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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 July 2015 (30.07.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/112718 A1

(51) International Patent Classification:
G06N 3/04 (2006.01)

(21) International Application Number:
PCT/US2015/012464

(22) International Filing Date:
22 January 2015 (22.01.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/163,921 24 January 2014 (24.01.2014) US

(71) Applicant: **QUALCOMM INCORPORATED** [US/US]; ATTN: International IP Administration, 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(72) Inventors: **CANOY, Michael-David Nakayoshi**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **LIU, Yinyin**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **BEHABADI, Bardia Fallah**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **RANGAN, Venkat**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(72) Inventor: **HUNZINGER, Jason Frank** *(deceased)*.

(74) Agents: **LENKIN, Alan M.** et al.; Seyfarth Shaw LLP, Suite 3500, 2029 Century Park East, Los Angeles, California 90067-3021 (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

[Continued on next page]

(54) **Title**: METHOD FOR CONVERTING VALUES INTO SPIKES

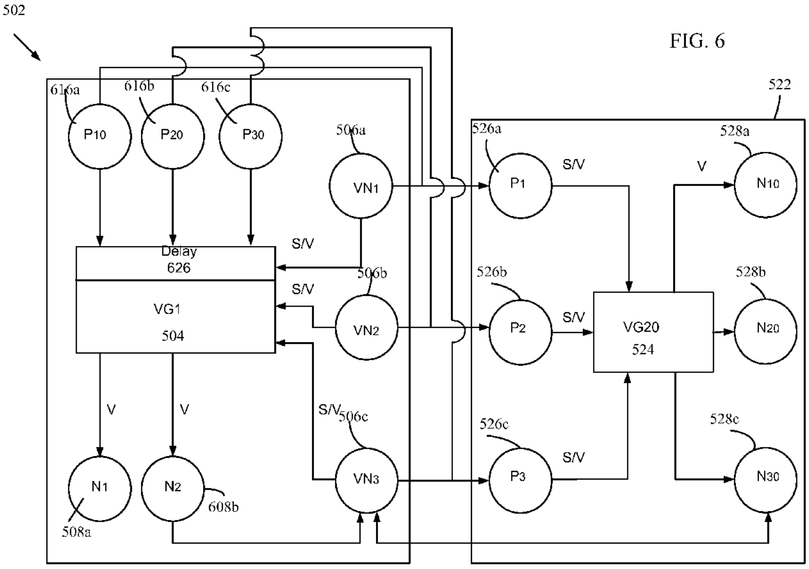


FIG. 6

(57) **Abstract**: A method for transmitting values in a neural network includes obtaining a parameter value. The method also includes encoding the parameter value based on at least one value used by a neuron. The encoding is based on a spike to be transmitted via a spike channel.

WO 2015/112718 A1

Published:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

METHOD FOR CONVERTING VALUES INTO SPIKES

BACKGROUND

Field

[0001] Aspects of the present disclosure generally relate to neural system engineering and, more particularly, to systems and methods for converting values into spikes for transmission in a neural network.

Background

[0002] An artificial neural network, which may comprise an interconnected group of artificial neurons (i.e., neuron models), is a computational device or represents a method to be performed by a computational device. Artificial neural networks may have corresponding structure and/or function in biological neural networks. However, artificial neural networks may provide innovative and useful computational techniques for certain applications in which traditional computational techniques are cumbersome, impractical, or inadequate. Because artificial neural networks can infer a function from observations, such networks are particularly useful in applications where the complexity of the task or data makes the design of the function by conventional techniques burdensome.

[0003] Execution of large neural models may span multiple neural processors. The information shared between neural processors may be limited to neural spikes. Still, the model may specify for the use of non-spikes values (e.g., neuromodulators) and for those values to be synchronized across neural processors for proper execution. Thus, it is desirable to provide a neuromorphic mechanism to synchronize values across neural processors of a neural network.

