(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 September 2020 (24.09.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/190696 A1

(51) International Patent Classification:
H04N 19/42 (2014.01) *G06N 3/08* (2006.01)

(21) International Application Number:
PCT/US2020/022585

(22) International Filing Date:
13 March 2020 (13.03.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/818,914 15 March 2019 (15.03.2019) US

(71) Applicant: **INTERDIGITAL VC HOLDINGS, INC.** [US/US]; 200 Bellevue Parkway, Suite 300, Wilmington, Delaware 19809 (US).

(72) Inventors: **RACAPE, Fabien**; Bld 4, Ste 200, 3000 El Camino Real, Palo Alto, California 94306 (US). **JAIN, Swayambhoo**; Bld 4, Ste 200, 3000 El Camino Real, Palo Alto, California 94306 (US). **HAMIDI-RAD, Shahab**; Bld 4, Ste 200, 3000 El Camino Real, Palo Alto, California 94306 (US). **PAPADIMITRIOU, Dimitris**; 920 Keeler Ave., Berkley, California 94708 (US).

(74) Agent: **KOLCZYNSKI, Ronald**; 200 Bellevue Parkway, Suite 300, Wilmington, Delaware 19809 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ,

(54) Title: LOW DISPLACEMENT RANK BASED DEEP NEURAL NETWORK COMPRESSION

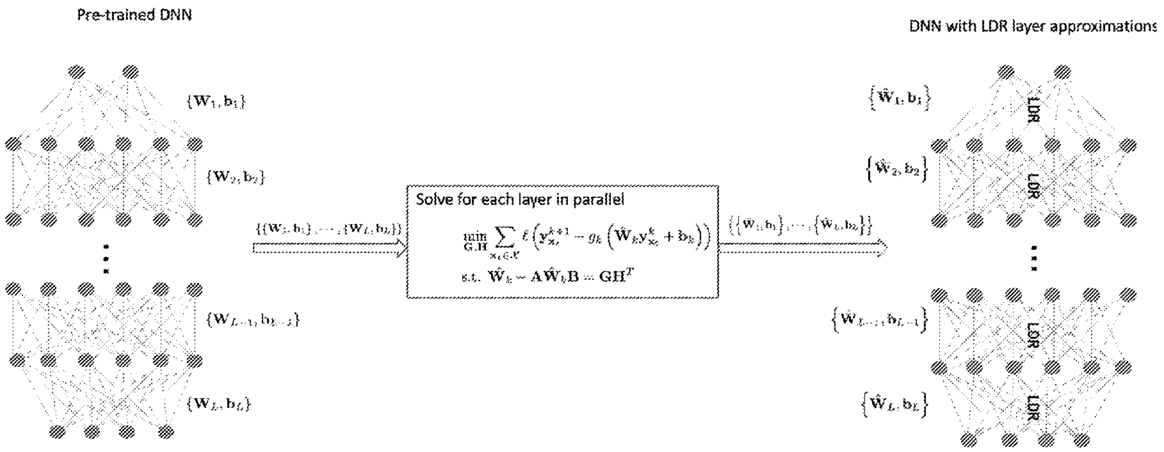
Pre-trained DNN
$\{W_1, b_1\}$
$\{W_2, b_2\}$
$\{W_{L-1}, b_{L-1}\}$
$\{W_L, b_L\}$
$\{\{W_1, b_1\}, \cdots, \{W_L, b_L\}\}$
Solve for each layer in parallel
$\min_{G,H} \sum_{x_i \in \mathcal{X}} \ell\left(y_{x_i}^{k+1} - g_k\left(\hat{W}_k y_{x_i}^k + b_k\right)\right)$
s.t. $\hat{W}_k - A\hat{W}_k B = GH^T$
$\{\{\hat{W}_1, b_1\}, \cdots, \{\hat{W}_L, b_L\}\}$
DNN with LDR layer approximations
$\{\hat{W}_1, b_1\}$
$\{\hat{W}_2, b_2\}$
$\{\hat{W}_{L-1}, b_{L-1}\}$
$\{\hat{W}_L, b_L\}$
LDR
LDR
LDR
LDR

Figure 2

(57) **Abstract:** A method and an apparatus for performing deep neural network compression use an approximation training set along with information, such as in matrices representing weights, biases and non-linearities, to iteratively compress a p re-trained deep neural network by low displacement rank based approximation of the network layer weight matrices. The low displacement rank approximation allows for replacement of an original layer weight matrices of the pre-trained deep neural network as the sum of a small number of structured matrices, allowing compression and low inference complexity.

WO 2020/190696 A1

[Continued on next page]

UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

LOW DISPLACEMENT RANK BASED DEEP NEURAL NETWORK COMPRESSION

TECHNICAL FIELD

At least one of the present embodiments generally relates to a method or an apparatus for video encoding or decoding.

BACKGROUND

To achieve high compression efficiency, image and video coding schemes usually employ prediction, including spatial and/or motion vector prediction, and transforms to leverage spatial and temporal redundancy in the video content. Generally, intra or inter prediction is used to exploit the intra or inter frame correlation, then the differences between the original image and the predicted image, often denoted as prediction errors or prediction residuals, are transformed, quantized, and entropy coded. To reconstruct the video, the compressed data are decoded by inverse processes corresponding to the entropy coding, quantization, transform, and prediction. Mapping and inverse mapping processes can be used in an encoder and decoder to achieve improved coding performance. Indeed, for better coding efficiency, signal mapping may be used. Mapping aims at better exploiting the samples codewords values distribution of the video pictures.

SUMMARY

Drawbacks and disadvantages of the prior art may be addressed by the general aspects described herein, which are directed intra prediction mode partitioning in encoding and decoding.

According to a first aspect, there is provided a method. The method comprises steps for obtaining information representative of weights, biases and non-linearities of a deep neural network, and an approximation training set for the deep neural network; generating output matrices of information for a layer of the deep neural network using said information and the approximation training set; and, updating said output matrices for additional layers by iteratively using output matrices of previous layers of the deep neural network and said approximation training set until a minimization criterion has been met for that layer, and until a last layer is reached to generate coefficients representative of said output matrices; and, performing quantization and entropy coding on said coefficients.

According to a second aspect, there is provided a method. The method comprises steps for performing entropy decoding on data from a bitstream; performing inverse quantization on said entropy decoded data; and, iteratively obtaining dequantized matrices representative of a deep neural network and building weight matrices until a last layer is reached.

According to another aspect, there is provided an apparatus. The apparatus comprises a processor. The processor can be configured to compress and decompress a deep neural network by executing any of the aforementioned methods.

According to another general aspect of at least one embodiment, there is provided a device comprising an apparatus according to any of the decoding embodiments; and at least one of (i) an antenna configured to receive a signal, the signal including the video block, (ii) a band limiter configured to limit the received signal to a band of frequencies that includes the video block, or (iii) a display configured to display an output representative of a video block.

According to another general aspect of at least one embodiment, there is provided a non-transitory computer readable medium containing data content generated according to any of the described encoding embodiments or variants.

According to another general aspect of at least one embodiment, there is provided a signal comprising video data generated according to any of the described encoding embodiments or variants.

According to another general aspect of at least one embodiment, a bitstream is formatted to include data content generated according to any of the described encoding embodiments or variants.

According to another general aspect of at least one embodiment, there is provided a computer program product comprising instructions which, when the program is executed by a computer, cause the computer to carry out any of the described decoding embodiments or variants.

These and other aspects, features and advantages of the general aspects will become apparent from the following detailed description of exemplary embodiments, which is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a pipeline for low displacement rank based neural network compression under the general aspects described.

Figure 2 shows low displacement rank approximation under the general aspects described.

Figure 3 shows a flow diagram for determining low displacement rank approximation layers under the general aspects described.

Figure 4 shows an example of a flow diagram for a training/update loop for low displacement rank approximation estimation under the general aspects described.

Figure 5 shows an example of a decoding process under the general aspects described.

Figure 6 shows one embodiment of an encoding method under the general aspects described.

Figure 7 shows one embodiment of a decoding method under the general aspects described.

Figure 8 shows one embodiment of an apparatus for encoding or decoding using intra prediction mode extensions.

Figure 9 shows a generic, standard encoding scheme.

Figure 10 shows a generic, standard decoding scheme.

Figure 11 shows a typical processor arrangement in which the described embodiments may be implemented.

DETAILED DESCRIPTION

Deep Neural Networks (DNNs) have shown state of the art performance in variety of domains such as computer vision, speech recognition, natural language processing, etc. This performance however comes at the cost of massive computational cost as DNNs tend to have a huge number of parameters often running into millions, and sometimes even billions. This leads to prohibitively high inference complexity -- the computational cost of applying trained DNN to test data for inference. This high inference complexity is the main challenge in bringing the performance of DNNs to mobile or embedded devices with resource limitations on battery size, computational power, and memory capacity etc.

This invention applies to compression of a pre-trained DNN so that it facilitates transmission and/or storage as well as low inference complexity. We propose compression of pre-trained DNN by Low Displacement Rank (LDR) based approximation of the network layer weight matrices. The LDR approximation allows for replacing the original layer weight matrices of the pre-trained DNN as the sum of small number of structured matrices. This decomposition into sum of structured matrices leads to simultaneous compression, and low inference complexity thereby enabling the power of Deep Learning in the resource limited devices.

Most of approaches for compression of DNNs are either based on sparsity-based assumption or low rank-based approximation. While these approaches lead to compression, they still suffer from the high inference complexity. The sparsity structure is difficult to implement in hardware as the performance depends critically on the pattern of sparsity, and the existing approaches do not have any control over the sparsity pattern. The low-rank matrices are still unstructured. Due to these reasons, these approaches do not necessarily lead to improvement in the inference complexity. The LDR based approximation proposed in this invention approximates the given layer weight matrix as a sum small number of structured matrices which allows for simultaneous compression and low inference complexity.

