



(51) International Patent Classification:

G06N 3/063 (2006.01)

G06F 13/42 (2006.01)

(21) International Application Number:

PCT/US2019/066981

(22) International Filing Date:

17 December 2019 (17.12.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

16/453,735

26 June 2019 (26.06.2019)

US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:

US

16/453,735 (CON)

Filed on

26 June 2019 (26.06.2019)

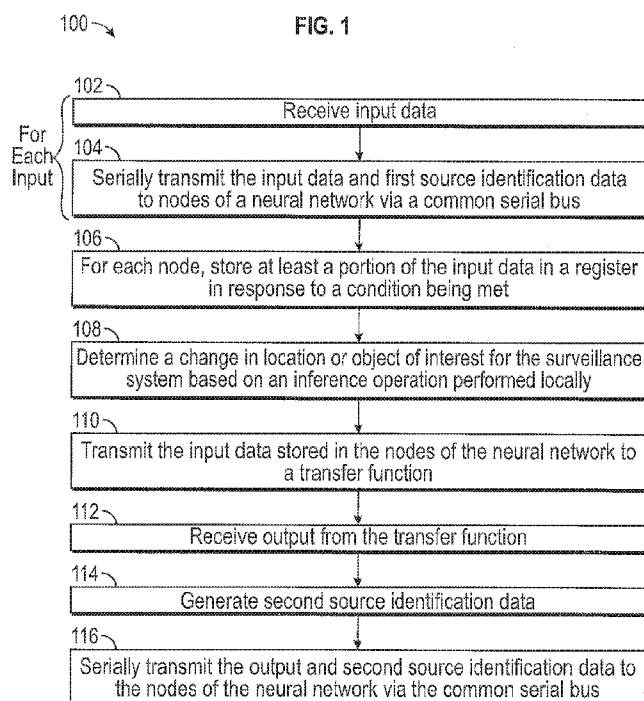
(71) Applicant: **WESTERN DIGITAL TECHNOLOGIES, INC.** [US/US]; 5601 Great Oaks Parkway, San Jose, California 95119 (US).

(72) Inventors: **OBUKHOV, Dmitry**; 5601 Great Oaks Parkway, San Jose, California 95119 (US). **SINGH, Anshuman**; 5601 Great Oaks Parkway, San Jose, California 95119 (US). **AWASTHI, Anuj**; 5601 Great Oaks Parkway, San Jose, California 95119 (US).

(74) Agent: **BARRY, Daniel, G.**; Barry IP Law, P.C., 27201 Puerta Real, Suite 300, Mission Viejo, CA 92691 (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, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(54) Title: NEURAL NETWORK ENGINE UTILIZING A SERIAL BUS



(57) Abstract: Embodiments of serial neural network configuration and processing via a common serial bus are disclosed. In some embodiments, the input data and source identification data is sent to nodes of the neural network serially. The nodes can determine whether the source identification data matches with an address for the node. If the address matches, the node can store the input data in its register for further processing. In some embodiments, the serial neural network engine can include a common serial bus that can broadcast data across multiple processor chips or cores.

(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, 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).

Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

TECHNICAL FIELD

[0001] The present disclosure relates to a neural network engine, and more particularly, to processing data serially in a neural network engine via a common serial bus.

BACKGROUND

[0002] Computer learning models can process large volumes of data. For example, a model may be implemented as an artificial neural network. Artificial neural networks are artificial in the sense that they are computational entities, inspired by biological neural networks but modified for implementation by computing devices. A neural network typically comprises an input layer, one or more hidden layer(s) and an output layer. The nodes in each layer connect to nodes in the subsequent layer and the strengths of these interconnections are typically learnt from data during the training process. Once trained, a neural network can be used for inference, that is, provided with new input data in order to predict the corresponding output.

[0003] Machine learning techniques, such as neural networks, are frequently being utilized by modern computing systems. These technologies can operate on large data sets and thus can require large amounts of storage space. However, current memory architectures do not allow for scalability of big data analysis. The present disclosure addresses these and other problems.

SUMMARY

[0004] The systems, methods, and devices of this disclosure each have several innovative aspects, no single one of which is solely responsible for the all of the desirable attributes disclosed herein.

[0005] A system of performing neural network computations can comprise: an input interface configured to receive input data for neural network processing; and a common serial bus configured to serially transmit the input data and a first source

identification data to a plurality of nodes of a neural network, wherein in response to the input data and the first source identification data being serially transmitted from the common serial bus, a count is incremented; wherein: each node of the plurality of nodes is configured to, in response to a condition being met, store at least a portion of the input data in a register of the node; and the neural network is configured to, in response to the count meeting a threshold: transmit, from the nodes that stored the input data, the input data to a transfer function; receive an output of the transfer function; add a second source identification data to the output of the transfer function; and transmit the output of the transfer function with the second source identification data to each node of the plurality of nodes via the common serial bus.

[0006] In the system of the preceding paragraph or any of the paragraphs herein, the system can further comprise a processor configured to load a pre-determined neural network with pre-determined weightings.

[0007] In the system of the preceding paragraph or any of the paragraphs herein, the system can further comprise a processor configured to: receive, via the input interface, one or more configuration parameters for the neural network; and dynamically configure weightings for the neural network based on the one or more configuration parameters.

[0008] In the system of the preceding paragraph or any of the paragraphs herein, the system can further comprise a processor that is configured to store data in response to the data corresponding to a particular source.

[0009] In the system of the preceding paragraph or any of the paragraphs herein, each node of the plurality of nodes can further be configured to store a plurality of data in response to the data corresponding to any of a plurality of predetermined sources.

[0010] In the system of the preceding paragraph or any of the paragraphs herein, the condition can include matching source data identifying a node to the first source identification data.

[0011] In the system of the preceding paragraph or any of the paragraphs herein, the common serial bus can be configured to transmit data to each node of the plurality of nodes without storing the data in memory.

[0012] A method of performing neural network computations can comprise: receiving input data for processing in a neural network comprising a plurality of nodes; transmitting the input data and first source identification data to the plurality of nodes via a common serial bus, wherein the common serial bus serially transmits the input data and the first source identification data to the plurality of nodes; for each node of the plurality of nodes, in response to a condition being met: storing at least a portion of the input data in a register of the node of the plurality of nodes; and incrementing a count; and in response to the count satisfying a threshold: transmitting, from the nodes that stored the input data, the input data to a transfer function; receiving an output of the transfer function; adding a second source identification data to the output of the transfer function; and transmitting the output of the transfer function with the second source identification data to each node of the plurality of nodes via the common serial bus.

[0013] In the method of the preceding paragraph or any of the paragraphs herein, the common serial bus can transmit the input data and the first source identification data to each node of the plurality of nodes without storing the input data or the first source identification data in memory.

[0014] In the method of the preceding paragraph or any of the paragraphs herein, the method can further comprise, by the common serial bus, broadcasting any data received to each of the plurality of nodes.

[0015] In the method of the preceding paragraph or any of the paragraphs herein, the plurality of nodes can include at least one node of an input layer of the neural network, at least one node of an output layer of the neural network, and at least one node of a hidden layer of the neural network.

[0016] In the method of the preceding paragraph or any of the paragraphs herein, the method can further comprise the input data can correspond to pixels of an image, and wherein the common serial bus transmitting the input data and the first source identification data serially includes transmitting each pixel of the image the input data serially to each node of the plurality of nodes.

[0017] In the method of the preceding paragraph or any of the paragraphs herein, the plurality of nodes can be disposed across a plurality of hardware processor chips or processor cores including a first and second hardware chip or processor core.

[0018] In the method of the preceding paragraph or any of the paragraphs herein, the method can further comprise, by the common serial bus, broadcasting the input data and the first source identification data to the nodes disposed on the first hardware chip or processor core and the second hardware chip or processor core.

[0019] In the method of the preceding paragraph or any of the paragraphs herein, the method can further comprise generating routing data, and broadcasting, by the common serial bus, the routing data with the input data and the first source identification data to each of the nodes of the plurality of nodes.

[0020] In the method of the preceding paragraph or any of the paragraphs herein, the method can further comprise, in response to receiving output of the transfer function, resetting the count.

[0021] In the method of the preceding paragraph or any of the paragraphs herein, the condition can include matching source data identifying a node to the first source identification data.

[0022] A system for performing neural network computations can comprise: means for receiving input data for processing by a neural network comprising a plurality of node; means for transmitting serially the input data and first source identification data to the plurality of nodes; and means for processing, for each node of the plurality of nodes, in response to a condition being met: storing at least a portion of the input data; and incrementing a count, means for processing further for, in response to the count meeting a threshold: receiving an output of a transfer function; and adding a second source identification data to the output of the transfer function, and means for transmitting further for serially transmitting the output of the transfer function with the second source identification data to each of the nodes of the plurality of nodes.

[0023] In the system of the preceding paragraph or any of the paragraphs herein, the condition can include matching source data identifying a node to the first source identification data.

