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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 July 2019 (25.07.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/141902 A1

(51) International Patent Classification:
G06N 3/04 (2006.01) *H04W 4/50* (2018.01)
H04W 8/24 (2009.01) *G06F 9/50* (2006.01)
H04L 29/08 (2006.01) *G06N 20/00* (2019.01)

(21) International Application Number:
PCT/FI2019/050028

(22) International Filing Date:
15 January 2019 (15.01.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
20185041 17 January 2018 (17.01.2018) FI

(71) Applicant: **NOKIA TECHNOLOGIES OY** [FI/FI]; Karaportti 3, 02610 Espoo (FI).

(72) Inventors: **AYTEKIN, Caglar**; Riihuhdankatu 13 B6, 33580 Tampere (FI). **FAN, Lixin**; Suoliojankatu 2, 33720 Tampere (FI). **CRICRI, Francesco**; Massunkuja 1 A 14, 33100 Tampere (FI). **AKSU, Emre**; Virrenkatu 3 D 20, 33800 Tampere (FI).

(74) Agent: **NOKIA TECHNOLOGIES OY** et al.; Ari Aarnio, IPR Department, Karakaari 7, 02610 Espoo (FI).

(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, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH,

(54) Title: AN APPARATUS, A METHOD AND A COMPUTER PROGRAM FOR RUNNING A NEURAL NETWORK

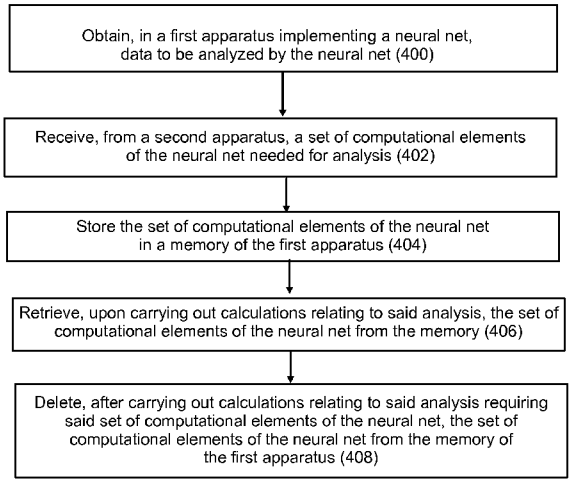

Fig. 4

(57) Abstract: A method comprising: obtaining, in a first apparatus implementing a neural net, data to be analyzed by the a neural net; receiving, from a second apparatus, a set of computational elements of the neural net needed for analysis; storing the set of computational elements of the neural net in a memory of the first apparatus; retrieving, upon carrying out calculations relating to said analysis, the set of computational elements of the neural net from the memory; and deleting, after carrying out calculations relating to said analysis requiring said set of computational elements of the neural net, the set of computational elements of the neural net from the memory of the first apparatus.

WO 2019/141902 A1

[Continued on next page]

GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, 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))*

AN APPARATUS, A METHOD AND A COMPUTER PROGRAM FOR RUNNING A NEURAL NETWORK

TECHNICAL FIELD

[0001] The present invention relates to an apparatus, a method and a computer program for running a neural network.

BACKGROUND

[0002] Recently, the development of various artificial neural network (NN) techniques, especially the ones related to deep learning, has enabled to learn algorithms for several tasks from the raw data, which algorithms may outperform algorithms which have been developed for many years using non-learning based methods.

[0003] Neural networks are used more and more in various types of devices, from smartphones to self-driving cars. Many small devices are typically very constrained in terms of memory, bandwidth and computation capacity. On the other hand, many small devices are configured to run applications, which could benefit from running the application or part of it as NN-based algorithms.

[0004] Due to high memory requirement and computationally costly implementation of typical neural networks, efficient execution of neural networks on low capacity devices confronts problems relating to how to send NN models to the bandwidth and memory limited devices and how to run the models efficiently on the computation-limited devices.

[0005] It has been suggested to deal with the above problem via compressing or quantizing or pruning the neural network weights. Performing these operations on the pre-trained original network results into a new network that requires less memory, and thus less bandwidth if the network is communicated, and less number of computational operations. Therefore, it may be easier to send and run the compressed or quantized or pruned neural network in bandwidth, memory and computation limited devices.

[0006] However, pruning or compressing a neural network may often reduce the accuracy of the resulting model. Furthermore, in some applications, it may be important to have comparable results when the same neural network is run by devices with different memory, bandwidth and computation capabilities.

SUMMARY

[0007] Now in order to at least alleviate the above problems, an enhanced method for running a neural network is introduced herein.

[0008] A method according to a first aspect comprises obtaining, in a first apparatus, data to be analyzed by a neural net; receiving, from a second apparatus, a set of computational elements of the neural net needed for analysis; storing the set of computational elements of the neural net in a memory of the first apparatus; retrieving, upon carrying out calculations relating to said analysis, the set of computational elements of the neural net from the memory; and deleting, after carrying out calculations relating to said analysis requiring said set of computational elements of the neural net, the set of computational elements of the neural net from the memory of the first apparatus.

[0009] According to an embodiment, the set of computational elements of the neural net comprise topology information of at least one layer of the neural net and/or a set of weights for the neural net.

[0010] According to an embodiment, the method further comprises carrying out said calculations based on the data to be analyzed, the topology information of the at least one layer of the neural net and the set of weights for the neural net.

[0011] According to an embodiment, the data to be analyzed is input data of the neural net or an activation resulting from a previous computation in the neural net.

[0012] According to an embodiment, the method further comprises obtaining the weights in the same sequential order as the weights are used in said calculations relating to said analysis.

[0013] According to an embodiment, the method further comprises receiving a first stream comprising weights of a first layer of the neural net needed for said analysis; carrying out the calculations using the weights of said first layer; storing activations from said first layer to be used as an input for a subsequent layer; deleting said weights of said first layer from the memory; and repeating the above steps for any subsequent layer of the neural net needed for said analysis.

[0014] According to an embodiment, the method further comprises receiving a first stream comprising only a first convolutional filter of a first layer of the neural net needed for said analysis; carrying out the calculations using said first convolutional filter of said first layer; storing activations resulting from using said first convolutional filter to be used as an input for a subsequent calculation; deleting said using said first convolutional filter of said first layer

from the memory; and repeating the above steps for any subsequent convolutional filter of said first layer of the neural net needed for said analysis.

[0015] An apparatus according to a second aspect comprises means for obtaining data to be analyzed by a neural net; means for receiving a set of computational elements of the neural net needed for analysis; means for storing the set of computational elements of the neural net in a memory; means for retrieving, upon carrying out calculations relating to said analysis, the set of computational elements of the neural net from the memory; and means for deleting, after carrying out calculations relating to said analysis requiring said set of computational elements of the neural net, the set of computational elements of the neural net from the memory.

[0016] A third aspect relates to a method comprising: receiving, by a second apparatus, a request from a first apparatus to provide a set of computational elements of a neural net needed for an analysis; providing a first subset of the computational elements to the first apparatus; receiving, from the first apparatus, a request for at least a second subset of computational elements; and providing at least said second subset of the computational elements to the first apparatus.

[0017] An apparatus according to a fourth aspect comprises means for receiving a request from a remote apparatus to provide a set of computational elements of a neural net needed for an analysis; means for providing a first subset of the computational elements to the remote apparatus; means for receiving, from the remote apparatus, a request for at least a second subset of computational elements; and means for providing at least said second subset of the computational elements to the remote apparatus.

[0018] The apparatuses as described above, are arranged to carry out the above methods and one or more of the embodiments related thereto.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] For better understanding of the present invention, reference will now be made by way of example to the accompanying drawings in which:

[0020] Figure 1 shows schematically an electronic device employing embodiments of the invention;

[0021] Figure 2 shows schematically a user equipment suitable for employing embodiments of the invention;

[0022] Figure 3 further shows schematically electronic devices employing embodiments of the invention connected using wireless and wired network connections;

[0023] Figure 4 shows a flow chart of a method for running a neural network according to an embodiment of the invention;

[0024] Figure 5 shows a block diagram for a layer-wise streaming of layer weights according to an embodiment of the invention; and

[0025] Figure 6 shows a block diagram for a filter-wise streaming of layer weights according to an embodiment of the invention.