Suppose we are provided with a L-Layer pre-trained DNN with weight matrices $\{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_L\}$, biases $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L\}$, and non-linearities $\{g_1, \ldots, g_L\}$. With these weights, biases, and non-linearities, the output of kth layer $\boldsymbol{y}^{k+1}$ is written as follows (where $\boldsymbol{y^1} = \boldsymbol{x}$ is the input to the DNN):

$$\boldsymbol{y}^{k+1} = g_k(\boldsymbol{W}_k \, \boldsymbol{y}^k + \boldsymbol{b}_k)$$

We propose to approximate the layers of pre-trained DNN $\{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_L\}$ with LDR matrices $\{ \widehat{\boldsymbol{W}}_1, \ldots, \ \widehat{\boldsymbol{W}}_L\}$. Suppose the k^{th} layer weights matrix $\widehat{\boldsymbol{W}}_k$ has low rank $r_k \ll min\,\{m, n\}$, then it implies that

$$L_{A,B}(\widehat{\boldsymbol{W}}_k) = \boldsymbol{W}_k - \boldsymbol{A}\boldsymbol{W}_k\boldsymbol{B} = \boldsymbol{G}_k\, \boldsymbol{H}_k^T,$$

where $\boldsymbol{A}, \boldsymbol{B}$ are square matrices of size $m \times m$, $n \times n$ respectively, $\boldsymbol{G_k}$ is a $m \times r_k$ matrix, $\boldsymbol{H}_k$ is $n \times r_k$ matrix, and m, n are the number of rows and columns of the original weight matrix $\boldsymbol{W}_k$. The columns of G and H can represent some sort of filters, for example. And depending on A and B they can interact in different ways end-to-end.

Here, the displacement r_k is a parameter of choice. A small r_k implies more compression and computational efficiency. Alternatively, the LDR can also be sought in an equivalent but an alternative expression as

$$L_{A,B}(\widehat{\boldsymbol{W}}_k) = \boldsymbol{A}\boldsymbol{W}_k - \boldsymbol{W}_k\boldsymbol{B} = \boldsymbol{G}_k\, \boldsymbol{H}_k^T.$$

By different choices of $\boldsymbol{A}, \boldsymbol{B}$ the LDR structure is general enough so that it covers a whole host of other structures such as Toeplitz, circulant, Hankel, etc. For the proposes of describing the invention we will use the first expression for displacement rank. In order to obtain such an approximation the proposed approach requires an approximation training set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$ which can be chosen as a subset of original training set used to train the given DNN or it may be chosen as set examples on the DNN it is supposed to operate on. Using the approximation training set $\mathcal{X}$, we can obtain the output and input of each layer of the original pre-trained DNN. The input and output of the k^{th} layer for a given example x_t in the approximating set $\mathcal{X}$ are denoted as $\boldsymbol{y}_{x_t}^k$ and $\boldsymbol{y}_{x_t}^{k+1}$. With these, we solve the following optimization problem to obtain $\boldsymbol{G}_k, \boldsymbol{H}_k$:

$$\min_{\{\boldsymbol{G}_k, \boldsymbol{H}_k\}} \sum\nolimits_{x_t \in \mathcal{X}} \boldsymbol{l}(\boldsymbol{y}_{x_t}^{k+1} - \ \boldsymbol{g}_k(\widehat{\boldsymbol{W}}_k\, \boldsymbol{y}_{x_t}^k + \boldsymbol{b}_k) \ ) \ \ \boldsymbol{s.t.}\ \ \widehat{\boldsymbol{W}}_k - \boldsymbol{A}\widehat{\boldsymbol{W}}_k\boldsymbol{B} = \boldsymbol{G}_k\, \boldsymbol{H}_k^T \qquad (1)$$

The above problem can be approximately solved by using stochastic gradient descent algorithm. The term in the summation is being minimized. The equality constraint in above problem can be handled using the inversion formulae from “Inversion of Displacement Operators” by Pan and Wang. An exemplary over-all architecture for compressing a neural network is shown in Figure 1. It shows the DNN training stage that involves training a DNN on the given training data. Next block takes the pre-trained DNN represented by weight matrices $\{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_L\}$, biases $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L\}$, and non-linearities $\{g_1, \ldots, g_L\}$ as the input, and the approximation training set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$ and could work with an approximate training set that is independent from the training set. The first sub-step in LDR based compression block is the LDR based approximation, which is the object of this invention.

After this, the weight matrices G_k and H_k of each layer's LDR based approximation are quantized and this followed by lossless coefficient compression for each layer.

This results in a bitstream that may be stored or transmitted. The resulting bitstream along with meta data involves matrices $\boldsymbol{A}$, $\boldsymbol{B}$, the bias vectors $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L\}$, and description of non-linearities are sent. The compressed bitstream is decompressed using the metadata, and for inference the DNN is loaded into memory for inference on test data for the application at hand.

Encoder

Figure 2 shows the LDR base approximation subblock in the Figure 1 in detail. The LDR approximation for each layer can be obtained in parallel as shown in Figure 2 using the approximation training set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$.

The LDR approximation of pre-trained DNN at the encoder is depicted in Figure 3. Using the approximation training set $\mathcal{X}$, we can obtain the output and input of each layer of the original pre-trained DNN. The input and output of the k^{th} layer for a given example x_t in the approximating set $\mathcal{X}$ are denoted as $\boldsymbol{y}_{\boldsymbol{x}_t}^k$ and $\boldsymbol{y}_{\boldsymbol{x}_t}^{k+1}$. Each layer can be accessed at step 101, potentially in parallel, looping from step 104 depending on computer resources, until last layer is processed. $\boldsymbol{G}_k$ and $\boldsymbol{H}_k$ are computed per layer at step 102 and 103.

The step 103 is further described in Figure 4. For each layer, the approximation set can be split in batches and several iterations, or epochs, can be performed over the set (204). For each iteration, the current batch of input/output data for the layer is accessed (201), the minimization problem of figure (1) over this dataset (202), and the matrices $\boldsymbol{G}_k$ and $\boldsymbol{H}_k$ are updated (203).

The termination criterion (204) can be based on number of training steps in terms number of epochs or if matrices $\boldsymbol{G}_k$ and $\boldsymbol{H}_k$ in consecutive train steps are numerically close to each other with a chosen parameter of closeness, the matrices $\boldsymbol{G}_k$ and $\boldsymbol{H}_k$ are output for all k.

This parallel version allows for efficient computing at the cost of considering non-updated inputs to subsequent layers after the modification of the $\boldsymbol{G}_k$ and $\boldsymbol{H}_k$ matrices. The parallel version processes inputs and outputs of each layer in the original networks which are obtained once in the beginning and they are used to parallelly update the layers. The serial version is mentioned as a variant later in this description. As a variant, the output of each layer *k* can be updated using training data so that the LDR computation of layer *k+1* can be performed using the proper input.

Decoder

To decode the produced bitstream, a compatible decoder needs to perform the inverse compression steps. Figure 5 details the different steps. The symbols of the input bitstream are first extracted from the entropy decoding engine 301, then inverse quantized (302). For each layer (305), the dequantized matrices are accessed (303) and each matrix $\boldsymbol{W}_{\boldsymbol{k}}^{dec}$is derived from the obtained $\boldsymbol{G}_k$ and $\boldsymbol{H}_k$ using the inversion formulae previously mentioned.

Hence, to decode and obtain the reconstructed DNN, decoders that would implement a standard compression scheme, such as the future MPEG NNR standard, will require to include the proposed method.

In one variant, using one of the frameworks mentioned above, it is possible to use different $\boldsymbol{A}$ and $\boldsymbol{B}$ for each layer. This will change the meta data that is needed to be transmitted from the encoder. The decoder while decoding the layers will use specific $\boldsymbol{A}$ and $\boldsymbol{B}$ for that layer.

In a second variant, the LDR approximation can also be done in a serial fashion in which the next layer are approximated by taken into account the approximation in the previous layers.

One embodiment of a method 600 using the general aspects described here is shown in Figure 6. The method commences at Start block 601 and control proceeds to function block 610 for obtaining information representative of weights, biases and non-linearities of a deep neural network, and an approximation training set for the deep neural network. Control then proceeds from block 610 to block 620 for generating output matrices of information for a layer of the deep neural network using the information and the approximation training set. Control then proceeds from block 620 to block 630 for updating output matrices for additional layers by iteratively using output matrices of previous layers of the deep neural network and the approximation training set until a minimization criterion has been met for that layer, and until a last layer is reached to generate coefficients representative of said output matrices. Control then proceeds from block 630 to block 640 for performing quantization and entropy coding on the coefficients.

One embodiment of a method 700 using the general aspects described here is shown in Figure 7. The method commences at Start block 701 and control proceeds to function block 710 for performing entropy decoding on data from a bitstream. Control then proceeds from block 710 to block 720 for performing inverse quantization on the entropy decoded data. Control then proceeds from block 720 to block 730 for iteratively obtaining dequantized matrices representative of a deep neural network and building weight matrices until a last layer is reached.

Figure 8 shows one embodiment of an apparatus 800 for compressing, encoding or decoding a deep neural network in a bitstream. The apparatus comprises Processor 810 and can be interconnected to a memory 820 through at least one port. Both Processor 810 and memory 820 can also have one or more additional interconnections to external connections.

Processor 810 is also configured to either insert or receive parameters in a bitstream and, either compressing, encoding or decoding a deep neural network using the parameters.

This application describes a variety of aspects, including tools, features, embodiments, models, approaches, etc. Many of these aspects are described with specificity and, at least to show the individual characteristics, are often described in a manner that may sound limiting. However, this is for purposes of clarity in description, and does not limit the application or scope of those aspects. Indeed, all of the different aspects can be combined and interchanged to provide further aspects. Moreover, the aspects can be combined and interchanged with aspects described in earlier filings as well.

The aspects described and contemplated in this application can be implemented in many different forms. Figures 9, 10, and 11 provide some embodiments, but other embodiments are contemplated and the discussion of Figures 9, 10, and 11 does not limit the breadth of the implementations. At least one of the aspects generally relates to video encoding and decoding, and at least one other aspect generally relates to transmitting a bitstream generated or encoded. These and other aspects can be implemented as a method, an apparatus, a computer readable storage medium having stored thereon instructions for encoding or decoding video data according to any of the methods described, and/or a computer readable storage medium having stored thereon a bitstream generated according to any of the methods described.