[0024] In the system of the preceding paragraph or any of the paragraphs herein, the means for transmitting can be further for transmitting the input data and the first source identification data to each node of the neural network without storing the data in memory.

BRIEF DESCRIPTION OF DRAWINGS

[0025] Embodiments of various inventive features will now be described with reference to the following drawings and appendices. Throughout the drawings, reference numbers may be re-used to indicate correspondence between referenced elements. The drawings are provided to illustrate example embodiments described herein and are not intended to limit the scope of the disclosure.

[0026] Figure 1 depicts a flow diagram of serially processing data through a neural network engine.

[0027] Figure 2 depicts a block diagram of a serial neural network engine.

[0028] Figure 3 depicts a block diagram of a single node of the serial neural network engine.

[0029] Figure 4A depicts a block diagram of serial broadcasting and storage of input data.

[0030] Figure 4B depicts a block diagram of a neural network.

[0031] Figure 5A depicts a block diagram of serial broadcasting and storage of input data with two hidden layers.

[0032] Figure 5B depicts a block diagram of a neural network with two hidden layers.

[0033] Figure 6 depicts a block diagram of serially broadcasting across multiple chips.

DETAILED DESCRIPTION

[0034] While certain embodiments are described, these embodiments are presented by way of example only, and are not intended to limit the scope of protection. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms. Furthermore, various omissions, substitutions, and changes in the form of the methods and systems described herein may be made without departing from the scope of protection.

[0035] Existing neural networks require a lot of connections between nodes, especially for wide networks with lots of hidden layers, and many nodes within each layer.

As such, for each connection between nodes, existing neural networks may require a dynamic random access memory (DRAM) bus and store the data in DRAM intermittently as a means to pass data in the connection. In some implementations where nodes are implemented in certain logic configurations, each connection between the nodes may require a physical connection between the nodes. Moreover, networks typically require all input information upfront and at the same time. However, not all information is often available at the same time, for example, because a lot of other processing is performed on the device at different speeds and data may be passed via serial interfaces. Another issue with existing neural networks is that the processing done in parallel often requires multiple processors, which leads to high computational power requirements.

[0036] The systems and methods described herein can reduce and/or eliminate the drawbacks described above. A serial neural network can pass information to the nodes via a common serial bus. The data can be passed serially. For example, in certain embodiments, some or all of the nodes of the neural network can receive first data via the common serial bus. Each of some or all of the nodes can determine whether it is to store the first data in the node's register for future processing. For example, the node can assess source identification information corresponding to the first data and determine whether the source of the first data matches an address for that node. The serial neural network can receive second data and pass the second data to each node, and each node can determine whether the node should store the second data based on the source identification corresponding to the second data. Then when all of the information sufficient for processing is received, the neural network can process the data. The address for the node indicates one or more addresses that the node is to store data for. For example, all nodes in the first hidden layer of a neural network correspond to addresses for all nodes of the input layer, and thus these nodes only store data coming from the input layer. This data can include the first and second data described above, that can be passed serially to the nodes, and can indicate that the first and second data corresponds to an address of the input layer. Second hidden layer nodes correspond with addresses of the first hidden layer nodes. Thus, the second hidden layer nodes store data coming from nodes of the first hidden layer.

[0037] Generally, as described herein, neural networks, including but not limited to deep neural networks (“DNNs”), have multiple layers of nodes. Illustratively, a neural network may include an input layer, and output layer, and any number of intermediate, internal, or “hidden” layers between the input and output layers. The individual layers may include any number of separate nodes. Nodes of adjacent layers may be logically connected to each other, and each logical connection between the various nodes of adjacent layers may be associated with a respective weight. Conceptually, a node may be thought of as a computational unit that computes an output value as a function of a plurality of different input values. Nodes may be considered to be “connected” when the input values to the function associated with a current node include the output of functions associated with nodes in a previous layer, multiplied by weights associated with the individual “connections” between the current node and the nodes in the previous layer. Each node passes the result of this weighted sum of inputs through a non-linear function (such as a sigmoid, tangent hyperbolic, RELU, and/or the like) before sending its output to the next layer nodes.

[0038] The nodes in each layer connect to some (e.g., partially connected layers) or all (e.g., fully connected layers) of the nodes in the subsequent layer and the weights of these connections are typically learnt from data during the training process, for example through backpropagation in which the network parameters are tuned to produce expected outputs given corresponding inputs in labeled training data. Thus, the artificial neural network is an adaptive system that is configured to change its structure (e.g., the connection configuration and/or weights) based on information that flows through the network during training, and the weights of the hidden layers can be considered as an encoding of meaningful patterns in the data.

[0039] The parameters of a neural network can be set in a process referred to as training. For example, a neural network can be trained using training data that includes input data and the correct or preferred output of the model for the corresponding input data. Sets of individual input vectors (“mini-batches”) may be processed at the same time by using an input matrix instead of a single input vector, which may speed up training. The neural network can repeatedly process the input data, and the parameters (e.g., the weight matrices) of the neural network can be modified in what amounts to a trial-and-

error process until the model produces (or “converges” on) results substantially close to the correct or preferred output. The modification of weight values may be performed through a process referred to as “backpropagation.” Backpropagation includes determining the difference between the expected model output and the obtained model output, and then determining how to modify the values of some or all parameters of the model to reduce the difference between the expected model output and the obtained model output.

[0040] Although described within the context of neural networks, it will be appreciated that the training and/or inference concepts described herein could be applied to other machine learning models including decision trees, semi-supervised expectation maximization functions, and/or the like.

[0041] Various aspects of the disclosure will now be described with regard to certain examples and embodiments, which are intended to illustrate but not limit the disclosure. Although the examples and embodiments described herein will focus, for the purpose of illustration, specific calculations and algorithms, one of skill in the art will appreciate the examples are illustrative only, and are not intended to be limiting.

OVERVIEW

[0042] Figure 1 depicts a flow diagram 100 of serially processing data through a neural network engine according to one embodiment. The process 100 can be implemented by any system that can process a neural network. For example, the process 100, in whole or in part, can be implemented by the serial neural network engine or other computing system. Although any number of systems, in whole or in part, can implement the process 100, to simplify discussion, the process 100 will be described with respect to particular systems. Further, although the process 100 may be performed with respect to variations of systems comprising neural network engines, to simplify discussion, the process 100, will be described with respect to the serial neural network engine.

[0043] At block 102, for each input, the process 100 can receive input data. For example, the process 100 can receive data at a serial neural network engine interface and/or an input layer of the neural network. The data can include data for processing

through the neural network. For example, the process 100 can receive image data for inference operations to be performed in the neural network.

[0044] At block 104, for each input, process 100 can serially transmit the input data and/or first source identification data to nodes of the neural network via a common serial bus. The process 100 can receive data serially, such as receiving data serially at block 102, and a common serial bus of a serial neural network engine can transmit the data serially to each node of the neural network. For example, each pixel of the input data image can be received by the process 100, and the common serial bus can transmit each pixel data to each node with the source identification data.

[0045] At block 106, each node of the process 100 can receive each data serially with the source identification information. Each node can store at least a portion of the input data in a register in response to a condition being met. For example, each node can be configured to store data received if the source identification data matches a predetermined source. Although all nodes may receive the input data, only the nodes that are supposed to process the data will store the data for processing.

[0046] At block 108, the process 100 can increment a count for each data passed serially to each neural network. For example, after a first pixel is passed, the count is increased to four. For a four-pixel image, the process 100 can transmit each pixel serially, and the process 100 can increment the count up to four.

[0047] At block 110, the process 100 can transmit the input data stored in the nodes of the neural network to one or more transfer functions to perform analysis on the received data. In line with the example above, for a four-pixel image input, the process 100 can continue to increment the count after passing each pixel data serially. Once the count is four, the process 100 can initiate processing of the four-pixel data by transmitting the input data to a transfer function.

[0048] At block 112, the process 100 can receive output data from the transfer function. The transfer function can represent a function performed for one or more nodes and/or a function for a layer of the neural network. For example, the one or more transfer functions can include an inference operation performed for the first layer of the serial neural network engine.

[0049] At block 114, the process 100 can generate second source identification data. If a first node stores the input data at block 106, and the input data is processed by a transfer function, the process 100 can generate second source identification data including an address and/or other identification information for the first node.

[0050] At block 116, the process 100 can serially transmit the output and the second source identification data to the nodes of the neural network via a common serial bus. For example, after processing the first layer of the neural network via the one or more transfer functions at block 110 and 112, the process 100 can generate source identification information that indicates which node the output of the first layer came from. The process 100 can then transmit the output of the first layer serially to every node of the neural network. The nodes on the second layer of the neural network can store the input data in its register because the received input data is from the nodes of the first layer, indicated by the second source identification data.