DETAILED DESCRIPTON OF SOME EXAMPLE EMBODIMENTS

[0026] The following describes in further detail suitable apparatus and possible mechanisms for running a neural network according to embodiments. In this regard reference is first made to Figures 1 and 2, where Figure 1 shows an example block diagram of an apparatus 50. The apparatus may be an Internet of Things (IoT) apparatus configured to perform various functions, such as for example, gathering information by one or more sensors, receiving or transmitting information, analyzing information gathered or received by the apparatus, or the like. The apparatus may comprise a video coding system, which may incorporate a codec. Figure 2 shows a layout of an apparatus according to an example embodiment. The elements of Figs. 1 and 2 will be explained next.

[0027] The electronic device 50 may for example be a mobile terminal or user equipment of a wireless communication system, a sensor device, a tag, or other lower power device. However, it would be appreciated that embodiments of the invention may be implemented within any electronic device or apparatus which may process data by neural networks.

[0028] The apparatus 50 may comprise a housing 30 for incorporating and protecting the device. The apparatus 50 further may comprise a display 32 in the form of a liquid crystal display. In other embodiments of the invention the display may be any suitable display technology suitable to display an image or video. The apparatus 50 may further comprise a keypad 34. In other embodiments of the invention any suitable data or user interface mechanism may be employed. For example the user interface may be implemented as a virtual keyboard or data entry system as part of a touch-sensitive display.

[0029] The apparatus may comprise a microphone 36 or any suitable audio input which may be a digital or analogue signal input. The apparatus 50 may further comprise an audio output device which in embodiments of the invention may be any one of: an earpiece 38, speaker, or an analogue audio or digital audio output connection. The apparatus 50 may also comprise a battery (or in other embodiments of the invention the device may be powered by any suitable mobile energy device such as solar cell, fuel cell or clockwork generator). The

apparatus may further comprise a camera capable of recording or capturing images and/or video. The apparatus 50 may further comprise an infrared port for short range line of sight communication to other devices. In other embodiments the apparatus 50 may further comprise any suitable short range communication solution such as for example a Bluetooth wireless connection or a USB/firewire wired connection.

[0030] The apparatus 50 may comprise a controller 56, processor or processor circuitry for controlling the apparatus 50. The controller 56 may be connected to memory 58 which in embodiments of the invention may store both data in the form of image and audio data and/or may also store instructions for implementation on the controller 56. The controller 56 may further be connected to codec circuitry 54 suitable for carrying out coding and decoding of audio and/or video data or assisting in coding and decoding carried out by the controller.

[0031] The apparatus 50 may further comprise a card reader 48 and a smart card 46, for example a UICC and UICC reader for providing user information and being suitable for providing authentication information for authentication and authorization of the user at a network.

[0032] The apparatus 50 may comprise radio interface circuitry 52 connected to the controller and suitable for generating wireless communication signals for example for communication with a cellular communications network, a wireless communications system or a wireless local area network. The apparatus 50 may further comprise an antenna 44 connected to the radio interface circuitry 52 for transmitting radio frequency signals generated at the radio interface circuitry 52 to other apparatus(es) and for receiving radio frequency signals from other apparatus(es).

[0033] The apparatus 50 may comprise a camera capable of recording or detecting individual frames which are then passed to the codec 54 or the controller for processing. The apparatus may receive the video image data for processing from another device prior to transmission and/or storage. The apparatus 50 may also receive either wirelessly or by a wired connection the image for coding/decoding. The structural elements of apparatus 50 described above represent examples of means for performing a corresponding function.

[0034] With respect to Figure 3, an example of a system within which embodiments of the present invention can be utilized is shown. The system 10 comprises multiple communication devices which can communicate through one or more networks. The system 10 may comprise any combination of wired or wireless networks including, but not limited to a wireless cellular telephone network (such as a GSM, UMTS, CDMA, 4G, 5G network etc.), a wireless local area network (WLAN) such as defined by any of the IEEE 802.x standards, a Bluetooth

personal area network, an Ethernet local area network, a token ring local area network, a wide area network, and the Internet.

[0035] The system 10 may include both wired and wireless communication devices and/or apparatus 50 suitable for implementing embodiments of the invention.

[0036] For example, the system shown in Figure 3 shows a mobile telephone network 11 and a representation of the internet 28. Connectivity to the internet 28 may include, but is not limited to, long range wireless connections, short range wireless connections, and various wired connections including, but not limited to, telephone lines, cable lines, power lines, and similar communication pathways.

[0037] The example communication devices shown in the system 10 may include, but are not limited to, an electronic device or apparatus 50, a combination of a personal digital assistant (PDA) and a mobile telephone 14, a PDA 16, an integrated messaging device (IMD) 18, a desktop computer 20, a notebook computer 22. The apparatus 50 may be stationary or mobile when carried by an individual who is moving. The apparatus 50 may also be located in a mode of transport including, but not limited to, a car, a truck, a taxi, a bus, a train, a boat, an airplane, a bicycle, a motorcycle or any similar suitable mode of transport.

[0038] The embodiments may also be implemented in a set-top box; i.e. a digital TV receiver, which may/may not have a display or wireless capabilities, in tablets or (laptop) personal computers (PC), which have hardware and/or software to process neural network data, in various operating systems, and in chipsets, processors, DSPs and/or embedded systems offering hardware/software based coding.

[0039] Some or further apparatus may send and receive calls and messages and communicate with service providers through a wireless connection 25 to a base station 24. The base station 24 may be connected to a network server 26 that allows communication between the mobile telephone network 11 and the internet 28. The system may include additional communication devices and communication devices of various types.

[0040] The communication devices may communicate using various transmission technologies including, but not limited to, code division multiple access (CDMA), global systems for mobile communications (GSM), universal mobile telecommunications system (UMTS), time divisional multiple access (TDMA), frequency division multiple access (FDMA), transmission control protocol-internet protocol (TCP-IP), short messaging service (SMS), multimedia messaging service (MMS), email, instant messaging service (IMS), Bluetooth, IEEE 802.11 and any similar wireless communication technology. A communications device involved in implementing various embodiments of the present

invention may communicate using various media including, but not limited to, radio, infrared, laser, cable connections, and any suitable connection.

[0041] In telecommunications and data networks, a channel may refer either to a physical channel or to a logical channel. A physical channel may refer to a physical transmission medium such as a wire, whereas a logical channel may refer to a logical connection over a multiplexed medium, capable of conveying several logical channels. A channel may be used for conveying an information signal, for example a bitstream, from one or several senders (or transmitters) to one or several receivers.

[0042] The embodiments may also be implemented in so-called IoT devices. The Internet of Things (IoT) may be defined, for example, as an interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure. The convergence of various technologies has and will enable many fields of embedded systems, such as wireless sensor networks, control systems, home/building automation, etc. to be included the Internet of Things (IoT). In order to utilize Internet IoT devices are provided with an IP address as a unique identifier. IoT devices may be provided with a radio transmitter, such as WLAN or Bluetooth transmitter or a RFID tag. Alternatively, IoT devices may have access to an IP-based network via a wired network, such as an Ethernet-based network or a power-line connection (PLC).

[0043] Recently, the development of various artificial neural network (NN) techniques, especially the ones related to deep learning, has enabled to learn algorithms for several tasks from the raw data, which algorithms may outperform algorithms which have been developed for many years using traditional (non-learning based) methods.