In the present application, the terms "reconstructed" and "decoded" may be used interchangeably, the terms "pixel" and "sample" may be used interchangeably, the terms "image," "picture" and "frame" may be used interchangeably. Usually, but not necessarily, the term "reconstructed" is used at the encoder side while "decoded" is used at the decoder side.

Various methods are described herein, and each of the methods comprises one or more steps or actions for achieving the described method. Unless a specific order of steps or actions is required for proper operation of the method, the order and/or use of specific steps and/or actions may be modified or combined.

Various methods and other aspects described in this application can be used to modify modules, for example, the intra prediction, entropy coding, and/or decoding modules (160, 360, 145, 330), of a video encoder 100 and decoder 200 as shown in FIG. 9 and FIG. 10. Moreover, the present aspects are not limited to VVC or HEVC, and can be applied, for example, to other standards and recommendations, whether pre-existing or future-developed, and extensions of any such standards and recommendations (including VVC and HEVC). Unless indicated otherwise, or technically precluded, the aspects described in this application can be used individually or in combination.

Various numeric values are used in the present application. The specific values are for example purposes and the aspects described are not limited to these specific values.

FIG. 9 illustrates an encoder 100. Variations of this encoder 100 are contemplated, but the encoder 100 is described below for purposes of clarity without describing all expected variations.

Before being encoded, the video sequence may go through pre-encoding processing (101), for example, applying a color transform to the input color picture (e.g., conversion from RGB 4:4:4 to YCbCr 4:2:0), or performing a remapping of the input picture components in order to get a signal distribution more resilient to compression (for instance using a histogram equalization of one of the color components). Metadata can be associated with the pre-processing and attached to the bitstream.

In the encoder 100, a picture is encoded by the encoder elements as described below. The picture to be encoded is partitioned (102) and processed in units of, for example, CUs. Each unit is encoded using, for example, either an intra or inter mode. When a unit is encoded in an intra mode, it performs intra prediction (160). In an inter mode, motion estimation (175) and compensation (170) are performed. The encoder decides (105) which one of the intra mode or inter mode to use for encoding the unit, and indicates the intra/inter decision by, for example, a prediction mode flag. Prediction residuals are calculated, for example, by subtracting (110) the predicted block from the original image block.

The prediction residuals are then transformed (125) and quantized (130). The quantized transform coefficients, as well as motion vectors and other syntax elements, are entropy coded (145) to output a bitstream. The encoder can skip the transform and apply quantization directly to the non-transformed residual signal. The encoder can bypass both transform and quantization, i.e., the residual is coded directly without the application of the transform or quantization processes.

The encoder decodes an encoded block to provide a reference for further predictions. The quantized transform coefficients are de-quantized (140) and inverse transformed (150) to decode prediction residuals. Combining (155) the decoded prediction residuals and the predicted block, an image block is reconstructed. In-loop filters (165) are applied to the reconstructed picture to perform, for example, deblocking/SAO (Sample Adaptive Offset) filtering to reduce encoding artifacts. The filtered image is stored at a reference picture buffer (180).

FIG. 10 illustrates a block diagram of a video decoder 200. In the decoder 200, a bitstream is decoded by the decoder elements as described below. Video decoder 200 generally performs a decoding pass reciprocal to the encoding pass as described in FIG.

9. The encoder 100 also generally performs video decoding as part of encoding video data.

In particular, the input of the decoder includes a video bitstream, which can be generated by video encoder 100. The bitstream is first entropy decoded (230) to obtain transform coefficients, motion vectors, and other coded information. The picture partition information indicates how the picture is partitioned. The decoder may therefore divide (235) the picture according to the decoded picture partitioning information. The transform coefficients are de-quantized (240) and inverse transformed (250) to decode the prediction residuals. Combining (255) the decoded prediction residuals and the predicted block, an image block is reconstructed. The predicted block can be obtained (270) from intra prediction (260) or motion-compensated prediction (i.e., inter prediction) (275). In-loop filters (265) are applied to the reconstructed image. The filtered image is stored at a reference picture buffer (280).

The decoded picture can further go through post-decoding processing (285), for example, an inverse color transform (e.g. conversion from YCbCr 4:2:0 to RGB 4:4:4) or an inverse remapping performing the inverse of the remapping process performed in the pre-encoding processing (101). The post-decoding processing can use metadata derived in the pre-encoding processing and signaled in the bitstream.

FIG. 11 illustrates a block diagram of an example of a system in which various aspects and embodiments are implemented. System 1000 can be embodied as a device including the various components described below and is configured to perform one or more of the aspects described in this document. Examples of such devices include, but are not limited to, various electronic devices such as personal computers, laptop computers, smartphones, tablet computers, digital multimedia set top boxes, digital television receivers, personal video recording systems, connected home appliances, and servers. Elements of system 1000, singly or in combination, can be embodied in a single integrated circuit (IC), multiple ICs, and/or discrete components. For example, in at least one embodiment, the processing and encoder/decoder elements of system 1000 are distributed across multiple ICs and/or discrete components. In various embodiments, the system 1000 is communicatively coupled to one or more other systems, or other electronic devices, via, for example, a communications bus or through dedicated input and/or output ports. In various embodiments, the system 1000 is configured to implement one or more of the aspects described in this document.

The system 1000 includes at least one processor 1010 configured to execute instructions loaded therein for implementing, for example, the various aspects described in this document. Processor 1010 can include embedded memory, input output interface, and various other circuitries as known in the art. The system 1000 includes at least one memory 1020 (e.g., a volatile memory device, and/or a non-volatile memory device). System 1000 includes a storage device 1040, which can include non-volatile memory and/or volatile memory, including, but not limited to, Electrically Erasable Programmable Read-Only Memory (EEPROM), Read-Only Memory (ROM), Programmable Read-Only Memory (PROM), Random Access Memory (RAM), Dynamic Random Access Memory (DRAM), Static Random Access Memory (SRAM), flash, magnetic disk drive, and/or optical disk drive. The storage device 1040 can include an internal storage device, an attached storage device (including detachable and non-detachable storage devices), and/or a network accessible storage device, as non-limiting examples.

System 1000 includes an encoder/decoder module 1030 configured, for example, to process data to provide an encoded video or decoded video, and the encoder/decoder module 1030 can include its own processor and memory. The encoder/decoder module 1030 represents module(s) that can be included in a device to perform the encoding and/or decoding functions. As is known, a device can include one or both of the encoding and decoding modules. Additionally, encoder/decoder module 1030 can be implemented as a separate element of system 1000 or can be incorporated within processor 1010 as a combination of hardware and software as known to those skilled in the art.

Program code to be loaded onto processor 1010 or encoder/decoder 1030 to perform the various aspects described in this document can be stored in storage device 1040 and subsequently loaded onto memory 1020 for execution by processor 1010. In accordance with various embodiments, one or more of processor 1010, memory 1020, storage device 1040, and encoder/decoder module 1030 can store one or more of various items during the performance of the processes described in this document. Such stored items can include, but are not limited to, the input video, the decoded video or portions of the decoded video, the bitstream, matrices, variables, and intermediate or final results from the processing of equations, formulas, operations, and operational logic.

In some embodiments, memory inside of the processor 1010 and/or the encoder/decoder module 1030 is used to store instructions and to provide working memory for processing that is needed during encoding or decoding. In other embodiments, however, a memory external to the processing device (for example, the processing device can be either the processor 1010 or the encoder/decoder module 1030) is used for one or more of these functions. The external memory can be the memory 1020 and/or the storage device 1040, for example, a dynamic volatile memory and/or a non-volatile flash memory. In several embodiments, an external non-volatile flash memory is used to store the operating system of, for example, a television. In at least one embodiment, a fast external dynamic volatile memory such as a RAM is used as working memory for video coding and decoding operations, such as for MPEG-2 (MPEG refers to the Moving Picture Experts Group, MPEG-2 is also referred to as ISO/IEC 13818, and 13818-1 is also known as H.222, and 13818-2 is also known as H.262), HEVC (HEVC refers to High Efficiency Video Coding, also known as H.265 and MPEG-H Part 2), or VVC (Versatile Video Coding, a new standard being developed by JVET, the Joint Video Experts Team).

The input to the elements of system 1000 can be provided through various input devices as indicated in block 1130. Such input devices include, but are not limited to, (i) a radio frequency (RF) portion that receives an RF signal transmitted, for example, over the air by a broadcaster, (ii) a Component (COMP) input terminal (or a set of COMP input terminals), (iii) a Universal Serial Bus (USB) input terminal, and/or (iv) a High *Definition* Multimedia Interface (HDMI) input terminal. Other examples, not shown in FIG. 11, include composite video.

In various embodiments, the input devices of block 1130 have associated respective input processing elements as known in the art. For example, the RF portion can be associated with elements suitable for (i) selecting a desired frequency (also referred to as selecting a signal, or band-limiting a signal to a band of frequencies), (ii) downconverting the selected signal, (iii) band-limiting again to a narrower band of frequencies to select (for example) a signal frequency band which can be referred to as a channel in certain embodiments, (iv) demodulating the downconverted and band-limited signal, (v) performing error correction, and (vi) demultiplexing to select the desired stream of data packets. The RF portion of various embodiments includes one or more elements to perform these functions, for example, frequency selectors, signal selectors, band-limiters, channel selectors, filters, downconverters, demodulators, error correctors, and demultiplexers. The RF portion can include a tuner that performs various of these functions, including, for example, downconverting the received signal to a lower frequency (for example, an intermediate frequency or a near-baseband frequency) or to baseband. In one set-top box embodiment, the RF portion and its associated input processing element receives an RF signal transmitted over a wired (for example, cable) medium, and performs frequency selection by filtering, downconverting, and filtering again to a desired frequency band. Various embodiments rearrange the order of the above-described (and other) elements, remove some of these elements, and/or add other elements performing similar or different functions. Adding elements can include inserting elements in between existing elements, such as, for example, inserting amplifiers and an analog-to-digital converter. In various embodiments, the RF portion includes an antenna.

