(12) UK Patent (19) GB (11) 2587021 (13) B

(45) Date of B Publication 13.09.2023

(54) Title of the Invention: Physical implementation of artificial neural networks

(51) INT CL: ***G06N 3/065*** (2023.01) ***G06N 3/045*** (2023.01)

(21) Application No: **1913275.2**

(22) Date of Filing: **13.09.2019**

(43) Date of A Publication **17.03.2021**

(56) Documents Cited:
JOKSAS et al., "Committee Machines - A Universal Method to Deal with Non-Idealities in RRAM-Based Neural Networks", 14 September 2019, available at https://arxiv.org/abs/1909.06658
MEHONIC et al., Frontiers in Neuroscience, 2019, volume 13 page 593, "Simulation of Inference Accuracy Using Realistic RRAM Devices"
LI et al., "2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC", 2015, IEEE, "Merging the Interface: Power, Area and Accuracy Co-optimization for RRAM Crossbar-based Mixed-Signal Computing System"
AGRAWAL et al., "X-CHANGR: Changing Memristive Crossbar Mapping for Mitigating Line-Resistance Induced Accuracy Degradation in Deep Neural Networks", 26 June 2019, available at https://arxiv.org/abs/1907.00285
PAYVAND et al., "A neuromorphic systems approach to in-memory computing with non-ideal memristive devices: From mitigation to exploitation", 13 July 2018, available at https://arxiv.org/abs/1807.05128

(58) Field of Search:
As for published application 2587021 A viz:
INT CL **G06G, G06N, G11C**
Other: **WPI, EPODOC, INSPEC**
updated as appropriate

Additional Fields
Other: **None**

(72) Inventor(s):
Adnan Mehonic
Dovydas Joksas
Anthony J Kenyon

(73) Proprietor(s):
UCL Business Ltd.
The Network Building, 97 Tottenham Court Road,
LONDON, W1T 4TP, United Kingdom

(74) Agent and/or Address for Service:
D Young & Co LLP
120 Holborn, LONDON, EC1N 2DY, United Kingdom

