neural network circuit according to the present invention is configured from: a means for multiplying input data by a rounded value of the mantissa part of weight data; a means for shifting the multiplication result by the number of bits of the rounded value; a means for adding the shifted result to the original input data; and a means for shifting the addition result by the number of bits of the exponent part of the weight.

801

| W | Wb | FORMULA 1 |
|---|---|---|
| $2^0 \sim 0$ (SAME ON ONE SIDE) | 0.75 | $3*2^{-2}$ |
| | 0.5 | $2*2^{-2}$ |
| | 0.25 | $1*2^{-2}$ |
| | 0 | $0*2^{-2}$ |

802

| W | Wb | FORMULA 2 |
|---|---|---|
| $2^{-1} \sim 0$ (SAME ON ONE SIDE) | 0.375 | $2^{-1}$*FORMULA 1 |
| | 0.25 | |
| | 0.125 | |
| | 0 | |

803

| W | Wb | FORMULA 3 |
|---|---|---|
| $2^{-2} \sim 0$ (SAME ON ONE SIDE) | 0.1875 | $2^{-2}$*FORMULA 1 |
| | 0.125 | |
| | 0.0625 | |
| | 0 | |

FIG. 1

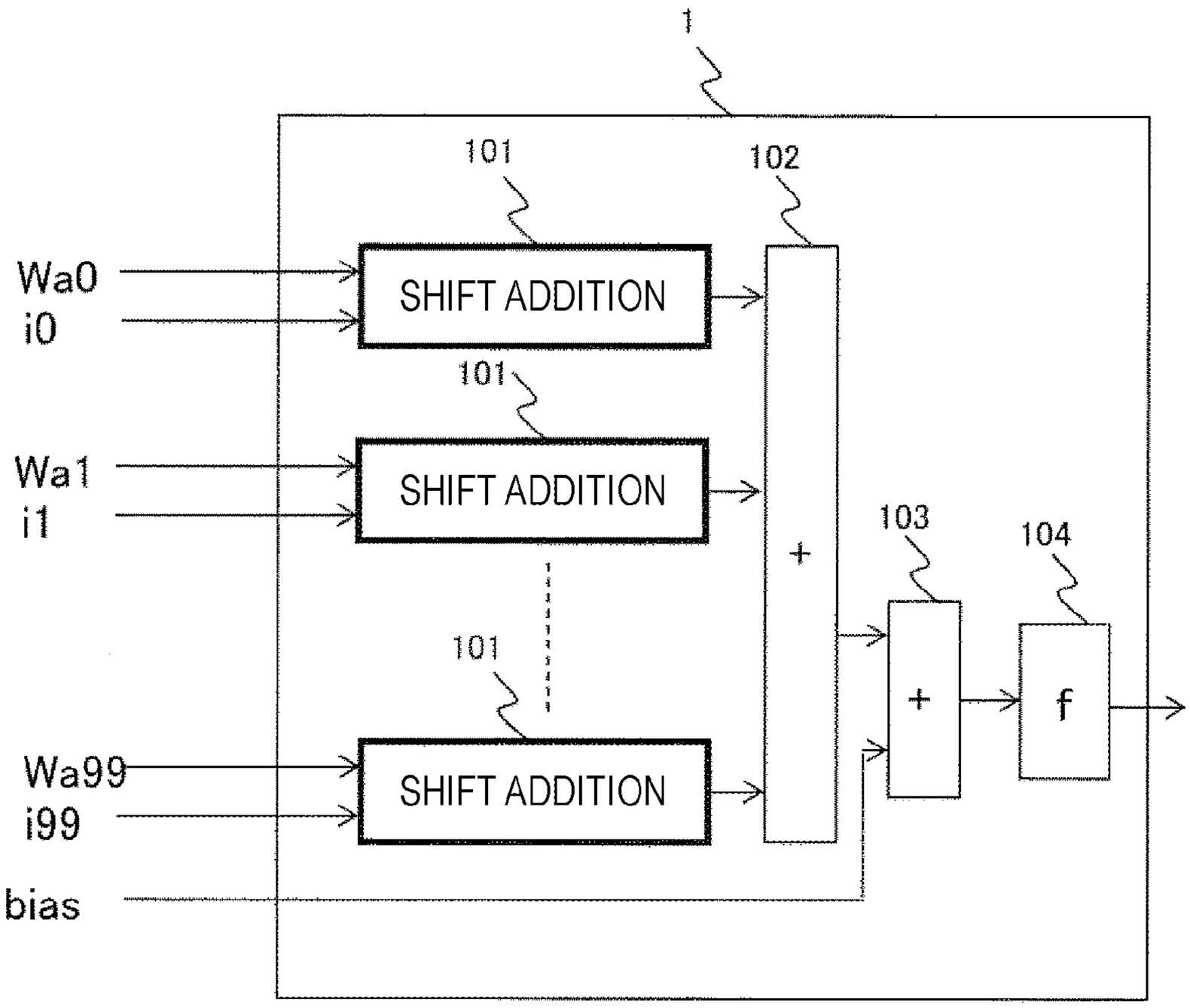

1
101
102
Wa0
i0
SHIFT ADDITION
101
Wa1
i1
SHIFT ADDITION
+
103
104
101
+
f
Wa99
i99
SHIFT ADDITION
bias