Additionally, the USB and/or HDMI terminals can include respective interface processors for connecting system 1000 to other electronic devices across USB and/or HDMI connections. It is to be understood that various aspects of input processing, for example, Reed-Solomon error correction, can be implemented, for example, within a separate input processing IC or within processor 1010 as necessary. Similarly, aspects of USB or HDMI interface processing can be implemented within separate interface ICs or within processor 1010 as necessary. The demodulated, error corrected, and demultiplexed stream is provided to various processing elements, including, for example, processor 1010, and encoder/decoder 1030 operating in combination with the memory and storage elements to process the datastream as necessary for presentation on an output device.

Various elements of system 1000 can be provided within an integrated housing, Within the integrated housing, the various elements can be interconnected and transmit data therebetween using suitable connection arrangement, for example, an internal bus as known in the art, including the Inter-IC (I2C) bus, wiring, and printed circuit boards.

The system 1000 includes communication interface 1050 that enables communication with other devices via communication channel 1060. The communication interface 1050 can include, but is not limited to, a transceiver configured to transmit and to receive data over communication channel 1060. The communication interface 1050 can include, but is not limited to, a modem or network card and the communication channel 1060 can be implemented, for example, within a wired and/or a wireless medium.

Data is streamed, or otherwise provided, to the system 1000, in various embodiments, using a wireless network such as a Wi-Fi network, for example IEEE 802.11 (IEEE refers to the Institute of Electrical and Electronics Engineers). The Wi-Fi signal of these embodiments is received over the communications channel 1060 and the communications interface 1050 which are adapted for Wi-Fi communications. The communications channel 1060 of these embodiments is typically connected to an access point or router that provides access to external networks including the Internet for allowing streaming applications and other over-the-top communications. Other embodiments provide streamed data to the system 1000 using a set-top box that delivers the data over the HDMI connection of the input block 1130. Still other embodiments provide streamed data to the system 1000 using the RF connection of the input block 1130. As indicated above, various embodiments provide data in a non-streaming manner. Additionally, various embodiments use wireless networks other than Wi-Fi, for example a cellular network or a Bluetooth network.

The system 1000 can provide an output signal to various output devices, including a display 1100, speakers 1110, and other peripheral devices 1120. The display 1100 of various embodiments includes one or more of, for example, a touchscreen display, an organic light-emitting diode (OLED) display, a curved display, and/or a foldable display. The display 1100 can be for a television, a tablet, a laptop, a cell phone (mobile phone), or other device. The display 1100 can also be integrated with other components (for example, as in a smart phone), or separate (for example, an external monitor for a laptop). The other peripheral devices 1120 include, in various examples of embodiments, one or more of a stand-alone digital video disc (or digital versatile disc) (DVR, for both terms), a disk player, a stereo system, and/or a lighting system. Various embodiments use one or more peripheral devices 1120 that provide a function based on the output of the system 1000. For example, a disk player performs the function of playing the output of the system

1000.

In various embodiments, control signals are communicated between the system 1000 and the display 1100, speakers 1110, or other peripheral devices 1120 using signaling such as AV.Link, Consumer Electronics Control (CEC), or other communications protocols that enable device-to-device control with or without user intervention. The output devices can be communicatively coupled to system 1000 via dedicated connections through respective interfaces 1070, 1080, and 1090. Alternatively, the output devices can be connected to system 1000 using the communications channel 1060 via the communications interface 1050. The display 1100 and speakers 1110 can be integrated in a single unit with the other components of system 1000 in an electronic device such as, for example, a television. In various embodiments, the display interface 1070 includes a display driver, such as, for example, a timing controller (T Con) chip.

The display 1100 and speaker 1110 can alternatively be separate from one or more of the other components, for example, if the RF portion of input 1130 is part of a separate set-top box. In various embodiments in which the display 1100 and speakers 1110 are external components, the output signal can be provided via dedicated output connections, including, for example, HDMI ports, USB ports, or COMP outputs.

The embodiments can be carried out by computer software implemented by the processor 1010 or by hardware, or by a combination of hardware and software. As a non-limiting example, the embodiments can be implemented by one or more integrated circuits. The memory 1020 can be of any type appropriate to the technical environment and can be implemented using any appropriate data storage technology, such as optical memory devices, magnetic memory devices, semiconductor-based memory devices, fixed memory, and removable memory, as non-limiting examples. The processor 1010 can be of any type appropriate to the technical environment, and can encompass one or more of microprocessors, general purpose computers, special purpose computers, and processors based on a multi-core architecture, as non-limiting examples.

Various implementations involve decoding. “Decoding”, as used in this application, can encompass all or part of the processes performed, for example, on a received encoded sequence to produce a final output suitable for display. In various embodiments, such processes include one or more of the processes typically performed by a decoder, for example, entropy decoding, inverse quantization, inverse transformation, and differential decoding. In various embodiments, such processes also, or alternatively, include processes performed by a decoder of various implementations described in this application.

As further examples, in one embodiment "decoding" refers only to entropy decoding, in another embodiment "decoding" refers only to differential decoding, and in another embodiment "decoding" refers to a combination of entropy decoding and differential decoding. Whether the phrase "decoding process" is intended to refer specifically to a subset of operations or generally to the broader decoding process will be clear based on the context of the specific descriptions and is believed to be well understood by those skilled in the art.

Various implementations involve encoding. In an analogous way to the above discussion about "decoding", "encoding" as used in this application can encompass all or part of the processes performed, for example, on an input video sequence to produce an encoded bitstream. In various embodiments, such processes include one or more of the processes typically performed by an encoder, for example, partitioning, differential encoding, transformation, quantization, and entropy encoding. In various embodiments, such processes also, or alternatively, include processes performed by an encoder of various implementations described in this application.

As further examples, in one embodiment "encoding" refers only to entropy encoding, in another embodiment "encoding" refers only to differential encoding, and in another embodiment "encoding" refers to a combination of differential encoding and entropy encoding. Whether the phrase "encoding process" is intended to refer specifically to a subset of operations or generally to the broader encoding process will be clear based on the context of the specific descriptions and is believed to be well understood by those skilled in the art.

Note that the syntax elements as used herein are descriptive terms. As such, they do not preclude the use of other syntax element names.

When a figure is presented as a flow diagram, it should be understood that it also provides a block diagram of a corresponding apparatus. Similarly, when a figure is presented as a block diagram, it should be understood that it also provides a flow diagram of a corresponding method/process.

Various embodiments may refer to parametric models or rate distotion optimization. In particular, during the encoding process, the balance or trade-off between the rate and distortion is usually considered, often given the constraints of computational complexity. It can be measured through a Rate Distortion Optimization (RDO) metric, or through Least Mean Square (LMS), Mean of Absolute Errors (MAE), or other such measurements. Rate distortion optimization is usually formulated as minimizing a rate distortion function, which is a weighted sum of the rate and of the distortion. There are different approaches to solve the rate distortion optimization problem. For example, the approaches may be based on an extensive testing of all encoding options, including all considered modes or coding parameters values, with a complete evaluation of their coding cost and related distortion of the reconstructed signal after coding and decoding. Faster approaches may also be used, to save encoding complexity, in particular with computation of an approximated distortion based on the prediction or the prediction residual signal, not the reconstructed one. Mix of these two approaches can also be used, such as by using an approximated distortion for only some of the possible encoding options, and a complete distortion for other encoding options. Other approaches only evaluate a subset of the possible encoding options. More generally, many approaches employ any of a variety of techniques to perform the optimization, but the optimization is not necessarily a complete evaluation of both the coding cost and related distortion.

The implementations and aspects described herein can be implemented in, for example, a method or a process, an apparatus, a software program, a data stream, or a signal. Even if only discussed in the context of a single form of implementation (for example, discussed only as a method), the implementation of features discussed can also be implemented in other forms (for example, an apparatus or program). An apparatus can be implemented in, for example, appropriate hardware, software, and firmware. The methods can be implemented in, for example, , a processor, which refers to processing devices in general, including, for example, a computer, a microprocessor, an integrated circuit, or a programmable logic device. Processors also include communication devices, such as, for example, computers, cell phones, portable/personal digital assistants ("PDAs"), and other devices that facilitate communication of information between endusers.

Reference to “one embodiment” or “an embodiment” or “one implementation” or “an implementation”, as well as other variations thereof, means that a particular feature, structure, characteristic, and so forth described in connection with the embodiment is included in at least one embodiment. Thus, the appearances of the phrase “in one embodiment” or “in an embodiment” or “in one implementation” or “in an implementation”, as well any other variations, appearing in various places throughout this application are not necessarily all referring to the same embodiment.

Additionally, this application may refer to “determining” various pieces of information. Determining the information can include one or more of, for example, estimating the information, calculating the information, predicting the information, or retrieving the information from memory.

Further, this application may refer to “accessing” various pieces of information. Accessing the information can include one or more of, for example, receiving the information, retrieving the information (for example, from memory), storing the information, moving the information, copying the information, calculating the information, determining the information, predicting the information, or estimating the information.

Additionally, this application may refer to “receiving” various pieces of information. Receiving is, as with “accessing”, intended to be a broad term. Receiving the information can include one or more of, for example, accessing the information, or retrieving the information (for example, from memory). Further, “receiving” is typically involved, in one way or another, during operations such as, for example, storing the information, processing the information, transmitting the information, moving the information, copying the information, erasing the information, calculating the information, determining the information, predicting the information, or estimating the information.

It is to be appreciated that the use of any of the following “/”, “and/or”, and “at least one of”, for example, in the cases of “A/B”, “A and/or B” and “at least one of A and B”, is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further example, in the cases of “A, B, and/or C” and “at least one of A, B, and C”, such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C) only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended, as is clear to one of ordinary skill in this and related arts, for as many items as are listed.

Also, as used herein, the word "signal" refers to, among other things, indicating something to a corresponding decoder. For example, in certain embodiments the encoder signals a particular one of a plurality of transforms, coding modes or flags. In this way, in an embodiment the same transform, parameter, or mode is used at both the encoder side and the decoder side. Thus, for example, an encoder can transmit (explicit signaling) a particular parameter to the decoder so that the decoder can use the same particular parameter. Conversely, if the decoder already has the particular parameter as well as others, then signaling can be used without transmitting (implicit signaling) to simply allow the decoder to know and select the particular parameter. By avoiding transmission of any actual functions, a bit savings is realized in various embodiments. It is to be appreciated that signaling can be accomplished in a variety of ways. For example, one or more syntax elements, flags, and so forth are used to signal information to a corresponding decoder in various embodiments. While the preceding relates to the verb form of the word "signal", the word "signal" can also be used herein as a noun.