[0044] Artificial neural networks, or simply neural networks, are parametric computation graphs consisting of units and connections. The units may be arranged in successive layers, and in some neural network architectures only units in adjacent layers are connected. Each connection has an associated parameter or weight, which defines the strength of the connection. The weight gets multiplied by the incoming signal in that connection. In fully-connected layers of a feedforward neural network, each unit in a layer is connected to each unit in the following layer. So, the signal which is output by a certain unit gets multiplied by the connections connecting that unit to another unit in the following layer. The latter unit then may perform a simple operation such as a sum of the weighted signals.

[0045] Apart from fully-connected layers, there are different types of layers, such as but not limited to convolutional layers, non-linear activation layers, batch-normalization layers, and pooling layers.

[0046] The input layer receives the input data, such as images, and the output layer is task-specific and outputs an estimate of the desired data, for example a vector whose values represent a class distribution in the case of image classification. The "quality" of the neural network's output is evaluated by comparing it to ground-truth output data. This comparison would then provide a "loss" or "cost" function.

[0047] The weights of the connections represent the biggest part of the learnable parameters of a neural network. Other learnable parameters may be for example the parameters of the batch-normalization layer.

[0048] The parameters are learned by means of a training algorithm, where the goal is to minimize the loss function on a training dataset. The training dataset is regarded as a representative sample of the whole data. One popular learning approach is based on iterative local methods, where the loss is minimized by following the negative gradient direction. Here, the gradient is understood to be the gradient of the loss with respect to the weights of the neural network. The loss is represented by the reconstructed prediction error. Computing the gradient on the whole dataset may be computationally too heavy, thus learning is performed in sub-steps, where at each step a mini-batch of data is sampled and gradients are computed from the mini-batch. This is regarded to as stochastic gradient descent. The gradients are usually computed by back-propagation algorithm, where errors are propagated from the output layer to the input layer, by using the chain rule for differentiation. If the loss function or some components of the neural network are not differentiable, it is still possible to estimate the gradient of the loss by using policy gradient methods, such as those used in reinforcement learning. The computed gradients are then used by one of the available optimization routines (such as stochastic gradient descent, Adam, RMSprop, etc.), to compute a weight update, which is then applied to update the weights of the network. After a full pass over the training dataset, the process is repeated several times until a convergence criterion is met, usually a generalization criterion. The gradients of the loss, i.e., the gradients of the reconstructed prediction error with respect to the weights of the neural network, may be referred to as the training signal.

[0049] Online training consists of learning the parameters of the neural network continuously on a stream of data, as opposed to the case where the dataset is finite and given in its entirety and the network is used only after training has completed.

[0050] In the neural networks described above, a large memory storage is typically needed for the weights of the neural network. Moreover, the number of operations to be performed is also very high.

[0051] Some neural networks, for example feedforward neural networks, which represent the vast majority of neural networks, can operate in a sequential manner, where each layer takes input from a single previous layer and gives output to a single subsequent layer. In the case of a convolution layer consisting of several convolution filters, for the calculations within a layer, operations are conducted separately for each filter in the layer, i.e., independently given the input from the previous layer. Usually these filter-wise operations correspond to applying several convolutions separately to an input. Usually, in order to exploit this structure more efficiently, GPUs are used for parallel computing.

[0052] Neural networks are used more and more in various types of devices, from smartphones to self-driving cars. One very important category of devices is represented by very small devices such as IoT devices. Many IoT devices, such as the ones mentioned above, are typically very constrained in terms of memory, bandwidth and computation capacity. On the other hand, many IoT devices are configured to run applications, which could benefit from running the application or part of it as NN-based algorithms. An example of such an application is object recognition in IoT devices which are capable of media acquisition, such as IoT devices provided with a camera.

[0053] Due to high memory requirement and computationally costly implementation of typical neural networks, there are at least the following challenges to efficiently run neural networks on low capacity devices.

- how to send NN models to the bandwidth limited devices;
- how to store the NN models inside the memory limited devices;
- how to run the models efficiently on the computation-limited devices.

[0054] The above problem may be alleviated via compressing or quantizing or pruning the neural network weights. Performing these operations on the pre-trained original network results into a new network that requires less memory and less number of computational operations. Therefore, it may be easier to send and run the neural network in bandwidth, memory and computation limited devices.

[0055] However, pruning or compressing a neural network may often reduce the accuracy of the resulting model.

[0056] Now an enhanced method for running the neural network especially in bandwidth, memory and computation limited devices is introduced.

[0057] The method, which is depicted in the flow chart of Figure 4, comprises obtaining (400), in a first apparatus implementing a neural net, data to be analyzed by the neural net; receiving (402), from a second apparatus, a set of computational elements of the neural net

needed for analysis; storing (404) the set of computational elements of the neural net in a memory of the first apparatus; retrieving (406), upon carrying out calculations relating to said analysis, the set of computational elements of the neural net from the memory; and deleting (408), after carrying out calculations relating to said analysis requiring said set of computational elements of the neural net, the set of computational elements of the neural net from the memory of the first apparatus.

[0058] The method may further comprise storing the results of the calculations relating to said analysis requiring said set of computations elements of the neural net. The set of computational elements may comprise a subset of computational elements of the neural network, for example, computational elements corresponding to one layer of the neural network. A computational element may comprise a numerical value and/or a mathematical operation associated with a node or neuron of the neural network.

[0059] Thus, the method is based on a simultaneous transmission/reception and running of only subsets of neural networks, whereupon the benefits are most prominent in computation, bandwidth and memory limited devices. Accordingly, only a necessary set of computational elements of a neural network is sent to the device running a neural net-based algorithm. The device may not permanently store the set of computational elements, but may only store them temporarily, for example in a cache, and as soon as related computations due to the set of computational elements are completed, the device releases (e.g. deletes) the set of computational elements. The approach enables efficient implementation of neural networks without compressing them in computation, bandwidth and memory limited devices, thus being able to retain the original accuracy of the model. According to an embodiment, the sequential transfer of the computational elements may be combined with compressing the computational elements before the transfer.

[0060] Herein, the first apparatus running the neural net-based algorithm may also be referred to as “a layer user device”. The second apparatus sending the necessary sets of computational elements to the first device may also be referred to as “a layer provider device”. The first apparatus comprises a communication module allowing it to communicate with one or more other devices, at least with said second apparatus (“the layer provider device”). Thus, the first apparatus is configured to perform an analysis of data using a neural network.

[0061] According to an embodiment, the data to be analyzed is input data (e.g., gathered from a data source) of the neural net or a set of neural network activations resulting from a previous computation in a subset of the neural net. The first apparatus may be capable of

capturing the input data, or input data may be streamed to the first apparatus from a separate data source. On the other hand, the data to be analyzed may be a result of a previous calculation of a subset of the neural network, such as a set of activations from a previous layer of the neural network.

[0062] According to an embodiment, the set of computational elements of the neural net comprise topology information of at least one layer of the neural net and a set of weights related to that one layer for the neural net. Thus, the first apparatus is configured to store a layer, which consists of both topology information and weights information. The first apparatus is also configured to store either the input data or the activations computed at a previous computation step. Herein, the computation step refers to the step during which one set of received computational elements (for example one layer) is run on data (either input data or previous activations).

[0063] According to an embodiment, said calculations are carried out based on the data to be analyzed, the topology information of the at least one layer of the neural net and the set of weights for the neural net. Consequently, the first apparatus is configured to perform computation of a neural network layer, given the data or previous activations, and the layer's topology and weights.

[0064] According to an embodiment, the first apparatus obtains the weights in the same sequential order as the weights are used in said calculations relating to said analysis. Thus, the second apparatus sends, for example streams, the weights necessary at a particular stage of the neural net-based algorithm to the first apparatus. The weights may be streamed in the same sequential order that they are used in the network.

[0065] According to an embodiment, the sequential order is determined layer-wise or filter-wise.

[0066] According to an embodiment, upon streaming the weights layer-wise, the method further comprises receiving a first stream comprising weights of a first layer of the neural net needed for said analysis; carrying out the calculations using the weights of said first layer; storing activations from said first layer to be used as an input for a subsequent layer; deleting said weights of said first layer from the memory; and repeating the above steps for any subsequent layer of the neural net needed for said analysis.