SUMMARY

[0004] In an aspect of the present disclosure, a method for transmitting values in a neural network is disclosed. The method includes obtaining a parameter value and encoding the parameter value based on at least one value used by a neuron. The encoding is based on a spike(s) to be transmitted via a spike channel.

[0005] In another aspect of the present disclosure, a method for receiving parameter values in a neural network is disclosed. The method includes determining which neuron will receive a spike representing an encoded value. The method also includes decoding a spike(s) to determine a parameter value used by a neuron.

[0006] In yet another aspect of the present disclosure, an apparatus for transmitting values in a neural network is disclosed. The apparatus includes a memory and a processor(s) coupled to the memory. The processor(s) is (are) configured to obtain a parameter value. The processor(s) is (are) also configured to encode the parameter value based on a value(s) used by a neuron. The encoding of the parameter value is based on a spike(s) to be transmitted via a spike channel.

[0007] In still another aspect of the present disclosure, an apparatus for receiving parameter values in a neural network is disclosed. The apparatus includes a memory and a processor(s) coupled to the memory. The processor(s) is (are) configured to determine which neuron will receive a spike representing an encoded value. The processor is further configured to decode at least one spike to determine a parameter value used by a neuron.

[0008] In yet still another aspect of the present disclosure, an apparatus for transmitting values in a neural network is disclosed. The apparatus includes means for obtaining a parameter value. The apparatus also includes means for encoding the parameter value based on at least one value used by a neuron. The encoding is based on a spike(s) to be transmitted via a spike channel.

[0009] In a further aspect of the present disclosure, an apparatus for receiving parameter values in a neural network is disclosed. The apparatus includes means for determining which neuron will receive a spike representing an encoded value. The apparatus also includes means for decoding a spike(s) to determine a parameter value used by a neuron.

[0010] In an aspect of the present disclosure, a computer program product for transmitting values in a neural network is disclosed. The computer program product includes a non-transitory computer readable medium having encoded thereon program code. The program code includes program code to obtain a parameter value and program code to encode the parameter value based on at least one value used by a neuron. The encoding is based on a spike(s) to be transmitted via a spike channel.

[0011] In yet another aspect, a computer program product for receiving parameter values in a neural network is disclosed. The computer program product includes a non-transitory computer readable medium having encoded thereon program code. The program code includes program code to determine which neuron will receive a spike representing an encoded value. The program code also includes program code to decode a spike(s) to determine a parameter value used by a neuron.

[0012] This has outlined, rather broadly, the features and technical advantages of the present disclosure in order that the detailed description that follows may be better understood. Additional features and advantages of the disclosure will be described below. It should be appreciated by those skilled in the art that this disclosure may be readily utilized as a basis for modifying or designing other structures for carrying out the same purposes of the present disclosure. It should also be realized by those skilled in the art that such equivalent constructions do not depart from the teachings of the disclosure as set forth in the appended claims. The novel features, which are believed to be characteristic of the disclosure, both as to its organization and method of operation, together with further objects and advantages, will be better understood from the following description when considered in connection with the accompanying figures. It is to be expressly understood, however, that each of the figures is provided for the purpose of illustration and description only and is not intended as a definition of the limits of the present disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The features, nature, and advantages of the present disclosure will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference characters identify correspondingly throughout.

[0014] FIGURE 1 illustrates an example network of neurons in accordance with certain aspects of the present disclosure.

[0015] FIGURE 2 illustrates an example of a processing unit (neuron) of a computational network (neural system or neural network) in accordance with certain aspects of the present disclosure.

[0016] FIGURE 3 illustrates an example of a spike-timing dependent plasticity (STDP) curve in accordance with certain aspects of the present disclosure.

[0017] FIGURE 4 illustrates an example of a positive regime and a negative regime for defining behavior of a neuron model in accordance with certain aspects of the present disclosure.

[0018] FIGURE 5 is a high level block diagram illustrating an exemplary system architecture for synchronizing values between neural processors in a neural network in accordance with aspects of the present disclosure.