FIG. 2

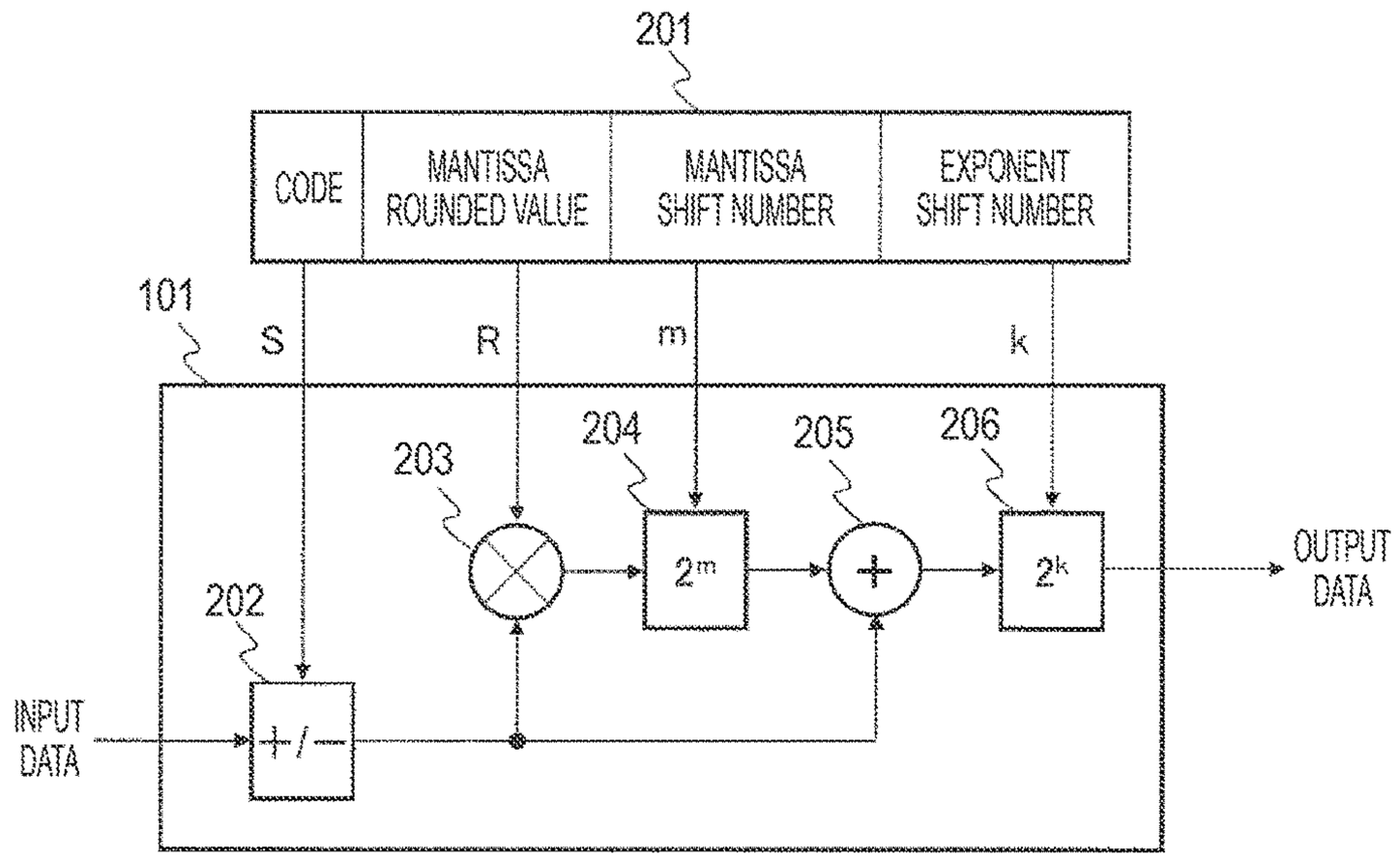



FIG. 3

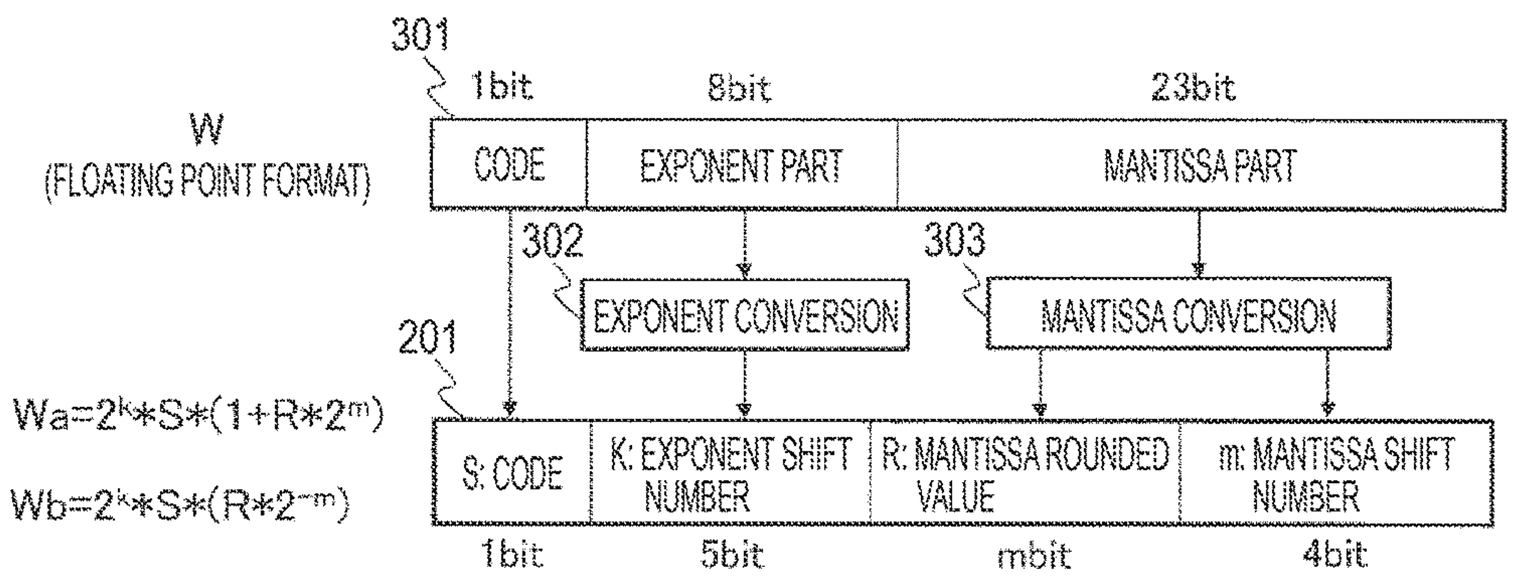



FIG. 4

401

| W | Wa | FORMULA 1 |
|---|---|---|
| $2^1 \sim 2^0$ (SAME ON ONE SIDE) | 1.75 | $1+3*2^{-2}$ |
| | 1.5 | $1+2*2^{-2}$ |
| | 1.25 | $1+1*2^{-2}$ |
| | 1.0 | $1+0*2^{-2}$ |