SERIAL NEURAL NETWORK ENGINE

[0051] Figure 2 depicts a block diagram 200 of a serial neural network engine according to one embodiment. In the block diagram 200, the serial neural network engine can include an input layer 204, a serial bus 206 such as a common serial bus, a neural network 208, and an output layer 210. In some cases, the serial neural network of block diagram 200 can include more or less components. In some cases, the features described for Figure 2 can include one or more features for the serial neural network of block diagram described in other figures. For example, the common serial bus described in Figure 2 can include one or more features for the common serial bus described in other figures.

[0052] In some cases, the input data 202A, 202B, 202C, 202D (collectively referred to herein as input data 202) can include data for processing in a neural network. For example, the input data 202 can include a four-pixel image, a first pixel 202A, a second pixel 202B, a third pixel 202C, and a fourth pixel 202D. The pixels 202 can be sent to the input layer 204 of the neural network and/or an input interface for the serial neural network engine. The input layer 204 can receive the pixels 202 serially.

[0053] In some cases, the serial neural network engine can include a single interface for input data, where the data is received serially at the input layer. The serial neural network engine can pass the input data 202 serially to the serial bus 206. For example, a 2x2 pixel input can be broken into 4 messages, and each of the 4 messages can be sent to the serial bus 206 serially, one by one.

[0054] In some cases, the serial bus 206 can broadcast the input data 202 serially to each node of the neural network 208. Each node of some or all of the nodes of the neural network 208 can receive the input data 202 and check the corresponding source identification data with one or more addresses. If the address matches, the node stores the input data 202 in a register. That way, the nodes that are supposed to process the data that was sent from the serial bus 206 save the input data 202, for example, into the node's corresponding register. While other saving mechanisms such as volatile or non-volatile memory may be used in some embodiments, the description below will use the register implementation for illustration. In some cases, the serial bus 206 can broadcast a message with the input data 202 and source identification information serially.

[0055] In some cases, the node can save the input data 202 into a register if a destination identification data matches one or more addresses stored for the node. For example, a destination identification data can be sent with the input data 202 that indicates the destination node that the data is to be stored. In some cases, the destination identification data can be used by a node on a different chip or processor core to determine whether the destination identification data matches an address corresponding to the chip or processor core, as described in Figure 6.

[0056] In some cases, the serial neural network engine can determine to process a subset of data. For example, the input data can include 64 pixels. However, the serial neural network engine can process the input data after sending 60 pixels of data serially to each node. Advantageously, the serial neural network engine can process data faster on a subset of data that may be sufficient for neural network processing.

[0057] In some cases, the serial neural network engine can include one or more processors 214. The one or more processors 214 can execute a counter, such as a counter for each time data is passed through the serial bus 206. The one or more

processors 214 can determine that after 60 counts, the neural network 208 can process the data.

[0058] In some cases, the serial neural network engine can include an output layer 210 that can receive output data of the neural network. The output layer 210 can check whether the received data is for the output layer 210, such as by checking if the source identification data matches with one or more corresponding addresses for the output layer 210. The output later 210 can send the data to an actuator 212. The actuator 212 can include a one-bit actuator control. In some cases, the serial neural network engine can include a plurality of actuators.

[0059] In some cases, the serial neural network engine does not store data between layers in an external memory of the neural network, such as a dynamic random-access memory (DRAM). Instead, the serial neural network engine stores the input data in registers within the neural network 208. In some cases, the registers include temporary memory where the next set of data that is stored overwrites the previously stored data. As DRAM can consume a high amount of power, this can provide a significant power saving over implementations that require DRAM support.

[0060] In some cases, the serial neural network can load the addresses for each node, such that each node can check the source identification information corresponding to the input data received to see if the node should store the received input data in its register.

SINGLE NODE OF THE SERIAL NEURAL NETWORK ENGINE

[0061] Figure 3 depicts a block diagram 300 of a single node of the serial neural network engine, such as of the engine 200, according to one embodiment. In the block diagram 300, the node 301 of the serial neural network engine can connect to a serial bus 302, and/or can include a receiver 304, source addresses 306A, 306B, 306C, 306D (collectively referred to herein as source addresses 306), weights 308A, 3068B, 308C, 308D (collectively referred to herein as weights 308), saved values 310A, 310B, 3100C, 310D (collectively referred to herein as values 310), a ready state 312A, 312B, 312C, 312D (collectively referred to herein as ready states 312), a count 314 from a counter, a transfer function 318, a count threshold 316, source ID data 320, and a transmitter 322.

In some cases, a node 301 for the serial neural network of block diagram 300 can include more or fewer components. For example, the configuration of four elements of each of source address, weight, value and ready state shown in the figure is an example, as some nodes may have a different number of elements. In some cases, the features described for Figure 3 can include one or more features for the serial neural network of block diagram described in other figures. For example, the common serial bus and node configuration described in Figure 3 can include one or more features for the common serial bus engine described in Figure 2, and/or vice versa.

[0062] In some cases, the receiver 304 can receive data from the serial bus 302. The node 301 checks the received data with source addresses 306 with source identification data received from the serial bus 302. If the source identification data matches any of the source addresses 306, the node 301 can store the corresponding value 310 of the input data in a register, and the ready state 312 can be activated.

[0063] In some cases, the node 301 can increment a count 314 of the counter. The node 301 can have a count threshold 316, such as 60. The node 301 can process the stored values 310 in a transfer function 318 when all inputs are received and stored in the registers. The transfer function 318 can process the stored data with weights corresponding to the node 301.

[0064] In some cases, the output of the transfer function 318 is sent to a transmitter 322. The transmitter 322 can combine the output of the transfer function 318 with source identification data 320 to generate a message, and send the message to the serial bus to be transmitted to the nodes of the neural network.

[0065] In some cases, when the message is sent from the transmitter 322 to the serial bus 302, the count 314 is reset, such as being set to 0, and/or the ready states 312 can be turned off. In some cases, the ready states 312 can include a binary flag, such as on for a ready state and off for a not ready state.

SERIAL BROADCASTING AND STORAGE OF INPUT DATA

[0066] Figure 4A depicts a block diagram 400 of serial broadcasting and storage of input data according to one embodiment. In the block diagram 400, the serial neural network engine can include an input layer 412, a process layer 418, an output

layer 422, and a serial bus 416. The input layer 412 can include input nodes 414A, 414B, 414C, 414D (collectively referred to herein as input nodes 414). The process layer 418 can include process nodes 421A, 421B, 421C, 421D (collectively referred to herein as process nodes 421) that each have registers 420A, 420B, 420C, 420D, 420E, 420F, 420G, 420H, 420I, 420J, 420K, 420L, 420M, 420N, 420O, 420P (collectively referred to herein as registers 420). For example, the first node N1 421A includes four registers 420A, 420E, 420I, 420M. The output layer 422 can include registers 424A, 424B, 424C, 424D (collectively referred to herein as registers 424) and an output node 426.

[0067] Figure 4B depicts a block diagram 450 of a neural network according to one embodiment. The neural network of block diagram 450 can include four input nodes 452A, 452B, 452C, 452D (collectively referred to herein as input nodes 452), a hidden layer with hidden layer nodes 454A, 454B, 454C, 454D (collectively referred to herein as hidden layer nodes 454), and an output node 456.

[0068] In some cases, the serial neural network of block diagram 400 and/or the neural network of block diagram 450 can include more or less components. In some cases, the features described for Figures 4A and 4B can include one or more features for the serial neural network of block diagram described in other figures. For example, the common serial bus described in Figure 4A can include one or more features for the common serial bus described in other figures.

[0069] In some cases, input data I1, I2, I3, I4 are received by the input nodes 414. The serial bus 416 can receive each of the input data I1, I2, I3, I4 serially. The serial bus 416 can broadcast each of the input data I1, I2, I3, I4 serially to nodes connected to the serial bus, such as the process nodes 421. The first input I1 can be stored in the registers 420A, 420B, 420C, 420D corresponding to the process nodes 421. The nodes 421 can determine that the source identification data indicates that the data came from the input layer 412, and determine that the source identification data matches with an address for the node. For example, in relation to the block diagram 450 of Figure 4B, the hidden layer nodes 454 are connected to each of the input nodes 452. Thus, each of the hidden layer nodes 454 should store data in their corresponding registers that have source identification data to any of the input nodes 452. For Figure 4A, the same is repeated for the other input data I2, I3, I4, and stored in the process nodes 420.

[0070] In some cases, the serial neural network engine also broadcasts the input data to the output node 426. However, the output node 426 checks the source identification data and determines that the data came from the input layer 412, and thus does not store the data in its registers 424. In relation to Figure 4B, the output node 456 is connected with the hidden layer nodes 454, and thus, only stores data received from the hidden layer nodes 454, and not data received from the input nodes 452. In other words, the source identification data can be used to represent the connections in the neural network.