[0019] FIGURE 6 is a high level block diagram illustrating an exemplary system architecture for synchronizing values between neural processors in a neural network in accordance with aspects of the present disclosure.

[0020] FIGURE 7A is a high level block diagram illustrating an exemplary system for encoding and decoding spikes in accordance with aspects of the present disclosure.

[0021] FIGURE 7B shows a pair of graphs illustrating exemplary encoding techniques in accordance with aspects of the present disclosure.

[0022] FIGURE 8 illustrates an example implementation of a method for synchronizing values across processing blocks in a neural network using a general-purpose processor in accordance with certain aspects of the present disclosure.

[0023] FIGURE 9 illustrates an example implementation for synchronizing values across processing blocks of the neural network in accordance with certain aspects of the present disclosure.

[0024] FIGURE 10 illustrates an example implementation of the aforementioned method for synchronizing values across processing blocks of a neural network in accordance with certain aspects of the present disclosure.

[0025] FIGURE 11 illustrates a method for converting values to spikes for transmission in a neural network in accordance with certain aspects of the present disclosure.

[0026] FIGURE 12 illustrates a method for receiving a parameter value in a neural network in accordance with certain aspects of the present disclosure.

[0027] FIGURE 13 illustrates an example implementation of a neural network in accordance with certain aspects of the present disclosure.

DETAILED DESCRIPTION

[0028] The detailed description set forth below, in connection with the appended drawings, is intended as a description of various configurations and is not intended to represent the only configurations in which the concepts described herein may be practiced. The detailed description includes specific details for the purpose of providing a thorough understanding of the various concepts. However, it will be apparent to those skilled in the art that these concepts may be practiced without these specific details. In some instances, well-known structures and components are shown in block diagram form in order to avoid obscuring such concepts.

[0029] Based on the teachings, one skilled in the art should appreciate that the scope of the disclosure is intended to cover any aspect of the disclosure, whether implemented independently of or combined with any other aspect of the disclosure. For example, an apparatus may be implemented or a method may be practiced using any number of the aspects set forth. In addition, the scope of the disclosure is intended to cover such an apparatus or method practiced using other structure, functionality, or structure and functionality in addition to or other than the various aspects of the disclosure set forth. It should be understood that any aspect of the disclosure disclosed may be embodied by one or more elements of a claim.

[0030] The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any aspect described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other aspects.

[0031] Although particular aspects are described herein, many variations and permutations of these aspects fall within the scope of the disclosure. Although some benefits and advantages of the preferred aspects are mentioned, the scope of the disclosure is not intended to be limited to particular benefits, uses or objectives. Rather, aspects of the disclosure are intended to be broadly applicable to different technologies, system configurations, networks and protocols, some of which are illustrated by way of example in the figures and in the following description of the preferred aspects. The detailed description and drawings are merely illustrative of the disclosure rather than limiting, the scope of the disclosure being defined by the appended claims and equivalents thereof.

AN EXAMPLE NEURAL SYSTEM, TRAINING AND OPERATION

[0032] FIGURE 1 illustrates an example artificial neural system 100 with multiple levels of neurons in accordance with certain aspects of the present disclosure. The neural system 100 may have a level of neurons 102 connected to another level of neurons 106 through a network of synaptic connections 104 (i.e., feed-forward connections). For simplicity, only two levels of neurons are illustrated in FIGURE 1, although fewer or more levels of neurons may exist in a neural system. It should be noted that some of the neurons may connect to other neurons of the same layer through lateral connections. Furthermore, some of the neurons may connect back to a neuron of a previous layer through feedback connections.

[0033] As illustrated in FIGURE 1, each neuron in the level 102 may receive an input signal 108 that may be generated by neurons of a previous level (not shown in FIGURE 1). The signal 108 may represent an input current of the level 102 neuron. This current may be accumulated on the neuron membrane to charge a membrane potential. When the membrane potential reaches its threshold value, the neuron may fire and generate an output spike to be transferred to the next level of neurons (e.g., the level 106). In some modeling approaches, the neuron may continuously transfer a signal to the next level of neurons. This signal is typically a function of the membrane potential. Such behavior can be emulated or simulated in hardware and/or software, including analog and digital implementations such as those described below.