402

| W | Wa | FORMULA 2 |
|---|---|---|
| $2^0 \sim 2^{-1}$ (SAME ON ONE SIDE) | 0.875 | $2^{-1}$*FORMULA 1 |
| | 0.75 | |
| | 0.625 | |
| | 0.5 | |

403

| W | Wa | FORMULA 3 |
|---|---|---|
| $2^{-1} \sim 2^{-2}$ (SAME ON ONE SIDE) | 0.4375 | $2^{-2}$*FORMULA 1 |
| | 0.375 | |
| | 0.3125 | |
| | 0.25 | |

FIG. 5

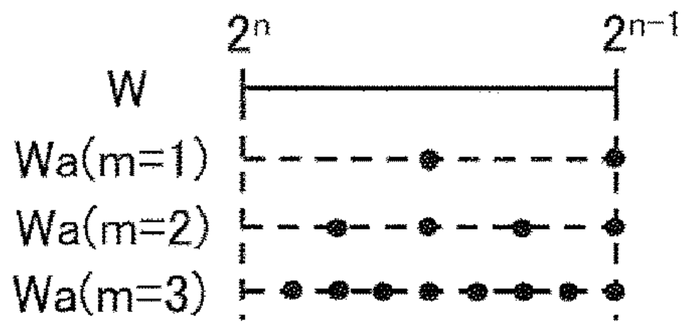



FIG. 6

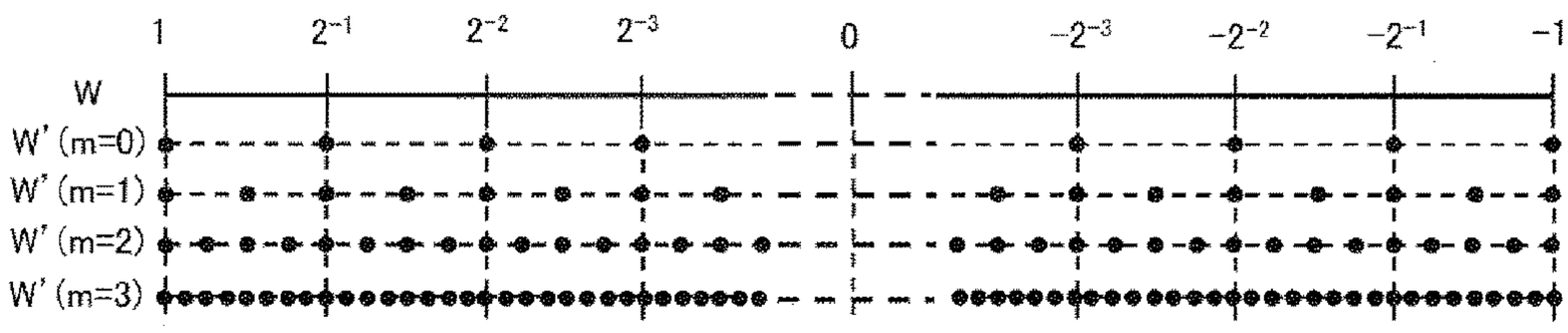



FIG. 7

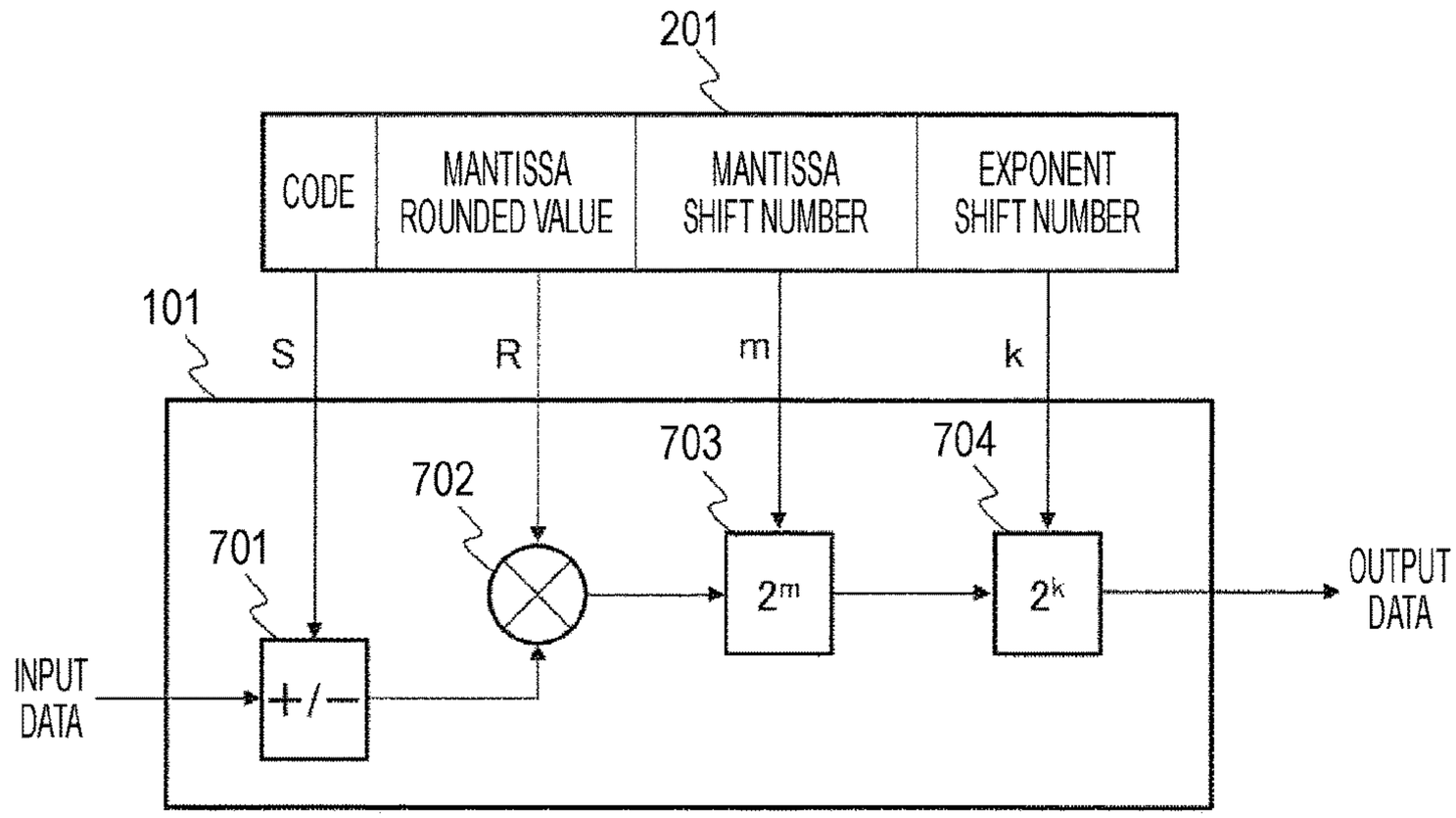


FIG. 8

801

| W | Wb | FORMULA 1 |
|---|---|---|
| $2^0 \sim 0$ (SAME ON ONE SIDE) | 0.75 | $3*2^{-2}$ |
| | 0.5 | $2*2^{-2}$ |
| | 0.25 | $1*2^{-2}$ |
| | 0 | $0*2^{-2}$ |

802

| W | Wb | FORMULA 2 |
|---|---|---|
| $2^{-1} \sim 0$ (SAME ON ONE SIDE) | 0.375 | $2^{-1}$*FORMULA 1 |
| | 0.25 | |
| | 0.125 | |
| | 0 | |

803

| W | Wb | FORMULA 3 |
|---|---|---|
| $2^{-2} \sim 0$ (SAME ON ONE SIDE) | 0.1875 | $2^{-2}$*FORMULA 1 |
| | 0.125 | |
| | 0.0625 | |
| | 0 | |

