

METHOD AND APPARATUS FOR QUANTIZING ARTIFICIAL NEURAL NETWORK

CROSS-REFERENCE TO RELATED APPLICATION

[1] This application claims priority from Korean Patent Application No. 10-2017-0150707, filed on November 13, 2017 in the Korean Intellectual Property Office, the disclosure of which is incorporated herein in its entirety by reference.

TECHNICAL FIELD

[2] Apparatuses and methods consistent with one or more exemplary embodiments relate to an artificial neural network, and more particularly, to a method and apparatus for quantizing an artificial neural network.

BACKGROUND

[3] An artificial neural network (ANN) may refer to a computing device or a method performed by a computing device to generate interconnected sets of artificial neurons (or neuron models). Artificial neurons may generate output data by performing simple operations on input data, and the output data may be transmitted to other artificial neurons. As an example of an ANN, a deep neural network (DNN) or deep learning architecture may have a multi-layer structure. Layers of an ANN may have parameters and multiple sample distributions may be generated, by training each of the layers according to multiple samples.

SUMMARY

[4] Aspects of one or more exemplary embodiments provide a method and apparatus for quantizing an artificial neural network (ANN), by which an arbitrary ANN is quantized with high accuracy.

[5] According to an aspect of an exemplary embodiment, there is provided an artificial neural network (ANN) quantization method for generating an output ANN by quantizing an input ANN, the ANN quantization method including: obtaining second parameters by quantizing first parameters of the input ANN; obtaining a sample distribution from an intermediate ANN in which the obtained second parameters have been applied to the input ANN; and obtaining a fractional length for the obtained sample distribution by quantizing the obtained sample distribution.

[6] According to an aspect of another exemplary embodiment, there is provided an apparatus for generating an output artificial neural network (ANN) by quantizing an input ANN, the apparatus including: a memory storing computer-executable instructions; and at least one processor configured to execute the stored computer-executable instructions to implement: a parameter quantizer configured to obtain second parameters by quantizing first parameters of the input ANN; a neural network interface configured to obtain a sample distribution from an intermediate ANN in which the obtained second parameters have been applied to the input ANN; and a sample quantizer configured to obtain a fractional length for the obtained sample distribution.

[7] According to an aspect of another exemplary embodiment, there is provided a method of quantizing a floating-point neural network, the method including: obtaining quantized parameters by quantizing parameters in a same category in the floating-point neural network; obtaining a sample distribution from a semifixed-point artificial neural

network (ANN) in which the obtained quantized parameters have been applied to the floating-point neural network; and obtaining a fractional length for the obtained sample distribution.

[8] According to an aspect of another exemplary embodiment, there is provided an apparatus for quantizing a floating-point neural network (ANN), the apparatus including: a memory storing computer-executable instructions; and at least one processor configured to execute the stored computer-executable instructions to implement: a parameter quantizer configured to obtain quantized parameters by quantizing parameters in a same category in the floating-point neural network; a neural network interface configured to obtain a sample distribution from a semifixed-point artificial neural network (ANN) in which the obtained quantized parameters have been applied to the floating-point neural network; and a sample quantizer configured to obtain a fractional length for the obtained sample distribution.

BRIEF DESCRIPTION OF THE DRAWINGS

[9] One or more exemplary embodiments will be more clearly understood from the following detailed description taken in conjunction with the accompanying drawings in which:

[10] FIG. 1 is a block diagram of a quantization system and a neural network system, according to an exemplary embodiment;

[11] FIG. 2 is a diagram illustrating an example of an artificial neural network (ANN) according to an exemplary embodiment;

[12] FIG. 3 is a flowchart of a method of quantizing an ANN, according to an exemplary embodiment;

[13] FIG. 4 is a flowchart of an example of operation S200 of FIG. 3, according to an exemplary embodiment;

[14] FIG. 5 is a flowchart of an example of operation S600 of FIG. 3, according to an exemplary embodiment;

[15] FIGS. 6A-6C illustrate an example of operation S620 of FIG. 5, according to an exemplary embodiment;

[16] FIG. 7 is a flowchart of an example of operation S660 of FIG. 5, according to an exemplary embodiment;

[17] FIG. 8 is a flowchart of an example of operation S680 of FIG. 3, according to an exemplary embodiment;

[18] FIG. 9 is a flowchart of an example of operation S682 of FIG. 8, according to an exemplary embodiment;

[19] FIGS. 10A and 10B are flowcharts of examples of operation S684 of FIG. 8, according to one or more exemplary embodiments;

[20] FIG. 11 is a block diagram of a computing system according to an exemplary embodiment; and

[21] FIG. 12 is a block diagram of a mobile computing device according to an exemplary embodiment.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[22] Hereinafter, one or more exemplary embodiments will be described in detail in conjunction with the accompanying drawings, wherein like reference numerals refer to like elements throughout. As used herein, the term “and/or” includes any and all combinations of one or more of the associated listed items. Similarly, expressions such as

“at least one of,” when preceding a list of elements, modify the entire list of elements and do not modify the individual elements of the list.

[23] FIG. 1 is a block diagram of a quantization system 100 and a neural network system 200 according to an exemplary embodiment. The neural network system 200 may provide an artificial neural network (ANN), and the quantization system 100 may quantize the ANN provided by the neural network system 200 and provide the neural network system 200 with an ANN of which at least a portion has been quantized.

[24] The ANN may refer to a computing system inspired from a biological neural network that forms an animal’s brain. The ANN may learn to perform tasks by considering multiple samples (or examples), unlike classical algorithms that perform tasks according to predefined conditions, such as rule-based programming. The ANN may have a structure in which artificial neurons (or neurons) are connected to one another. A connection between the neurons may be referred to as a synapse. Each of the neurons may process a received signal and transmit a processed signal to another neuron through the synapse. The output of the neuron may be referred to as “activation.” The neuron and/or the synapse may have varying weights, and the influence of a signal processed by the neuron may increase or decrease depending on its weight. In particular, the weight associated with an individual neuron may be referred to as a bias.

[25] A deep neural network (DNN) or deep learning architecture may have a layer structure, and an output of a particular layer may be an input of a subsequent layer. In such a multi-layered structure, each layer may be trained according to multiple samples. An ANN, such as the DNN, may be implemented by a plurality of processing nodes, each corresponding to an artificial neuron, and may utilize high computational complexity and computing resources to obtain good (e.g., high-accuracy) results.