10 12 20

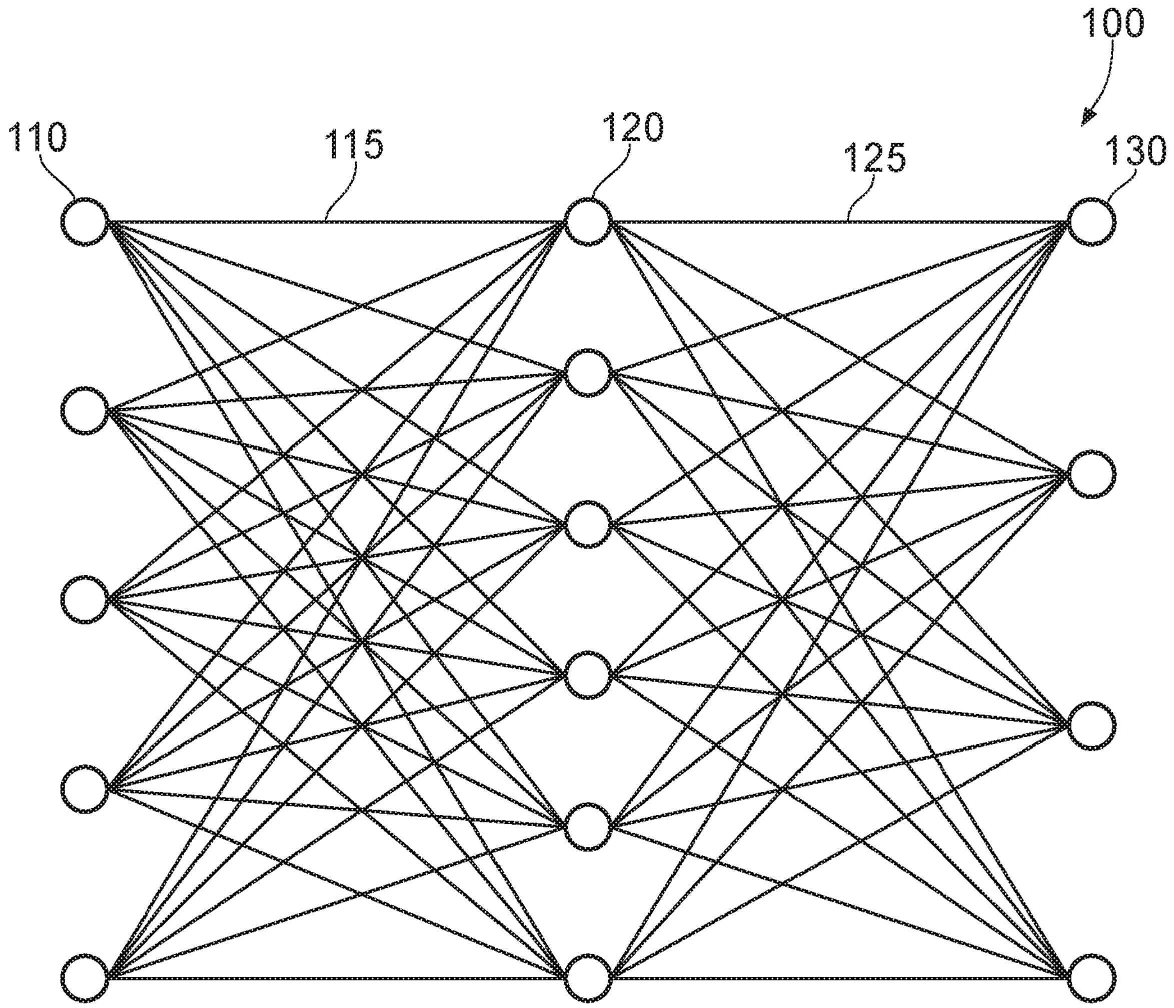

FIG. 1

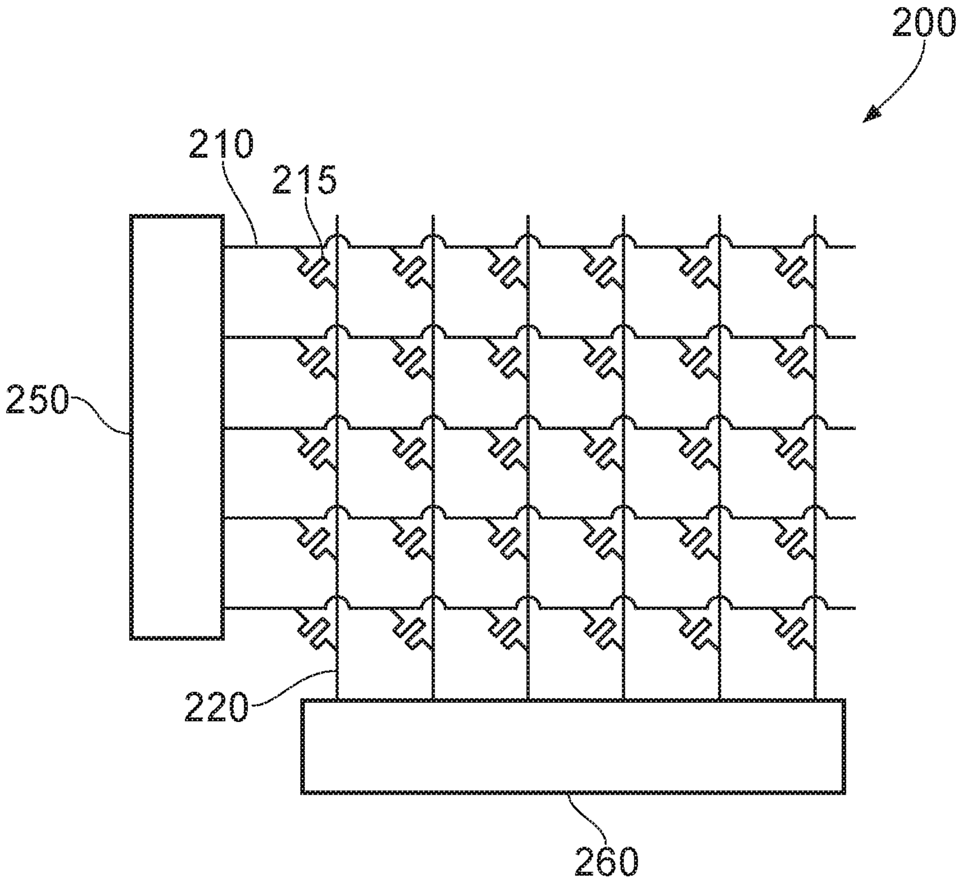

FIG. 2

10 12 20

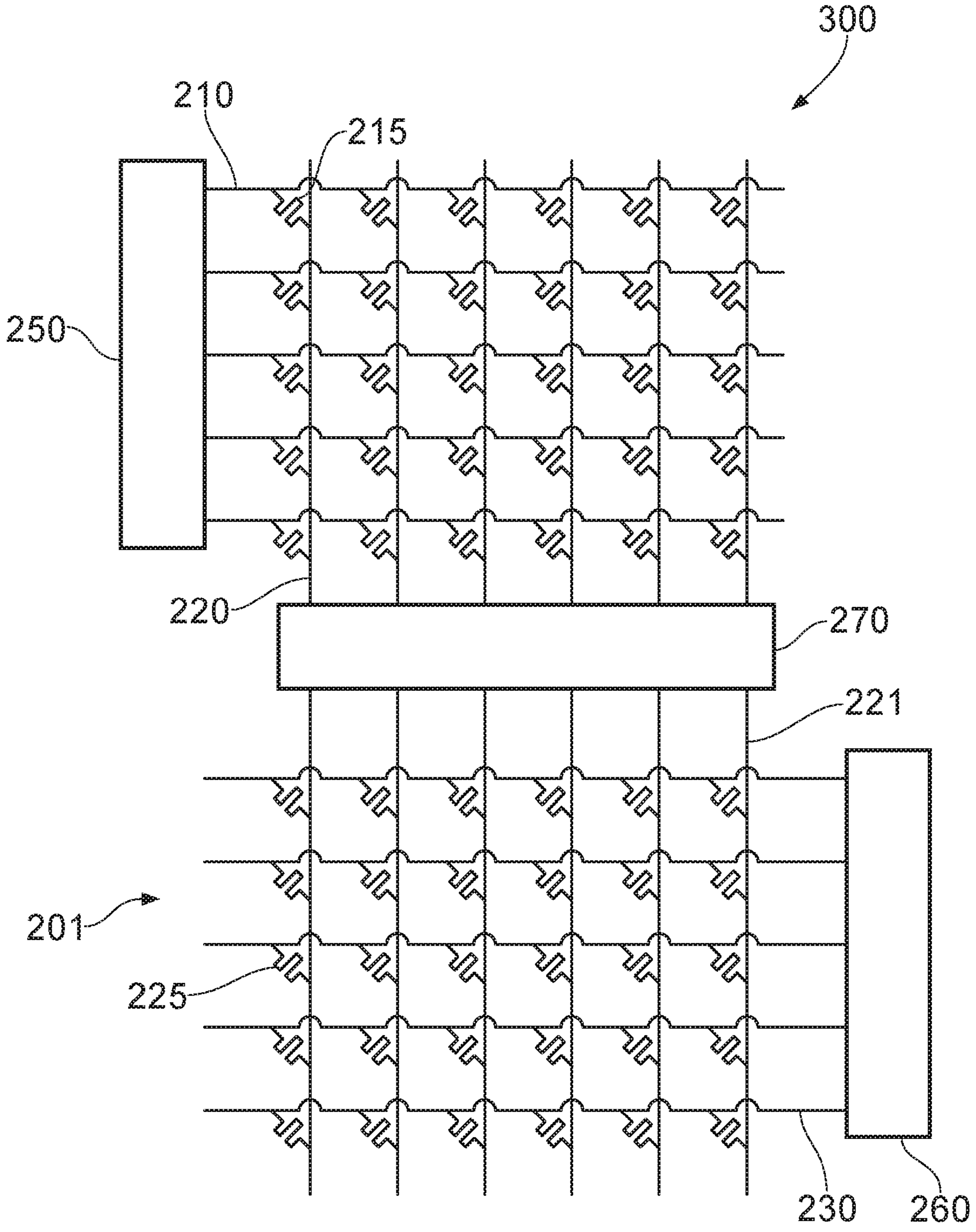

FIG. 3

10 12 20

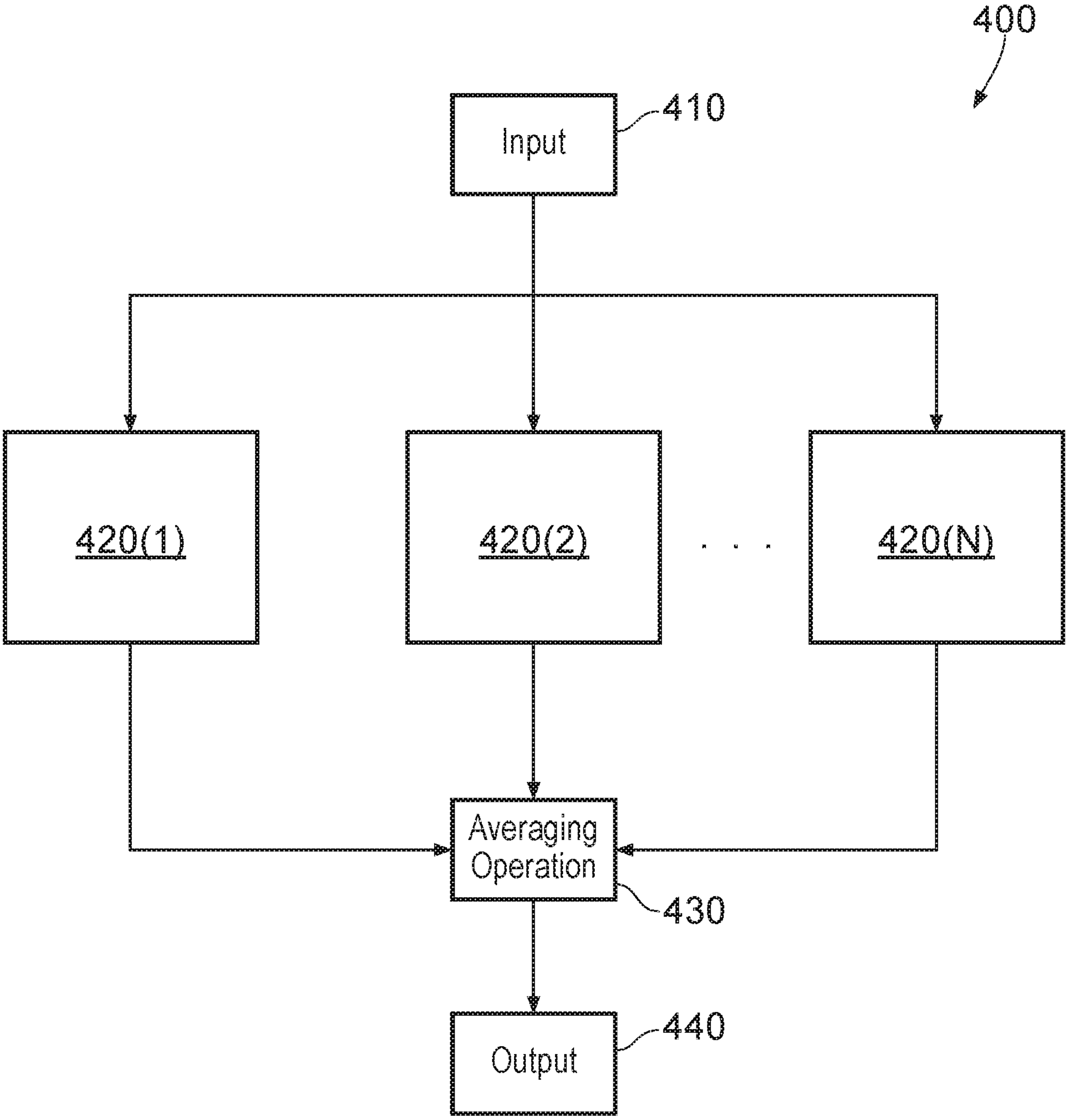
400
Input
410
420(1)
420(2)
. . .
420(N)
Averaging Operation
430
Output
440