FIG. 9

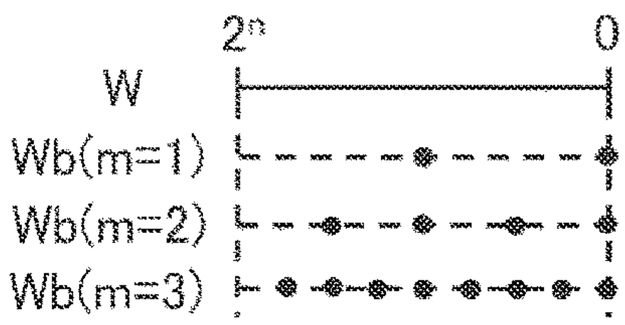



FIG. 10

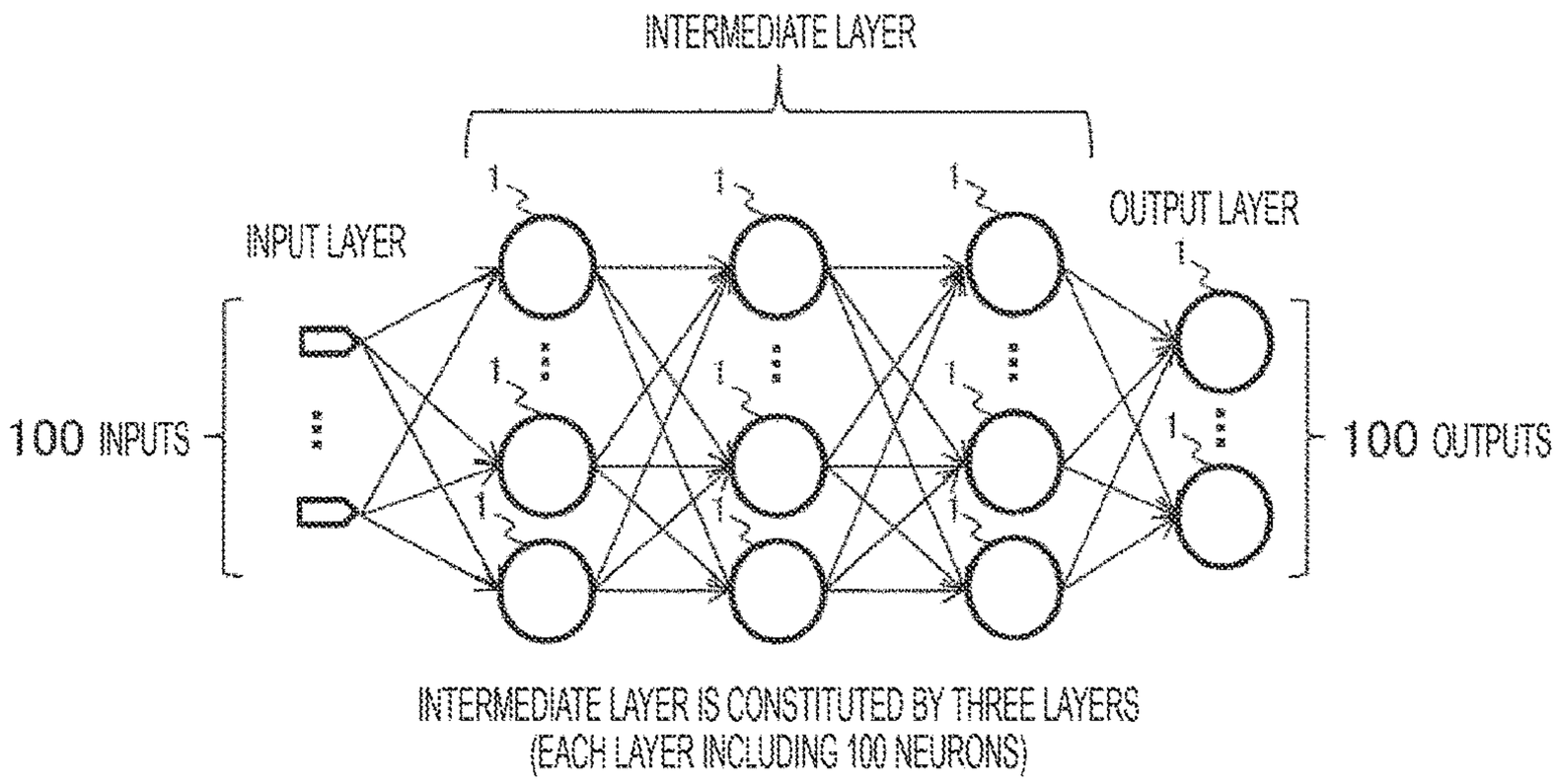
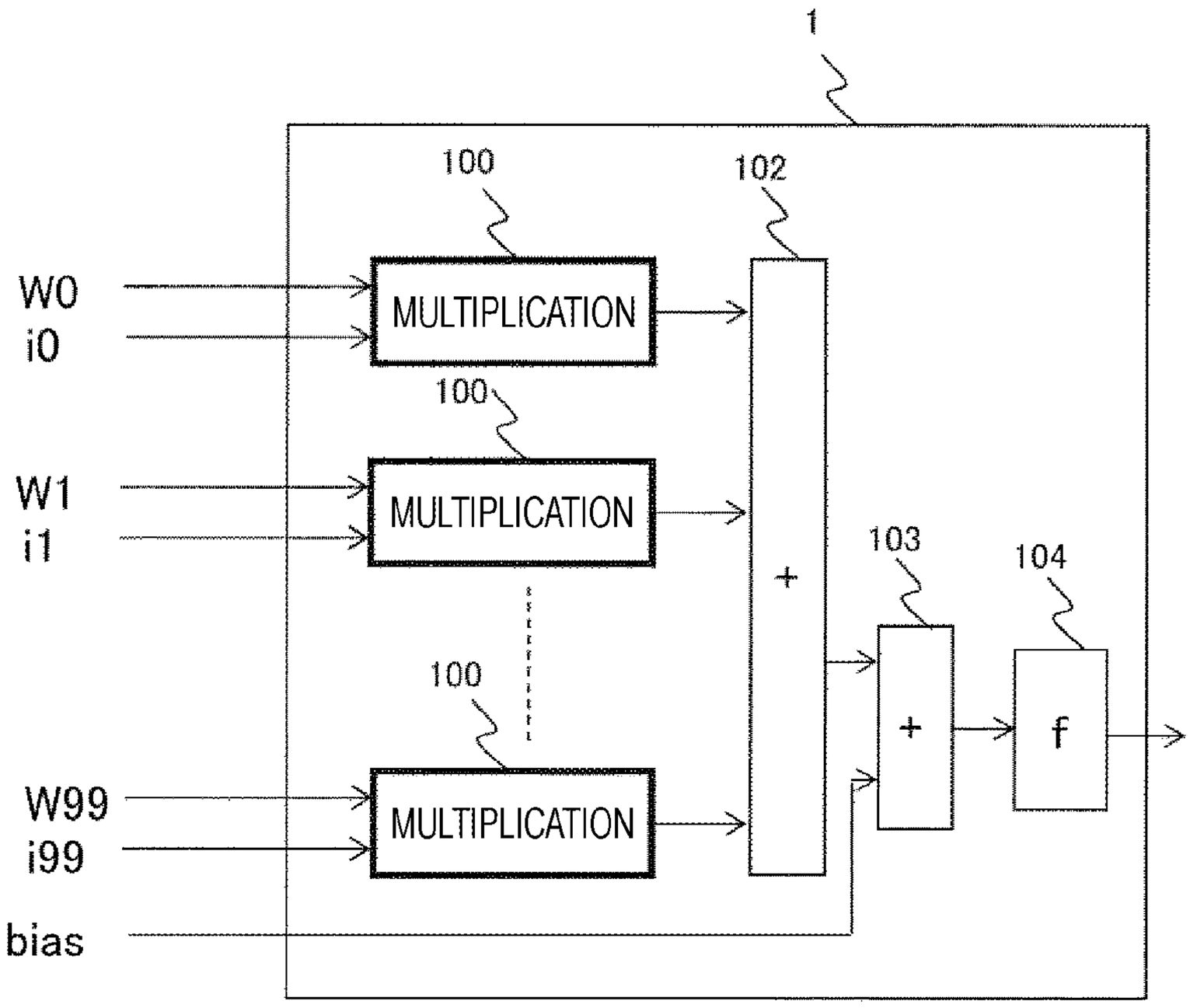


FIG. 11


1
100
102
W0
i0
MULTIPLICATION
100
W1
i1
MULTIPLICATION
+
103
104
100
W99
i99
MULTIPLICATION
+
f
bias

# NEURAL NETWORK CIRCUIT

## TECHNICAL FIELD

**[0001]** The present invention relates to a neural network circuit.

## BACKGROUND ART

**[0002]** As a circuit configuration method of neurons in a neural network, a device as disclosed in JP H4-51384 A (PTL 1) is disclosed. In PTL 1, weight data is approximated by one power of two or a sum of a plurality of powers of two. Illustrated is an example in which a power-of-two operation is configured using a bit shift circuit, and results of the operation are added by an adder so that the multiplication of input data and weight data is approximated by a small-scale circuit.