[26] To reduce computational complexity, an ANN may be quantized. Quantization may refer to a process of mapping input values to a smaller number of values than the number of input values, like mapping a real number to an integer through rounding. For example, quantization in an ANN may be applied to layer activations, weights, biases, thresholds, input/output distributions, and so on. A quantized ANN may enable use of a reduced number of computing resources.

[27] Quantization of an ANN may involve a process of converting a floating-point neural network into a fixed-point neural network. Converting a floating-point neural network into a fixed-point neural network may include determining (e.g., obtaining) a fractional length of the fixed-point neural network. Floating-point numbers may include signs, exponents, and significant digits (or significand), whereas fixed-point numbers may include integer parts and fraction parts. In some examples, the integer parts of the fixed-point numbers may include sign bits. An ANN using floating-point numbers, i.e., a floating-point neural network, may have high accuracy and high computational complexity, whereas an ANN using fixed-point numbers, i.e., a fixed-point neural network, may have reduced accuracy and reduced computational complexity.

[28] Due to a trade-off relationship between accuracy of results and computational complexity in the ANN, quantization of the ANN may result in a variation in accuracy and the variation in the accuracy may depend on a quantization method. As will be described below with reference to drawings, the quantization system 100 according to an exemplary embodiment may provide an efficient means for quantization of an ANN. Accordingly, the quantization system 100 may provide quantization with respect to an arbitrary ANN, and may provide a quantized ANN having the same performance as the performance of the arbitrary ANN. Due to the quantized ANN having high accuracy, computing resources for

implementing the ANN may be decreased, and as a result, the application network of the ANN may be significantly expanded.

[29] The neural network system 200 may be any system that provides (or drives) an ANN, and may be referred to as a neural network apparatus. For example, the neural network system 200 may be a computing system that includes at least one processor and a memory. As a non-limiting example, the neural network system 200 may be a mobile computing system, such as a laptop computer, a tablet computer, or a smart phone, as well as a stationary computing system, such as a desktop computer, a work station, or a server.

[30] According to one or more exemplary embodiments, the neural network system 200 may drive a floating-point neural network and may provide information about the floating-point neural network to the quantization system 100. According to one or more exemplary embodiments, the neural network system 200 may drive an ANN according to information received from the quantization system 100 and may provide information about the driven ANN to the quantization system 100. In FIG. 1, the neural network system 200 and the quantization system 100 are separate systems. However, in one or more exemplary embodiments, the neural network system 200 and the quantization system 100 may be formed as a single system.

[31] The quantization system 100 may be any system that performs quantization according to one or more exemplary embodiments, and may be referred to as a quantization apparatus. For example, the quantization system 100 may be a computing system that includes at least one processor and a memory. The quantization system 100 may be a stationary computing system or a mobile computing system. The quantization system 100 may quantize an ANN, based on information about the ANN provided by the neural network system 200. Herein, an ANN to be quantized may be referred to as an input ANN, and an ANN created by quantizing the input ANN may be referred to as an output ANN. According

to one or more exemplary embodiments, the input ANN may include a floating-point neural network, and the output ANN may include a fixed-point neural network.

[32] As illustrated in FIG. 1, the quantization system 100 may include a neural network interface 120, a parameter quantizer 140, and a sample quantizer 160. According to one or more exemplary embodiments, each of the neural network interface 120, the parameter quantizer 140, and the sample quantizer 160 may be implemented with a logic block implemented through logic synthesis, a software block executed by a processor, or a combination thereof. According to one or more exemplary embodiments, each of the neural network interface 120, the parameter quantizer 140, and the sample quantizer 160 may be a procedure as a set of instructions that are executed by a processor, and may be stored in a memory accessible by the processor.

[33] The neural network interface 120 may provide an interface for the neural network system 200 to the parameter quantizer 140 and the sample quantizer 160. For example, the neural network interface 120 may provide the parameters of an input ANN received from the neural network system 200 to the parameter quantizer 140, and may provide quantized parameters received from the parameter quantizer 140 to the neural network system 200. In addition, the neural network interface 120 may provide a sample distribution received from the neural network system 200 to the sample quantizer 160, and may provide a quantized sample distribution received from the sample quantizer 160 or a fractional length (e.g., FL_{sam} of FIG. 5) to the neural network system 200.

[34] The parameter quantizer 140 may generate (e.g., obtain) the quantized parameters (hereinafter, referred to as output parameters or second parameters) from parameters received from the neural network system 200 via the neural network interface 120 (hereinafter, referred to as input parameters or first parameters). As described below with reference to FIG. 2, an ANN may include various parameters, and quantization of the

ANN may include quantization of these parameters. The parameter quantizer 140 may quantize at least some of the parameters of the input ANN, and accordingly, the parameters of the input ANN may be quantized prior to the sample distribution. As described below with reference to FIG. 2, because parameters and sample distributions in an ANN may have different characteristics, quantization of the parameters may be separated from quantization of the sample distributions. The parameter quantizer 140 may provide the quantized parameters to the neural network system 200 via the neural network interface 120. Examples of an operation of the parameter quantizer 140 will be described below with reference to FIG. 4.

[35] The sample quantizer 160 may generate (e.g., obtain) quantized sample distributions (hereinafter, referred to as output sample distributions) from sample distributions received from the neural network system 200 via the neural network interface 120 (hereinafter, referred to as input sample distributions). The input sample distributions may have a different form than related art distributions, such as a Gaussian distribution, a Laplacian distribution, and a Gamma distribution. As will be described below, the sample quantizer 160 according to one or more exemplary embodiments may not be limited with respect to form of the sample distributions and may provide quantization having improved accuracy with respect to a sample distribution having an arbitrary form.

[36] According to one or more exemplary embodiments, the sample quantizer 160 may receive sample distributions generated from an ANN obtained by applying the quantized parameters obtained by the parameter quantizer 140 to the input ANN. In the present specification, the ANN obtained by applying the quantized parameters obtained by the parameter quantizer 140 to the input ANN may be referred to as an intermediate ANN. In particular, while a floating-point neural network is being converted into a fixed-point neural network, an ANN obtained by applying quantized parameters to the floating-point

neural network may be referred to as a semifixed-point neural network. The sample quantizer 160 quantizes sample distributions generated by the quantized parameters which are to be applied to the quantized ANN, namely, the output ANN. Thus, quantization accuracy with respect to the sample distributions of the output ANN may be improved. In other words, quantization errors with respect to the sample distributions may be reduced. Examples of an operation of the sample quantizer 160 will be described below with reference to FIGS. 5, 6A-6C, 7-9, and 10A-10B.

[37] FIG. 2 is a diagram illustrating an example of an ANN according to an exemplary embodiment. In detail, FIG. 2 is a diagram schematically illustrating the structure of a DNN 20 as an example of an ANN.