[0071] In some cases, the serial neural network engine increases a count for each time data is passed through the serial bus 416. The serial neural network engine can determine when a count meets a count threshold, such as when the count is 4, the data is processed via a transfer function for the nodes 421. The nodes 421 then transmit the output of the transfer function to the serial bus 416. The serial bus 416 then broadcasts the output of the transfer function to the output layer 422. The output layer 422 identifies that the source identification data corresponds to the hidden layer nodes 421 and thus stores the output of the transfer function from the hidden layer nodes into its registers 424. Once the count for the output layer meets a count threshold, such as 4 after each of the outputs of the transfer functions are saved in the registers 424 serially, then the output layer 422 transmits the output.

SERIAL BROADCASTING AND STORAGE OF INPUT DATA

[0072] Figure 5A depicts a block diagram 500 of serial broadcasting and storage of input data with two hidden layers according to one embodiment. In the block diagram 500, the serial neural network engine can include an input layer 502, a process layer 508, an output layer 512, and a serial bus 504. The input layer 502 can include input nodes 506A, 506B, 506C, 506D (collectively referred to herein as input nodes 506). The process layer 508 can include process nodes 511A-511H (collectively referred to herein as process nodes 511) each with corresponding registers 510A, 510B... 510AF (collectively referred to herein as registers 510). The output layer 512 can include registers 514A, 514B, 514C, 514D (collectively referred to herein as registers 514) and an output node 516.

[0073] Figure 5B depicts a block diagram 550 of a neural network with two hidden layers according to one embodiment. The neural network of block diagram 550 can include four input nodes 552A, 552B, 552C, 552D (collectively referred to herein as input nodes 552), a first hidden layer with first hidden layer nodes 554A, 554B, 554C, 554D (collectively referred to herein as first hidden layer nodes 554), a second hidden layer with second hidden layer nodes 556A, 556B, 556C, 556D (collectively referred to herein as second hidden layer nodes 556), and an output node 558.

[0074] In some cases, the serial neural network of block diagram 500 and/or the neural network of block diagram 550 can include more or less components. In some cases, the features described for Figures 5A and 5B can include one or more features for the serial neural network of block diagram described in other figures. For example, the common serial bus described in Figure 5A can include one or more features for the common serial bus described in other figures.

[0075] In some cases, the serial neural network of block diagram 500 receives the input data I1, I2, I3, I4, transmits the input data I1, I2, I3, I4 from the input layer 502 to the serial bus 504, is received by nodes corresponding to the first hidden layer 511A, 511B, 511C, 511D, is saved in their corresponding registers 510, and is processed through transfer functions of the nodes 511A, 511B, 511C, 511D, similar to Figure 4A.

[0076] In some cases, the serial neural network of block diagram 500 includes a second hidden layer. As shown in Figure 5B, the second hidden layer nodes 556 are connected to the first hidden layer nodes 554. Thus, data that is received by the second hidden layer nodes 556 with source identification data that corresponds to any of the first hidden layer nodes 554 are saved in corresponding registers. For example, the output of the transfer functions for nodes 511A, 511B, 511C, 511D (which correspond to the first hidden layer nodes) are sent to the serial bus 504, and the serial bus broadcasts these outputs to all nodes. However, only the nodes that correspond to the second hidden layer, nodes 511E, 511F, 511G, 511H, save the outputs in their registers, . Likewise, the output nodes of the output layer 512 store data that corresponds to source identification data of the second hidden layer nodes 511E, 511F, 511G, 511H.

SERIALLY BROADCASTING ACROSS MULTIPLE CHIPS

[0077] Figure 6 depicts a block diagram 600 of serially broadcasting across multiple chips according to one embodiment. In the block diagram 600, the serial neural network engine can include an input layer 602, a first hardware chip 606 (such as, an application specific integrated circuit (ASIC), field programmable gate array (FPGA), or the like), a second hardware chip 610 (such as, an ASIC, FPGA, or the like), an output layer 614, and a serial bus 618. The input layer 602 can include input nodes 604A, 604B, 604C, 604D (collectively referred to herein as input nodes 604). The first chip 606 can include first chip nodes 609A-609D (collectively referred to herein as first chip nodes 609) each with corresponding first chip registers 608A-608P (collectively referred to herein as first chip registers 608). The second chip 610 can include second chip nodes 613A-613D (collectively referred to herein as first chip nodes 613) each with corresponding second chip registers 612A-612P (collectively referred to herein as second chip registers 612). The output layer 614 can include registers 616A, 616B, 616C, 616D (collectively referred to herein as registers 616) and an output node 618. In other embodiments, a chip may be substituted by a processor core.

[0078] In some cases, the serial neural network of block diagram 600 can include more or less components. In some cases, the features described for Figure 6 can include one or more features for the serial neural network of block diagram described in other figures. For example, the common serial bus described in Figure 6 can include one or more features for the common serial bus described in other figures.

[0079] In some cases, a neural network can be distributed across multiple chips. For example, some nodes can be configured on a first chip 606 and other nodes can be configured on a second chip 610. The serial bus 618 can connect to each of the chips to facilitate sending the outputs of certain nodes to the nodes that are to facilitate receiving such outputs.

[0080] In some cases, the destination identification data can be used by a node on a different chip to determine whether the destination identification data matches an address corresponding to the chip. In some cases, the serial bus 618 can determine which chip to broadcast the data based on the destination identification data. For

example, the serial bus 618 can receive input data from the input nodes 604 along with destination identification data that indicates that the input data should be broadcast to the first chip 606. The serial bus 618 can selectively broadcast the input data only to the nodes of the first chip 606, and not to the second chip 610. Likewise, when output data is received from the nodes 609 of the first chip 606, the output data can be combined with destination identification data that indicates the output data should be sent to the second chip 610 for further processing, such as for a second hidden layer. In some cases, the destination identification data can indicate a plurality of destinations, such as a plurality of chips to broadcast the data to.

OTHER VARIATIONS

[0081] Those skilled in the art will appreciate that additional system components can be utilized, and disclosed system components can be combined or omitted. Although some embodiments describe video data transmission, disclosed systems and methods can be used for transmission of any type of data. The actual steps taken in the disclosed processes may differ from those shown in the figures. Depending on the embodiment, certain of the steps described above may be removed, others may be added. Accordingly, the scope of the present disclosure is intended to be defined only by reference to the appended claims.

[0082] While certain embodiments have been described, these embodiments have been presented by way of example only, and are not intended to limit the scope of the protection. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms. Furthermore, various omissions, substitutions and changes in the form of the methods and systems described herein may be made without departing from the spirit of the protection. The accompanying claims and their equivalents are intended to cover such forms or modifications as would fall within the scope and spirit of the protection. For example, the systems and methods disclosed herein can be applied to hard disk drives, hybrid hard drives, and the like. In addition, other forms of storage (such as, DRAM or SRAM, battery backed-up volatile DRAM or SRAM devices, EPROM, EEPROM memory, etc.) may additionally or alternatively be used. As another example, the various components illustrated in the figures may be implemented as software and/or

firmware on a processor, ASIC/FPGA, or dedicated hardware. Also, the features and attributes of the specific embodiments disclosed above may be combined in different ways to form additional embodiments, all of which fall within the scope of the present disclosure.

[0083] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of this disclosure. As used herein, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will further be understood that the terms “comprises” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof. Further, references to “a method” or “an embodiment” throughout are not intended to mean the same method or same embodiment, unless the context clearly indicates otherwise.

[0084] The corresponding structures, materials, acts, and equivalents of all means or step plus function elements in the claims below are intended to include any structure, material, or act for performing the function in combination with other claimed elements as specifically claimed. The description of the various embodiments of the present disclosure has been presented for purposes of illustration and description, but is not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of this disclosure. The example embodiments were chosen and described in order to best explain the principles of this disclosure and the practical application, and to enable others of ordinary skill in the art to understand this disclosure for various embodiments with various modifications as are suited to the particular use contemplated.

[0085] Although the present disclosure provides certain preferred embodiments and applications, other embodiments that are apparent to those of ordinary skill in the art, including embodiments which do not provide all of the features and advantages set forth herein, are also within the scope of this disclosure. Accordingly, the scope of the present disclosure is intended to be defined only by reference to the appended claims.

WHAT IS CLAIMED IS:

1. A system of performing neural network computations, the system comprising:

an input interface configured to receive input data for neural network processing;
and

a common serial bus configured to serially transmit the input data and a first source identification data to a plurality of nodes of a neural network, wherein in response to the input data and the first source identification data being serially transmitted from the common serial bus, a count is incremented;

wherein:

each node of the plurality of nodes is configured to, in response to a condition being met, store at least a portion of the input data in a register of the node; and

the neural network is configured to, in response to the count meeting a threshold:

transmit, from the nodes that stored the input data, the input data to a transfer function;

receive an output of the transfer function;

add a second source identification data to the output of the transfer function; and

transmit the output of the transfer function with the second source identification data to each node of the plurality of nodes via the common serial bus.

2. The system of claim 1 further comprising a processor configured to load a pre-determined neural network with pre-determined weightings.

3. The system of any one of claims 1-2 further comprising a processor configured to:

receive, via the input interface, one or more configuration parameters for the neural network; and

dynamically configure weightings for the neural network based on the one or more configuration parameters.