## CITATION LIST

### Patent Literature

**[0003]** PTL 1: JP H4-51384 A

## SUMMARY OF INVENTION

### Technical Problem

**[0004]** There is a deep neural network as one method of machine learning. A neuron, which is a basic unit of the neural network, is configured to multiply a plurality of pieces of input data by corresponding weight factors, respectively, add the multiplication results, and output the addition result. Thus, in the case of implementation with a logic circuit such as an FPGA, a large number of multipliers are required so that an increase in circuit scale becomes a problem.

**[0005]** Therefore, a problem is to implement a neural network using a small-scale circuit by simplifying the multiplication of input data by weight data.

### Solution to Problem

**[0006]** The present invention has been made in view of such circumstances. As an example, a neural network circuit according to the present invention is configured from: a means for multiplying input data by a rounded value of the mantissa part of weight data; a means for shifting the multiplication result by the number of bits of the rounded value; a means for adding the shifted result to the original input data; and a means for shifting the addition result by the number of bits of the exponent part of the weight.

### Advantageous Effects of Invention

**[0007]** According to the present invention, it is possible to implement a neural network using a small-scale circuit by simplifying the multiplication of the input data by weight data.

## BRIEF DESCRIPTION OF DRAWINGS

**[0008]** FIG. **1** is a block diagram illustrating a neuron in an embodiment.

**[0009]** FIG. **2** is a block diagram illustrating a shift addition means in the embodiment.

**[0010]** FIG. **3** is a block diagram illustrating conversion of weight factors in the embodiment.

**[0011]** FIG. **4** is a table illustrating values of the weight factors in the embodiment.

**[0012]** FIG. **5** is a view illustrating the weight factors according to a mantissa shift number for a range of $2^n$ to $2^{n-1}$ in the embodiment.

**[0013]** FIG. **6** is a view illustrating the weight factors according to the mantissa shift number for a range of weights from 1 to −1 in the embodiment.

**[0014]** FIG. **7** is a block diagram illustrating a shift addition means in an embodiment.

**[0015]** FIG. **8** is a table illustrating values of weight factors in the embodiment.

**[0016]** FIG. **9** is a view illustrating the weight factors according to the mantissa shift number for a range of $2^n$ to 0 in the embodiment.

**[0017]** FIG. **10** is a block diagram illustrating an example of a general neural network circuit

**[0018]** FIG. **11** is a block diagram illustrating an example of a conventional neuron.

## DESCRIPTION OF EMBODIMENTS

**[0019]** Hereinafter, an embodiment of the present invention will be described with reference to FIGS. **1** to **11**.

**[0020]** FIG. **10** is a block diagram of a neural network circuit on which the embodiment is mounted. As an example, a neural network that recognizes a character from an image of the character written manually. In FIG. **10**, in an input layer, image data indicating gray values of a total of 100 pixels of 10 pixels×10 pixels in the vertical direction is input as 100 signals. Next, three intermediate layers are configured using 100 neurons **1** for each layer. An output of each neuron is input to the 100 neurons **1** of the next layer, and the 100 neurons **1** of an output layer finally output a recognition result.

**[0021]** For example, when an output of the neuron **1** at the top of the output layer is a character "A", it is configured to output a larger value than the neurons **1** of the other output layers. In addition, when an output of the second neuron **1** from the top is a character "B", it is configured to output a larger value than neurons in the other output layers. As described above, an identification result of the character is obtained depending on which number of neuron outputs the maximum value.

**[0022]** FIG. **11** is a block diagram of a conventional general neuron **1**. Multiplication **100** performs multiplication of an input in and a weight Wn. An adder **102** is a means for obtaining a sum of multiplication results of inputs i0 to i99 by the respective pieces of weight data. An adder **103** is a means for adding an output of the adder **102** and bias data. An output function **104** is a means for performing a certain function operation on an output value of the adder **103** and outputting the result.

**[0023]** For the purpose of processing the neural network circuit of FIG. **10** at high speed, the single neuron **1** requires 100 multipliers in the case of implementation with a logic circuit. Even when only one intermediate layer is subjected to the operation at one time, 100 neurons×100=10,000 multipliers are required. The number of bits of the weight W to be input to the multiplier **100** requires 8 bits, and the number of bits of image input data requires 8 bits in order to obtain high character recognition performance, and a circuit scale becomes enormous if these are implemented using the logic circuit.

**[0024]** Therefore, one purpose of the present embodiment is to implement a neural network circuit with a small-scale circuit while maintaining performance by performing such multiplication using an operation of a combination of addition and bit shift.

**[0025]** FIG. **1** is a block diagram of a neuron in an embodiment. A shift addition **101** executes a multiplication operation of an input in and a weight factor Wa. Incidentally, details will be described later with reference to FIG. **2**. An adder **102** is a means for obtaining a sum of shift addition results of inputs i0 to i99 by the respective weight factors. The adder **102** is a means for adding an output of the adder **102** and bias data. An output function **104** is a means for performing a certain function operation on an output value of the adder **103** and outputting the result.

**[0026]** FIG. **2** is a block diagram of a shift addition means in the embodiment. A weight factor storage unit **201** is a means for storing weight factors. The shift addition **101** obtains the weight factor Wa from (Formula a) $Wa=S*(1+R*2^{-m})*2^{k}$, and multiply the input in by the obtained weight factor Wa. Here, S indicates a positive or negative sign of a weight, and is 1 in the case of the positive sign and −1 in the case of the negative sign. Further, m is an integer of 0 or more and indicates accuracy of a weight factor, and has $2^{m}$ values in a range of $2^{n}$ to $2^{n+1}$. Details will be described later with reference to FIG. **5**.

**[0027]** FIG. **5** is a view illustrating the weight factor Wa according to the number of mantissa shifts m in a range of $2^{n}$ to $2^{n-1}$. In the range of $2^{n}$ to $2^{n-1}$, the weight factor has two points when m=1, four points when m=2, and eight points when m=3. As described above, the number of points of the weight factors is $2^{m}$.