FIG. 4

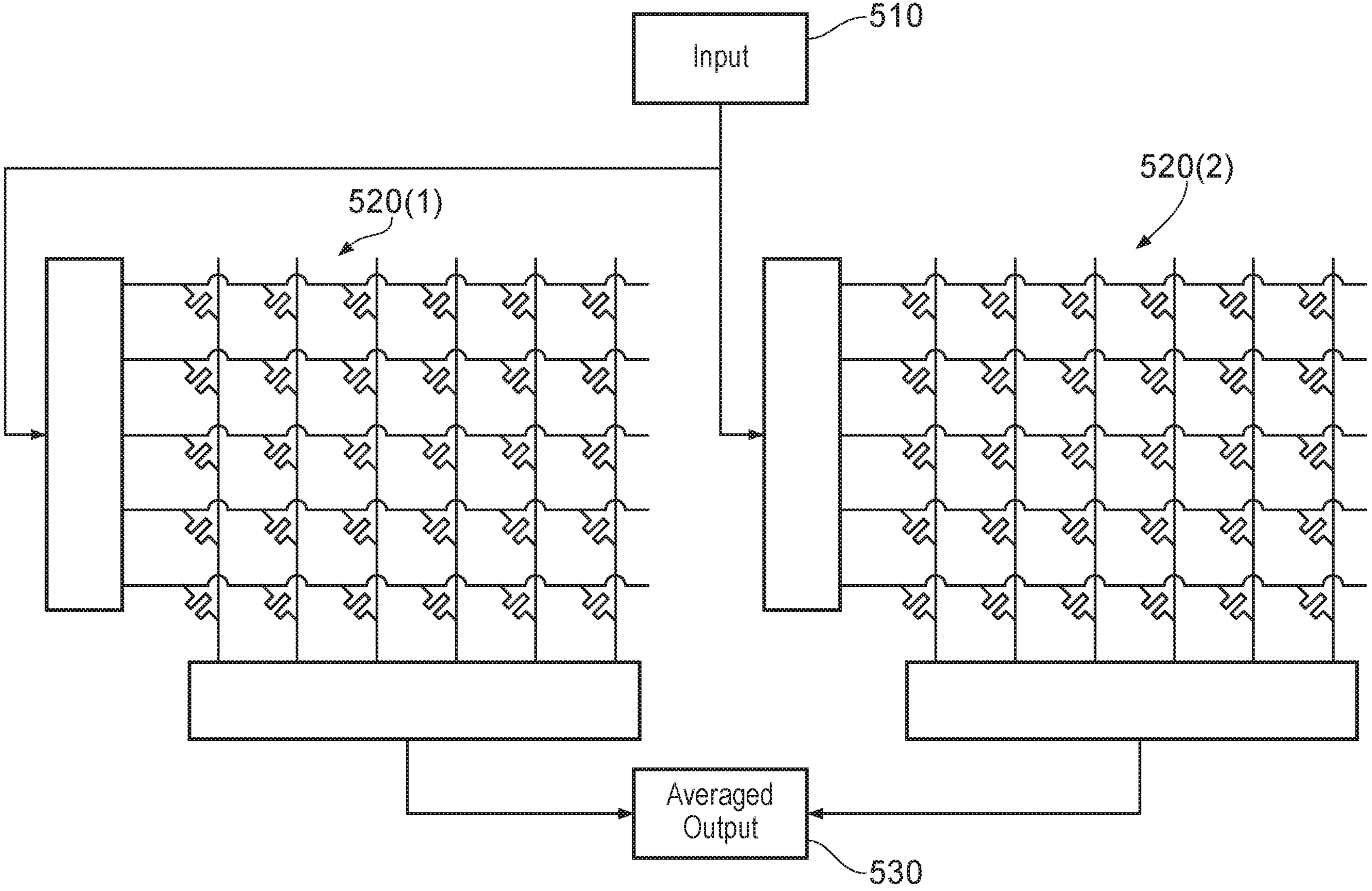

FIG. 5

10 12 20

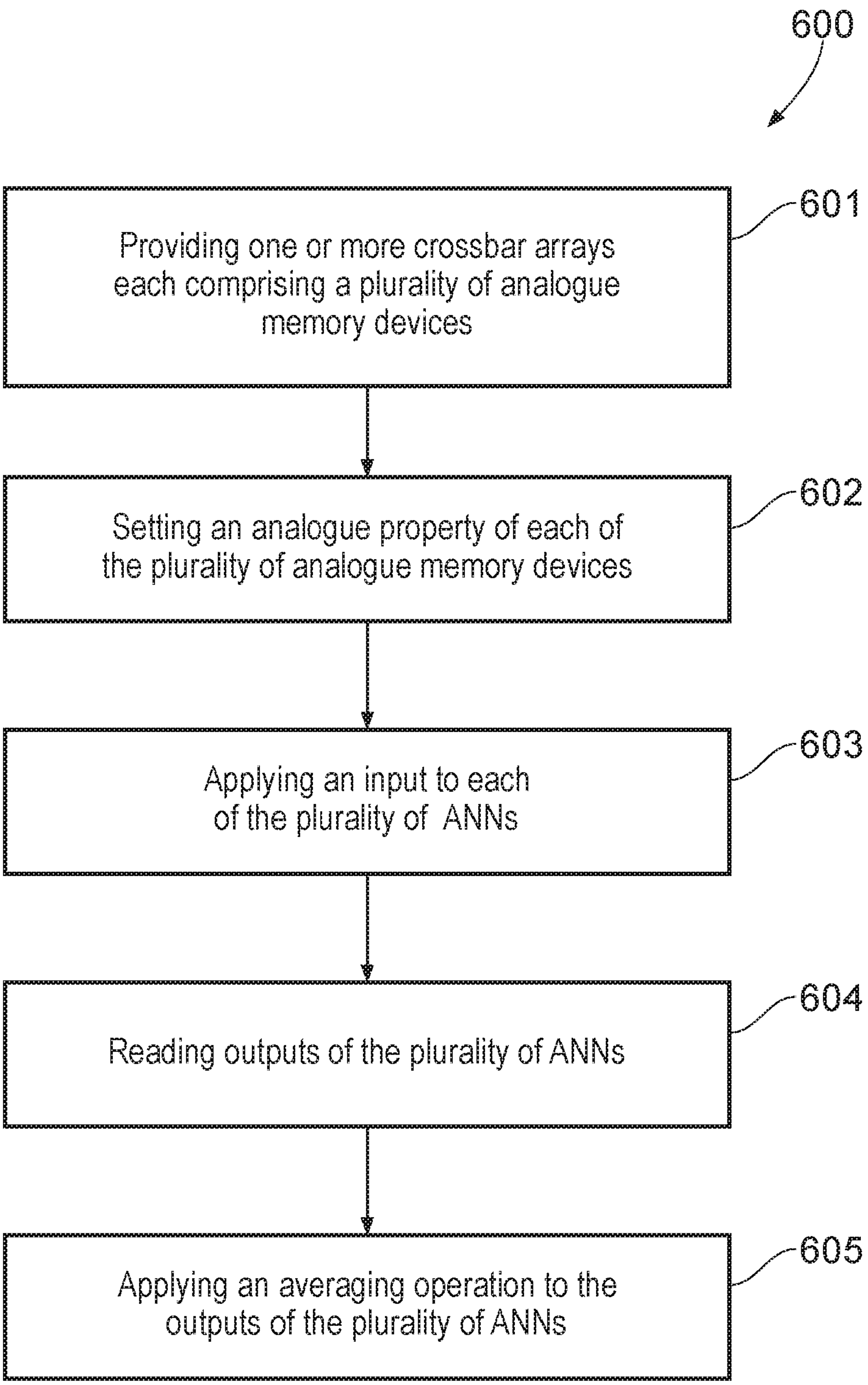
600
601
Providing one or more crossbar arrays each comprising a plurality of analogue memory devices
602
Setting an analogue property of each of the plurality of analogue memory devices
603
Applying an input to each of the plurality of ANNs
604
Reading outputs of the plurality of ANNs
605
Applying an averaging operation to the outputs of the plurality of ANNs

FIG. 6

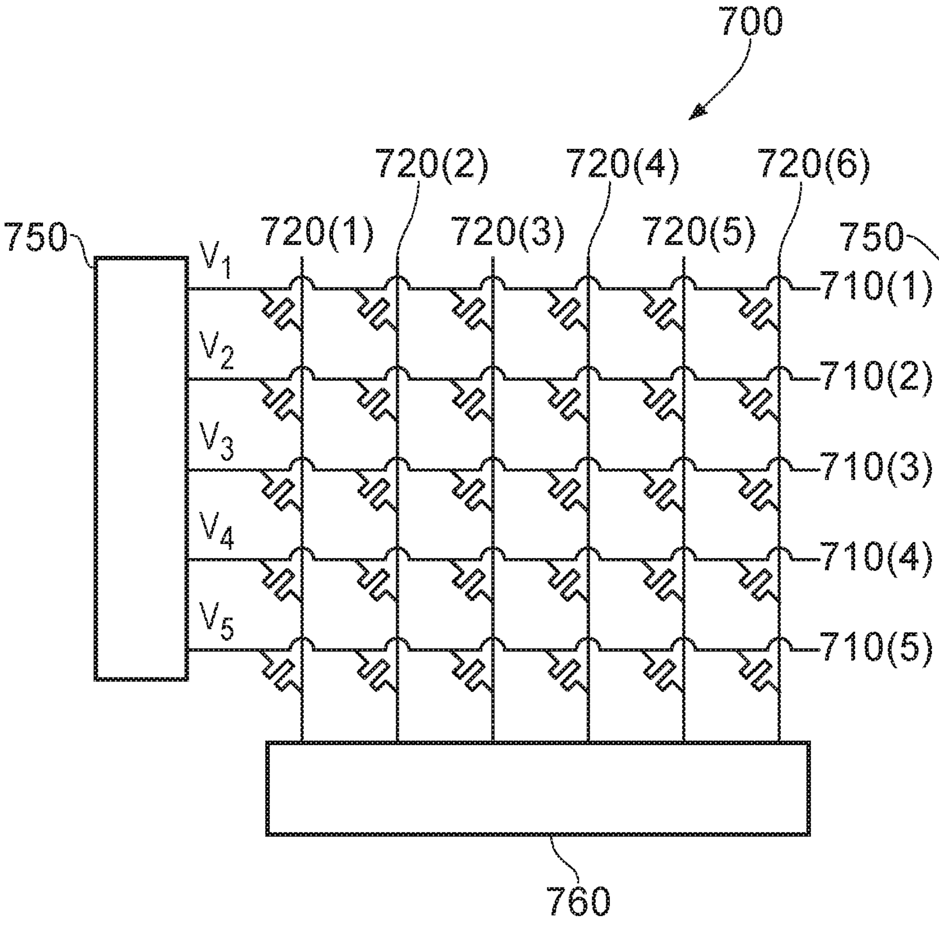
700
750
V1
V2
V3
V4
V5
720(1)
720(2)
720(3)
720(4)
720(5)
720(6)
710(1)
710(2)
710(3)
710(4)
710(5)
760