4. The system of any one of claims 1-3 further comprising a processor that is configured to store data in response to the data corresponding to a particular source.

5. The system of any one of claims 1-4, wherein each node of the plurality of nodes is further configured to store a plurality of data in response to the data corresponding to any of a plurality of predetermined sources.

6. The system of any one of claims 1-5, wherein the condition includes matching source data identifying a node to the first source identification data.

7. The system of any one of claims 1-6, wherein the common serial bus is configured to transmit data to each node of the plurality of nodes without storing the data in memory.

8. A method of performing neural network computations, the method comprising:

receiving input data for processing in a neural network comprising a plurality of nodes;

transmitting the input data and first source identification data to the plurality of nodes via a common serial bus, wherein the common serial bus serially transmits the input data and the first source identification data to the plurality of nodes;

for each node of the plurality of nodes, in response to a condition being met:

storing at least a portion of the input data in a register of the node of the plurality of nodes; and

incrementing a count; and

in response to the count satisfying a threshold:

transmitting, from the nodes that stored the input data, the input data to a transfer function;

receiving an output of the transfer function;

adding a second source identification data to the output of the transfer function; and

transmitting the output of the transfer function with the second source identification data to each node of the plurality of nodes via the common serial bus.

9. The method of claim 8, wherein the common serial bus transmits the input data and the first source identification data to each node of the plurality of nodes without storing the input data or the first source identification data in memory.

10. The method of any one of claims 8-9, further comprising, by the common serial bus, broadcasting any data received to each of the plurality of nodes.

11. The method of any one of claims 8-10, wherein the plurality of nodes include at least one node of an input layer of the neural network, at least one node of an output layer of the neural network, and at least one node of a hidden layer of the neural network.

12. The method of any one of claims 8-11, wherein the input data corresponds to pixels of an image, and wherein the common serial bus transmitting the input data and the first source identification data serially includes transmitting each pixel of the image the input data serially to each node of the plurality of nodes.

13. The method of any one of claims 8-12, wherein the plurality of nodes are disposed across a plurality of hardware processor chips or processor cores including a first and second hardware chip or processor core.

14. The method of claim 13, further comprising, by the common serial bus, broadcasting the input data and the first source identification data to the nodes disposed on the first hardware chip or processor core and the second hardware chip or processor core.

15. The method of any one of claims 8-14, further comprising generating routing data, and broadcasting, by the common serial bus, the routing data with the input data and the first source identification data to each of the nodes of the plurality of nodes.

16. The method of any one of claims 8-15, wherein the method further comprises, in response to receiving output of the transfer function, resetting the count.

17. The method of any one of claims 8-16, wherein the condition includes matching source data identifying a node to the first source identification data.

18. A system for performing neural network computations, the system comprising:

means for receiving input data for processing by a neural network comprising a plurality of node;

means for transmitting serially the input data and first source identification data to the plurality of nodes; and

means for processing, for each node of the plurality of nodes, in response to a condition being met:

storing at least a portion of the input data; and

incrementing a count,

means for processing further for, in response to the count meeting a threshold:

receiving an output of a transfer function; and

adding a second source identification data to the output of the transfer function, and

means for transmitting further for serially transmitting the output of the transfer function with the second source identification data to each of the nodes of the plurality of nodes.

19. The system of claim 18, wherein the condition includes matching source data identifying a node to the first source identification data.

20. The system of any one of claims 18-19, wherein the means for transmitting is further for transmitting the input data and the first source identification data to each node of the neural network without storing the data in memory.