[0067] In some neural networks, for example feed-forward neural networks, in order to compute the activations of a certain layer, it is not necessary to store the filters (weights) of the neural network for other layers (previous or later). It is only needed to store the activations from the layers which the current layer will use as input.

[0068] Therefore, a layer-wise streaming of neural networks may be applied such that in each stream, one or more (network) layers comprising topology and weights are sent to the first apparatus and the first apparatus stores the topology and the weights only temporarily, for example in a cache, to maintain the topology and the weights during the computation and releases the topology and the weights as soon as the related computations are completed. The first apparatus then stores the activations and, if the computation is not completed, waits for the next stream to provide the next layer's weights to be stored in the cache. The process is repeated until the entire computation of the neural network is completed.

[0069] According to an embodiment, topology information may be predetermined or signalled beforehand to the layer user apparatus. In such case the stream may include the weight information without topology information. The layer user device may store activations of the layers that are going to be used later due to connections from earlier layer. For example at layer N , the network may use activation output of layer K (K<N). In that case, it may be preferred to know beforehand which layers to keep, since it may use them later. In this embodiment, topology information can be sent beforehand, or the server can only send another signal when sending the layer K in order to let the layer user device know that this layer activation will be used in layer N, so that the layer user device may avoid deleting those activations after implementation of layer K+1.

[0070] Figure 5 illustrates a simplified block chart for implementing caching when performing the layer-wise streaming. The second apparatus ("layer provider device") 500 sends the topology of the current layer and its weights to the first apparatus ("layer user device") 510, wherein the layer's topology and weights are stored in a first cache 520. It is noted that the topology information of the layer and the weights may be transmitted either together or separately. It is also noted that the topology information of the layer and the weights may be sent as pushed by the second apparatus 500 or after an initial first transmission, as requested by the first apparatus. The first apparatus comprises a layer assembler 530, which assembles the layer needed for the computation by using the topology and weights. Herein the topology may refer, for example, to the type of the layer, the number of neurons, the number of filters (in case of a convolution layer), or the type of non-linearity. For example, the assembler may create the computation graph defined by the topology and weights, and which is then run on either the input data or the previous activations. The layer computation block 540 carries out the actual execution of the layer on given the input controlled by a switch 550. The input may be either the data from a data source or activations

output at a previous computation step. The previous activations may be temporarily stored in a second cache 560.

[0071] According to an embodiment, for example when the one or more layers are convolution layers consisting of several convolution filters, upon streaming the weights filter-wise, the method further comprises receiving a first stream comprising only a first convolutional filter of a first layer of the neural net needed for said analysis; carrying out the calculations using said first convolutional filter of said first layer; storing activations resulting from using said first convolutional filter to be used as an input for a subsequent calculation; deleting said using said first convolutional filter of said first layer from the memory; and repeating the above steps for any subsequent convolutional filter of said first layer of the neural net needed for said analysis.

[0072] In some neural networks, for example feed-forward neural networks, during the computation of a layer, there often exists several computations that are independent of each other. One example is multiple convolutions to be performed on the input separately. If parallel processing units, such as GPUs, are not available, these computations are performed at different times. Hence, it is not necessary to store filters in a layer other than the one that is used for the current computation.

[0073] This enables to use a filter-wise streaming for neural networks. In each stream, only one convolutional filter is sent to the first apparatus and the first apparatus stores the first filter only temporarily, for example in a cache, to maintain the first filter during the computation and releases the filter. The first apparatus then obtains a subsequent filter from the next stream, wherein the first and the subsequent filters are related to the same layer. During streaming, the second apparatus may include a variable into the stream to indicate the layer of the convolutional filter so that the first apparatus knows the input of the convolutional filter. The process is repeated until all filters are sent and the computation is completed.

[0074] Figure 6 illustrates a simplified block chart for implementing caching when performing the filter-wise streaming. The second apparatus (herein referred to as "filter provider device") 600 sends the layer's topology and the weights of the current feature to the first apparatus ("layer user device") 610, wherein the layer's topology and weights of the current feature are stored in a first cache 620. Similarly to the layer-wise implementation, the topology may refer, for example, to the type of the layer, the number of neurons, the number of filters (in case of a convolution layer), or the type of non-linearity. The feature computation block 630 carries out the actual execution of the feature on given the input controlled by a switch 640. The input may be either the data from a data source or activations

output at a previous computation step. The previous activations may be temporarily stored in a second cache 650. After necessary filters for the particular layer are executed, an output assembler 660 assembles the filter outputs (cached in the second cache 660) from said layer according to the layer topology (cached in the first cache 620).

[0075] As it becomes evident from the above, both the layer-wise and the filter-wise streaming of weights and the related caching schemes provide the benefit of allowing dynamic configuration of neural network architectures, which is an advantageous feature for deep learning optimization.

[0076] The second apparatus provides the first apparatus ("layer user device") with the topology information of network layer.

[0077] According to an embodiment, the topology information comprises a topology layer type and a description for the entire network. The second apparatus may send the whole topology information in advance. In such case, the signalling may be implemented, for example, as follows:

- Topology Layer Type Signal: (C-C-C-C-D), wherein four consecutive Convolutional layers (C) are followed by a dense layer (D). For example, the signaling may be (0,0,0,0,1)
- Topology Layer Description Signal: (Fnum-Fsize1- Fsize2-FStride1-FStride2-A-P-Psize1-Psize2-Pstride1-Pstride2- Fnum-Fsize1- Fsize2-Stride1-Stride2-A-P-Psize1-Psize2-Pstride1-Pstride2- Fnum-Fsize1- Fsize2-Stride1-Stride2-A-P-Psize1-Psize2-Pstride1-Pstride2-DSize), where Fnum= number of filters, Fsize1,2= size of the filter in the two spatial dimensions, FStride= stride of the convolution, A= type of activation (0:no activation,1:relu,2 sigmoid for example), P= type of pooling (0: no pooling, 1, avg pooling, 2: max pooling for example), Psize= size of pooling, Pstride= stride of pooling, and DSize= neuron number in dense layer.

[0078] According to an embodiment, the topology information comprises a topology layer type and a description for one layer of the network. The second apparatus may send the topology layer type and the description for each layer sequentially. In such case, the first signalling may include, for example, one or more of the following:

- Type of Layer: C (convolutional)
- Description Signal: Fnum-Fsize1- Fsize2-FStride1-FStride2-A-P-Psize1-Psize2-Pstride1-Pstride2, where the above semantic applies

[0079] After the first signal, the first apparatus may request for either layer-wise or feature-wise streaming with a request signal to the second apparatus. The request signal may be, for

example, 1 for layer wise 0 for filter wise. In general, the request signal may indicate a type of neural network streaming. It is possible that the first apparatus first evaluates its capability regarding the network topology informed in the first signaling, and only then decides whether to request for a layer-wise or a feature-wise streaming or neither.

[0080] If the first apparatus requests layer-wise streaming, the second apparatus sends the first layer, where the layer signal includes the weights of the first layer.

[0081] Based on the initially sent topology, the first apparatus knows what the numbers in the stream mean. For example, for 16 filters, 3x3 filter size example, the first 9 (3x3) numbers in the stream corresponds to the first filter.

[0082] After the computations related to the layer is completed, the first apparatus sends a next request signal to obtain weights for a subsequent layer. This process continues until the complete execution of the network.

[0083] The first apparatus may conclude that there are no more layers or filters to execute by one of the following alternatives:

- The full topology or full sequence of layers or filters was received in advance.
- The first apparatus has sent a request signal and in return it receives a stop signal comprising an indication that no more layers or filters will be provided.
- When receiving the last layer, the stream may also include a signaling filed indicating that it is the last layer to execute.
- Other similar mechanisms may be used.

[0084] Once the last layer is executed, the first apparatus provides the final network output to its client, which may be a communication module, a storage device, a UI output device, etc.