[38] As shown in FIG. 2, the DNN 20 may include a plurality of layers, i.e., first to n-th layers L1, L2, L3, ..., and Ln, and outputs of each of the first to n-th layers L1, L2, L3, ..., and Ln may be input to a subsequent layer through at least one channel. For example, the first layer L1 may provide outputs to the second layer L2 through a plurality of channels CH11 to CH1x by processing a sample SAM, and the second layer L2 may also provide outputs to the third layer L3 through a plurality of channels CH21 to CH2y. Finally, the n-th layer Ln may output a result RES, and the result RES may include at least one value associated with the sample SAM. The numbers of channels through which the outputs of the first to n-th layers L1, L2, L3, ..., and Ln are respectively delivered may be equal to or different from one another. For example, the number of channels CH21 to CH2y of the second layer L2 and the number of channels CH31 to CH3z of the third layer L3 may be equal to or different from each other.

[39] The sample SAM may be input data that is processed by the DNN 20. For example, the sample SAM may be an image including a letter written by a person with a pen, and the DNN 20 may output the result RES containing a value representing the letter by

recognizing the letter from the image. The result RES may include a plurality of probabilities corresponding to different letters, and the most likely letter among the different letters may correspond to a highest probability. Each of the first to n-th layers L1, L2, L3, ..., and Ln of the DNN 20 may generate its own outputs by processing the sample SAM or outputs of a previous layer based on parameters (e.g., weights, biases, thresholds, etc.) generated by learning a plurality of images including letters or predefined parameters (e.g., weights, biases, thresholds, etc.). For example, as shown in FIG. 2, the second layer L2 may include a weight set W including a plurality of weights w_1 , ..., and w_p , a bias set B including a plurality of biases b_1 , ..., and b_q , and a threshold set T including a plurality of thresholds t_1 , ..., and t_r , and may further include other parameters not specifically shown in FIG. 2.

[40] According to one or more exemplary embodiments, at least some of the parameters of the DNN 20 may be quantized by the parameter quantizer 140 of FIG. 1. According to one or more exemplary embodiments, the parameter quantizer 140 may independently quantize parameters for each of the plurality of layers L1, L2, L3, ..., and Ln, or may independently quantize parameters for each of the plurality of channels (for example, CH21, ..., and CH2y). According to one or more exemplary embodiments, the parameters of at least two layers may be commonly quantized, and the parameters of at least two channels may be commonly quantized. According to one or more exemplary embodiments, the weights, the biases, and the thresholds may be quantized independently from each other, or parameters of at least two different types may be commonly quantized. One or more exemplary embodiments will now be described by mainly referring to an illustration of quantizing weights as the parameters, though it is understood that one or more other exemplary embodiments are not limited thereto.

[41] The DNN 20 may include an activation function between layers. The activation function may convert an output of a previous layer to an input of a next layer. For

example, as shown in FIG. 2, an activation function f_2 may be a non-linear function, for example, a rectified linear unit (ReLU), a parametric rectified linear unit (PReLU), a hyperbolic tangent (tanh), or a sigmoid function, and may nonlinearly convert an output of the second layer L2 between the second layer L2 and the third layer L3.

[42] According to one or more exemplary embodiments, the sample quantizer 160 of FIG. 1 may receive not only an input of an activation function, namely, a pre-activation value, but also an output of the activation function, as sample distributions, and perform quantization. In addition, according to one or more exemplary embodiments, the neural network interface 120 of FIG. 1 may obtain an input distribution in units of layers and/or channels of the ANN (e.g., the DNN 20). For example, the sample distributions received by the sample quantizer 160 of FIG. 1 may be obtained from outputs produced by one layer through a plurality of channels, or may be obtained from an output corresponding to one channel.

[43] FIG. 3 is a flowchart of a method of quantizing an ANN, according to an exemplary embodiment. For example, the method of FIG. 3 may be performed by the quantization system 100 of FIG. 1. Hereinafter, FIG. 3 will be described with reference to FIG. 1.

[44] Referring to FIG. 3, in operation S200, parameters of the ANN may be quantized. For example, the parameter quantizer 140 of FIG. 1 may generate quantized parameters corresponding to given fractional lengths, and a final fractional length (for example, FL_{par} of FIG. 4) for parameters may be determined based on a quantization error of the quantized parameters. An example of operation S200 will be described below with reference to FIG. 4.

[45] In operation S400, sample distributions may be obtained from an ANN to which the quantized parameters have been applied. For example, the parameter quantizer

140 of FIG. 1 may provide the quantized parameters to the neural network system 200 via the neural network interface 120, and the neural network system 200 may drive an intermediate ANN obtained by applying the quantized parameters to an input ANN. The neural network interface 120 may receive a sample distribution of the intermediate ANN from the neural network system 200, and may provide the received sample distribution as an input sample distribution to the sample quantizer 160.

[46] In operation S600, the sample distributions may be quantized. For example, the sample quantizer 160 of FIG. 1 may split the input sample distribution having an arbitrary shape into at least two sample groups, and may independently approximate each of the sample groups to a continuous probability distribution (CPD). The sample quantizer 160 may calculate (e.g., obtain) an optimal step size, based on the CPD, and may determine (e.g., obtain) a fractional length (for example, FL_{sam} of FIG. 5) for quantizing the input sample distribution from the calculated step size. Examples of operation S600 will be described below with reference to FIGS. 5, 6A-6C, 7-9, and 10A-10B.

[47] FIG. 4 is a flowchart of an example of operation S200 of FIG. 3, according to an exemplary embodiment. As described above with reference to FIG. 3, in operation S200' of FIG. 4, the parameters of the ANN may be quantized. As shown in FIG. 4, operation S200' may include a plurality of operations S210 through S260. According to one or more exemplary embodiments, operation S200' of FIG. 4 may be performed by the parameter quantizer 140 of FIG. 1. FIG. 4 will now be described with reference to FIG. 1.

[48] In operation S210, initialization may be performed by setting a variable i to be 1. The variable i may be an index that indicates each of a plurality of fractional lengths, and evaluation values respectively corresponding to the plurality of fractional lengths may be calculated in subsequent operations. According to one or more exemplary embodiments, the plurality of fractional lengths may be determined (e.g., obtained) by a system on which

a quantized ANN is to be mounted (for example, 400 of FIG. 12). For example, a plurality of available fractional lengths may be determined according to a bit width of the system. In the example of FIG. 4, when k is a positive integer, k fractional lengths $FL_1, \dots, \text{and } FL_k$ may be used, and the variable i may sequentially increase from 1 to k .