1/8

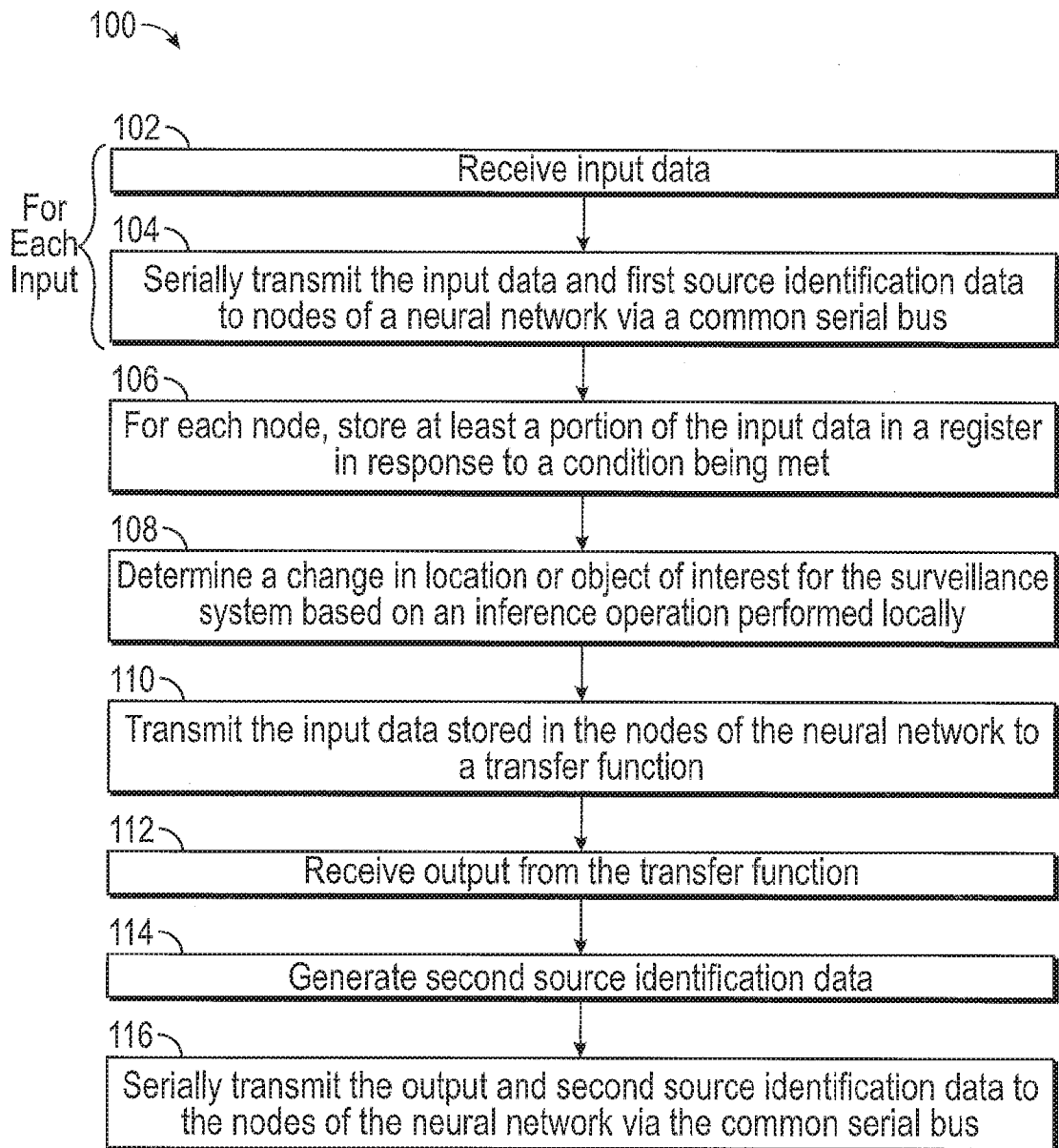


FIG. 1

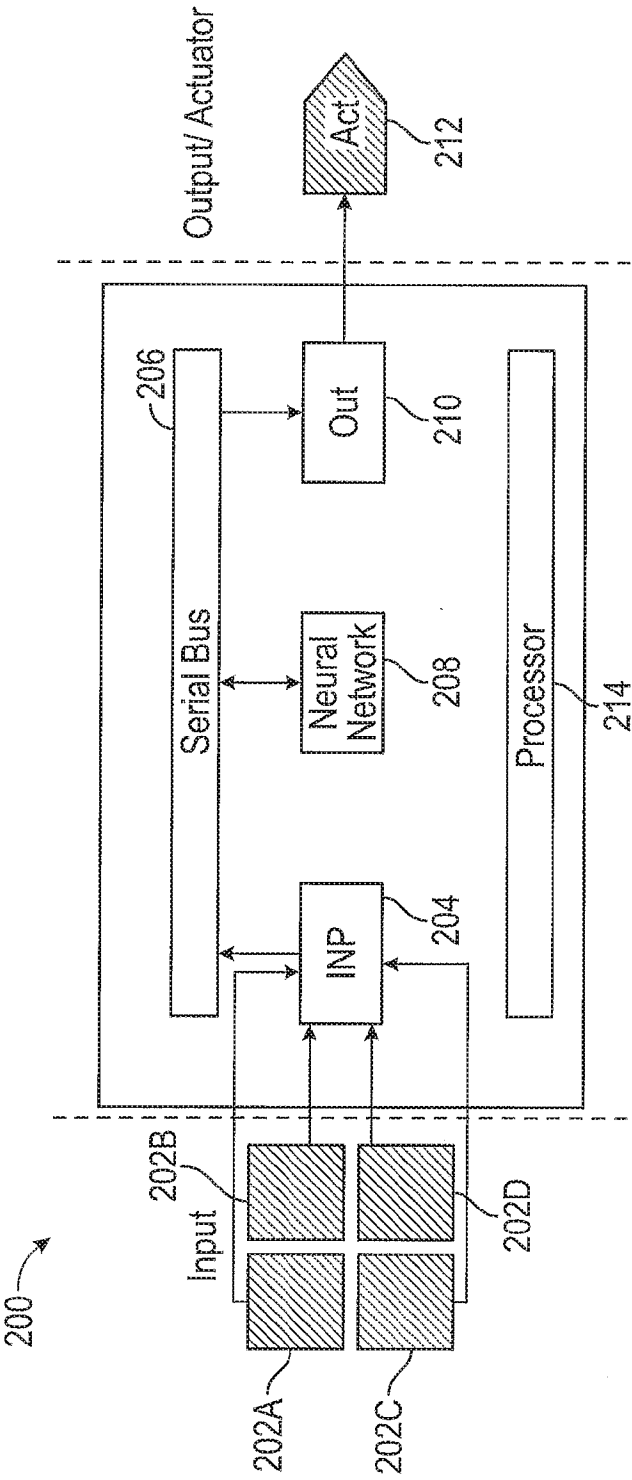


FIG. 2

3/8

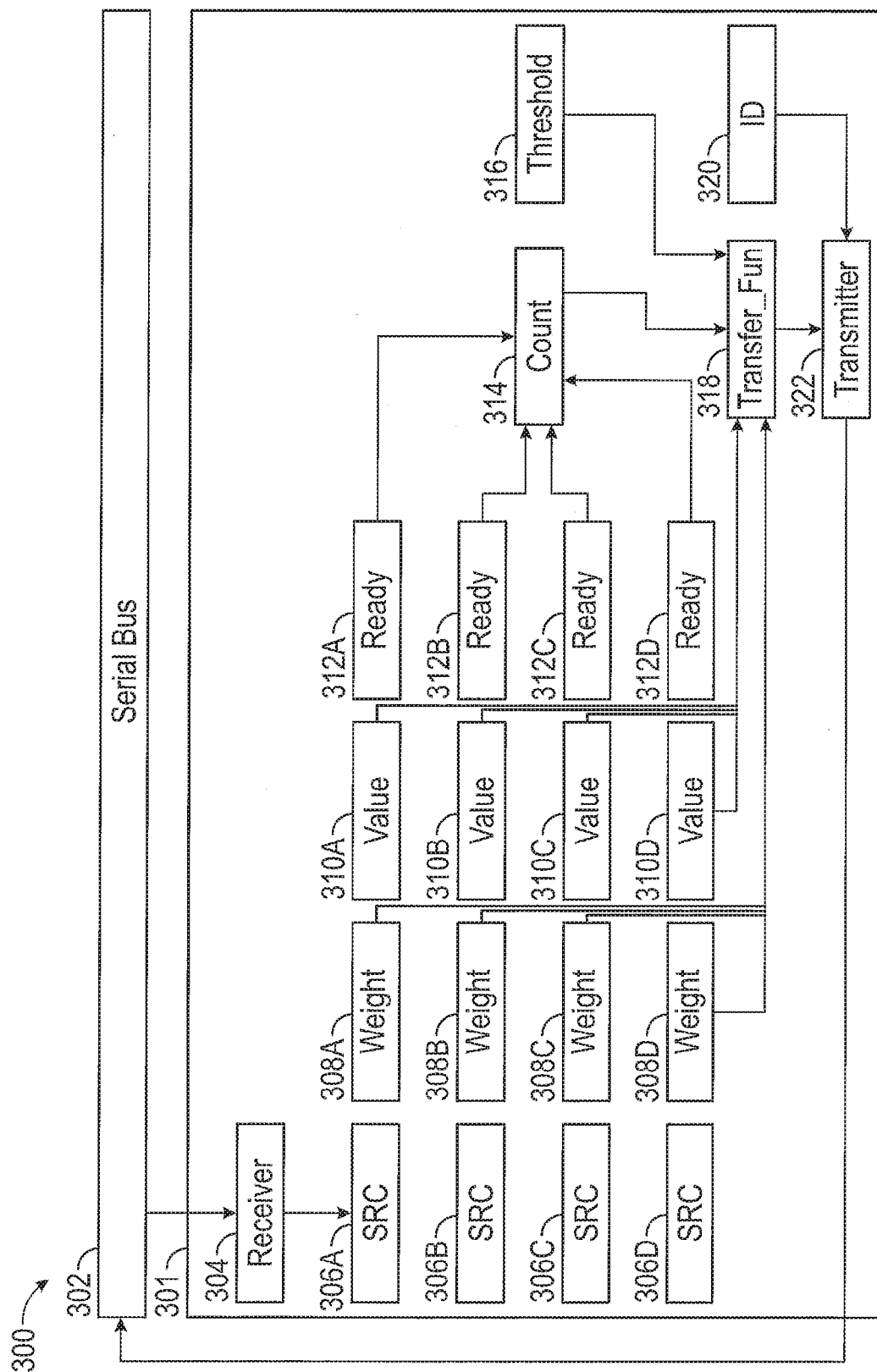


FIG. 3

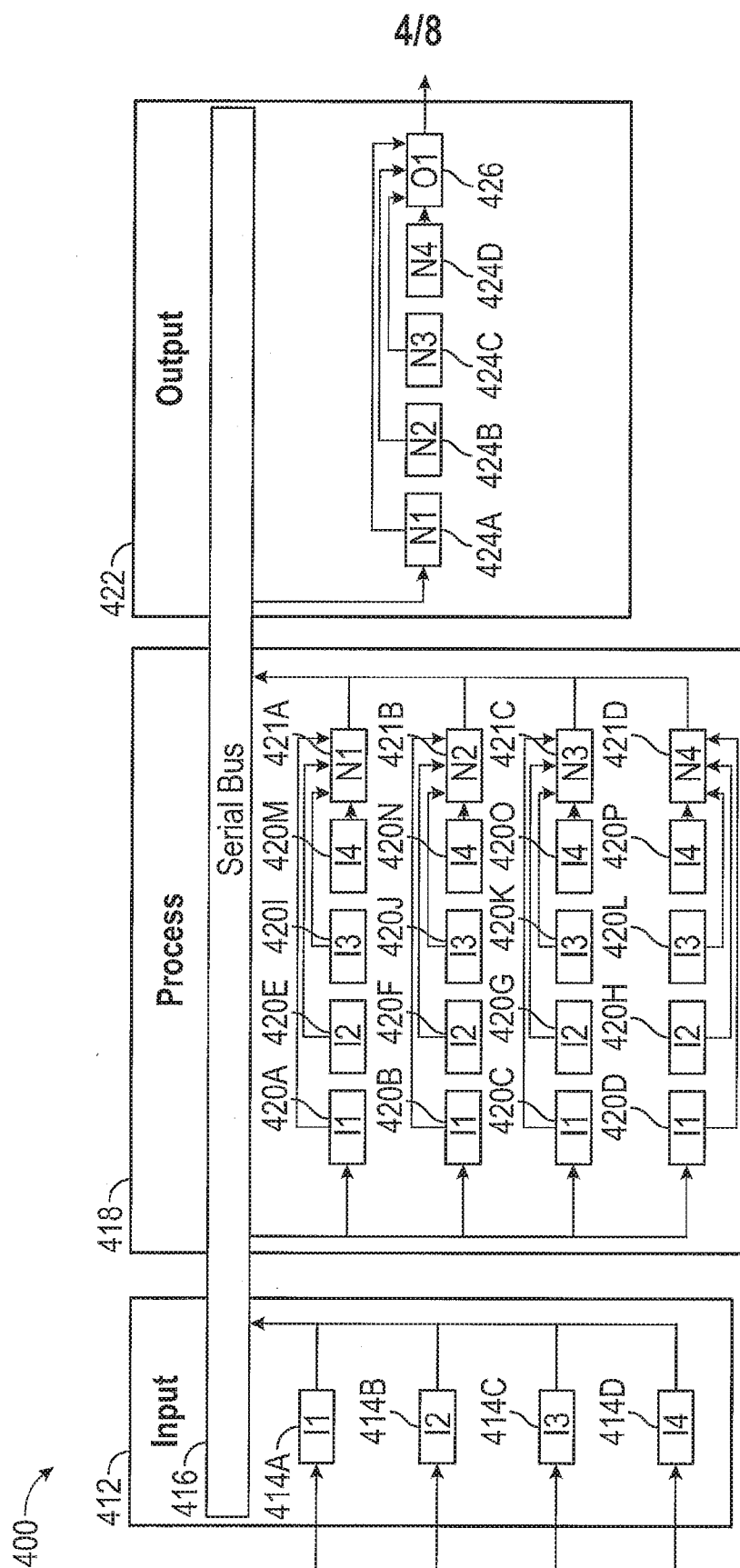


FIG. 4A

5/8

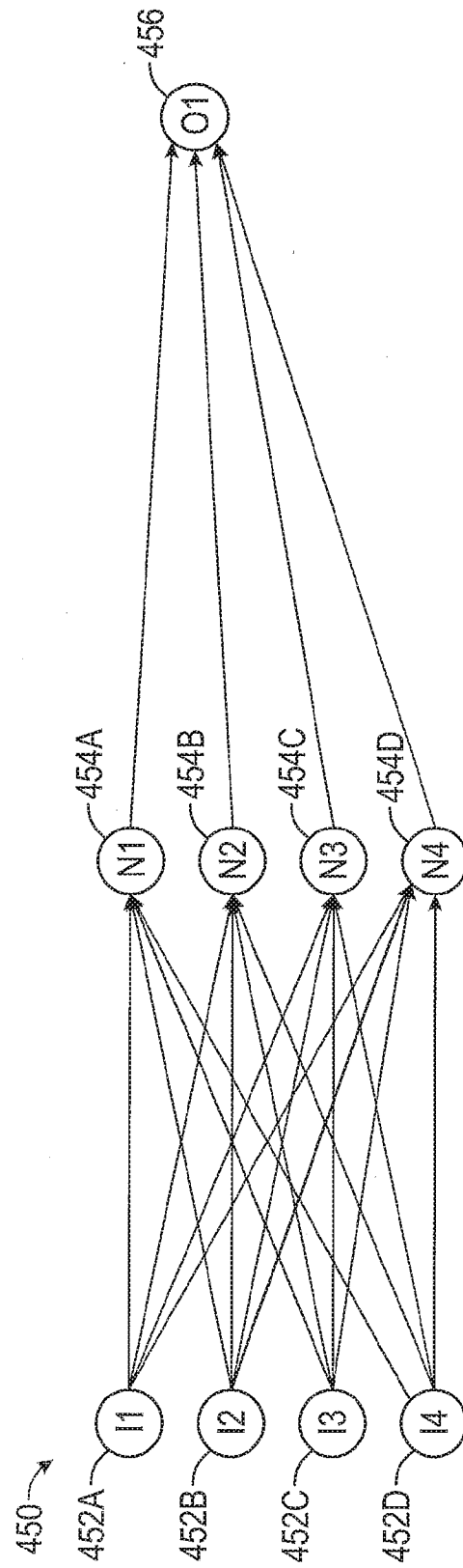


FIG. 4B

6/8

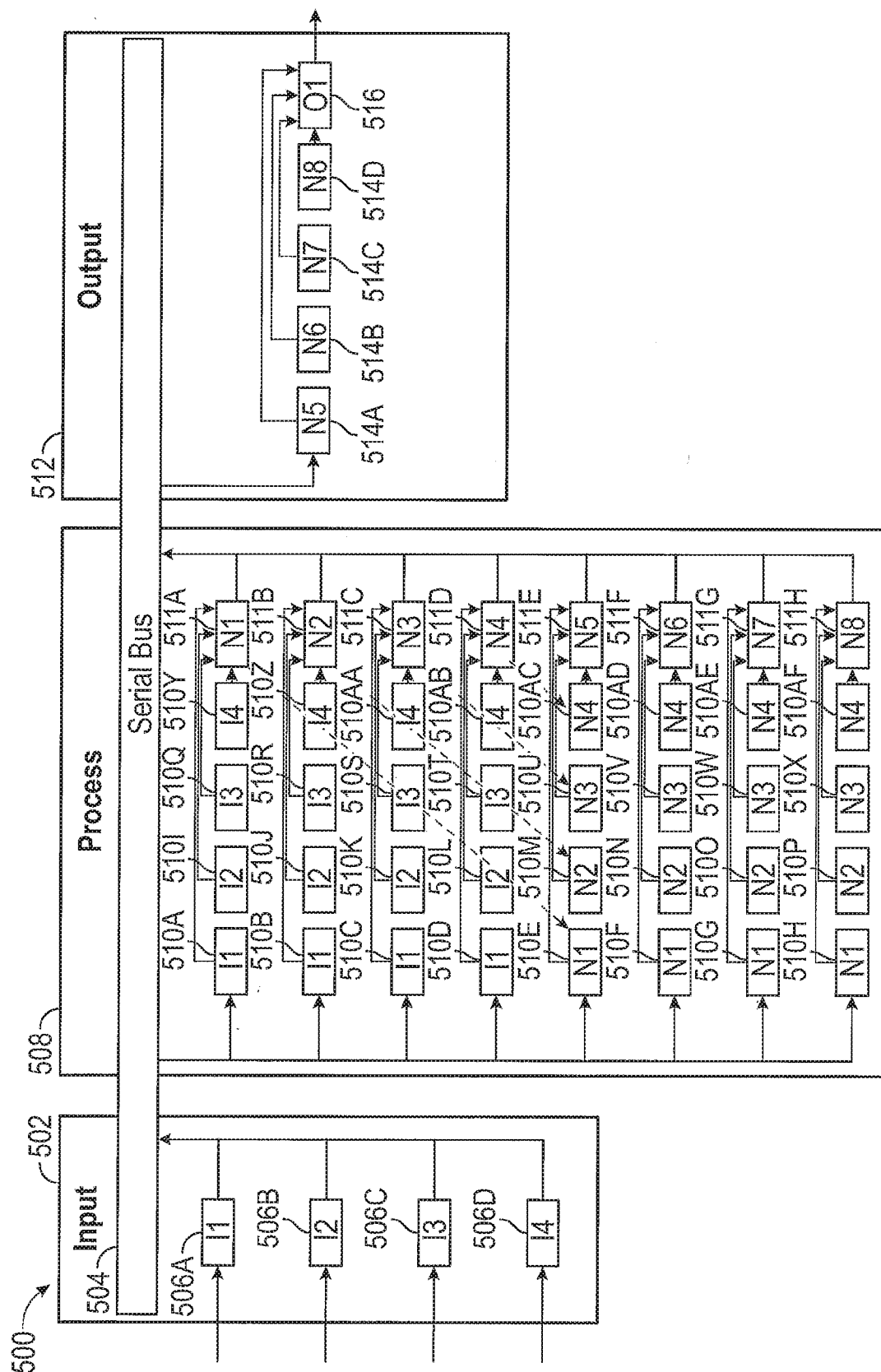


FIG. 5A

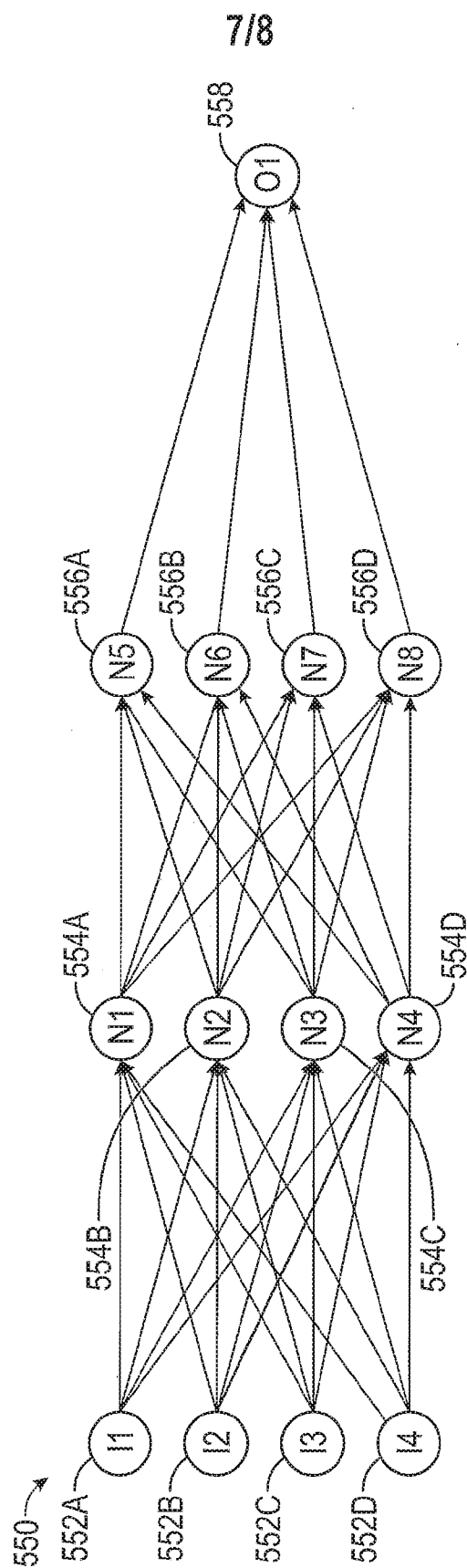


FIG. 5B

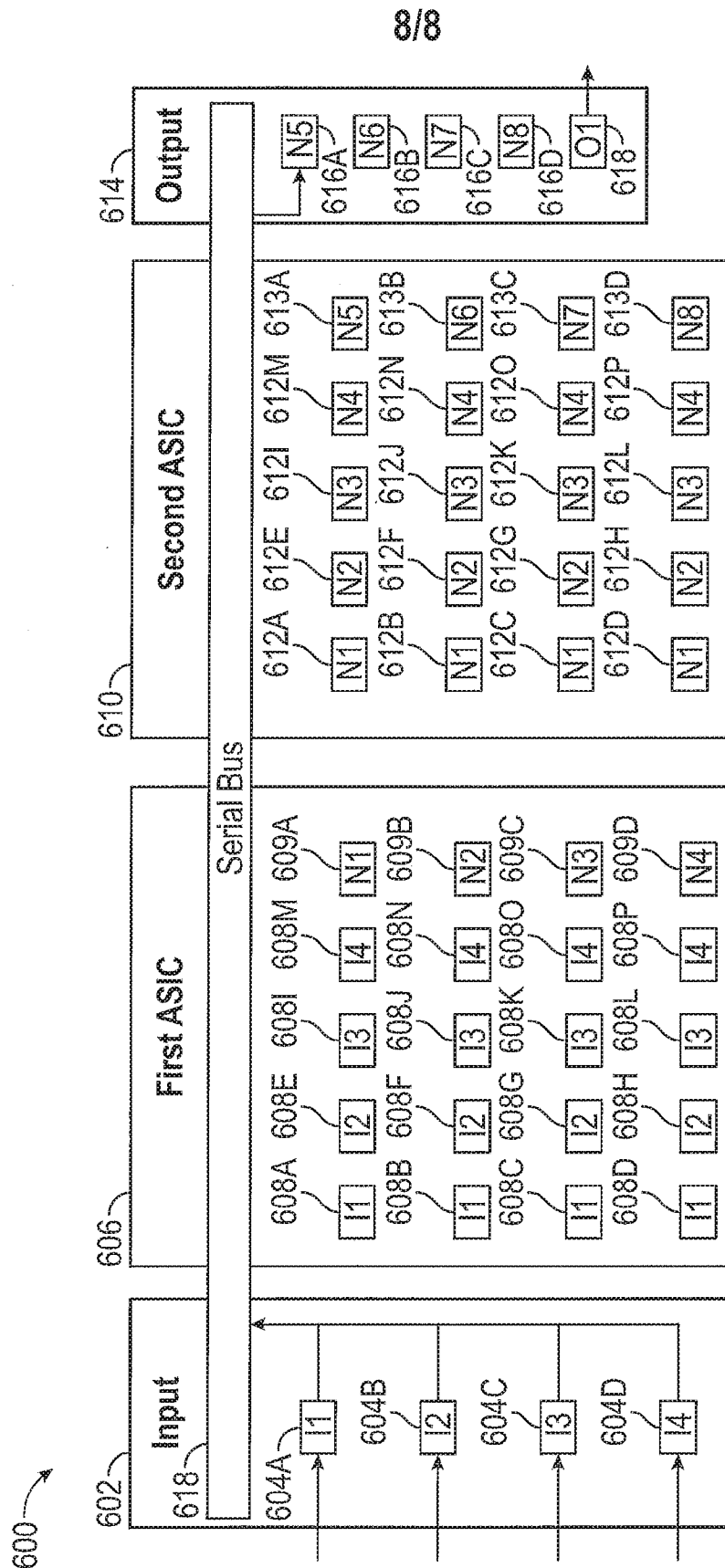


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2019/066981

A. CLASSIFICATION OF SUBJECT MATTER IPC (20200101) G06N 3/063, G06F 13/42 CPC (20130101) G06N 3/063, G06F 13/4282 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 (20200101) G06N 3/063, G06F 13/42 CPC (20130101) G06N 3/063, G06F 13/4282 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) Databases consulted: Esp@cenet, Google Patents, Orbit Search terms used: neural network, architecture, serial bus,		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5337395 A Vassiliadis et al. 09 Aug 1994 (1994/08/09) the whole document	1-20
A	US 2004181497 A1 Dodgson 16 Sep 2004 (2004/09/16) the whole document	1-20
A	US 2019050717 A1 Temam et al. 14 Feb 2019 (2019/02/14) the whole document	1-20
☐ 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 "D" document cited by the applicant in the international application "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 08 Mar 2020		Date of mailing of the international search report 09 Mar 2020