[0085] If, on the other hand, the first apparatus requests filter-wise streaming, the second apparatus may send the first filter, where the filter signal includes the weights of the first filter.

[0086] Based on the initially sent topology, the first apparatus knows how to convert the 1D stream to 2D filters. After the computations related to the feature is completed, the first apparatus sends a request signal to obtain a subsequent filter. This process continues until the complete execution of the network.

[0087] The computational elements may be packetized or grouped in the stream such that the layer user device is able to distinguish different computational elements from each other. For example a layer identifier and/or a filter identifier may be associated with a group of computational elements. The topology information may indicate the layer identifiers or filter

identifier belonging to the neural network. Based on this information a layer user device may determine if necessary computational elements have been received and/or whether they have been received in the correct order.

[0088] Further aspects of the invention relate to the operation of the second apparatus ("layer/filter provider device"). In accordance with what has been disclosed above, the operation of the second apparatus may be defined by a method comprising: receiving, by a second apparatus, a request from a first apparatus to provide a subset of computational elements of a neural net; providing a first subset of the computational elements to the first apparatus; receiving, from the first apparatus a request for at least a second subset of computational elements; and providing at least said second subset of the computational elements to the first apparatus.

[0089] In general, the various embodiments of the invention may be implemented in hardware or special purpose circuits, software, logic or any combination thereof. For example, some aspects may be implemented in hardware, while other aspects may be implemented in firmware or software which may be executed by a controller, microprocessor or other computing device, although the invention is not limited thereto. While various aspects of the invention may be illustrated and described as block diagrams, flow charts, or using some other pictorial representation, it is well understood that these blocks, apparatus, systems, techniques or methods described herein may be implemented in, as non-limiting examples, hardware, software, firmware, special purpose circuits or logic, general purpose hardware or controller or other computing devices, or some combination thereof.

[0090] The embodiments of this invention may be implemented by computer software executable by a data processor of the mobile device, such as in the processor entity, or by hardware, or by a combination of software and hardware. Further in this regard it should be noted that any blocks of the logic flow as in the Figures may represent program steps, or interconnected logic circuits, blocks and functions, or a combination of program steps and logic circuits, blocks and functions. The software may be stored on such physical media as memory chips, or memory blocks implemented within the processor, magnetic media such as hard disk or floppy disks, and optical media such as for example DVD and the data variants thereof, CD.

[0091] The memory may be of any type suitable to the local technical environment and may be implemented using any suitable data storage technology, such as semiconductor-based memory devices, magnetic memory devices and systems, optical memory devices and systems, fixed memory and removable memory. The data processors may be of any type

suitable to the local technical environment, and may include one or more of general purpose computers, special purpose computers, microprocessors, digital signal processors (DSPs) and processors based on multi-core processor architecture, as non-limiting examples.

[0092] Embodiments of the inventions may be practiced in various components such as integrated circuit modules. The design of integrated circuits is by and large a highly automated process. Complex and powerful software tools are available for converting a logic level design into a semiconductor circuit design ready to be etched and formed on a semiconductor substrate.

[0093] Programs, such as those provided by Synopsys, Inc. of Mountain View, California and Cadence Design, of San Jose, California automatically route conductors and locate components on a semiconductor chip using well established rules of design as well as libraries of pre-stored design modules. Once the design for a semiconductor circuit has been completed, the resultant design, in a standardized electronic format (e.g., Opus, GDSII, or the like) may be transmitted to a semiconductor fabrication facility or "fab" for fabrication.

[0094] The foregoing description has provided by way of exemplary and non-limiting examples a full and informative description of the exemplary embodiment of this invention. However, various modifications and adaptations may become apparent to those skilled in the relevant arts in view of the foregoing description, when read in conjunction with the accompanying drawings and the appended claims. However, all such and similar modifications of the teachings of this invention will still fall within the scope of this invention.

CLAIMS:

1. A method comprising

obtaining, in a first apparatus, data to be analyzed by a neural net;

receiving, from a second apparatus over a network connection, a set of computational elements of the neural net needed for analysis;

storing the set of computational elements of the neural net in a memory of the first apparatus;

retrieving, upon carrying out calculations relating to said analysis, the set of computational elements of the neural net from the memory; and

deleting, after carrying out calculations relating to said analysis requiring said set of computational elements of the neural net, the set of computational elements of the neural net from the memory of the first apparatus.

2. The method according to claim 1, wherein the set of computational elements of the neural net comprise topology information of at least one layer of the neural net and/or a set of weights for the neural net.

3. The method according to claim 2, further comprising

carrying out said calculations based on the data to be analyzed, the topology information of the at least one layer of the neural net and the set of weights for the neural net.

4. The method according to any preceding claim, wherein the data to be analyzed is input data of the neural net or an activation resulting from a previous computation in the neural net.

5. The method according to claim 2 or 3, further comprising

obtaining the weights in the same sequential order as the weights are used in said calculations relating to said analysis.

6. The method according to claim 5, further comprising

receiving a first stream comprising weights of a first layer of the neural net needed for said analysis;

carrying out the calculations using the weights of said first layer;

storing activations from said first layer to be used as an input for a subsequent layer;

deleting said weights of said first layer from the memory; and

repeating the above steps for any subsequent layer of the neural net needed for said analysis.

7. The method according to claim 5, further comprising

receiving a first stream comprising only a first convolutional filter of a first layer of the neural net needed for said analysis;

carrying out the calculations using said first convolutional filter of said first layer;

storing activations resulting from using said first convolutional filter to be used as an input for a subsequent calculation;

deleting said using said first convolutional filter of said first layer from the memory; and

repeating the above steps for any subsequent convolutional filter of said first layer of the neural net needed for said analysis.

8. An apparatus comprising

means for obtaining data to be analyzed by a neural net;

means for receiving, over a network connection, a set of computational elements of the neural net needed for analysis;

means for storing the set of computational elements of the neural net in a memory;

means for retrieving, upon carrying out calculations relating to said analysis, the set of computational elements of the neural net from the memory; and

means for deleting, after carrying out calculations relating to said analysis requiring said set of computational elements of the neural net, the set of computational elements of the neural net from the memory.

9. The apparatus according to claim 8, wherein the set of computational elements of the neural net comprise topology information of at least one layer of the neural net and/or a set of weights for the neural net.

10. The apparatus according to claim 9, further comprising

means for carrying out said calculations based on the data to be analyzed, the topology information of the at least one layer of the neural net and the set of weights for the neural net.

11. The apparatus according to any of claims 8 - 10, wherein the data to be analyzed is input data of the neural net or an activation resulting from a previous computation in the neural net.

12. The apparatus according to claim 9 or 10, further comprising

means for obtaining the weights in the same sequential order as the weights are used in said calculations relating to said analysis.

13. The apparatus according to claim 12, further comprising

means for receiving a first stream comprising weights of a first layer of the neural net needed for said analysis;

means for carrying out the calculations using the weights of said first layer;

means for storing activations from said first layer to be used as an input for a subsequent layer;

means for deleting said weights of said first layer from the memory; and

means for repeating the above steps for any subsequent layer of the neural net needed for said analysis.

14. The apparatus according to claim 12, further comprising

means for receiving a first stream comprising only a first convolutional filter of a first layer of the neural net needed for said analysis;

means for carrying out the calculations using said first convolutional filter of said first layer;

means for storing activations resulting from using said first convolutional filter to be used as an input for a subsequent calculation;

means for deleting said using said first convolutional filter of said first layer from the memory; and

means for repeating the above steps for any subsequent convolutional filter of said first layer of the neural net needed for said analysis.

15. The apparatus of according to any of claims 8 - 14, wherein the means comprises at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes the performance of the apparatus.

16. A method comprising:

receiving, by a second apparatus over a network connection, a request from a first apparatus to provide a set of computational elements of a neural net needed for an analysis;

providing a first subset of the computational elements to the first apparatus;

receiving, from the first apparatus, a request for at least a second subset of computational elements; and

providing at least said second subset of the computational elements to the first apparatus.