[49] In operation S220, the parameters may be quantized according to an i -th fractional length FL_i , and a quantization error may be calculated. When a fractional length is given, the parameters may be quantized, and the quantization error may be calculated from not-yet-quantized parameters (i.e., input parameters or first parameters) and quantized parameters (i.e., output parameters or second parameters). For example, when a weight quantized from a weight w_j according to the i -th fractional length FL_i is $Q(w_j, FL_i)$, a quantization error E_{ij} may be calculated using [Equation 1].

[50] **【Equation 1】**

$$E_{ij} = |w_j - Q(w_j, FL_i)|$$

[51] In operation S230, an evaluation value corresponding to the i -th fractional length FL_i may be calculated. The evaluation value may indicate a value that represents the performance of quantization based on the quantization error of [Equation 1] or a quantization error defined differently from the quantization error of [Equation 1]. As a non-limiting example, the evaluation value may be a mean squared error (MSE), a signal to quantization noise ratio (SQNR), or the like. According to one or more exemplary embodiments, an evaluation value $D_w(FL_i)$ for the weights included in the weight set W in the given i -th fractional length FL_i may be calculated using [Equation 2].

[52] **【Equation 2】**

$$D_w(FL_i) = \sum_{w_j \in W} (E_{ij})^2$$

[53] In operation S240, it may be determined whether the variable i is identical to k . In other words, it may be checked whether evaluation values have been calculated for all of the k fractional lengths $FL_1, \dots$, and FL_k . When the variable i is not identical to k , in operation S250, the variable i may be increased by 1, and then operation S220 may be performed. On the other hand, when the variable i is identical to k , operation S260 may be performed.

[54] In operation S260, a fractional length FL_{par} for the parameters may be determined (e.g., obtained) based on the evaluation values. In other words, in operations S220 through S240, k evaluation values may be calculated, and, in operation S260, one of the k fractional lengths $FL_1, \dots$, and FL_k may be determined to be the fractional length FL_{par} for the parameters. For example, when an evaluation value is calculated using [Equation 2], a fractional length corresponding to a lowest evaluation value from among the k evaluation values may be selected. Unlike the sample distributions, the parameters of the input ANN may be deterministic values and, as described above, a fractional length that minimizes a quantization error may be determined. The method of FIG. 4 is merely an example of operation S200 of FIG. 3, and various quantization methods similar to or different from the method of FIG. 4 within the technical scope of the inventive concept may be employed to quantize the parameters.

[55] FIG. 5 is a flowchart of an example of operation S600 of FIG. 3, according to an exemplary embodiment. As described above with reference to FIG. 3, in operation S600' of FIG. 6, sample distributions may be quantized. In detail, FIG. 5 illustrates an example of quantizing a single sample distribution and, as shown in FIG. 5, operation S600'

may include a plurality of operations S620, S640, S660, and S680. According to one or more exemplary embodiments, operation S600' of FIG. 5 may be performed by the sample quantizer 160 of FIG. 1. FIG. 5 will now be described with reference to FIG. 1.

[56] In operation S620, a sample distribution (or an input sample distribution) may be split into at least two sample groups. According to an exemplary embodiment, statistical characteristics may be used to quantize the sample distribution. Accordingly, to derive statistical characteristics of the sample distribution, the sample distribution may be split into at least two sample groups $SG_1, SG_2, \dots$, and the sample groups $SG_1, SG_2, \dots$ may be independently approximated. An example of operation S620 will be described below with reference to FIG. 6.

[57] In operation S640, the sample groups may be approximated to CPDs, respectively. According to one or more exemplary embodiments, the sample groups $SG_1, SG_2, \dots$ generated in operation S620 may be respectively approximated to the CPDs by estimating at least one parameter of each of the CPDs. For example, each of the sample groups may be approximated to a generalized gamma distribution, a Gaussian distribution, or a Laplacian distribution. To estimate the parameters of the CPDs, a moment using method using a mean, a variance, or the like, a maximum likelihood method, a numerical calculation-based method, or the like may be used as a non-limiting example. Accordingly, at least two CPDs $CPD_1, CPD_2, \dots$ having parameters with unique values may be derived from the at least two sample groups $SG_1, SG_2, \dots$.

[58] According to one or more exemplary embodiments, the sample groups $SG_1, SG_2, \dots$ may each be approximated to a generalized gamma distribution $p(x)$ having 4 parameters α, β, λ , and μ , as in [Equation 3].

[59] **【Equation 3】**

$$p(x) = \mu|x|^\beta \cdot e^{-\lambda|x|^\alpha}, \text{ where } -\infty < x < \infty, \beta > -1, \text{ and } \alpha > 0,$$

where α and β may be exponential decay parameters. When $\alpha=1$, the 4 parameters α , β , λ , and μ may be derived from a mean m_x and a variance σ_x^2 according to the moment using method as shown in [Equation 4].

[60] **【Equation 4】**

$$\alpha = 1, \beta = m_x^2/\sigma_x^2 - 1, \lambda = m_x/\sigma_x^2, \text{ and } \mu = \frac{\lambda^{m_x^2/\sigma_x^2}}{2\Gamma(m_x^2/\sigma_x^2)}$$

[61] In operation S660, step sizes corresponding to the sample groups may be calculated. In other words, the step sizes may be calculated from the sample groups and the CPDs derived from the sample groups. For example, a quantization error may be calculated from a sample group and a CPD, and a step size having a low quantization error may be calculated. Accordingly, step sizes $\Delta_1, \Delta_2, \dots$ respectively corresponding to the sample groups may be calculated. An example of operation S660 will be described below with reference to FIG. 7.

[62] In operation S680, a fractional length for the sample distribution may be selected. For example, at least one candidate fractional length may be derived from the step sizes calculated in operation S660, and one of the at least one candidate fractional length may be selected as a fractional length FL_{sam} for the sample distribution. Examples of operation S800 will be described below with reference to FIGS. 8, 9, 10A, and 10B.

[63] FIGS. 6A-6C illustrate an example of operation S620 of FIG. 5, according to an exemplary embodiment. As described above with reference to FIG. 5, in operation S620, a sample distribution may be split into at least two sample groups. In detail, FIGS. 6A-6C

illustrate an example of an operation of splitting a sample distribution SD into two sample groups SG_1 and SG_2 .