Name and mailing address of the ISA: Israel Patent Office Technology Park, Bldg.5, Malcha, Jerusalem, 9695101, Israel Email address: pctoffice@justice.gov.il		Authorized officer ROASH Yoaela Telephone No. 972-73-3927256

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2019/066981

Patent document cited search report			Publication date	Patent family member(s)			Publication Date
US	5337395	A	09 Aug 1994	US	5337395	A	09 Aug 1994
				EP	0459222	A2	04 Dec 1991
				EP	0459222	A3	30 Mar 1994
				EP	0484479	A1	13 May 1992
				EP	0484479	A4	13 Apr 1994
				EP	0484506	A1	13 May 1992
				EP	0484506	A4	23 Mar 1994
				EP	0484507	A1	13 May 1992
				EP	0484507	A4	15 Jul 1992
				EP	0484522	A1	13 May 1992
				EP	0484522	A4	23 Mar 1994
				EP	0486635	A1	27 May 1992
				EP	0486635	A4	23 Mar 1994
				EP	0486684	A1	27 May 1992
				EP	0486684	A4	23 Mar 1994
				EP	0564847	A2	13 Oct 1993
				EP	0564847	A3	20 Jul 1994
				JP	H04507027	A	03 Dec 1992
				JP	2502867	B2	29 May 1996
				JP	H04507025	A	03 Dec 1992
				JP	2575565	B2	29 Jan 1997
				JP	H0675986	A	18 Mar 1994
				JP	2647327	B2	27 Aug 1997
				JP	H05500429	A	28 Jan 1993
				JP	2663995	B2	15 Oct 1997
				JP	H04505824	A	08 Oct 1992
				JP	2663996	B2	15 Oct 1997
				JP	H04506879	A	26 Nov 1992
				JP	2666830	B2	22 Oct 1997
				JP	H04507026	A	03 Dec 1992

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2019/066981

Patent document cited search report	Publication date	Patent family member(s)	Publication Date
		JP 2746350 B2	06 May 1998
		JP H04232562 A	20 Aug 1992