17. The method according to claim 16, wherein the set of computational elements of the neural net comprise topology information of at least one layer of the neural net and/or a set of weights for the neural net.

18. The method according to claim 17, further comprising

providing the weights in the same sequential order as the weights are used in calculations carried out by the first apparatus.

19. An apparatus comprising:

means for receiving, over a network connection, a request from a remote apparatus to provide a set of computational elements of a neural net needed for an analysis;

means for providing a first subset of the computational elements to the remote apparatus;

means for receiving, from the remote apparatus, a request for at least a second subset of computational elements; and

means for providing at least said second subset of the computational elements to the remote apparatus.

20. The apparatus according to claim 16, wherein the set of computational elements of the neural net comprise topology information of at least one layer of the neural net and/or a set of weights for the neural net.

21. The apparatus according to claim 17, further comprising

means for providing the weights in the same sequential order as the weights are used in calculations carried out by the remote apparatus.

22. The apparatus of according to any of claims 19 - 21, wherein the means comprises at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes the performance of the apparatus

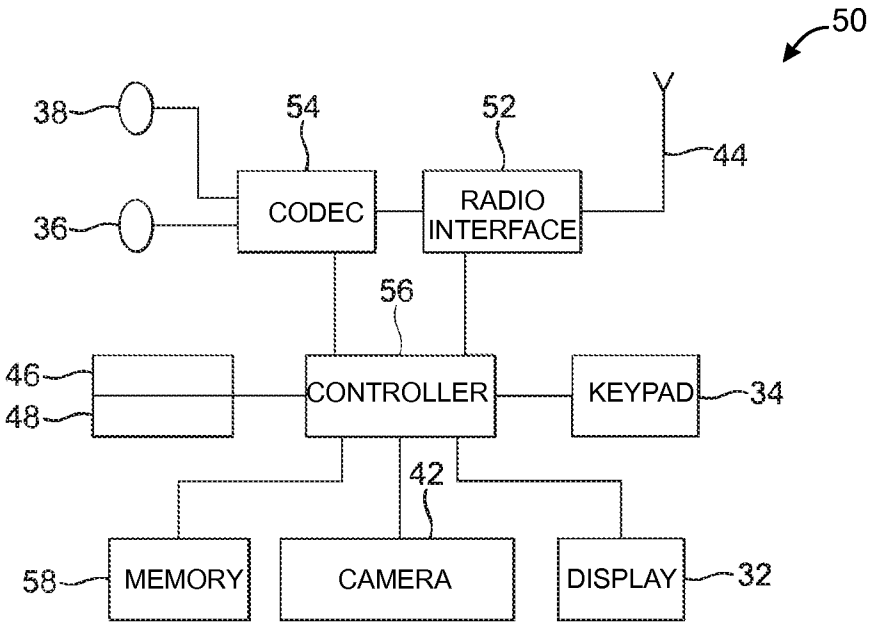
50
38
54
52
44
36
CODEC
RADIO INTERFACE
56
46
48
CONTROLLER
KEYPAD
34
42
58
MEMORY
CAMERA
DISPLAY
32