FIG. 7A

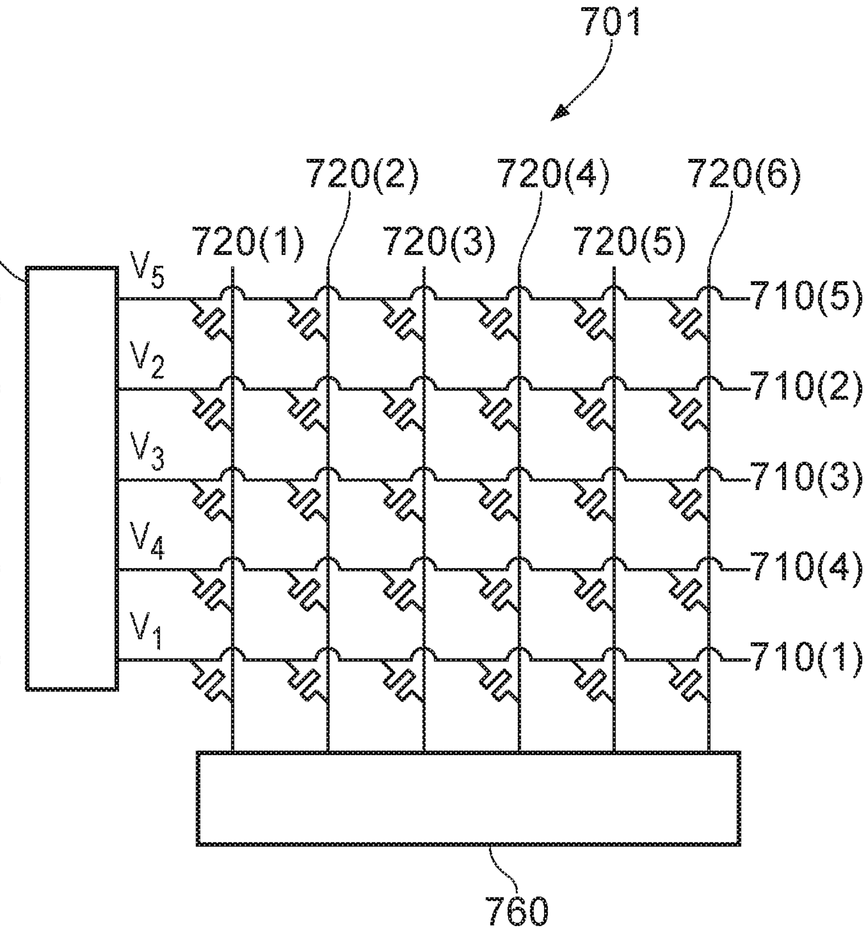
701
750
V5
V2
V3
V4
V1
720(1)
720(2)
720(3)
720(4)
720(5)
720(6)
710(5)
710(2)
710(3)
710(4)
710(1)
760

FIG. 7B

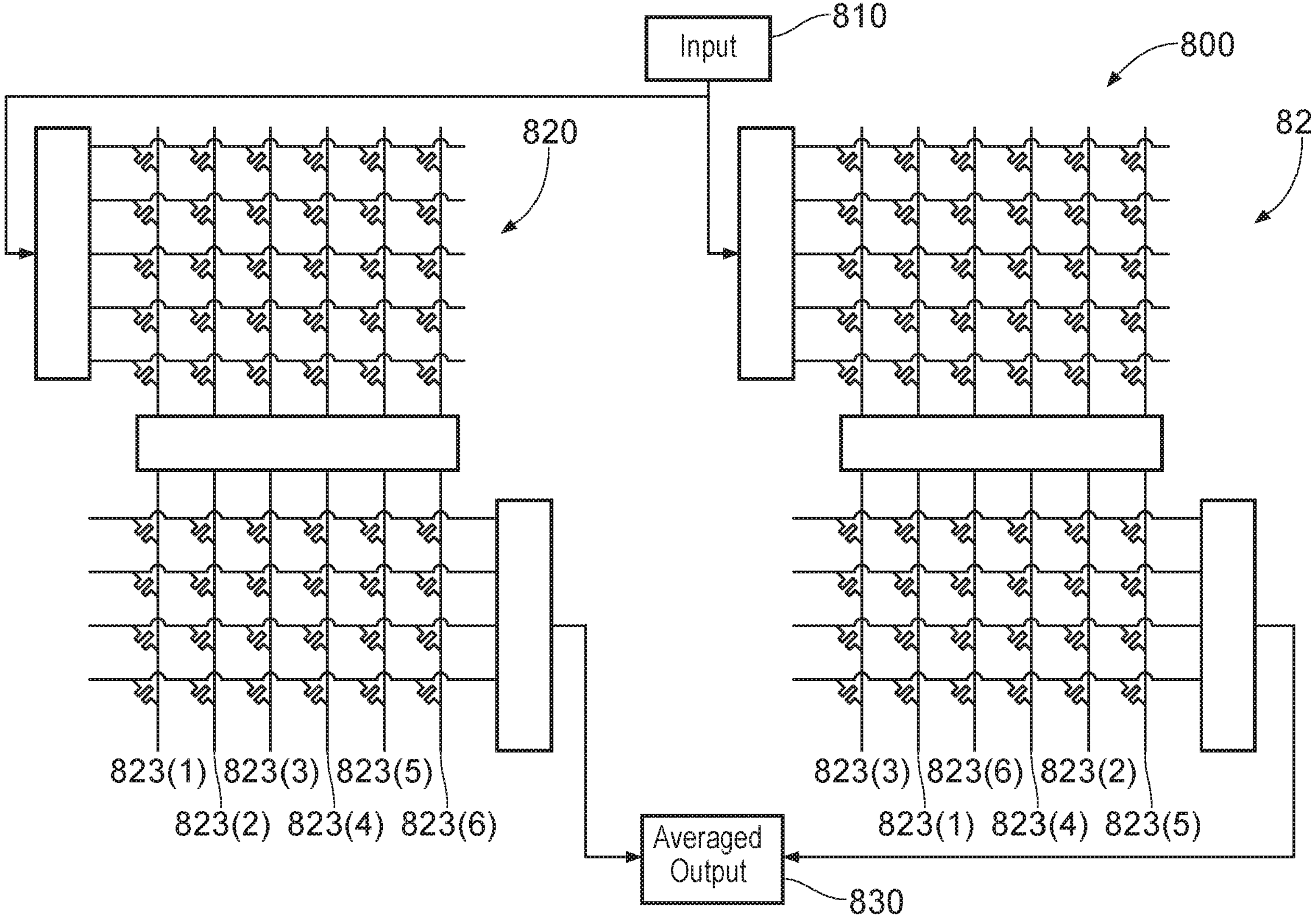
Input
810
800
820
821
823(1)
823(2)
823(3)
823(4)
823(5)
823(6)
823(3)
823(1)
823(6)
823(4)
823(2)
823(5)
Averaged Output
830

FIG. 8

Physical Implementation of Artificial Neural Networks

Technical Field

The invention relates particularly, but not exclusively, to apparatus and methods for physically implementing artificial neural networks.

Background

In recent years, artificial neural networks have been employed as a means of programming computing systems to perform certain tasks, such as image recognition or playing games, which are challenging to program effectively using traditional algorithms. Artificial neural networks include neurons (nodes) and synapses (connections between these nodes) which draw inspiration from the neurons and synapses within the human brain.

In artificial neural networks, neurons are arranged in layers with synapses connecting the neurons of a particular layer with the nodes of a different layer. Inputs are received at neurons in an input layer and the input values are combined with weightings for the synapses and signals are forwarded to neurons in an output layer according to the combination of the input and synaptic weights, potentially through a number of additional intermediate, or “hidden”, layers, which perform similar processes.

Artificial neural networks are “trained” to perform tasks using examples which are used to adjust the synaptic weights to alter how signals are propagated through the network. Accordingly, the synaptic weights can converge to values which allow the network to accurately perform the task. Once trained, the neural network can then be used for inference purposes with high accuracy.