**[0028]** Next, in the above Formula a, R is a rounded value of a weight, and is an integer in a range of less than 0 and $R<2^{m}$. Further, K is a bit shift number corresponding to the exponent of weight and is an integer.

**[0029]** The shift addition **101** of FIG. **2** illustrates a specific implementation method of the above-described operation expression. A code conversion **202** is a function of converting the input in either positive or negative data. The input in is directly output if the code is 0, and the input in is multiplied by −1 and the multiplication result is output if the code is 1.

**[0030]** A multiplier **203** is a means for multiplying a mantissa rounded value which is an output from the weight factor storage unit **201**.

**[0031]** A mantissa shift **204** is a means for bit-shifting an output of the multiplier **203** according to a mantissa shift number which is an output from the weight factor storage unit **201**. If the mantissa shift number is a positive value, the shift is performed to the left. In addition, the shift is performed to the right if the mantissa shift number is a negative value.

**[0032]** An adder **205** is a means for adding an output of the mantissa shift **204** and an output of the code conversion **202**.

**[0033]** The exponent shift **206** is a means for bit-shifting an output of the adder **205** according to an exponent shift number which is an output from the weight factor storage unit **201**. If the exponent shift number is a positive value, the shift is performed to the left. In addition, the shift is performed to the right if the mantissa shift number is a negative value.

**[0034]** The weight factor according to the above Formula a is reliably applied from the time of first learning to obtain the weight factor of each neuron in the neural network of FIG. **10**. Incidentally, learning to obtain a weight factor may be performed by a floating-point operation using a computer, and the obtained weight factor may be approximated to the weight factor obtained by the above formula so that the neural network may be implemented by a small-scale logic circuit as illustrated in FIG. **3**.

**[0035]** FIG. **3** is a block diagram illustrating a correspondence between a conventional weight factor **301** described in a floating-point format and the weight factor **201** used for the shift addition processing of the present embodiment.

**[0036]** Although the present embodiment is directed to a fixed point operation, an example for easily obtaining the weight factor to be used for the shift addition processing from the weight factor of the floating-point format is illustrated. This example is advantageous in a case where, for example, learning to obtain a weight factor is performed by a floating-point operation using a computer, and the obtained weight factor is converted into the weight factor of the present invention so that a neural network is implemented by a small-scale logic circuit.

**[0037]** In the weight factor **201**, the reference sign S is the same as the reference sign of the weight factor storage unit **301** in the floating-point format.

**[0038]** An exponent shift number K is generated by an exponent conversion **302** based on exponent part data of the weight factor storage unit **301**. Specifically, a value of the exponent of the floating-point format is a value obtained by adding **127** as an offset. Therefore, a value obtained by subtracting **127** is set as the exponent shift number in 2's complement notation in the exponent conversion **302**.

**[0039]** The mantissa rounded value R is generated by a mantissa conversion **303** based on mantissa part data of the weight factor storage unit **301**. Specifically, the upper m bits of the mantissa part data are used as the mantissa rounded value.

**[0040]** The mantissa shift number m is generated by the mantissa conversion **303** based on mantissa part data of the weighting data **301**. Specifically, the number of bits of the mantissa rounded value R is set as m.

**[0041]** FIG. **4** is a table illustrating an example of the weight factor. In the present embodiment, a formula to be calculated is determined depending on which range of powers of 2 the weight factor exists on the basis of the power of 2. A specific numerical example will be described hereinafter.

**[0042]** First, a table **401** illustrates values of the weight factor Wa in the case where a weight range W is in a range of 2 to 1 and the mantissa shift number m=2, and Formula 1 to obtain the values. According to Formula 1, a value of R can take four integer values from 0 to 3, and thus, the weight factor Wa takes four values from 1.0 to 1.75 by an increment of 0.25.

**[0043]** Next, a table **402** illustrates values of a weight approximate value Wa in the case where the weight range W is in a range of 1.0 to 0.5 and the mantissa shift number m=2, and Formula 2 to obtain the values. Formula 1 is multiplied by $2^{-1}$ by setting the exponent shift number K=−1, thereby obtaining the result of Formula 2.

**[0044]** Next, a table **403** illustrates values of the weight factor Wa in the case where the weight range W is in a range of 0.5 to 0.25 and the mantissa shift numbers m=2, and Formula 3 to obtain the values. Formula 1 is multiplied by

$2^{-2}$ by setting the exponent shift number K=−2, thereby obtaining the result of Formula 3.

**[0045]** As described above, a feature of the present embodiment is that the bit shift is performed according to the range of powers of 2 including the weight factor based on the weight factor value obtained by Formula 1 so that the weight value is not rounded to 0 although being a value close to 0.

**[0046]** FIG. **6** is a view illustrating the weight factor Wa according to the mantissa shift number m in the range of 1.0 to −1.0. Since the number of weight factors is always constant in the range of $2^n$ to $2^{n-1}$, the weight factor is not rounded to 0 although being a value close to 0, and the multiplication with input data can be performed with high accuracy.

**[0047]** FIG. **7** is a block diagram of a shift addition means according to another embodiment. A weight factor storage unit **201** is a means for storing weight factor values. A shift addition **101** obtains a weight factor value Wb for an input in from (Formula b) $Wb=S*R*2^{-m}*2^{k}$, and multiply the input in by the obtained weight factor value Wb. Here, S indicates a positive or negative sign of a weight, and is 1 in the case of the positive sign and −1 in the case of the negative sign. Further, m is an integer of 0 or more and indicates accuracy of a weight factor, and has $2^m$ weight factors in a range of $2^n$ to 0. Details will be described later with reference to FIG. **9**.

**[0048]** FIG. **9** is a view illustrating the weight factor Wb according to a mantissa shift number m with respect to the weight factors in the range of $2^n$ to 0. In the range of $2^n$ to 0, there are two points when m=1, four points when m=2, and eight points when m=3. As described above, the number of points of the weight factors is $2^m$.