[0034] In biological neurons, the output spike generated when a neuron fires is referred to as an action potential. This electrical signal is a relatively rapid, transient, nerve impulse, having an amplitude of roughly 100 mV and a duration of about 1 ms. In a particular embodiment of a neural system having a series of connected neurons (e.g., the transfer of spikes from one level of neurons to another in FIGURE 1), every action potential has basically the same amplitude and duration, and thus, the information in the signal may be represented only by the frequency and number of spikes, or the time of spikes, rather than by the amplitude. The information carried by an action potential may be determined by the spike, the neuron that spiked, and the time of the spike relative to other spike or spikes. The importance of the spike may be determined by a weight applied to a connection between neurons, as explained below.

[0035] The transfer of spikes from one level of neurons to another may be achieved through the network of synaptic connections (or simply "synapses") 104, as illustrated in FIGURE 1. Relative to the synapses 104, neurons of level 102 may be considered pre-synaptic neurons and neurons of level 106 may be considered post-synaptic neurons. The synapses 104 may receive output signals (i.e., spikes) from the level 102 neurons and scale those signals according to adjustable synaptic weights $w_1^{(i,i+1)}, \ldots, w_P^{(i,i+1)}$ where P is a total number of synaptic connections between the neurons of levels 102 and 106 and is an indicator of the neuron level. For example, in the example of FIGURE 1, i represents neuron level 102 and i+1 represents neuron level 106. Further, the scaled signals may be combined as an input signal of each neuron in the level 106. Every neuron in the level 106 may generate output spikes 110 based on the corresponding combined input signal. The output spikes 110 may be transferred to another level of neurons using another network of synaptic connections (not shown in FIGURE 1).

[0036] Biological synapses can mediate either excitatory or inhibitory (hyperpolarizing) actions in postsynaptic neurons and can also serve to amplify neuronal signals. Excitatory signals depolarize the membrane potential (i.e., increase the membrane potential with respect to the resting potential). If enough excitatory signals are received within a certain time period to depolarize the membrane potential above a threshold, an action potential occurs in the postsynaptic neuron. In contrast, inhibitory signals generally hyperpolarize (i.e., lower) the membrane potential. Inhibitory signals, if strong enough, can counteract the sum of excitatory signals and prevent the membrane potential from reaching a threshold. In addition to counteracting synaptic excitation, synaptic inhibition can exert powerful control over spontaneously active neurons. A spontaneously active neuron refers to a neuron that spikes without further input, for example due to its dynamics or a feedback. By suppressing the spontaneous generation of action potentials in these neurons, synaptic inhibition can shape the pattern of firing in a neuron, which is generally referred to as sculpturing. The various synapses 104 may act as any combination of excitatory or inhibitory synapses, depending on the behavior desired.

[0037] The neural system 100 may be emulated by a general purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device (PLD), discrete gate or transistor logic, discrete hardware components, a software module executed by a processor, or any combination thereof. The neural system 100 may be utilized in a large range of applications, such as image and pattern recognition, machine learning, motor control, and alike. Each neuron in the neural system 100 may be implemented as a neuron circuit. The neuron membrane charged to the threshold value initiating the output spike may be implemented, for example, as a capacitor that integrates an electrical current flowing through it.

[0038] In an aspect, the capacitor may be eliminated as the electrical current integrating device of the neuron circuit, and a smaller memristor element may be used in its place. This approach may be applied in neuron circuits, as well as in various other applications where bulky capacitors are utilized as electrical current integrators. In addition, each of the synapses 104 may be implemented based on a memristor element, where synaptic weight changes may relate to changes of the memristor resistance. With nanometer feature-sized memristors, the area of a neuron circuit and synapses may be substantially reduced, which may make implementation of a large-scale neural system hardware implementation more practical.