Fig. 1

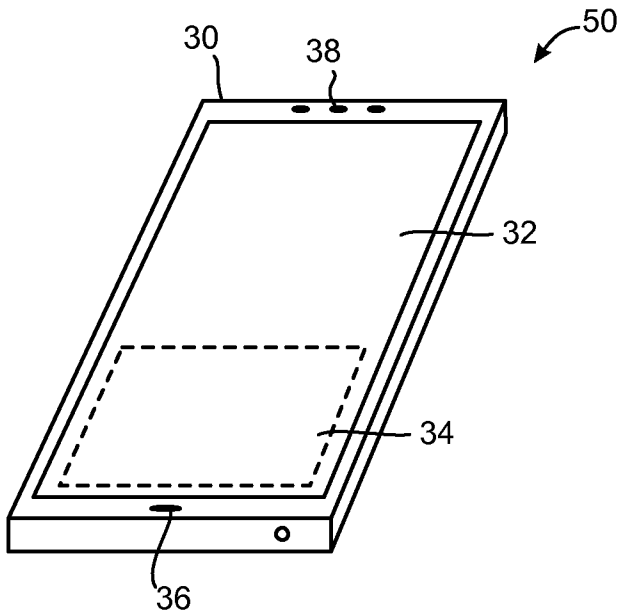
50
30
38
32
34
36

Fig. 2

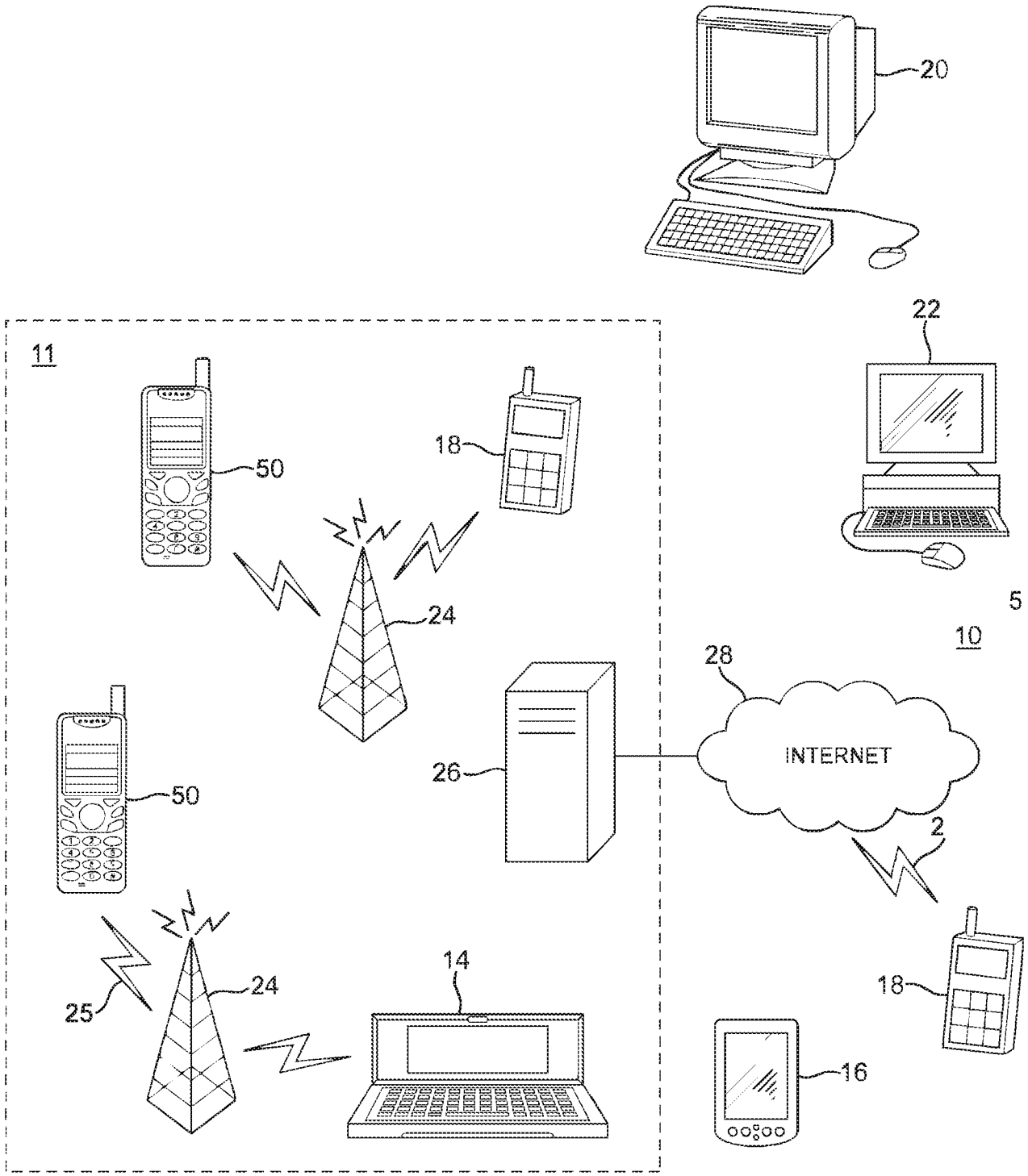

Fig. 3

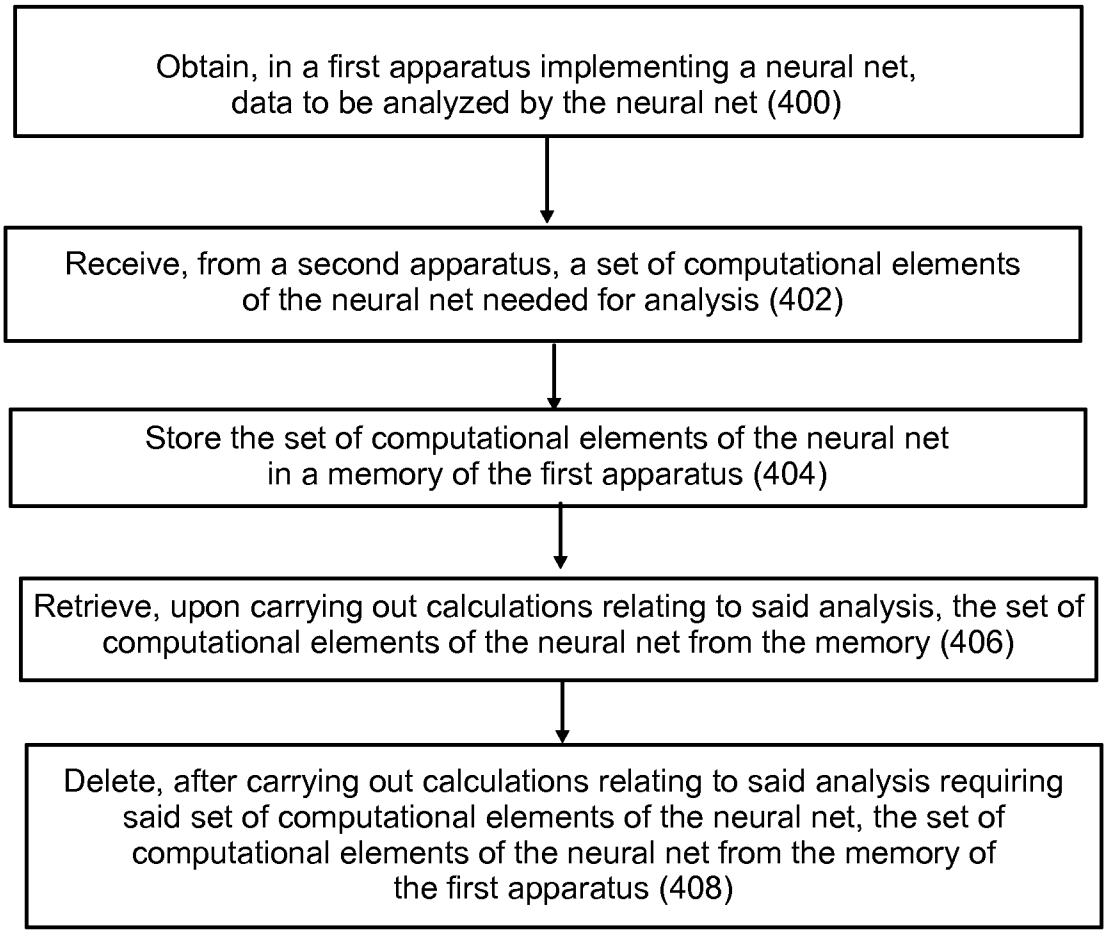
Obtain, in a first apparatus implementing a neural net, data to be analyzed by the neural net (400)
Receive, from a second apparatus, a set of computational elements of the neural net needed for analysis (402)
Store the set of computational elements of the neural net in a memory of the first apparatus (404)
Retrieve, upon carrying out calculations relating to said analysis, the set of computational elements of the neural net from the memory (406)
Delete, after carrying out calculations relating to said analysis requiring said set of computational elements of the neural net, the set of computational elements of the neural net from the memory of the first apparatus (408)

Fig. 4

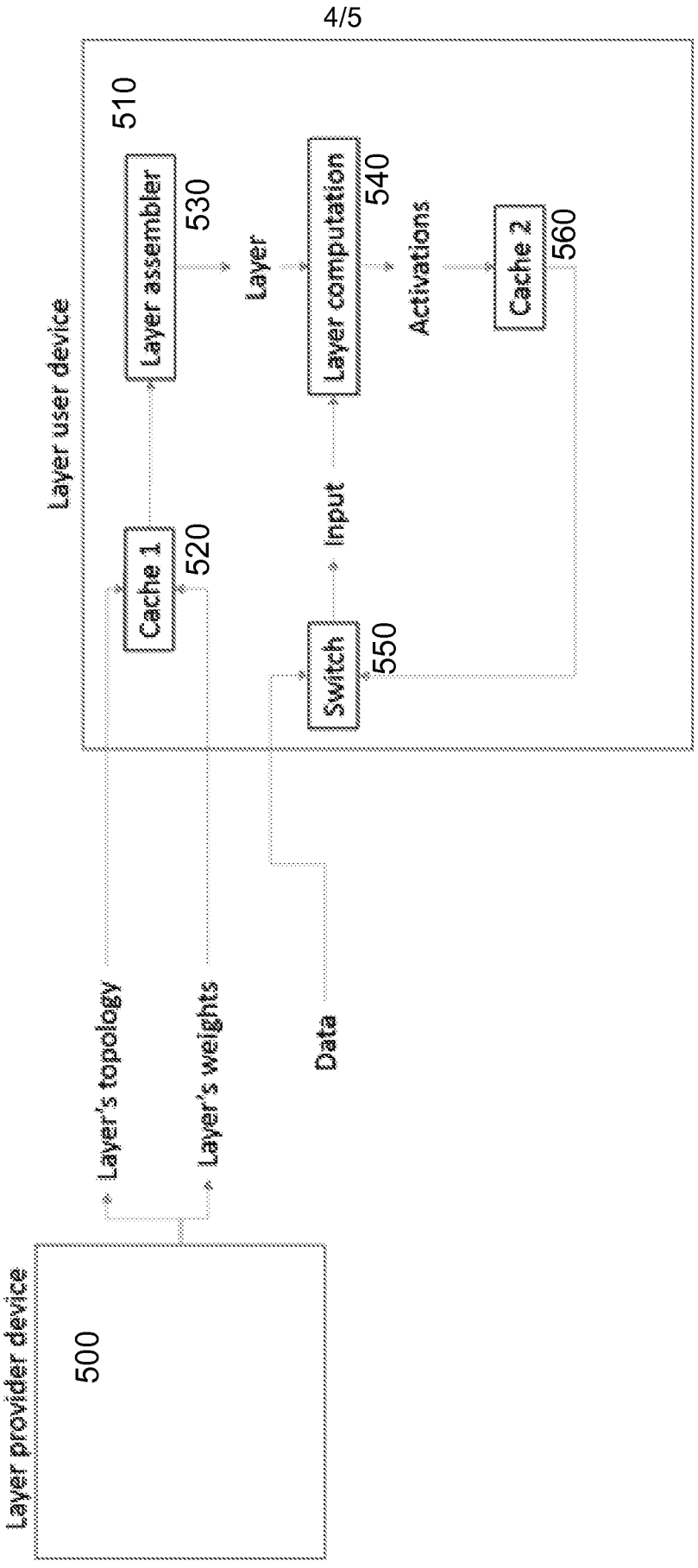
Layer provider device
500
Layer's topology
Layer's weights
Data
Layer user device
510
Layer assembler
530
Layer
Layer computation
540
Activations
Cache 2
560
Cache 1
520
Input
Switch
550