As will be evident to one of ordinary skill in the art, implementations can produce a variety of signals formatted to carry information that can be, for example, stored or transmitted. The information can include, for example, instructions for performing a method, or data produced by one of the described implementations. For example, a signal can be formatted to carry the bitstream of a described embodiment. Such a signal can be formatted, for example, as an electromagnetic wave (for example, using a radio frequency portion of spectrum) or as a baseband signal. The formatting can include, for example, encoding a data stream and modulating a carrier with the encoded data stream. The information that the signal carries can be, for example, analog or digital information. The signal can be transmitted over a variety of different wired or wireless links, as is known. The signal can be stored on a processor-readable medium.

We describe a number of embodiments, across various claim categories and types. Features of these embodiments can be provided alone or in any combination. Further, embodiments can include one or more of the following features, devices, or aspects, alone or in any combination, across various claim categories and types:

- A process or device to perform encoding and decoding with deep neural network compression of a pre-trained deep neural network.
- A process or device to perform encoding and decoding with inserted information in a bitstream representative of parameters to implement deep neural network compression of a pre-trained deep neural network comprising one or more layers.
- A process or device to perform encoding and decoding with inserted information in a bitstream representative of parameters to implement deep neural network compression of a pre-trained deep neural network until a compression criterion is reached.
- A bitstream or signal that includes one or more of the described syntax elements, or variations thereof.
- A bitstream or signal that includes syntax conveying information generated according to any of the embodiments described.
- Creating and/or transmitting and/or receiving and/or decoding according to any of the embodiments described.
- A method, process, apparatus, medium storing instructions, medium storing data, or signal according to any of the embodiments described.
- Inserting in the signaling syntax elements that enable the decoder to determine coding mode in a manner corresponding to that used by an encoder.
- Creating and/or transmitting and/or receiving and/or decoding a bitstream or signal that includes one or more of the described syntax elements, or variations thereof.
- A TV, set-top box, cell phone, tablet, or other electronic device that performs transform method(s) according to any of the embodiments described.
- A TV, set-top box, cell phone, tablet, or other electronic device that performs transform method(s) determination according to any of the embodiments

described, and that displays (e.g. using a monitor, screen, or other type of display) a resulting image.

- A TV, set-top box, cell phone, tablet, or other electronic device that selects, bandlimits, or tunes (e.g. using a tuner) a channel to receive a signal including an encoded image, and performs transform method(s) according to any of the embodiments described.
- A TV, set-top box, cell phone, tablet, or other electronic device that receives (e.g. using an antenna) a signal over the air that includes an encoded image, and performs transform method(s).

CLAIMS:

1. A method, comprising:

obtaining information representative of weights, biases and non-linearities of a deep neural network, and an approximation training set for the deep neural network;

generating output matrices of information for a layer of the deep neural network using said information and the approximation training set; and,

updating said output matrices for additional layers by iteratively using output matrices of previous layers of the deep neural network and said approximation training set until a minimization criterion has been met for that layer, and until a last layer is reached to generate coefficients representative of said output matrices; and,

performing quantization and entropy coding on said coefficients.

2. An apparatus, comprising:

a processor, configured to perform:

obtaining matrices representative of weight, biases and non-linearities of a deep neural network;

generating an output for a layer of the deep neural network using said matrices, an approximation set from training data, and an input to the layer; and;

updating said output matrices for additional layers by iteratively using output matrices of previous layers of the deep neural network and said approximation training set until a minimization criterion has been met for that layer, and until a last layer is reached to generate coefficients representative of said output matrices; and,

performing quantization and entropy coding on said coefficients.

3. A method, comprising:

performing entropy decoding on data from a bitstream;

performing inverse quantization on said entropy decoded data; and, iteratively obtaining dequantized matrices representative of a deep neural network and building weight matrices until a last layer is reached.

4. An apparatus, comprising:

a processor, configured to perform:

performing entropy decoding on data from a bitstream;

performing inverse quantization on said entropy decoded data; and,

iteratively obtaining dequantized matrices representative of a deep neural network and building weight matrices until a last layer is reached.

5. The method of claim 1, or the apparatus of claim 2, wherein said approximation comprises a low displacement rank based approximation.

6. The method of claim 1, or the apparatus of claim 2, wherein said approximation training set comprises a subset of an original training set of the deep neural network.

7. The method of claim 1, or the apparatus of claim 2, wherein said approximation training set comprises set examples on the deep neural network.

8. The method of claim 1, or the apparatus of claim 2, wherein said minimization using a stochastic gradient descent scheme.

9. The method of claim 1, or the apparatus of claim 2, wherein said minimization using an inversion scheme.

10. The method of claim 1, or the apparatus of claim 2, wherein said updating is performed using training data.

11. The method of claim 1, or the apparatus of claim 2, wherein said updating is performed using different metadata for each layer

12. A device comprising:

an apparatus according to any of claims 4 through 11; and

at least one of (i) an antenna configured to receive a signal, the signal including the video block, (ii) a band limiter configured to limit the received signal to a band of frequencies that includes the video block, and (iii) a display configured to display an output representative of a video block.

13. A non-transitory computer readable medium containing data content generated according to the method of any one of claims 1 and 5 to 11, or by the apparatus of any one of claims 2 and 5 to 11, for playback using a processor.

14. A signal comprising video data generated according to the method of any one of claims 1 and 5 to 11, or by the apparatus of any one of claims 2 and 5 to 11, for playback using a processor.

15. A computer program product comprising instructions which, when the program is executed by a computer, cause the computer to carry out the method of any one of claim 1, 3 and 5 to 11.

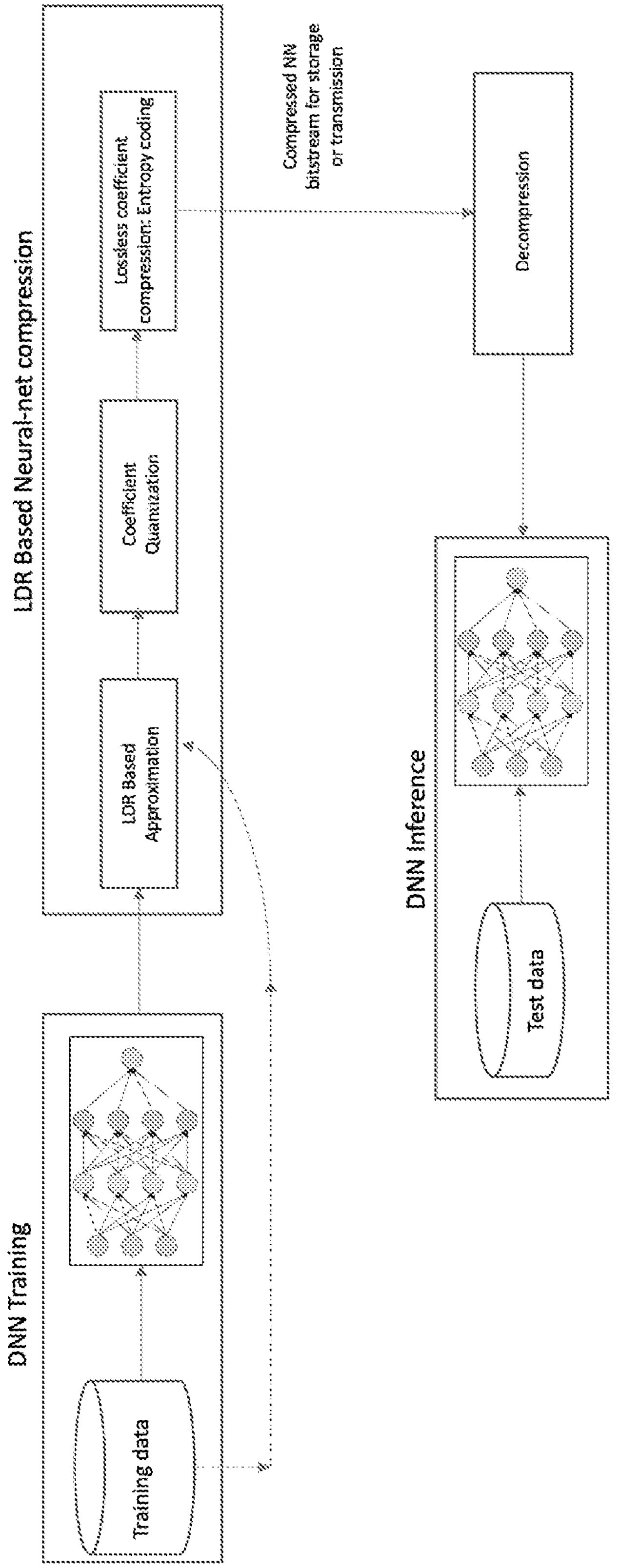
DNN Training
Training data
LDR Based Neural-net compression
LDR Based Approximation
Coefficient Quantization
Lossless coefficient compression: Entropy coding
Compressed NN bitstream for storage or transmission
Decompression
DNN Inference
Test data