[0039] Functionality of a neural processor that emulates the neural system 100 may depend on weights of synaptic connections, which may control strengths of connections between neurons. The synaptic weights may be stored in a non-volatile memory in order to preserve functionality of the processor after being powered down. In an aspect, the synaptic weight memory may be implemented on a separate external chip from the main neural processor chip. The synaptic weight memory may be packaged separately from the neural processor chip as a replaceable memory card. This may provide diverse functionalities to the neural processor, where a particular functionality may be based on synaptic weights stored in a memory card currently attached to the neural processor.

[0040] FIGURE 2 illustrates an example 200 of a processing unit (e.g., a neuron or neuron circuit) 202 of a computational network (e.g., a neural system or a neural network) in accordance with certain aspects of the present disclosure. For example, the neuron 202 may correspond to any of the neurons of levels 102 and 106 from FIGURE 1. The neuron 202 may receive multiple input signals 204_1-204_N (X_1-X_N), which may be signals external to the neural system, or signals generated by other neurons of the same neural system, or both. The input signal may be a current, a conductance, or a voltage, real-valued or complex-valued. The input signal may comprise a numerical value with a fixed-point or a floating-point representation. These input signals may be delivered to the neuron 202 through synaptic connections that scale the signals according to adjustable synaptic weights 206_1-206_N (W_1-W_N), where N may be a total number of input connections of the neuron 202.

[0041] The neuron 202 may combine the scaled input signals and use the combined scaled inputs to generate an output signal 208 (i.e., a signal Y). The output signal 208 may be a current, a conductance, or a voltage, real-valued or complex-valued. The output signal may be a numerical value with a fixed-point or a floating-point representation. The output signal 208 may be then transferred as an input signal to other neurons of the same neural system, or as an input signal to the same neuron 202, or as an output of the neural system.

[0042] The processing unit (neuron) 202 may be emulated by an electrical circuit, and its input and output connections may be emulated by electrical connections with synaptic circuits. The processing unit 202 and its input and output connections may also be emulated by a software code. The processing unit 202 may also be emulated by an electric circuit, whereas its input and output connections may be emulated by a software code. In an aspect, the processing unit 202 in the computational network may be an analog electrical circuit. In another aspect, the processing unit 202 may be a digital electrical circuit. In yet another aspect, the processing unit 202 may be a mixed-signal electrical circuit with both analog and digital components. The computational network may include processing units in any of the aforementioned forms. The computational network (neural system or neural network) using such processing units may be utilized in a large range of applications, such as image and pattern recognition, machine learning, motor control, and the like.

[0043] During the course of training a neural network, synaptic weights (e.g., the weights $w_1^{(i,i+1)}, \ldots, w_P^{(i,i+1)}$ from FIGURE 1 and/or the weights 206_1-206_N from FIGURE 2) may be initialized with random values and increased or decreased according to a learning rule. Those skilled in the art will appreciate that examples of the learning rule include, but are not limited to the spike-timing-dependent plasticity (STDP) learning rule, the Hebb rule, the Oja rule, the Bienenstock-Copper-Munro (BCM) rule, etc. In certain aspects, the weights may settle or converge to one of two values (i.e., a bimodal distribution of weights). This effect can be utilized to reduce the number of bits for each synaptic weight, increase the speed of reading and writing from/to a memory storing the synaptic weights, and to reduce power and/or processor consumption of the synaptic memory.

Synapse Type

[0044] In hardware and software models of neural networks, processing of synapse related functions can be based on synaptic type. Synapse types may comprise non-plastic synapses (no changes of weight and delay), plastic synapses (weight may change), structural delay plastic synapses (weight and delay may change), fully plastic synapses (weight, delay and connectivity may change), and variations thereupon (e.g., delay may change, but no change in weight or connectivity). The advantage of this is that processing can be subdivided. For example, non-plastic synapses may not require plasticity functions to be executed (or waiting for such functions to complete). Similarly, delay and weight plasticity may be subdivided into operations that may operate together or separately, in sequence or in parallel. Different types of synapses may have different lookup tables or formulas and parameters for each of the different plasticity types that apply. Thus, the methods would access the relevant tables, formulas, or parameters for the synapse's type.