[64] Referring to FIG. 6A, the sample distribution SD may be provided from an ANN to which quantized parameters have been applied. Referring to FIG. 6B, the sample distribution SD may be split into the two sample groups SG_1 and SG_2 , based on zero. According to one or more exemplary embodiments, the sample distribution SD may be split into a first sample group SG_1 including negative samples and zero and a second sample group SG_2 including positive samples. According to one or more exemplary embodiments, the sample distribution SD may be split into a first sample group SG_1 including negative samples and a second sample group SG_2 including zero and positive samples. According to one or more exemplary embodiments, the sample distribution SD may be split into a first sample group SG_1 including negative samples and a second sample group SG_2 including positive samples, and zero may be excluded. Referring to FIG. 6C, the first sample group SG_1 and the second sample group SG_2 may be independently approximated to a first CPD CPD_1 and a second CPD CPD_2 , respectively. According to one or more exemplary embodiments, each of the first sample group SG_1 and the second sample group SG_2 may be approximated to a generalized gamma distribution, and parameters of the first CPD CPD_1 and the second CPD CPD_2 may be estimated using [Equation 3] and [Equation 4].

[65] FIG. 7 is a flowchart of an example of operation S660 of FIG. 5, according to an exemplary embodiment. As described above with reference to FIG. 5, in operation S660' of FIG. 7, step sizes corresponding to the sample distributions may be calculated. In detail, FIG. 7 illustrates an example of calculating a step size, based on a single sample group SG_j and a CPD CPD_j to which the sample group SG_j has been approximated. As shown in FIG. 7, operation S660' may include operation S662 and operation S664.

[66] In operation S662, an overload distortion and a granular distortion may be calculated. The overload distortion may represent an error caused by an input that deviates from a restricted region, when the input is quantized in the restricted region. Spacing between quantization results within the restricted region of the input may be referred to granularity, and the granular distortion may represent an error caused by this spacing. According to one or more exemplary embodiments, when a $(-L, L)$ section is uniformly quantized, a quantization level S_i may be represented as in [Equation 5].

[67] **【Equation 5】**

$$S_i = \left(y_i - \frac{\Delta}{2}, y_i + \frac{\Delta}{2} \right), \quad 1 \leq i \leq N$$

[68] Accordingly, an overload distortion $D_N^{over}(L)$ may be defined as in [Equation 6].

[69] **【Equation 6】**

$$D_N^{over}(L) = \int_L^\infty (x - y_N)^2 p(x) dx, \quad y_i = -L + \left(i - \frac{1}{2} \right) \Delta$$

[70] A granular distortion $D_N^{gran}(L)$ may be defined as in [Equation 7].

[71] **【Equation 7】**

$$D_N^{gran}(L) = \sum_{i=1}^N \int_{S_i} (x - y_i)^2 p(x) dx, \quad y_i = -L + \left(i - \frac{1}{2} \right) \Delta$$

[72] Accordingly, the overload distortion D_j^{over} and the granular distortion D_j^{gran} corresponding to the sample group SG_j and the CPD CPD_j may be calculated.

[73] In operation S664, a step size may be calculated from the overload distortion and the granular distortion. For example, as shown in [Equation 8], an overall distortion D_j may be calculated from the overload distortion D_j^{over} and the granular distortion D_j^{gran} .

[74] **【Equation 8】**

$$D_j = D_j^{over} + D_j^{gran}$$

[75] According to one or more exemplary embodiments, when the overload distortion $D_N^{over}(L)$ and the granular distortion $D_N^{gran}(L)$ are calculated using [Equation 6] and [Equation 7], a step size Δ_N that minimizes the overall distortion D_j , which is a sum of the overload distortion $D_N^{over}(L)$ and the granular distortion $D_N^{gran}(L)$, may be calculated. For example, the step size Δ_N that minimizes the overall distortion D_j may be calculated using [Equation 9] when the CPD CPD_j is a generalized gamma distribution.

[76] **【Equation 9】**

$\Delta_N = 2L_N/N$, where

$$L_N = \left[\frac{2 \ln N}{\lambda} - \left(2 - \frac{1 + \beta}{\alpha} \right) \frac{\ln \ln N}{\lambda} - \frac{1}{\lambda} \ln \left(\frac{2^{1-(1+\beta)/\alpha} \alpha^2 \lambda^{(1+\beta)/\alpha}}{3\mu} \right) \right]^{1/\alpha} + \epsilon_N$$

$\epsilon_N \equiv$

$$\frac{1}{\lambda} \ln \left[\left(1 + \frac{2\alpha \ln N}{N} \right) \left(1 + \frac{3-3\alpha+2\beta}{2\alpha \ln N} \right) \cdot \left(1 + \frac{1}{2 \ln N} \left(\left(2 - \frac{1+\beta}{\alpha} \right) \ln \ln N + \ln \left(\frac{2^{1-(1+\beta)/\alpha} \alpha^2 \lambda^{(1+\beta)/\alpha}}{3\mu} \right) \right) \right)^{2-(1+\beta)/\alpha} \right]$$

[77] As such, the step size Δ_N that minimizes the overall distortion D_j may be derived based on a closed form. Unlike the above-description, according to one or more

exemplary embodiments, when a closed form is not induced from a CPD in order to calculate a step size, a numerical analysis may be used.

[78] As described above, the step size Δ_j corresponding to the sample group SG_j may be calculated in operation S660'. As a result, at least two step sizes respectively corresponding to at least two sample groups may be calculated. For example, when the sample distribution SD is split into the first sample group SG_1 and the second sample group SG_2 as described above with reference to FIGS. 6A-6C, the first step size Δ_1 and the second step size Δ_2 respectively corresponding to the first sample group SG_1 and the second sample group SG_2 may be calculated. FIGS. 8, 9, 10A, and 10B, illustrating examples of operation S680 of FIG. 5, performed after operation S660 of FIG. 5, will now be described supposing that the sample distribution SD is split into the first sample group SG_1 and the second sample group SG_2 and the first step size Δ_1 and the second step size Δ_2 are calculated. However, it is understood that one or more other exemplary embodiments are not limited thereto.

[79] FIG. 8 is a flowchart of an example of operation S680 of FIG. 3, according to an exemplary embodiment. As described above with reference to FIG. 5, in operation S680' of FIG. 8, the fractional length FL_{sam} for a sample distribution may be selected. As shown in FIG. 8, operation S680' may include operation S682 and operation S684.

[80] In operation S682, candidate fractional lengths may be generated. For example, a range of candidate fractional lengths may be determined (e.g., obtained) from the first step size Δ_1 and the second step size Δ_2 , and fractional lengths included in the range may be provided as the candidate fractional lengths. Accordingly, a candidate fractional length set FL_SET including at least one candidate fractional length may be generated. An example of operation S682 will be described below with reference to FIG. 9.