Figure 1

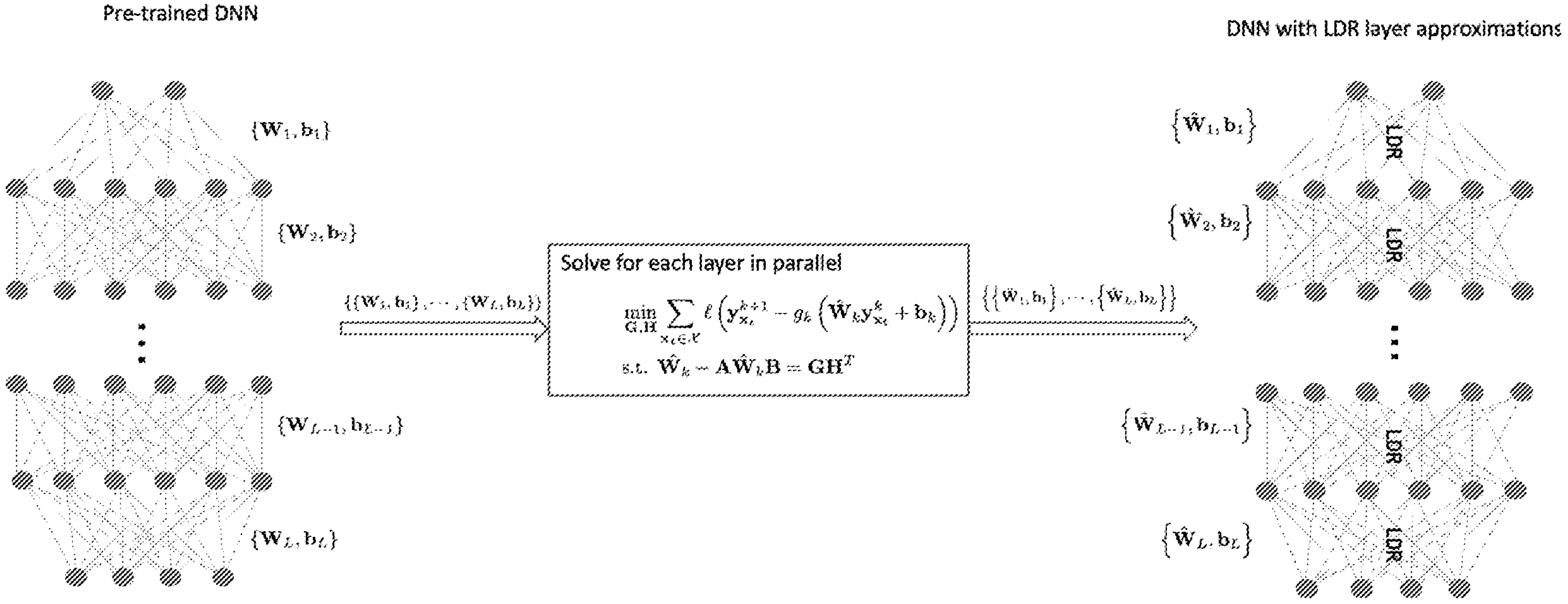
Pre-trained DNN
$\{\mathbf{W}_1, \mathbf{b}_1\}$
$\{\mathbf{W}_2, \mathbf{b}_2\}$
$\{\mathbf{W}_{L-1}, \mathbf{b}_{L-1}\}$
$\{\mathbf{W}_L, \mathbf{b}_L\}$
$\{\{\mathbf{W}_1, \mathbf{b}_1\}, \cdots, \{\mathbf{W}_L, \mathbf{b}_L\}\}$
Solve for each layer in parallel
$\min_{\mathbf{G},\mathbf{H}} \sum_{\mathbf{x}_i \in \mathcal{X}} \ell\left(\mathbf{y}_{\mathbf{x}_i}^{k+1} - g_k\left(\hat{\mathbf{W}}_k \mathbf{y}_{\mathbf{x}_i}^k + \mathbf{b}_k\right)\right)$
s.t. $\hat{\mathbf{W}}_k - \mathbf{A}\hat{\mathbf{W}}_k\mathbf{B} = \mathbf{G}\mathbf{H}^T$
$\{\{\hat{\mathbf{W}}_1, \mathbf{b}_1\}, \cdots, \{\hat{\mathbf{W}}_L, \mathbf{b}_L\}\}$
DNN with LDR layer approximations
$\{\hat{\mathbf{W}}_1, \mathbf{b}_1\}$
$\{\hat{\mathbf{W}}_2, \mathbf{b}_2\}$
$\{\hat{\mathbf{W}}_{L-1}, \mathbf{b}_{L-1}\}$
$\{\hat{\mathbf{W}}_L, \mathbf{b}_L\}$
LDR
LDR
LDR
LDR

Figure 2

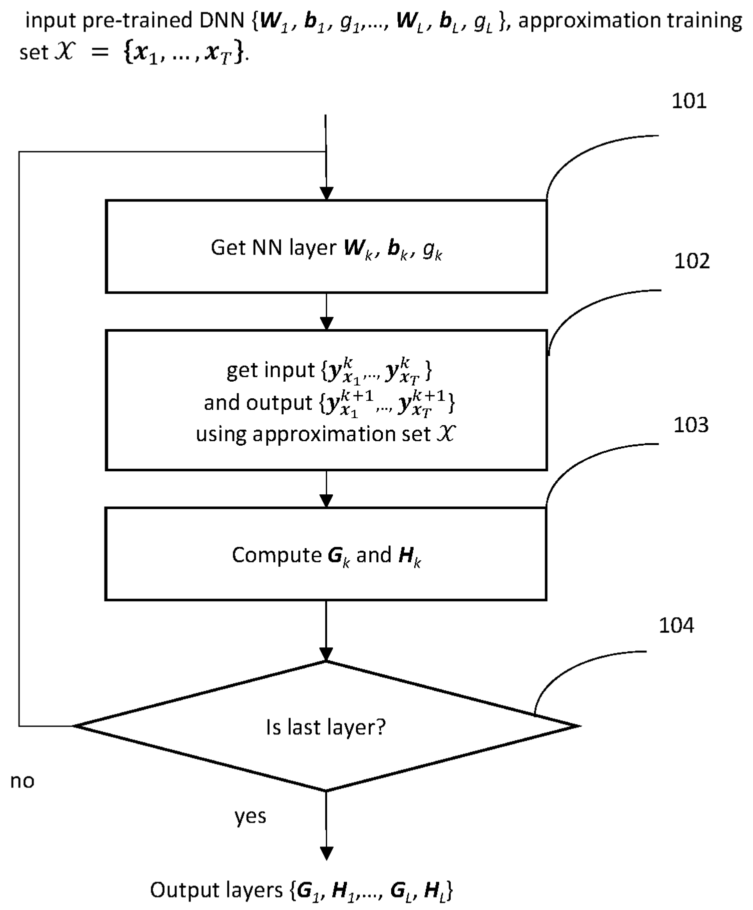
input pre-trained DNN $\{\boldsymbol{W}_1, \boldsymbol{b}_1, g_1,\ldots, \boldsymbol{W}_L, \boldsymbol{b}_L, g_L\}$, approximation training set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$.
101
Get NN layer $\boldsymbol{W}_k$, $\boldsymbol{b}_k$, g_k
102
get input $\{\boldsymbol{y}_{x_1}^{k},\ldots, \boldsymbol{y}_{x_T}^{k}\}$ and output $\{\boldsymbol{y}_{x_1}^{k+1},\ldots, \boldsymbol{y}_{x_T}^{k+1}\}$ using approximation set $\mathcal{X}$
103
Compute $\boldsymbol{G}_k$ and $\boldsymbol{H}_k$
104
Is last layer?
no
yes
Output layers $\{\boldsymbol{G}_1, \boldsymbol{H}_1,\ldots, \boldsymbol{G}_L, \boldsymbol{H}_L\}$

Figure 3

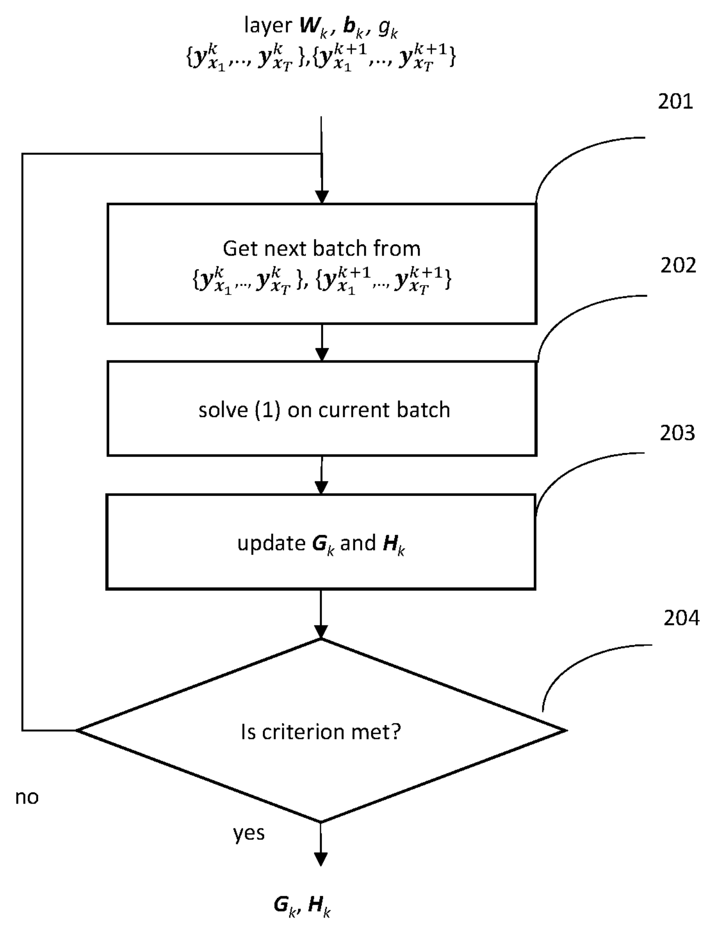
layer $\boldsymbol{W}_k$, $\boldsymbol{b}_k$, g_k
$\{\boldsymbol{y}^k_{x_1},\dots,\boldsymbol{y}^k_{x_T}\},\{\boldsymbol{y}^{k+1}_{x_1},\dots,\boldsymbol{y}^{k+1}_{x_T}\}$
201
Get next batch from $\{\boldsymbol{y}^k_{x_1},\dots,\boldsymbol{y}^k_{x_T}\}$, $\{\boldsymbol{y}^{k+1}_{x_1},\dots,\boldsymbol{y}^{k+1}_{x_T}\}$
202
solve (1) on current batch
203
update $\boldsymbol{G}_k$ and $\boldsymbol{H}_k$
204
Is criterion met?
no
yes
$\boldsymbol{G}_k$, $\boldsymbol{H}_k$