[0045] There are further implications of the fact that spike-timing dependent structural plasticity may be executed independently of synaptic plasticity. Structural plasticity may be executed even if there is no change to weight magnitude (e.g., if the weight has reached a minimum or maximum value, or it is not changed due to some other reason) since structural plasticity (i.e., an amount of delay change) may be a direct function of pre-post spike time difference. Alternatively, it may be set as a function of the weight change amount or based on conditions relating to bounds of the weights or weight changes. For example, a synapse delay may change only when a weight change occurs or if weights reach zero but not if they are maxed out. However, it can be advantageous to have independent functions so that these processes can be parallelized reducing the number and overlap of memory accesses.

DETERMINATION OF SYNAPTIC PLASTICITY

[0046] Neuroplasticity (or simply "plasticity") is the capacity of neurons and neural networks in the brain to change their synaptic connections and behavior in response to new information, sensory stimulation, development, damage, or dysfunction. Plasticity is important to learning and memory in biology, as well as for computational neuroscience and neural networks. Various forms of plasticity have been studied, such as synaptic plasticity (e.g., according to the Hebbian theory), spike-timing-dependent plasticity (STDP), non-synaptic plasticity, activity-dependent plasticity, structural plasticity and homeostatic plasticity.

[0047] STDP is a learning process that adjusts the strength of synaptic connections between neurons. The connection strengths are adjusted based on the relative timing of a particular neuron's output and received input spikes (i.e., action potentials). Under the STDP process, long-term potentiation (LTP) may occur if an input spike to a certain neuron tends, on average, to occur immediately before that neuron's output spike. Then, that particular input is made somewhat stronger. On the other hand, long-term depression (LTD) may occur if an input spike tends, on average, to occur immediately after an output spike. Then, that particular input is made somewhat weaker, and hence the name "spike-timing-dependent plasticity". Consequently, inputs that might be the cause of the post-synaptic neuron's excitation are made even more likely to contribute in the future, whereas inputs that are not the cause of the post-synaptic spike are made less likely to contribute in the future. The process continues until a subset of the initial set of connections remains, while the influence of all others is reduced to an insignificant level.

[0048] Since a neuron generally produces an output spike when many of its inputs occur within a brief period, i.e., being cumulative sufficient to cause the output, the subset of inputs that typically remains includes those that tended to be correlated in time. In addition, since the inputs that occur before the output spike are strengthened, the inputs that provide the earliest sufficiently cumulative indication of correlation will eventually become the final input to the neuron.

[0049] The STDP learning rule may effectively adapt a synaptic weight of a synapse connecting a pre-synaptic neuron to a post-synaptic neuron as a function of time difference between spike time t_{pre} of the pre-synaptic neuron and spike time t_{post} of the post-synaptic neuron (i.e., $t = t_{post} - t_{pre}$). A typical formulation of the STDP is to increase the synaptic weight (i.e., potentiate the synapse) if the time difference is positive (the pre-synaptic neuron fires before the post-synaptic neuron), and decrease the synaptic weight (i.e., depress the synapse) if the time difference is negative (the post-synaptic neuron fires before the pre-synaptic neuron).

[0050] In the STDP process, a change of the synaptic weight over time may be typically achieved using an exponential decay, as given by,

$$\Delta w(t) = \begin{cases} a_+ e^{-t/k_+} + \mu, t > 0 \\ a_- e^{t/k_-}, t < 0 \end{cases}, \qquad (1)$$

where k_+ and k_- are time constants for positive and negative time difference, respectively, a_+ and a_- are corresponding scaling magnitudes, and μ is an offset that may be applied to the positive time difference and/or the negative time difference.