[81] In operation S684, one candidate fractional length may be selected (e.g., obtained or determined). For example, a candidate fractional length that provides a smallest

quantization error may be selected from the candidate fractional length set FL_SET, and accordingly, the fractional length FL_{sam} for the sample distribution may be finally derived. According to one or more exemplary embodiments, the candidate fractional length may be selected using various methods, and some examples of operation S684 will be described below with reference to FIGS. 10A and 10B.

[82] FIG. 9 is a flowchart of an example of operation S682 of FIG. 8, according to an exemplary embodiment. As described above with reference to FIG. 8, in operation S682' of FIG. 9, candidate fractional lengths may be generated. As shown in FIG. 9, operation S682' may include operation S682_2 and operation S682_4.

[83] In operation S682_2, a range of the candidate fractional lengths may be determined. According to one or more exemplary embodiments, the fractional lengths of step sizes adjacent to the first and second step sizes Δ_1 and Δ_2 may be calculated. For example, as in [Equation 10], 4 fractional lengths corresponding to the step sizes adjacent to the first step size Δ_1 and the second step size Δ_2 may be calculated.

[84] **【Equation 10】**

$$FL_{1, \text{ceil}} = -\text{ceil}(\log_2 \Delta_1)$$

$$FL_{1, \text{floor}} = -\text{floor}(\log_2 \Delta_1)$$

$$FL_{2, \text{ceil}} = -\text{ceil}(\log_2 \Delta_2)$$

$$FL_{2, \text{floor}} = -\text{floor}(\log_2 \Delta_2)$$

[85] In [Equation 10], “ceil()” may be a ceiling function, and “floor()” may be a floor function. Next, the range of the candidate fractional lengths may be determined based on a maximum fractional length and a minimum fractional length from among the calculated

fractional lengths. For example, the candidate fractional length set FL_SET may be defined as in [Equation 11].

[86] **【Equation 11】**

$$FL_SET = \left\{ x \left| \begin{array}{l} \min(FL_{1, \text{ ceil}}, FL_{1, \text{ floor}}, FL_{2, \text{ ceil}}, FL_{2, \text{ floor}}) - M_{\text{margin}} \leq x \\ \leq \max(FL_{1, \text{ ceil}}, FL_{1, \text{ floor}}, FL_{2, \text{ ceil}}, FL_{2, \text{ floor}}) + N_{\text{margin}} \end{array} \right. \right\}$$

[87] In [Equation 11], a first margin M_{margin} and a second margin N_{margin} may be used to correct an error that may be generated due to a difference between an input sample distribution and a sample distribution to which the input sample distribution has been approximated (i.e., a CPD). As in [Equation 11], a lower limit of the candidate fractional lengths may be determined as a value obtained by subtracting the first margin M_{margin} from a minimum value of the parameters of [Equation 10], and an upper limit of the candidate fractional lengths may be determined as a value obtained by adding the second margin N_{margin} from a maximum value of the parameters of [Equation 10]. According to one or more exemplary embodiments, the first margin M_{margin} and the second margin N_{margin} may be set to be maximum values within a limit allowed by calculation complexity or a performing ability (or computing resources) of the quantization system 100 of FIG. 1, and accordingly, the range of the candidate fractional lengths may be expanded.

[88] FIGS. 10A and 10B are flowcharts of examples of operation S684 of FIG. 8, according to one or more exemplary embodiments. In detail, FIG. 10A illustrates an example according to a closed form, and FIG. 10B illustrates an example according to a sample-based method. As described above with reference to FIG. 8, in operations S684a and S684b of FIGS. 10A and 10B, one candidate fractional length may be selected. Overlapping descriptions between FIGS. 10A and 10B will now be omitted.

[89] Referring to FIG. 10A, operation S684a may include operation S684_2a and operation S684_4a. In operation S684_2a, quantization errors according to candidate fractional lengths may be calculated. The quantization errors may be calculated using various methods, such as an MSE and an SQNR. A quantization error corresponding to one candidate fractional length FL from the candidate fractional length set FL_SET derived by a generalized gamma distribution as in, for example, [Equation 3] and [Equation 4] may be calculated using [Equation 12].

[90] **【Equation 12】**

$$\hat{D}_O(FL) = \rho \hat{D}_N(FL, \alpha_1, \beta_1, \lambda_1, \mu_1) + (1 - \rho) \hat{D}_N(FL, \alpha_2, \beta_2, \lambda_2, \mu_2)$$

$$\hat{D}_N(\hat{L}_N, \alpha, \beta, \lambda, \mu) = \frac{1}{12} \left(\frac{2\hat{L}_N}{N} \right)^2 + \frac{4\mu}{(\alpha\lambda)^3} \frac{e^{-\lambda L_N^\alpha}}{L_N^{3\alpha-\beta-3}}, \quad FL \in FL_SET$$

[91] In [Equation 12], ρ indicates a value that determines weights of the first sample group SG₁ and the second sample group SG₂, and may be defined as in, for example, [Equation 13].

[92] **【Equation 13】**

$$\rho = \frac{\# \text{ of negative samples}}{\# \text{ of total samples}}$$

[93] Accordingly, quantization errors corresponding to the candidate fractional lengths included in the candidate fractional length set FL_SET may be calculated.

[94] In operation S684_4a, one candidate fractional length may be selected based on the quantization errors. For example, based on the quantization errors calculated as in [Equation 12], a candidate fractional length that provides a smallest quantization error may

be selected as the fractional length FL_{sam} for the sample distribution. In other words, as in [Equation 14], the fractional length FL_{sam} for the sample distribution may be selected.

[95] **【Equation 14】**

$$FL_{sam} = \arg \min_{FL \in FL_SET} \hat{D}_O(FL)$$

[96] Referring to FIG. 10B, operation S684b may include operations S684_2b, S684_4b, and S684_6b. In operation S684_2b, test sample distributions according to candidate fractional lengths may be obtained. For example, the sample quantizer 160 of FIG. 1 may provide the candidate fractional lengths included in the candidate fractional length set FL_SET to the neural network system 200 of FIG. 1 via the neural network interface 120, and the neural network system 200 may generate sample distributions, namely, test sample distributions, by driving ANNs to which the candidate fractional lengths have been respectively applied. Accordingly, the sample quantizer 160 may receive test sample distributions SD_{test1} , SD_{test2} , ... from the neural network system 200 via the neural network interface 120.