Fig.5

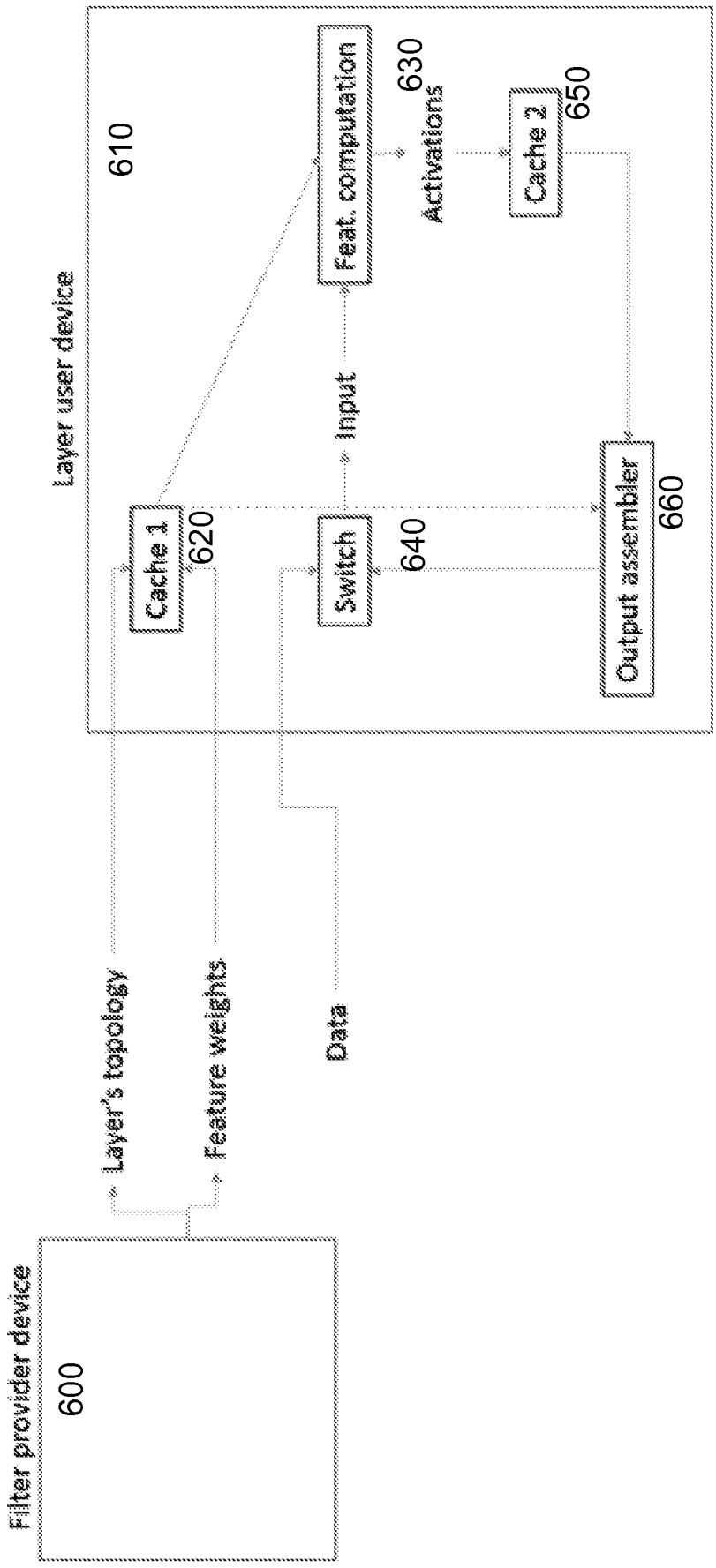
Filter provider device
600
Layer's topology
Feature weights
Data
Layer user device
610
Cache 1
620
Switch
640
Feat. computation
630
Activations
Cache 2
650
Input
Output assembler
660

Fig. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2019/050028

A. **CLASSIFICATION OF SUBJECT MATTER**

See extra sheet

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)

IPC: G06N, G06F, H04L, H04W

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

FI, SE, NO, DK

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

EPODOC, EPO-Internal full-text databases, Full-text translation databases from Asian languages, WPIAP, COMPDX, INSPEC, TDB, NPL, Internet

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ZOU, S. et al. Distributed Training Large-Scale Deep Architectures. arXiv.org, Ithaca, US: Cornell University Library, [online], 2017-08-10, pages 1-10, [retrieved on 2019-03-20]. Retrieved from <https://arxiv.org/pdf/1709.06622v1.pdf> ABSTRACT; page 2, column 1, lines 22-25; page 4, sections 2.2, 3; page 4, steps {(1), (3)-(4), (6)-(7)}; page 5, section 3.1.1; page 6, column 1; page 6, section 3.1.3; page 7, section 3.1.4; page 8, line 7; Figure 1	1-22
X	US 2016379111 A1 (BITTNER JR RAY A [US] et al.) 29 December 2016 (29.12.2016) abstract; paragraphs [0001], [0015]-[0017], [0021], [0028], [0042], [0052], [0057]; figures 1-3	1-22
A	US 2017221176 A1 (MUNTEANU MIHAI CONSTANTINE [RO] et al.) 03 August 2017 (03.08.2017) paragraph [0047]	

☒ 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	Date of mailing of the international search report
25 March 2019 (25.03.2019)	29 March 2019 (29.03.2019)

Name and mailing address of the ISA/FI
Finnish Patent and Registration Office
FI-00091 PRH, FINLAND
Facsimile No. +358 29 509 5328

Authorized officer
Mikko Flykt
Telephone No. +358 29 509 5000

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2019/050028

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2017124644 A1 (CAMBRICON TECH CO LTD [CN]) 27 July 2017 (27.07.2017) & abstract [online] EPOQUENET EPODOC & WPI & machine translation into English by the EPO [online] [retrieved 2018-09-03] claim 8	
A	US 2012011087 A1 (APARIN VLADIMIR [US]) 12 January 2012 (12.01.2012) paragraph [0006]	
A	AYTEKIN, C. et al. Memory-Efficient Deep Salient Object Segmentation Networks on Gridized Superpixels. arXiv.org, Ithaca, US: Cornell University Library, [online], 2017-12-27, pages 1-6, [retrieved on 2018-09-03]. Retrieved from <https://arxiv.org/abs/1712.09558v1> abstract	

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2019/050028

CLASSIFICATION OF SUBJECT MATTER

IPC
G06N 3/04 (2006.01)
H04W 8/24 (2009.01)
H04L 29/08 (2006.01)
H04W 4/50 (2018.01)
G06F 9/50 (2006.01)
G06N 20/00 (2019.01)

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2019/050028

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
US 2016379111 A1	29/12/2016	US 10140572 B2	27/11/2018
		CN 107735803 A	23/02/2018
		EP 3314543 A1	02/05/2018
		US 2019065954 A1	28/02/2019
		WO 2016210014 A1	29/12/2016
US 2017221176 A1	03/08/2017	CN 108701236 A	23/10/2018
		EP 3408798 A1	05/12/2018
		US 9665799 B1	30/05/2017
		WO 2017129325 A1	03/08/2017
WO 2017124644 A1	27/07/2017	CN 106991477 A	28/07/2017
		CN 108427990 A	21/08/2018
		US 2018330239 A1	15/11/2018
US 2012011087 A1	12/01/2012	US 8676734 B2	18/03/2014
		CN 102971754 A	13/03/2013
		CN 102971754 B	22/06/2016
		EP 2591449 A1	15/05/2013
		JP 2013534017 A	29/08/2013
		JP 2015167019 A	24/09/2015
		JP 2018014114 A	25/01/2018
		KR 20130036323 A	11/04/2013
		KR 101466251 B1	27/11/2014
		WO 2012006468 A1	12/01/2012