[0051] FIGURE 3 illustrates an example graph diagram 300 of a synaptic weight change as a function of relative timing of pre-synaptic and post-synaptic spikes in accordance with the STDP. If a pre-synaptic neuron fires before a post-synaptic neuron, then a corresponding synaptic weight may be increased, as illustrated in a portion 302 of the graph 300. This weight increase can be referred to as an LTP of the synapse. It can be observed from the graph portion 302 that the amount of LTP may decrease roughly exponentially as a function of the difference between pre-synaptic and post-synaptic spike times. The reverse order of firing may reduce the synaptic weight, as illustrated in a portion 304 of the graph 300, causing an LTD of the synapse.

[0052] As illustrated in the graph 300 in FIGURE 3, a negative offset μ may be applied to the LTP (causal) portion 302 of the STDP graph. A point of cross-over 306 of the x-axis (y=0) may be configured to coincide with the maximum time lag for considering correlation for causal inputs from layer i-1. In the case of a frame-based input (i.e., an input that is in the form of a frame of a particular duration comprising spikes or pulses), the offset value μ can be computed to reflect the frame boundary. A first input spike (pulse) in the frame may be considered to decay over time either as modeled by a post-synaptic potential directly or in terms of the effect on neural state. If a second input spike (pulse) in the frame is considered correlated or relevant of a particular time frame, then the relevant times before and after the frame may be separated at that time frame boundary and treated differently in plasticity terms by offsetting one or more parts of the STDP curve such that the value in the relevant times may be different (e.g., negative for greater than one frame and positive for less than one frame). For example, the negative offset μ may be set to offset LTP such that the curve actually goes below zero at a pre-post time greater than the frame time and it is thus part of LTD instead of LTP.

NEURON MODELS AND OPERATION

[0053] There are some general principles for designing a useful spiking neuron model. A good neuron model may have rich potential behavior in terms of two computational regimes: coincidence detection and functional computation. Moreover, a good neuron model should have two elements to allow temporal coding: arrival time of inputs affects output time and coincidence detection can have a narrow time window. Finally, to be computationally attractive, a good neuron model may have a closed-form solution in continuous time and stable behavior including near attractors and saddle points. In other words, a useful neuron model is one that is practical and that can be used to model rich, realistic and biologically-consistent behaviors, as well as be used to both engineer and reverse engineer neural circuits.

[0054] A neuron model may depend on events, such as an input arrival, output spike or other event whether internal or external. To achieve a rich behavioral repertoire, a state machine that can exhibit complex behaviors may be desired. If the occurrence of an event itself, separate from the input contribution (if any) can influence the state machine and constrain dynamics subsequent to the event, then the future state of the system is not only a function of a state and input, but rather a function of a state, event, and input.

[0055] In an aspect, a neuron *n* may be modeled as a spiking leaky-integrate-and-fire neuron with a membrane voltage $v_n(t)$ governed by the following dynamics,

$$\frac{dv_n(t)}{dt} = \alpha v_n(t) + \beta \sum_m w_{m,n} y_m(t - \Delta t_{m,n}), \qquad (2)$$

where α and β are parameters, $w_{m,n}$ is a synaptic weight for the synapse connecting a pre-synaptic neuron *m* to a post-synaptic neuron *n*, and $y_m(t)$ is the spiking output of the neuron *m* that may be delayed by dendritic or axonal delay according to $\Delta t_{m,n}$ until arrival at the neuron *n*'s soma.

[0056] It should be noted that there is a delay from the time when sufficient input to a post-synaptic neuron is established until the time when the post-synaptic neuron actually fires. In a dynamic spiking neuron model, such as Izhikevich's simple model, a time delay may be incurred if there is a difference between a depolarization threshold v_t and a peak spike voltage v_{peak}. For example, in the simple model, neuron soma dynamics can be governed by the pair of differential equations for voltage and recovery, i.e.,

$$\frac{dv}{dt} = (k(v - v_t)(v - v_r) - u + I)/C, \qquad (3)$$

$$\frac{du}{dt} = a(b(v - v_r) - u). \qquad (4)$$

where v is a membrane potential, u is a membrane recovery variable, k is a parameter that describes time scale of the membrane potential v, a is a parameter that describes time scale of the recovery variable u, b is a parameter that describes sensitivity of the recovery variable u to the sub-threshold fluctuations of the membrane potential v, v_r is a membrane resting potential, I is a synaptic current, and C is a membrane's capacitance. In accordance with this model, the neuron is defined to spike when $v > v_{peak}$.