The use of neural networks, however, can consume vast amounts of computing resources, which can lead to performance bottlenecks.

Summary of the Invention

Aspects of the invention are set out in the accompanying claims.

In a first aspect of the invention there is provided a device comprising: a plurality of crossbar arrays each comprising a plurality of analogue memory devices, the plurality of crossbar arrays being configured to implement a plurality of first artificial neural networks “ANNs” corresponding to one or more second ANNs trained on one or more other devices, wherein each of the plurality of first ANNs is implemented on a separate subset of the plurality of crossbar arrays, each subset comprising one or more crossbar arrays, wherein each crossbar array comprises a plurality of word lines and a plurality of bit lines logically connected via the plurality of analogue memory devices; setting circuitry configured to set an analogue property of each of the plurality of analogue memory devices of the plurality of crossbar arrays to correspond to a single synaptic weight of the one or more second ANNs; and output circuitry configured to: read output currents from readable portions of the plurality of bit lines of the one or more crossbar arrays of each subset of the plurality of crossbar arrays, to determine outputs of the plurality of first ANNs implemented on the plurality of crossbar arrays, and apply an ensemble averaging operation to the outputs of the plurality of first ANNs to generate an average output, wherein the ensemble averaging operation is a mean of the outputs of the plurality of first ANNs

Thus, according to the invention a device which allows ANNs to be implemented physically on analogue devices in a manner which provides comparable inference accuracy to digital ANNs without consuming large quantities of power and with reduced computation time is provided.

Advantageously, the averaging operation is an ensemble average of the outputs of the plurality of ANNs. Performing an ensemble average of the outputs of the ANNs provides a high degree of inference accuracy for physically implemented ANNs.

The device comprises a plurality of crossbar arrays, and each of the plurality of ANNs is implemented on a separate subset of the plurality of crossbar arrays, each subset comprising one or more crossbar arrays. By implementing each ANN on a separate subset of crossbar arrays, the physical implementation of the ANNs on a crossbar array can reduce construction complexity as compared to implementing an ANN on a single potentially large crossbar array.

The device comprise setting circuitry configured to set the analogue property of each of the plurality of analogue memory devices of the one or more crossbar arrays to correspond to the corresponding single synaptic weight of said one of the plurality of ANNs. Accordingly, the analogue properties of the analogue memory device can be adjusted based on further training or other use of either the physically implemented ANN or of a digital ANN.

Advantageously in some aspects, each crossbar array may comprise a plurality of word lines and a plurality of bit lines logically connected via the plurality of analogue memory devices; and the output circuitry may be configured to read the outputs of the plurality of ANNs from readable portions of bit lines of a said crossbar array. Such an arrangement allows ANNs with numerous input and output neurons to be physically implemented on crossbar arrays in a manner which reduces complexity relative to comparable systems.

The readable portions of the bit lines may be the portions of the bit line from which the output circuitry is arranged to read the output of the bit lines. Furthermore, additional components may be provided within the crossbar array, such as between the analogue memory devices and the word lines and bit lines, for example electrical contacts, diodes or any other suitable component, such that the analogue memory devices are not necessarily directly connected to the word lines and bit lines.

In such aspects, advantageously one or more first word lines of the plurality of word lines may be arranged to be geometrically closer to the readable portions of the plurality of bit lines than one or more second word lines of the plurality of word lines; and the one or more first word lines may be configured to receive an input voltage that is on average larger in magnitude than the one or more second word lines.

This arrangement of the word lines and bit lines prevents a large build-up of current in certain regions of the bit lines which would increase the effects of line resistance and therefore reduce the inference capabilities of the physically implemented ANN. The average values of the input voltages to the word lines can be determined, for example, through training or inference usage of the physical ANN or a digital ANN on which the physical ANN is based. This approach could be employed separately from the averaging of the outputs of multiple physical ANNs.

In some aspects each of the plurality of analogue memory devices is a resistive random-access memory device "RRAM"; and the analogue property of each RRAM may be a conductance of that RRAM. Such an arrangement can provide a physical implementation of an ANN which is reliable and can be reliably tuned by adjusting the resistance states of the RRAM devices, for example using voltage pulses.

In these aspects, at least one of the RRAMs may advantageously be a silicon oxide "SiOx" RRAM. Such devices can provide reliable resistive memory properties which can be readily adjusted. This allows the synaptic weights of the physically implemented ANN to be set accurately.

In some aspects, each of the plurality of ANNs may be implemented across two or more crossbar arrays; the bit lines of a first crossbar array of the two or more crossbar arrays may be electrically connected to the word lines of a second crossbar array of the two or more crossbar arrays; and wherein the bit lines of the second crossbar array may be provided with readable portions. Such an arrangement allows ANNs with multiple synaptic layers (i.e. with 'hidden neurons') to be implemented physically.

The bit lines of the first crossbar array and the word lines of the second crossbar array may be electrically connected, however this does not necessarily mean that they are directly connected to one another. For example, intermediate circuitry may (but not necessarily) be provided between the bit lines of the first crossbar array and the word lines of the second crossbar array. This intermediate circuitry may, for example, apply activation functions to the signals (either digitally or through analogue means) and/or convert a current output of the bit lines of the first crossbar array to a voltage input for the word lines of the second crossbar array, if necessary.

Advantageously, the one or more crossbar arrays for each ANN may each include a different geometric arrangement of the word lines relative to the bit lines. This arrangement results in non-idealities arising due to the geometric arrangement of the crossbar arrays producing different effects on each ANN, thus improving the inference accuracy of the system once the outputs of the physical ANNs have been averaged.

As an example, entire word lines or bit lines can be reordered, for example randomly, in each physical implementation of an ANN. This further improves the inference accuracy of the physical ANN system.

In a second aspect, there is provided a method, the method comprising: providing a plurality of crossbar arrays each comprising a plurality of analogue memory devices, the plurality of crossbar arrays being configured to implement a plurality of first artificial neural networks “ANNs” corresponding to one or more second ANNs trained on one or more other devices, wherein each of the plurality of first ANNs is implemented on a separate subset of the plurality of crossbar arrays, each subset comprising one or more crossbar arrays, and wherein each crossbar array comprises a plurality of word lines and a plurality of bit lines logically connected via the plurality of analogue memory devices; setting an analogue property of each of the plurality of analogue memory devices of plurality of crossbar arrays to correspond to a single synaptic weight of the one or more second ANNs; applying input voltages to the word lines of the one or more crossbar arrays of each subset of the plurality of crossbar arrays that implement the plurality of first ANNs; reading output currents from readable portions of the plurality of bit lines of the one or more crossbar arrays of each subset of the plurality of crossbar arrays, to determine outputs of the plurality of ANNs implemented on the plurality of crossbar arrays; and applying an ensemble averaging operation to the outputs of the first ANNs to generate an average output, wherein the ensemble averaging operation is a mean of the outputs of the plurality of first ANNs.

Thus, according to the present disclosure a method of implementing ANNs physically on analogue devices in a manner which provides comparable inference accuracy to digital ANNs whilst reducing power consumption relative to comparable systems is provided.

Advantageously, the averaging operation is an ensemble average of the outputs of the plurality of ANNs. Performing an ensemble average of the outputs of the ANNs provides improved inference accuracy for physically implemented ANNs.

In some aspects, the method further comprises: determining the synaptic weights of the plurality of ANNs by training the plurality of ANNs; and setting the analogue property of each of the one or more of the plurality of analogue memory devices of the plurality of crossbar arrays to correspond to a corresponding one of the determined synaptic weights. Training the ANNs in this manner improves the overall inference accuracy of the method and ensures that the physical ANN provides accurate inference capabilities. That is, the ANNs can, for example, be continuously trained and the analogue properties of the analogue memory device continually updated to increase the inference accuracy of the physically implemented ANN.

Advantageously, the method may further comprise training the plurality of ANNs on a digital processing device. This approach allows refined synaptic weights to be determined by training the ANNs digitally before implementing the refined synaptic weights on the physical ANN by setting the analogue properties of the analogue memory devices.

In some aspects, the synaptic weights of the plurality of ANNs may be determined by separately training each ANN. Accordingly, the ANNs implemented physically may be different from one another. This approach can provide high inference accuracies by combining the predictive capabilities of multiple ANNs and accounting for non-idealities in the physical implementations of the ANNs.

Advantageously, the plurality of ANNs may each have a different set of synaptic weights. In such cases, the ANNs have different idealised synaptic weights. That is, not only will the actual synaptic weights of the physical systems differ from the synaptic weights which are intended to be programmed onto the physical system, but the synaptic weights of the idealised ANNs which the physical ANNs implement are different.

In some aspects, the synaptic weights of each the plurality of ANNs may be determined by: determining the synaptic weights of a first ANN by training the first ANN; and setting the synaptic weights of the remaining ANNs of the plurality of ANNs to be equal to the synaptic weights of the first ANN. Accordingly, the inference capabilities of a single ANN can be improved by averaging the output of the physical implementations of the same ANN.