Figure 4

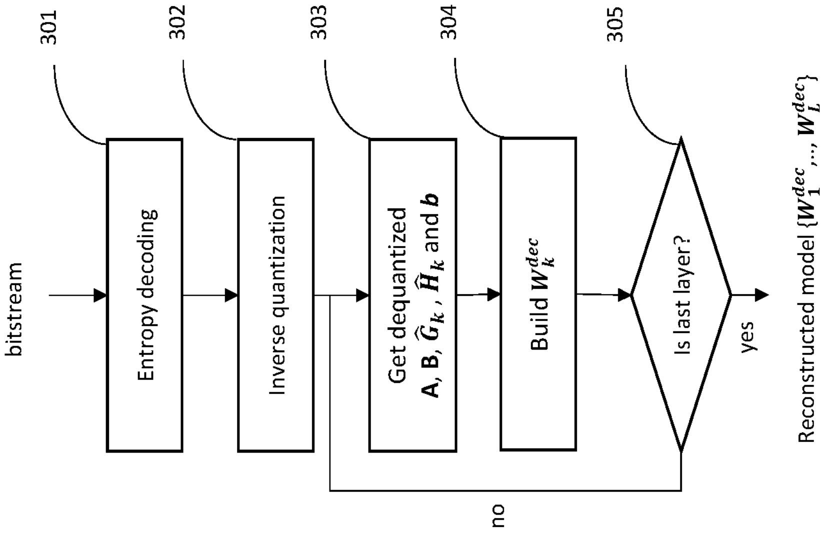
301
302
303
304
305
bitstream
Entropy decoding
Inverse quantization
Get dequantized
A, **B**, $\widehat{G}_k$, $\widehat{H}_k$ and $\boldsymbol{b}$
Build W_k^{dec}
Is last layer?
yes
no
Reconstructed model $\{W_1^{dec},.., W_L^{dec}\}$

Figure 5

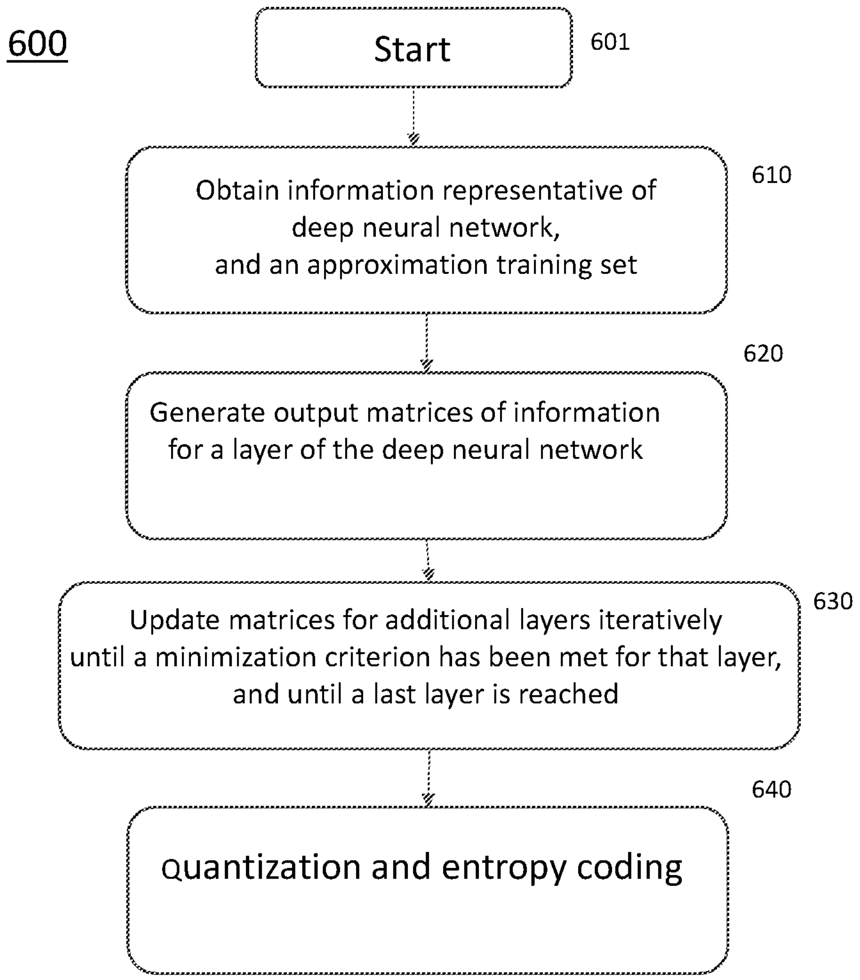
600
Start
601
Obtain information representative of
deep neural network,
and an approximation training set
610
Generate output matrices of information
for a layer of the deep neural network
620
Update matrices for additional layers iteratively
until a minimization criterion has been met for that layer,
and until a last layer is reached
630
Quantization and entropy coding
640

Figure 6

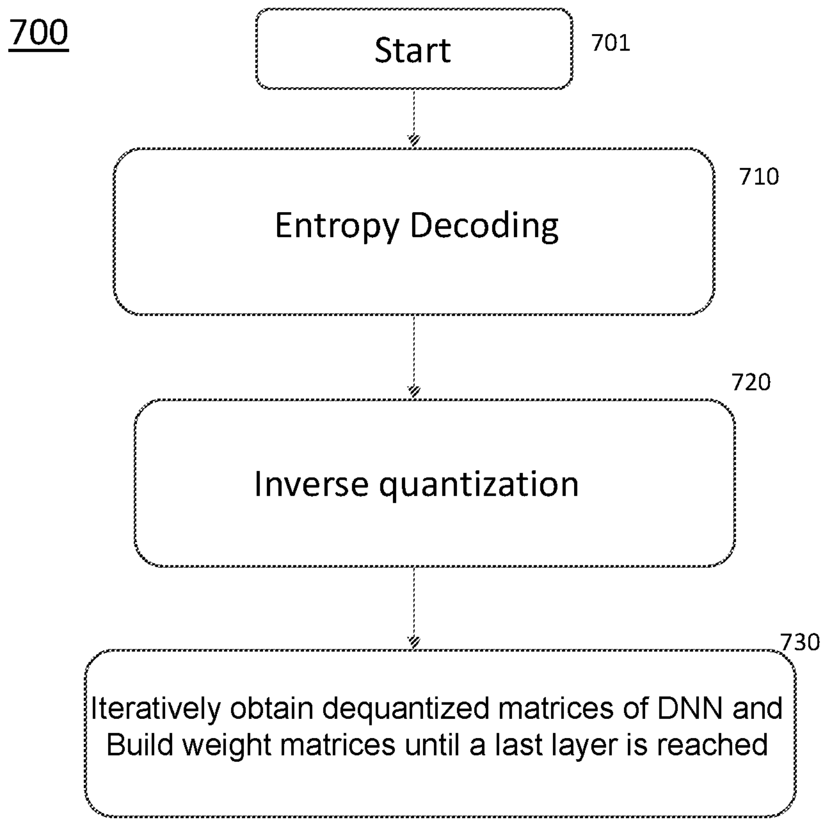
700
Start
701
Entropy Decoding
710
Inverse quantization
720
Iteratively obtain dequantized matrices of DNN and Build weight matrices until a last layer is reached
730