[97] In operation S684_4b, quantization errors according to the test sample distributions may be calculated. The quantization errors may be calculated from the test sample distributions and the sample distributions by using various methods, for example, an MSE and an SQNR. Accordingly, quantization errors corresponding to the candidate fractional lengths included in the candidate fractional length set FL_SET may be generated.

[98] In operation S684_6b, one candidate fractional length may be selected based on the quantization errors. For example, in operation S684_6b, a candidate fractional length that provides a smallest quantization error may be selected as the fractional length FL_{sam} for the sample distribution.

[99] FIG. 11 is a block diagram of a computing system 300 according to an exemplary embodiment. According to one or more exemplary embodiments, the quantization system 100 of FIG. 1 may be implemented using the computing system 300 of FIG. 11. Referring to FIG. 11, the computing system 300 may include a system memory 310, a processor 330, a storage 350, input/output (I/O) devices 370, and communication connections 390. The components of the computing system 300 may be interconnected, e.g., via a bus.

[100] The system memory 310 may include a program 312. The program 312 may allow the processor 330 to quantize an ANN according to one or more exemplary embodiments. For example, the program 312 may include a plurality of instructions executable by the processor 330, and an ANN may be quantized as the plurality of instructions of the program 312 are executed by the processor 330. Non-limiting examples of the system memory 310 may include a volatile memory, such as Static Random Access Memory (SRAM) and Dynamic Random Access Memory (DRAM), and a non-volatile memory, such as a flash memory.

[101] The processor 330 may include at least one core capable of executing any instruction set (e.g., Intel Architecture-32 (IA-32), 64-bit extension IA-32, x86-64, PowerPC, Sparc, MIPS, ARM, and IA-64). The processor 330 may execute instructions stored in the system memory 310 and may quantize an ANN by executing the program 312.

[102] The storage 350 may not lose stored data even if power supplied to the computing system 300 is cut off. For example, the storage 350 may be a non-volatile memory, such as electrically erasable programmable read-only memory (EEPROM), a flash memory, phase change random access memory (PRAM), resistance random access memory (RRAM), nano-floating gate memory (NFGM), polymer random access memory (PoRAM), magnetic random access memory (MRAM), ferroelectric random access memory (FRAM), or a

storage medium., such as a magnetic tape, an optical disk, or a magnetic disk. According to one or more exemplary embodiments, the storage 350 may be detachably attached to the computing system 300.

[103] According to one or more exemplary embodiments, the storage 350 may store a program 312 for quantization of an ANN according to an exemplary embodiment, and the program 312 or at least a portion thereof from the storage 350 may be loaded into the system memory 310 before the program 312 is executed by the processor 330. According to one or more exemplary embodiments, the storage 350 may store a file written in a programming language, and a program 312 generated by a compiler or the like or at least a portion of the program 312 from the file may be loaded into the system memory 310.

[104] According to one or more exemplary embodiments, the storage 350 may store data that is to be processed by the processor 330, and/or data processed by the processor 330. For example, the storage 350 may store the parameters and the sample distributions of FIG. 2, may store the quantized parameters and the quantized sample distributions, and may store data generated during quantization of a parameter or sample distribution, for example, a step size and a fractional length.

[105] The I/O devices 370 may include an input device, such as a keyboard or a pointing device, and an output device, such as a printer or a display. For example, a user may trigger the execution of the program 312 by the processor 330 via the I/O devices 370, may input a parameter and a sample distribution, and may check a quantized parameter, a quantized sample distribution, and/or an error message.

[106] The communications connections 390 may provide access to a network outside the computing system 300. For example, the network may include multiple computing systems and communication links, and the communication links may include wired links, optical links, wireless links, or any other type of links.

[107] FIG. 12 is a block diagram of a mobile computing device 400 according to an exemplary embodiment of the inventive concept. According to one or more exemplary embodiments, a fixed-point neural network quantized according to an exemplary embodiment may be implemented in the mobile computing device 400. As a non-limiting example, the mobile computing device 400 may be any of mobile electronic devices, such as a mobile phone, a tablet PC, a wearable device, and an object Internet device, which are powered by a battery, an external power supply, or via self power generation.

[108] Referring to FIG. 12, the mobile computing device 400 may include a memory subsystem 410, I/O devices 430, a processing unit 450, and a network interface 470. The memory subsystem 410, the I/O devices 430, the processing unit 450, and the network interface 470 may communicate with each other via a bus 490. According to one or more exemplary embodiments, at least two of the memory subsystem 410, the I/O devices 430, the processing unit 450, and the network interface 470 may be included in one package as a System-on-Chip (SoC).

[109] The memory subsystem 410 may include RAM 412 and a storage 414. The RAM 412 and/or the storage 414 may store instructions that are executed by the processing unit 450, and data that are processed by the processing unit 450. For example, the RAM 412 and/or the storage 414 may store variables such as signals, weights, biases, and thresholds of an ANN and may also store parameters of an artificial neuron (or computational node) of an ANN. According to one or more exemplary embodiments, the storage 414 may include a non-volatile memory.

[110] The processing unit 450 may include at least one of a central processing unit (CPU) 452, a graphics processing unit (GPU) 454, a digital signal processor (DSP) 456, and a neural processing unit (NPU) 458. For example, according to one or more exemplary

embodiments, the processing unit 450 may include one or only some of the CPU 452, the GPU 454, the DSP 456, and the NPU 458.

[111] The CPU 452 may control the overall operation of the mobile computing device 400. For example, the CPU 452 may directly perform a specific task in response to an external input received through the I/O devices 430, or may direct other components of the processing unit 450 to perform a task. The GPU 454 may generate data for an image output through a display included in the I/O devices 430 and may encode data received from a camera included in the I/O devices 430. The DSP 456 may generate useful data by processing a digital signal, e.g., a digital signal provided from the network interface 470.

[112] The NPU 458, which is dedicated hardware for an ANN, may include a plurality of computational nodes corresponding to at least some artificial neurons that constitute the ANN, and at least some of the plurality of computational nodes may process a signal in parallel. An ANN quantized according to an exemplary embodiment, for example, a DNN, has low computational complexity as well as high accuracy, and thus may be easily implemented in the mobile computing device 400 of FIG. 12, may have a fast processing speed, and may also be implemented by, for example, a simple and small-scale NPU 458.

[113] The I/O devices 430 may include input devices, such as a touch input device, a sound input device, and a camera, and output devices, such as a display and a sound output device. For example, when a user's voice is input through the sound input device, the voice may be recognized by a DNN implemented in the mobile computing device 400, and a corresponding operation may be triggered. In addition, when an image is input through the camera, an object in the image may be recognized by the DNN implemented in the mobile computing device 400, and an output such as virtual reality may be provided to a user. The network interface 470 may provide access to a mobile communication network, such as

Long Term Evolution (LTE) or 5G, to the mobile computing device 400 and may provide access to a local network such as Wi-Fi.