Advantageously, the plurality of ANNs may each have the same set of synaptic weights. In such cases, the ANNs each have the same idealised synaptic weights, as the imperfect physical implementations of the ANNs will inevitably result in the physically implemented ANNs having different. That is, the synaptic weights of the idealised ANNs which the physical ANNs implement are different

Brief Description of the Drawings

Embodiments of the invention will now be described, by way of example only, with reference to the following figures.

In accordance with one (or more) embodiments of the present invention the Figures show the following:

Figure 1 illustrates an example of an artificial neural network.

Figure 2 illustrates a crossbar array including analogue memory devices for implementing an artificial neural network.

Figure 3 illustrates multiple crossbar arrays for physically implementing an artificial neural network.

Figure 4 illustrates a method for averaging the outputs of multiple ANNs.

Figure 5 illustrates a physical implementation of a method of averaging the outputs of multiple ANNs.

Figure 6 illustrates a flow diagram including the steps of a method for averaging the outputs of multiple physically implemented artificial neural networks.

Figure 7A illustrates a crossbar array for physically implementing an artificial neural network including the voltages applied to each word line of the crossbar array.

Figure 7B illustrates a crossbar array for physically implementing an artificial neural network with reordered word lines to reduce the effects of line resistance in the crossbar array.

Figure 8 illustrates a method of ordering lines of crossbar arrays.

Any reference to prior art documents in this specification is not to be considered an admission that such prior art is widely known or forms part of the common general knowledge in the field.

As used in this specification, the words “comprises”, “comprising”, and similar words, are not to be interpreted in an exclusive or exhaustive sense. In other words, they are intended to mean “including, but not limited to”.

The invention is further described with reference to the following examples. It will be appreciated that the invention as claimed is not intended to be limited in any way by these examples. It will be further recognised that the skilled reader will understand from the teaching herein that integers and features of different embodiments may be used in any suitable and advantageous combination.

Detailed Description of the Drawings

Figure 1 illustrates an example of the structure of an artificial neural network (ANN) 100. An ANN 100 is a type of computer program which mimics the layout of the human brain to learn to perform certain tasks which can be challenging to program using traditional programming techniques. Such tasks include, for example, handwriting recognition, voice recognition, medical diagnosis and playing board and video games.

The example ANN 100 includes neurons 110, 120, 130 and synapses 115, 125 connecting the neurons 100, 120, 130 take inspiration from the neurons and synapses in the human brain in a simplified manner. A neuron 110 of the ANN 100 receives an input which it provides to synapses 115 with assigned weights, each synaptic weight representing the importance of the corresponding connection between the respective neurons.

The synapses 115 pass the resultant signals to neurons 120 which can apply their own activation functions (which transform the signals, for example, in a non-linear manner to improve the learning capabilities of the ANN or for normalisation purposes) to the signals, and output the signals to further synapses 125 or, alternatively, give the output signals as an output of the ANN 100. Neurons 120 which do not receive external input or produce external output are referred to as 'hidden neurons'.

Generally, an ANN 100 can be conceptualised in terms of layers, with a layer of input neurons 110 connected (via synapses) to a layer of output neurons 130 or hidden neurons 120 (which are in turn connected to a layer of output neurons or an additional layers of hidden neurons). An ANN 100 learns to perform its assigned tasks through training, in which the synaptic weights between layers of the ANN 100 are refined to increase the inference capabilities of the ANN 100 through exposure to large data sets.

The ANN 100 of Figure 1 includes an input layer of five input neurons 110, an output layer of four output neurons 130, and one hidden layer of six hidden neurons 120. Each input neuron 110 is connected to each hidden neuron 120 via synapses 115, and each hidden neuron 120 is connected to each output neuron 130 via synapses 125.

An activation function can be applied to the signal at the neurons, for example to normalise the signal between a value of 0 and 1. Example activation functions include the sigmoid function, the Heaviside step function, the identity function, and the Rectified Linear Unit (ReLU) function, however numerous other activation functions can be used in ANNs.

ANNs 100 can, in practice, include any number of neurons and any number of hidden layers, with the neurons of each hidden layer connected to the neurons of adjacent hidden layers. The number of neurons in the input and output layers can be determined based on the task the ANN performs. For example, a handwriting recognition ANN which recognises numbers may have a number of input neurons equal to the number of pixels of the input image to be analysed, and may have ten output neurons each corresponding to the numbers '0-9'.

The ANN 100 shown in Figure 1 is an example of a fully connected ANN in which each neuron of a particular layer is connected to every neuron of an adjacent layer. However, artificial neural networks can exist in other forms, such as convolutional neural networks. Furthermore, ANN 100 includes only one hidden layer, however any number of hidden layers could be present in the ANN.

When operating ANN 100, an input can be applied to the input layer in the form of a vector. This may comprise applying a scalar value to each input neuron 110The vector is then effectively multiplied by a matrix determined by synaptic weights of the synapses 115 and their arrangement between the input neurons 110 and the hidden neurons 120.

This multiplication can be said to yield a vector at hidden layer 120. This vector may be transformed by the activation function of the hidden neuron. The vector is then effectively multiplied by a further matrix determined by synaptic weights of the synapses 125 between the hidden neurons 120 and the output neurons 130, yielding a further vector which can be further transformed by the activation of the output neurons to produce a vector output at the output layer. An overall output of ANN 100 can then be determined by analysing the values of the output vector. For example, the overall output may be determined by choosing the label corresponding to the largest entry in the vector. As a specific example, where each output neuron corresponds to a digit from 0 to 9, the overall output may be the digit with the largest value at its corresponding output neuron.

ANNs such as ANN 100 described in Figure 1 can, however, consume vast amounts of power and other computing resources. One promising approach to overcoming these limitations is to implement artificial neural networks physically, rather than digitally, for example using non-von Neumann architecture. An example of non-von Neumann architecture which could be used for such purposes uses analogue memory devices as proxies for the synapses of the artificial neural network.

According to an aspect of the present disclosure, an ANN can be implemented within a crossbar array. Such a crossbar array can be considered to implement a non-von Neumann architecture. Figure 2 illustrates an example of a crossbar array implementing an ANN.

The crossbar array 200 shown in Figure 2 includes word lines 210 and bit lines 220. In the crossbar array 200 shown, each word line 210 is connected to each bit line 220 in a matrix arrangement via an analogue memory device 215. The crossbar array 200 also includes input circuitry 250 for applying inputs to the word lines 210, and output circuitry for reading the outputs of the bit lines 220 from readable portions of the bit lines 220. Although the input circuitry 250 and output circuitry 260 are shown as single entities, each of the input circuitry 250 and output circuitry 260 may comprise multiple circuitries such that separate circuitry is provided for each word line 210 and bit line 220 respectively.

The crossbar array 200 can be used to implement particular layers of ANN 100 shown in Figure 1. In the example crossbar array 200 shown in Figure 2, five word lines 210 act as the five input neurons 110, six bit lines 220 act as the six hidden neurons 120, and the analogue memory devices 215, connecting the word lines 210 and bit lines 220, act as the synapses 115. The crossbar array 200 can thus implement a two-layer ANN or, alternatively, can implement two layers of an ANN with greater than two layers.

In some examples, the number of word lines 210, bit lines 220 and analogue memory device 215 may differ from the number of neurons and synapses in the ANN. For example, a single neuron may correspond to two bit lines (or two word lines) such that a crossbar array can, for example, implement both positive and negative synaptic weights. Accordingly, multiple (for example two) analogue memory devices may be used to implement a single synapse of the ANN.

For the purposes of discussion, in such examples all word lines or bit lines which correspond to a single neuron are collectively discussed as a single word line or bit line, and all analogue memory devices which are used to implement a single synapse are collectively discussed as a single analogue memory device.

Analogue memory devices 215 can be implemented as various types of device which have an analogue property with a plurality of states. For example, analogue memory devices 215 can be non-volatile random-access memory (NVRAM) devices of various types, such as resistive random-access memory (RRAM), magnetoresistive random-access memory (MRAM), spin-transfer torque MRAM (STT-MRAM), ferroelectric random-access memory (FeRAM), or phase-change random-access memory (PCRAM).

Using RRAM as an example, a RRAM device includes multiple conductive states in which a conductance (or equivalently resistance, which is the inverse of conductance) of the device is different. Accordingly, the conductance of the RRAM devices can be tuned to multiple possible values, allowing the RRAM devices to act as synapses in an ANN, where the conductance relates to the synaptic weight of the synapse. This can be done by applying voltage pulses to the devices using setting circuitry to adjust the conductance of the RRAM device. For other physical implementations of an ANN, the setting circuitry may be any apparatus suitable for setting the analogue property of the analogue memory devices.