Figure 7

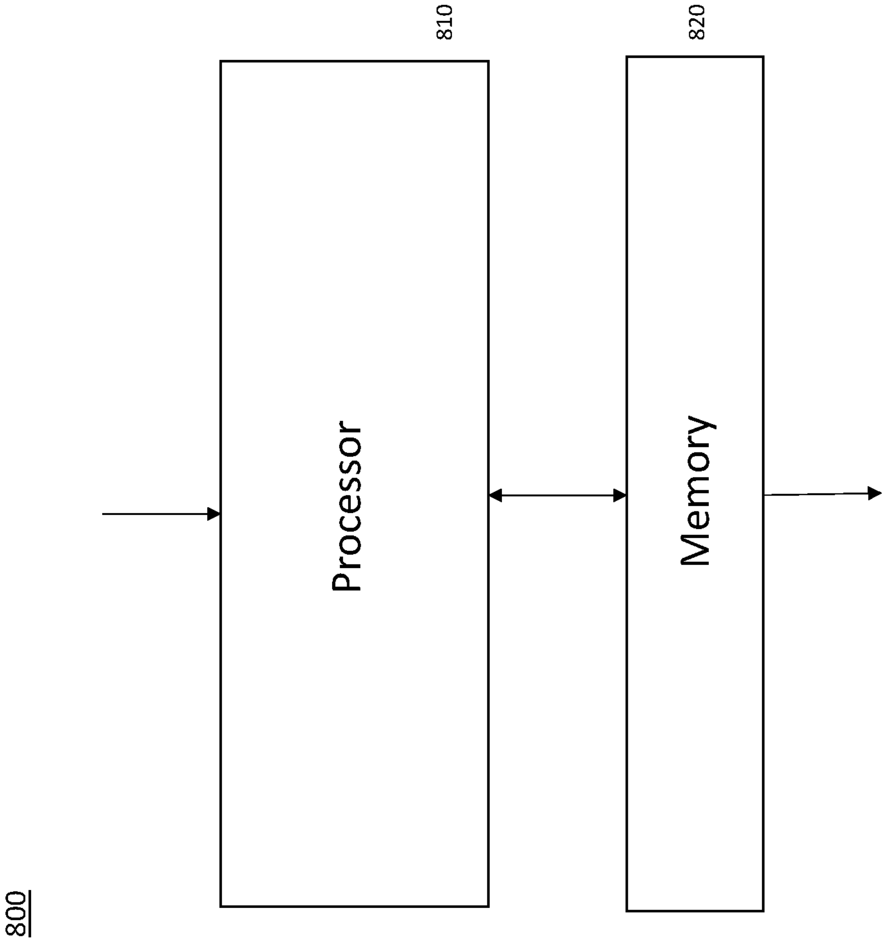

Figure 8

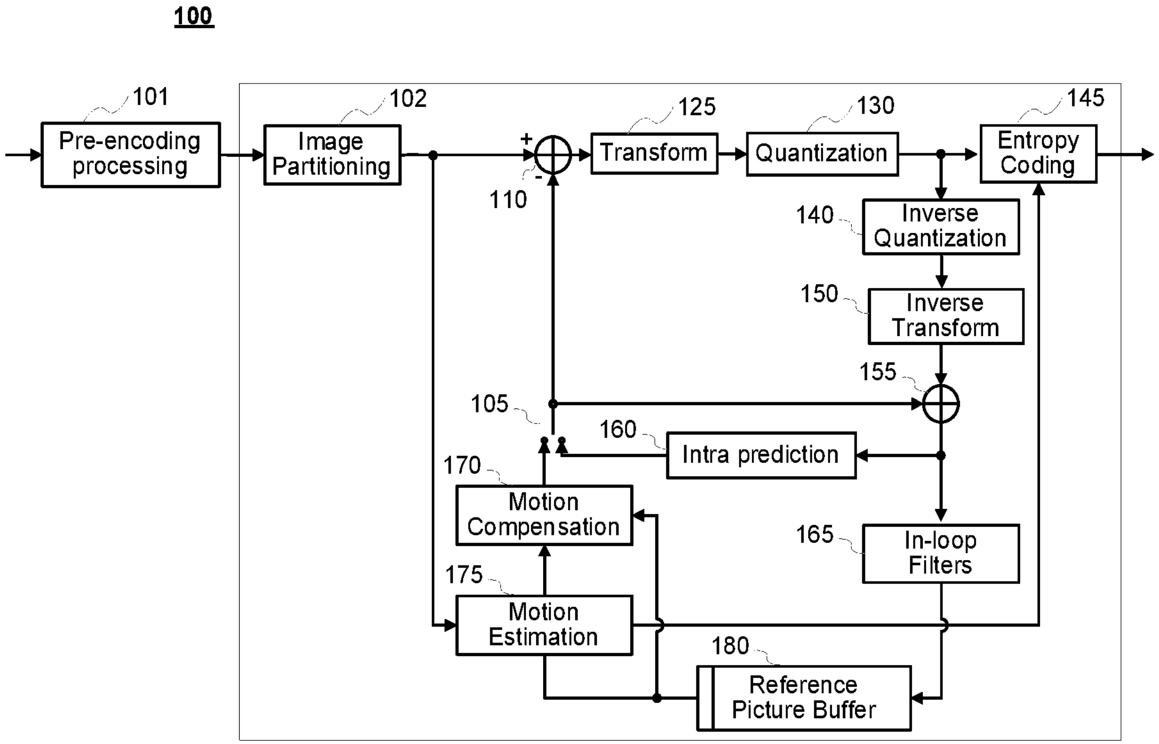
100
101
Pre-encoding processing
102
Image Partitioning
+
-
110
125
Transform
130
Quantization
145
Entropy Coding
140
Inverse Quantization
150
Inverse Transform
155
105
160
Intra prediction
170
Motion Compensation
165
In-loop Filters
175
Motion Estimation
180
Reference Picture Buffer

Figure 9

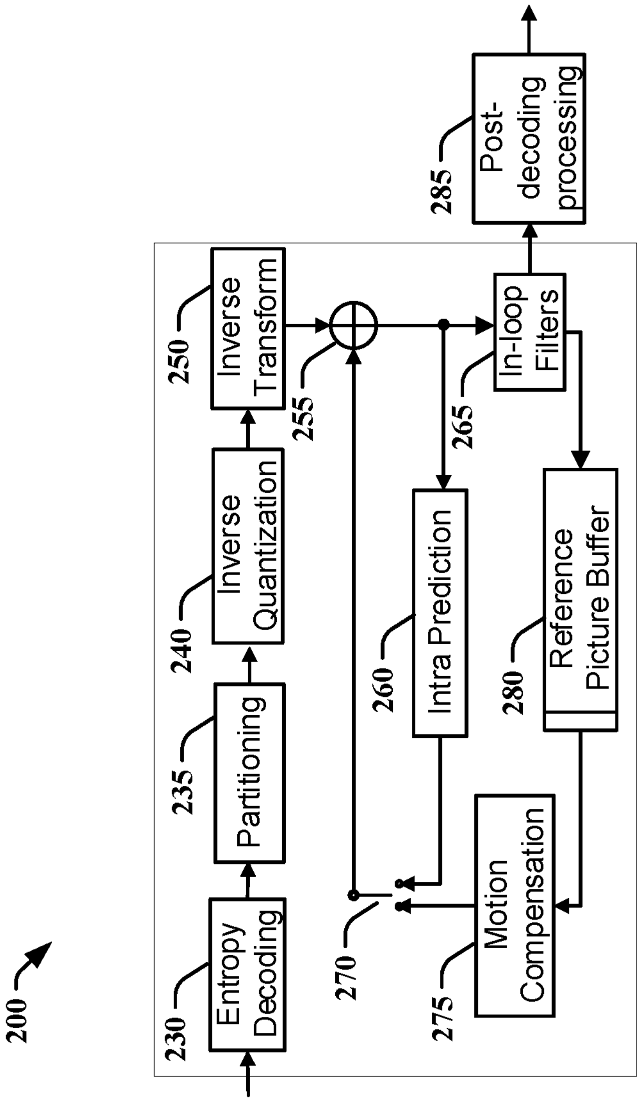
200
230
Entropy Decoding
235
Partitioning
240
Inverse Quantization
250
Inverse Transform
255
260
Intra Prediction
265
In-loop Filters
270
275
Motion Compensation
280
Reference Picture Buffer
285
Post-decoding processing

Figure 10

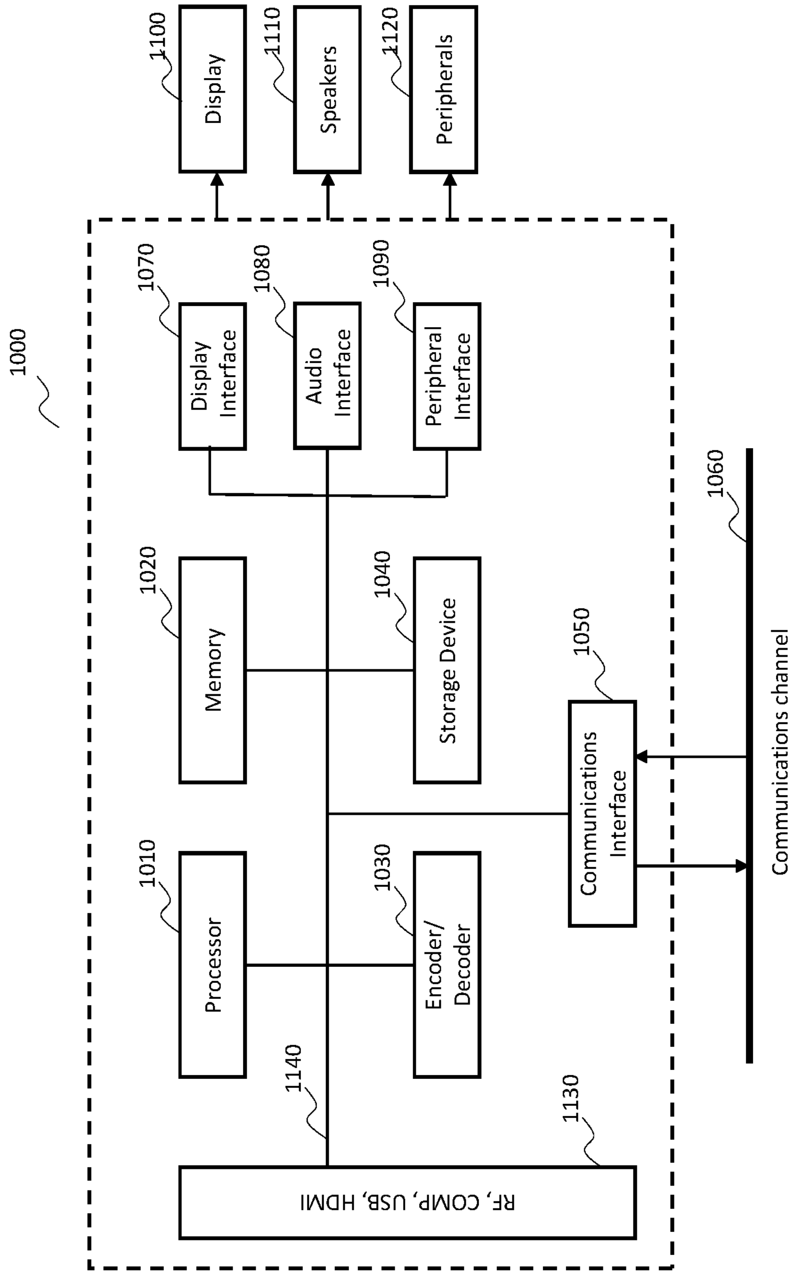
1000
1100
Display
1110
Speakers
1120
Peripherals
1070
Display Interface
1080
Audio Interface
1090
Peripheral Interface
1010
Processor
1020
Memory
1030
Encoder/ Decoder
1040
Storage Device
1050
Communications Interface
1060
Communications channel
1140
1130
RF, COMP, USB, HDMI

Figure 11

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2020/022585

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/42 G06N3/08
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Liang Zhao ET AL: "Theoretical Properties for Neural Networks with Weight Matrices of Low Displacement Rank", , 1 March 2017 (2017-03-01), XP055699025, Retrieved from the Internet: URL:https://arxiv.org/pdf/1703.00144.pdf [retrieved on 2020-05-27] abstract sections 1 and 3 -----	1-15
A	ANNA T THOMAS ET AL: "Learning Compressed Transforms with Low Displacement Rank", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 4 October 2018 (2018-10-04), XP081057560, sections 1 and 2 ----- -/--	1-15

☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 29 May 2020	Date of mailing of the international search report 10/06/2020
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer La, Valérie

1

International application No

PCT/US2020/022585

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2018/107925 A1 (CHOI YOO JIN [US] ET AL) 19 April 2018 (2018-04-19) abstract figures 1,2 -----	1-15

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2020/022585

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2018107925 A1	19-04-2018	CN 107967515 A	27-04-2018
		KR 20180043154 A	27-04-2018
		TW 201816669 A	01-05-2018
		US 2018107925 A1	19-04-2018