**[0049]** Next, in the above Formula b, R is a rounded value of a weight, and is an integer in a range of $0 \leq R < 2^m$.

**[0050]** Further, K is a bit shift number corresponding to the exponent of weight and is an integer. A shift addition **101** of FIG. **7** illustrates a specific implementation method of the above-described operation expression.

**[0051]** A code conversion **701** is a function of converting input data into either positive or negative data. The input data is directly output if the code is 0, and the input data is multiplied by −1 and the multiplication result is output if the code is 1.

**[0052]** A multiplier **702** is a means for multiplying a mantissa rounded value which is an output from the weight factor storage unit **201**.

**[0053]** A mantissa shift **703** is a means for bit-shifting an output of the multiplier **702** according to a mantissa shift number which is an output from the weight factor storage unit **201**. If the mantissa shift number is a positive value, the shift is performed to the left. In addition, the shift is performed to the right if the mantissa shift number is a negative value.

**[0054]** An exponent shift **704** is a means for bit-shifting an output from the mantissa shift **703** according to an exponent shift number. If the exponent shift number is a positive value, the shift is performed to the left. In addition, the shift is performed to the right if the mantissa shift number is a negative value.

**[0055]** The weight factor according to the above Formula b is reliably applied from the time of first learning to obtain the weight factor of each neuron in the neural network of FIG. **10**. Incidentally, learning to obtain a weight factor may be performed by a floating-point operation using a computer, and the obtained weight factor may be approximated to the weight factor obtained by the above formula so that the neural network may be implemented by a small-scale logic circuit as illustrated in FIG. **3**.

**[0056]** FIG. **8** is a table illustrating an example of the weight factor in FIG. **7**.

**[0057]** In the present embodiment, a formula to be calculated is determined depending on which range of powers of 2 including a weight value on the basis of the power of 2. A specific numerical example will be described hereinafter.

**[0058]** First, a table **801** illustrates values of the weight factor Wb in the case where a weight range W is in a range of 1 to 0 and the mantissa shift number m=2, and Formula 1 to obtain the values. According to Formula 1, a value of R can take four integer values from 0 to 3, and thus, the weight factor Wb takes four values from 1 to 0 by an increment of 0.25.

**[0059]** Next, a table **802** illustrates values of the weight factor Wb in the case where the weight range W is in a range of 0.5 to 0 and the mantissa shift numbers m=2, and Formula 2 to obtain the values. Formula 1 is multiplied by $2^{-1}$ by setting the exponent shift number K=−1, thereby obtaining the result of Formula 2.

**[0060]** Next, a table **803** illustrates values of the weight factor Wb in the case where the weight range W is in a range of 0.25 to 0 and the mantissa shift numbers m=2, and Formula 3 to obtain the values. Formula 1 is multiplied by $2^{-2}$ by setting the exponent shift number K=−2, thereby obtaining the result of Formula 3.

**[0061]** As described above, a feature of the present embodiment is that the bit shift is performed according to the range of powers of 2 including the weight factor based on the weight factor obtained by Formula 1 so that the weight value is not rounded to 0 although being a value close to 0.

**[0062]** According to the respective embodiments described above, it is possible to make the number of weight factor values in the range of $2^n$ to $2^{n-1}$ constant, to prevent a small weight factor value from being rounded to 0, and to implement the neural network with the small-scale circuit while maintaining the performance. In addition, the circuit system is generalized and it is easy to adjust the performance and the circuit scale in accordance with an application target of the DNN.

**[0063]** Although the embodiments of the present invention have been described above, the present invention is not limited to the above-described respective embodiments, and includes various modifications. For example, a part of each of the embodiments can be added, converted, deleted, or the like within a range where the effects of the present invention are exhibited. In addition, it is possible to replace a part of each of the embodiments.

**[0064]** That is, the above-described embodiments have been described in detail in order to facilitate the understanding of the present invention, and are not necessarily limited to one including the above-described configuration thereof.

## REFERENCE SIGNS LIST

**[0065]** **1** neuron
**[0066]** **100** multiplier
**[0067]** **101** shift addition
**[0068]** **102** adder
**[0069]** **103** adder
**[0070]** **104** output function

**[0071]** **201** weight factor storage unit of present invention
**[0072]** **202** code conversion
**[0073]** **203** multiplier
**[0074]** **204** bit shifter
**[0075]** **205** adder
**[0076]** **206** bit shifter
**[0077]** **301** weight factor storage unit of floating-point format
**[0078]** **401** to **403** weight factor table
**[0079]** **701** code conversion
**[0080]** **702** multiplier
**[0081]** **703** bit shifter
**[0082]** **704** bit shifter
**[0083]** **801** to **803** weight factor table

**1**. A neural network circuit comprising
a means for providing weight factors corresponding to a plurality of pieces of input data as each information indicating a weight rounded value and an amount of bit shift and performing a weighting operation on the input data by multiplication with the weight rounded value and the bit shift, in neurons constituting a neural network.

**2**. The neural network circuit according to claim **1**, wherein
in the neurons constituting the neural network, a value after the weighting operation on the input data is a result of the multiplication with the weight rounded value and the bit shift.

**3**. The neural network circuit according to claim **1**, wherein
a circuit on which the weight factors corresponding to the plurality of pieces of input data are mounted includes a means for converting the input data to the information indicating the weight rounded value and the amount of bit shift.

**4**. The neural network circuit according to claim **1**, wherein
the weight factors, obtained by an operation in which the number of the weight factors is constant in a range between $2^n$ and $2^{n-1}$, are provided.

**5**. The neural network circuit according to claim **1**, wherein
the weight factors, obtained by an operation in which the number of the weight factors is constant in a range between $2^n$ and 0, are provided.

**6**. A neural network circuit comprising:
a means for multiplying input data by a rounded value of a mantissa part of weight data;
a means for shifting the multiplication result by the number of bits of the rounded value;
a means for adding the shifted result to the original input data; and
a means for shifting the addition result by the number of bits of an exponent part of the weight.

* * * * *