RRAM devices can be formed of any suitable material. For example, a RRAM device can be formed of silicon oxide (SiOx) which could be sputtered onto electrodes made from, for example, gold or molybdenum. The word lines 210 and the bit lines 220 can be implemented by a conductive wire made from any suitable material such as, for example, gold. Furthermore, additional lines at ground potential could be provided in the crossbar array(s), if required. The crossbar arrays 200 may also include diodes or other electrical devices as appropriate, for example to reduce current leakage between bit lines.

In examples, the synaptic weights of the artificial neural network which are used to set the values of the analogue property of the analogue memory devices are determined through training the ANN. The ANN can be trained on a digital processing device before assigning the determined synaptic weights to the analogue memory devices. Alternatively, the ANN could be trained on an analogue processing device. This could involve the ANN being trained on any other analogue processing device.

To operate the crossbar array 200, input circuitry 250 provides a voltage input to each word line 210. The connections between the word lines 210 and bit lines 220 via the analogue memory devices 215 result in a current along each of the bit lines 220 which is read by output circuitry 260. Accordingly, the output circuitry 260 can read an output of the ANN implemented by the crossbar array 200. Furthermore, in examples where multiple bit lines correspond to a single neuron, the output circuitry may combine (through addition or subtraction) the output currents of multiple bit lines to determine an output of a particular output neuron of the ANN.

A further crossbar array can be provided to mimic the connections between hidden neurons 120 and output neurons 130 via synapses 125. Such an arrangement, according to an example of the present disclosure, is shown in Figure 3. The crossbar array 200 shown in Figure 2 is electrically connected to a second crossbar array 201. The second crossbar array 201 includes word lines 221, bit lines 230, and analogue memory devices 225 connecting the word lines 221 and bit lines 230 in a similar manner as described above in relation to crossbar array 200.

The second crossbar array 201 includes six word lines 221 and four bit lines 230. The word lines 221, bit lines 230, and analogue memory devices 225 mimic the hidden neurons 120, output neurons 130, and hidden neurons 125 of ANN 100 shown in Figure 1.

In the arrangement shown in Figure 3, the output circuitry 260 shown in Figure 2 is connected to the second crossbar array 201 to read the outputs of the word lines 230 of the second crossbar array 201, in order to read the output of the ANN 100 implemented across these two crossbar arrays 200 and 201.

The two crossbar arrays 200 and 201 can be electrically connected via intermediate circuitry 270 as shown. This intermediate circuitry 270 can be used to convert the output current of the bit lines 220 of the first crossbar array 200 to a voltage which can be input to the word lines 221 of the second crossbar array 201, if required.

Furthermore, the intermediate circuitry can apply an activation function to the signal, corresponding to an activation function of hidden nodes 120 of ANN 100, if required. For example, the intermediate circuitry may convert the analogue outputs of cross bar 200 to digital values, apply the activation function in software, and then convert the result to analogue voltages to be applied to the word lines 221 of the second crossbar array 201. Such an approach may be particularly useful in functions which are difficult to approximate, such as the sigmoid function. Alternatively, the activation function may be applied to the signal in an analogue manner. For example, the ReLU function could be implemented by adding a diode to each output of each crossbar array.

The above physical implementations of an ANN can yield large performance improvements over standard digital von-Neumann implementations in terms of resource consumption. For example, physical implementations of ANNs using crossbar arrays can reduce power consumption by orders of magnitude.

However, despite these benefits, implementing an ANN physically using crossbar arrays has drawbacks in inference accuracy as compared to the corresponding digital implementation of the ANN. These inference inaccuracies are caused by non-idealities in the physical implementation of the ANN which are not present in digital implementations of ANNs. These non-idealities disturb the synaptic weights of the ANN or disturb the current distribution within the ANN, and therefore impact inference accuracy.

In the case of RRAM devices, these non-idealities may include, for example: devices having a finite number of discrete resistance states, as opposed to a continuous tuneable resistance; devices having a small operational range of conductance modulation; faulty devices (for example stuck in a particular conductance state); current/voltage non-linearities; programming non-linearities; random telegraph noise; device-to-device variability; and line resistance in the wires forming the word lines and bit lines. Other analogue memory devices may include different non-idealities.

Various different steps can be taken to individually address each of these non-linearities, based generally on improving device fabrication or providing additional components, such as transistors, in the crossbar array arrangement. However, these non-idealities cannot be eliminated entirely and thus over time improved manufacturing methods produce diminishing returns in improving the inference accuracy of a physical ANN.

Figure 4 illustrates a method 400 for reducing the effects of all non-idealities in physically implemented ANNs, according to an example. First, the input 410 to be provided to an ANN is determined. This input 410 is then provided to multiple physically implemented ANNs 420. That is, physically implemented ANNs 420(1)-(N) are each provided with the same input values. Physically implemented ANNs 420(1)-(N) then each produce output values corresponding to the same set of possible outputs. As a specific example, where each output neuron corresponds to a digit from 0 to 9 each ANN outputs a value at each of the ten output neurons, each value corresponding to a degree of certainty that the input image contains a respective one of the digits 0-9.

The multiple physical implementations of ANN 420 may be provided on the same crossbar array or on multiple crossbar arrays. In the case of the multiple physical implementations being implemented on different crossbar arrays, the same set of input voltages are applied to the terminals of the crossbar arrays implementing the ANNs. The same ANN may be implemented physically multiple times, or alternatively different ANNs with different synaptic weights may be implemented physically. That is, as explained in more detail below, in some examples ANNs 420(1)-(N) have the same synaptic weights and in other examples ANNs 420(1)-(N) have different synaptic weights.

The outputs of the multiple physical implementations of an ANN 420 are then subject to an averaging operation 430. This, for example, may be an ensemble average in which a mean of the output values of corresponding output neurons of the multiple ANNs 420 is calculated, however other averaging operations could be used. For example, an output neuron with the largest value can be determined for each ANN 420, and a modal output neuron of the corresponding output neurons across the ANNs 420 determined. Such averaging operations can therefore be used to determine an overall output 440 of the ANNs 420.

Figure 5 illustrates a physical implementation of the averaging process shown in Figure 4. The same input 510 is provided to multiple physically implemented ANNs 520(1), 520(2). Although only a single crossbar array is shown for implementing each ANN 520 for ease of explanation, the number of crossbar arrays implementing each ANN 520 can be increased as required to implement the entire network. The outputs of the physically implemented ANNs 520 are then averaged to produce an averaged output 530.

This approach of averaging the output of multiple physically implemented ANNs 520 improves the overall accuracy of a physically implemented ANN 520 by reducing the effects of non-idealities in the physical implementations. The non-idealities will frequently be different in each physical implementation and thus the averaging operation prevents systematic non-idealities in the physical implementations. For example, faulty devices will not occur in the same location in different crossbar arrays and thus the averaging operation allows the effects of the faulty devices to be averaged out in the final output, thus improving inference accuracy. Furthermore, this approach is agnostic to the type of non-ideality in that all the effects of all non-idealities are reduced, thereby greatly improving the inference capabilities of physically implemented ANNs without requiring each potential non-ideality to be individually identified and corrected.

Furthermore, for a given total number of neurons and synapses, the overall inference capabilities of a system comprising multiple physically implemented ANNs is generally greater than that of a single physically implemented ANN. Accordingly, the inference capabilities can be improved without the need to increase the number of neurons and/or synapses, and thus this improvement is achieved with no, or minimal, increase in power, or other resource, consumption.

The averaging process can be implemented by mapping the same set of identical synaptic weights onto the analogue processing devices of the crossbar arrays which implement the ANNs. In other words, the same ANN is mapped onto each crossbar array. The non-idealities of the respective crossbar arrays will, however, cause the resultant physically implemented ANNs to differ from one another. Accordingly, the averaging process can improve the inference capabilities of the arrangement as a whole, as compared to the physical implementation of any one ANN, by averaging out the non-idealities.

Alternatively, rather than mapping the same set of synaptic weights onto crossbar arrays, different sets of synaptic weights can be mapped onto each crossbar array. In other words, a different ANN is mapped onto each crossbar array. These different synaptic weights can be determined, for example, by separately training the multiple ANNs to be physically implemented. Accordingly, the resultant physically implemented ANNs differ from one another not only due to the non-idealities of the physical system but also due to different programming of the analogue memory devices. Such an approach not only mitigates the effects of non-idealities (as when mapping the same ANN onto the crossbar arrays) but additionally combines the knowledge of each of the different ANNs, leading to improved inference capabilities. For example, a particular ANN may be particularly good at identifying the digit ‘2’, and by combining the inference capabilities of this ANN with other ANNs which are comparatively worse at identifying the digit ‘2’ but better at identifying other digits, the overall inference accuracy can be improved.

This approach can be further expanded to not only map different synaptic weights onto different crossbar arrays, but to average the outputs of ANNs with different layouts (for example different numbers of hidden neurons and synapses) in order to improve overall inference capabilities by providing a more diverse set of ANNs which are better at performing particular tasks.