		JP H07117948 B2	18 Dec 1995
		US 5065339 A	12 Nov 1991
		US 5146420 A	08 Sep 1992
		US 5146543 A	08 Sep 1992
		US 5148515 A	15 Sep 1992
		US 5243688 A	07 Sep 1993
		US 5251287 A	05 Oct 1993
		US 5325464 A	28 Jun 1994
		US 5329611 A	12 Jul 1994
		US 5483620 A	09 Jan 1996
		US 5509106 A	16 Apr 1996
		US 5517596 A	14 May 1996
		US 5542026 A	30 Jul 1996
		US 5577262 A	19 Nov 1996
		US 5612908 A	18 Mar 1997
		US 5613044 A	18 Mar 1997
		US 5617512 A	01 Apr 1997
		US 6405185 B1	11 Jun 2002
		WO 9118347 A1	28 Nov 1991
		WO 9118348 A1	28 Nov 1991
		WO 9118349 A1	28 Nov 1991
		WO 9118350 A1	28 Nov 1991
		WO 9118351 A1	28 Nov 1991
		WO 9201257 A1	23 Jan 1992
US 2004181497 A1	16 Sep 2004	US 2004181497 A1	16 Sep 2004
		GB 0303707 D0	19 Mar 2003
		GB 2398711 A	25 Aug 2004
		GB 2398711 B	12 Apr 2006

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2019/066981

Patent document cited search report			Publication date		Patent family member(s)		Publication Date	
US	2019050717	A1	14 Feb 2019	GB	0602348	D0	15 Mar 2006	
				GB	2419792	A	03 May 2006	
				GB	2419792	B	31 Jan 2007	
				KR	20040074630	A	25 Aug 2004	
				US	2019050717	A1	14 Feb 2019	
				US	10504022	B2	10 Dec 2019	
				CN	109389214	A	26 Feb 2019	
				DE	102018119513	A1	14 Feb 2019	
				DE	202018104609	U1	29 Nov 2018	
				GB	201812820	D0	19 Sep 2018	
				GB	2568776	A	29 May 2019	
				KR	20200005638	A	15 Jan 2020	
				TW	201911140	A	16 Mar 2019	
				TW	I679590	B	11 Dec 2019	
				US	2020005128	A1	02 Jan 2020	
				WO	2019032808	A1	14 Feb 2019	