[114] One or more exemplary embodiments have been particularly shown and described above with reference to the drawings. The terminology used herein is for the purpose of describing exemplary embodiments only and is not intended to be limiting of the inventive concept. Therefore, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the inventive concept as defined by the appended claims.

What is Claimed is:

1. An artificial neural network (ANN) quantization method for generating an output ANN by quantizing an input ANN, the ANN quantization method comprising:

obtaining second parameters by quantizing first parameters of the input ANN;

obtaining a sample distribution from an intermediate ANN in which the obtained second parameters have been applied to the input ANN; and

obtaining a fractional length for the obtained sample distribution by quantizing the obtained sample distribution.

2. The ANN quantization method of claim 1, wherein the obtaining the second parameters comprises:

obtaining quantized parameters by quantizing the first parameters according to a given fractional length, and calculating quantization errors between the first parameters and the obtained quantized parameters;

calculating an evaluation value of the given fractional length, based on the calculated quantization errors; and

obtaining a final fractional length for the second parameters, based on a plurality of evaluation values corresponding to a plurality of fractional lengths.

3. The ANN quantization method of claim 2, wherein:

the calculating the evaluation value comprises calculating, as the evaluation value, a sum of squares of the calculated quantization errors; and

the obtaining the final fractional length comprises determining, as the final fractional length, a fractional length corresponding to a minimum evaluation value from among the plurality of evaluation values.

4. The ANN quantization method of any one of claims 1 to 3, wherein:
the input ANN comprises layers and channels, each having at least one parameter;
and
the first parameters comprise one or more parameters from among the at least one parameter of each of the layers and the channels.

5. The ANN quantization method of claim 4, wherein the first parameters comprise at least one of weights, biases, and thresholds.

6. The ANN quantization method of any one of claims 1 to 5, wherein the obtaining the fractional length for the obtained sample distribution comprises:
splitting the obtained sample distribution into a plurality of sample groups;
approximating the plurality of sample groups to a plurality of continuous probability distributions (CPDs), respectively;
obtaining a plurality of step sizes, based on the plurality of sample groups and the plurality of CPDs; and
selecting the fractional length, based on the obtained plurality of step sizes.

7. The ANN quantization method of claim 6, wherein the splitting the obtained sample distribution into the plurality of sample groups comprises splitting the obtained sample distribution into a first sample group including negative samples and zero and a second sample group including positive samples.

8. The ANN quantization method of claim 6, wherein:

the splitting of the sample distribution into the plurality of sample groups comprises splitting the sample distribution into a first sample group and a second sample group; and the first sample group includes negative samples, and the second sample group includes zero and positive samples.

9. The ANN quantization method of claim 6, wherein:

the splitting the sample distribution into the plurality of sample groups comprises splitting the sample distribution into a first sample group and a second sample group; and the first sample group includes negative samples, and the second sample group includes positive samples.

10. The ANN quantization method of any one of claims 6 to 9, wherein the approximating the plurality of sample groups comprises approximating each of the plurality of sample groups to a generalized gamma distribution, a Gaussian distribution, or a Laplacian distribution.

11. The ANN quantization method of any one of claims 6 to 10, wherein the obtaining the plurality of step sizes comprises:

obtaining an overload distortion and a granular distortion according to a step size for each of the plurality of CPDs; and

obtaining each of the plurality of step sizes, based on the obtained overload distortion and the obtained granular distortion.

12. The ANN quantization method of any one of claims 6 to 11, wherein the selecting the fractional length comprises:

obtaining candidate fractional lengths, based on the obtained plurality of step sizes;
and

selecting, as the fractional length, one from among the obtained candidate fractional lengths.

13. The ANN quantization method of claim 12, wherein the obtaining the candidate fractional lengths comprises:

obtaining fractional lengths corresponding to step sizes adjacent to the obtained plurality of step sizes; and

determining a range of fractional lengths, based on the obtained fractional lengths.

14. The ANN quantization method of claim 13, wherein:

the determining the range of the fractional lengths comprises:

determining, as a lower limit of the range, a value obtained by subtracting a first margin from a minimum fractional length among the obtained fractional lengths, and

determining, as an upper limit of the range, a value obtained by adding a second margin from a maximum fractional length among the obtained fractional lengths;
and

the first margin and the second margin are determined based on a performing ability of the ANN quantization method.

15. The ANN quantization method of claim 13 or claim 14, wherein the selecting the one from among the obtained candidate fractional lengths comprises:

calculating errors corresponding to the obtained candidate fractional lengths, based on the obtained candidate fractional lengths and the plurality of CPDs; and

selecting the one candidate fractional length, based on the calculated errors.

16. The ANN quantization method of claim 13 or claim 14, wherein the selecting the one from among the obtained candidate fractional lengths comprises:

obtaining test sample distributions from test ANNs respectively depending on the obtained candidate fractional lengths;

calculating errors corresponding to the obtained test sample distributions, based on the obtained test sample distributions and the plurality of CPDs; and

selecting the one candidate fractional length, based on the calculated errors.

17. The ANN quantization method of any one of claims 1 to 16, further comprising obtaining a fixed-point ANN as the output ANN, based on the obtained second parameters and the obtained fractional length.

18. An apparatus for generating an output artificial neural network (ANN) by quantizing an input ANN, the apparatus comprising:

a memory storing computer-executable instructions; and

at least one processor configured to execute the stored computer-executable instructions to implement:

a parameter quantizer configured to obtain second parameters by quantizing first parameters of the input ANN;

a neural network interface configured to obtain a sample distribution from an intermediate ANN in which the obtained second parameters have been applied to the input ANN; and

a sample quantizer configured to obtain a fractional length for the obtained sample distribution.

19. The apparatus of claim 18, wherein:

the neural network interface is further configured to provide at least one candidate fractional length for the sample distribution, received from the sample quantizer, to a test ANN, and to obtain a test sample distribution from the test ANN; and

the sample quantizer is further configured to determine one from among the at least one candidate fractional length as the fractional length for the sample distribution, based on the obtained test sample distribution.

20. A method of quantizing a floating-point neural network, the method comprising:

obtaining quantized parameters by quantizing parameters in a same category in the floating-point neural network;

obtaining a sample distribution from a semifixed-point artificial neural network (ANN) in which the obtained quantized parameters have been applied to the floating-point neural network; and

obtaining a fractional length for the obtained sample distribution.