Figure 6 illustrates an example method, according to an example, for physically implementing artificial neural networks in a manner which reduces the effects of non-idealities in the physical implementations. The method includes providing 601 one or more crossbar arrays each comprising a plurality of analogue memory devices. The one or more crossbar arrays are configured to implement a plurality of ANNs. The method then includes setting 602 an analogue property of each of the plurality of analogue memory devices of the one or more crossbar arrays to correspond to a single synaptic weight of one of the plurality of ANNs.

Next, an input is applied 603 to each of the plurality of ANNs. The outputs of the plurality of ANNs implemented on the one or more crossbar arrays are then read 604. An averaging operation is then applied 605 to the outputs of the ANNs to generate an average output.

Accordingly, the effects on non-idealities on the inference capabilities of physically implemented ANNs can be mitigated.

In addition (or as an alternative) to averaging the outputs of the physical implementations of ANNs, non-idealities can be mitigated using additional techniques. Figures 7A and 7B illustrate one example of how this may be done. Through training of an ANN and/or use of an ANN for inference purposes, it can be determined that an input signal delivered to a particular input neuron is on average larger than an input signal delivered to different input neuron. Accordingly, for an ANN implemented physically on a crossbar array this will amount to an input voltage being delivered to a particular word line that is on average larger than an input signal delivered to a different word line.

For example, the crossbar array 701 shown in Figure 7A includes input circuitry 750, output circuitry 760, word lines 710(1)-(5), and bit lines 720(1)-(6). Through training of the ANN or use of the ANN for inference purposes, it is, for example, determined that the average input received by word lines 710(1)-(5) are V1-V5 respectively. In this example, V2 is approximately equal to V3 and V4, V1 is larger than V5 and V2-V4, and V5 is smaller than V2-V5 (i.e. V1 > V2 = V3 = V4 > V5).

Figure 7B illustrates a modified crossbar array 701 which minimises the effect of line resistance as compared to crossbar array 700. Here word line 710(1) which receives on average the largest input voltage V1 is positioned geometrically closer to the portion of bit lines 720(1)-(6) where the output circuitry 760 reads the output of the bit lines 720(1)-(6).

By positioning the word lines which receive larger input voltages closer to the output of the bit lines, the build-up of large currents along the length of the bit lines is prevented. This in turn reduces line resistance along the bit lines, thereby improving inference accuracy of the physically implemented ANN by making the physical implementation of the ANN closer to an idealised ANN in which the bit lines have zero resistance. While the reordering inputs of the crossbar arrays is discussed above in relation to an ANN implemented on a single crossbar array, the same method can be applied to ANNs implemented on multiple crossbar arrays, such as that shown in Figure 3.

Figure 8 illustrates a different method 800 for reducing non-idealities which can be used in combination with any of the techniques described herein. An input 810 is provided to multiple ANNs implemented on different sets of crossbar arrays 820 and 821. Where particular components of the sets of crossbar arrays 820 and 821 are not labelled, the components correspond to those of the crossbar arrays illustrated in Figure 3.

In this example, each of the word lines 823(1)-(6) of a particular crossbar array of the sets of crossbar arrays which physically implement the ANN are ordered randomly for each physically implemented ANN. That is, the geometric arrangement of the bit lines and word lines of different crossbar arrays of a particular ANN is different for different sets of crossbar arrays 820, 821.

Although Figure 8 depicts two physically implemented ANNs, this process can be applied to any number of physically implemented ANNs. Furthermore, while Figure 8 shows the word lines of a second crossbar array of the sets of crossbar arrays being reordered, this could be applied to any layer of the sets of crossbar arrays implementing an ANN, and to any number of layers in the sets of crossbar arrays.

This process increases the variability between the physical implementations of the ANNs, which allows the non-idealities arising from the layout of the crossbar array(s) to be averaged out, thereby improving the inference capabilities of the system.

Therefore, there has been described a device and method for physically implementing an artificial neural network. One or more crossbar arrays are provided which include analogue memory devices, the one or more crossbar arrays being configured to implement a plurality of artificial neural networks “ANNs”. Each of the analogue memory devices is configurable to correspond to a synaptic weight of the artificial neural network. Output circuitry is provided to read the output of the artificial neural network implemented on the one or more crossbar arrays.

Accordingly, the effects of non-idealities can be reduced, leading to improved inference accuracies for physically-implemented artificial neural networks which can match the inference accuracy of idealised digital artificial neural networks, and in some cases improve the overall inference accuracy.

Claims

1. A device comprising:

a plurality of crossbar arrays each comprising a plurality of analogue memory devices, the plurality of crossbar arrays being configured to implement a plurality of first artificial neural networks "ANNs" corresponding to one or more second ANNs trained on one or more other devices, wherein each of the plurality of first ANNs is implemented on a separate subset of the plurality of crossbar arrays, each subset comprising one or more crossbar arrays, wherein each crossbar array comprises a plurality of word lines and a plurality of bit lines logically connected via the plurality of analogue memory devices;

setting circuitry configured to set an analogue property of each of the plurality of analogue memory devices of the plurality of crossbar arrays to correspond to a single synaptic weight of the one or more second ANNs; and

output circuitry configured to:

read output currents from readable portions of the plurality of bit lines of the one or more crossbar arrays of each subset of the plurality of crossbar arrays, to determine outputs of the plurality of first ANNs implemented on the plurality of crossbar arrays, and

apply an ensemble averaging operation to the outputs of the plurality of first ANNs to generate an average output, wherein the ensemble averaging operation is a mean of the outputs of the plurality of first ANNs.

2. The device according to claim 1, wherein:

one or more first word lines of the plurality of word lines are arranged to be geometrically closer to the readable portions of the plurality of bit lines than one or more second word lines of the plurality of word lines; and

the one or more first word lines are configured to receive an input voltage that is on average larger in magnitude than the one or more second word lines.

3. The device according to any preceding claim, wherein:

each of the plurality of analogue memory devices is a resistive random-access memory device "RRAM"; and

the analogue property of each RRAM is a conductance of that RRAM.

4. The device according to claim 3, wherein at least one of the RRAMs is a silicon oxide "SiOx" RRAM.

5. The device according to any of claims 1-4, wherein:

each of the plurality of first ANNs are implemented across two or more crossbar arrays;

the bit lines of a first crossbar array of the two or more crossbar arrays are electrically connected to the word lines of a second crossbar array of the two or more crossbar arrays; and

wherein the bit lines of the second crossbar array are provided with readable portions.

6. The device according to any of claims 1-5, wherein the one or more crossbar arrays for each first ANN each includes a different geometric arrangement of the word lines relative to the bit lines.

7. A method, the method comprising:

providing a plurality of crossbar arrays each comprising a plurality of analogue memory devices, the plurality of crossbar arrays being configured to implement a plurality of first artificial neural networks “ANNs” corresponding to one or more second ANNs trained on one or more other devices,

wherein each of the plurality of first ANNs is implemented on a separate subset of the plurality of crossbar arrays, each subset comprising one or more crossbar arrays, and wherein each crossbar array comprises a plurality of word lines and a plurality of bit lines logically connected via the plurality of analogue memory devices;

setting an analogue property of each of the plurality of analogue memory devices of plurality of crossbar arrays to correspond to a single synaptic weight of the one or more second ANNs;

applying input voltages to the word lines of the one or more crossbar arrays of each subset of the plurality of crossbar arrays that implement the plurality of first ANNs;

reading output currents from readable portions of the plurality of bit lines of the one or more crossbar arrays of each subset of the plurality of crossbar arrays, to determine outputs of the plurality of ANNs implemented on the plurality of crossbar arrays; and

applying an ensemble averaging operation to the outputs of the first ANNs to generate an average output, wherein the ensemble averaging operation is a mean of the outputs of the plurality of first ANNs.

8. The method of claim 7, further comprising:

determining the synaptic weights of the one or more second ANNs by training the one or more second ANNs; and

setting the analogue property of each of the one or more of the plurality of analogue memory devices of the plurality of crossbar arrays to correspond to a corresponding one of the determined synaptic weights.

9. The method of claim 8, comprising training the one or more second ANNs on a digital processing device.

10. The method of claim 8-9, wherein the one or more second ANNs includes a plurality of second ANNs, and wherein the synaptic weights of the plurality of second ANNs are determined by separately training each second ANN.

11. The method of any of claims 8-9, wherein the one or more second ANNs includes a plurality of second ANNs each having a different set of synaptic weights.

12. The method of any of claims 8-9, wherein the one or more second ANNs includes a plurality of second ANNs, and wherein the synaptic weights of each the plurality of second ANNs are determined by:

determining the synaptic weights of a first second ANN by training the first second ANN; and

setting the synaptic weights of the remaining second ANNs of the plurality of second ANNs to be equal to the synaptic weights of the first second ANN.

13. The method of any of claims 8-9 or claim 12, wherein the one or more second ANNs each have the same set of synaptic weights.