Hunzinger Cold Model

[0057] The Hunzinger Cold neuron model is a minimal dual-regime spiking linear dynamical model that can reproduce a rich variety of neural behaviors. The model's one- or two-dimensional linear dynamics can have two regimes, wherein the time constant (and coupling) can depend on the regime. In the sub-threshold regime, the time constant, negative by convention, represents leaky channel dynamics generally acting to return a cell to rest in a biologically-consistent linear fashion. The time constant in the supra-threshold regime, positive by convention, reflects anti-leaky channel dynamics generally driving a cell to spike while incurring latency in spike-generation.

[0058] As illustrated in FIGURE 4, the dynamics of the model may be divided into two (or more) regimes. These regimes may be called the negative regime 402 (also interchangeably referred to as the leaky-integrate-and-fire (LIF) regime, not to be confused with the LIF neuron model) and the positive regime 404 (also interchangeably referred to as the anti-leaky-integrate-and-fire (ALIF) regime, not to be confused with the ALIF neuron model). In the negative regime 402, the state tends toward rest (v_-) at the time of a future event. In this negative regime, the model generally exhibits temporal input detection properties and other sub-threshold behavior. In the positive regime 404, the state tends toward a spiking event (v_s). In this positive regime, the model exhibits computational properties, such as incurring a latency to spike depending on subsequent input events. Formulation of dynamics in terms of events and separation of the dynamics into these two regimes are fundamental characteristics of the model.

[0059] Linear dual-regime bi-dimensional dynamics (for states v and u) may be defined by convention as,

$$\tau_\rho \frac{dv}{dt} = v + q_\rho \qquad (5)$$

$$-\tau_u \frac{du}{dt} = u + r \qquad (6)$$

where q_ρ and r are the linear transformation variables for coupling.

[0060] The symbol ρ is used herein to denote the dynamics regime with the convention to replace the symbol ρ with the sign "-" or "+" for the negative and positive regimes, respectively, when discussing or expressing a relation for a specific regime.

[0061] The model state is defined by a membrane potential (voltage) v and recovery current u. In basic form, the regime is essentially determined by the model state. There are subtle, but important aspects of the precise and general definition, but for the moment, consider the model to be in the positive regime 404 if the voltage v is above a threshold (v_+) and otherwise in the negative regime 402.

[0062] The regime-dependent time constants include τ_- which is the negative regime time constant, and τ_+ which is the positive regime time constant. The recovery current time constant τ_u is typically independent of regime. For convenience, the negative regime time constant τ_- is typically specified as a negative quantity to reflect decay so that the same expression for voltage evolution may be used as for the positive regime in which the exponent and τ_+ will generally be positive, as will be τ_u.

[0063] The dynamics of the two state elements may be coupled at events by transformations offsetting the states from their null-clines, where the transformation variables are

$$q_\rho = -\tau_\rho \beta u - v_\rho \qquad (7)$$

$$r = \delta(v + \varepsilon) \qquad (8)$$

where δ, ε, β and v_-, v_+ are parameters. The two values for v_ρ are the base for reference voltages for the two regimes. The parameter v_- is the base voltage for the negative regime, and the membrane potential will generally decay toward v_- in the negative regime. The parameter v_+ is the base voltage for the positive regime, and the membrane potential will generally tend away from v_+ in the positive regime.

[0064] The null-clines for v and u are given by the negative of the transformation variables q_ρ and r, respectively. The parameter δ is a scale factor controlling the slope of the u null-cline. The parameter ε is typically set equal to $-v_-$. The parameter β is a resistance value controlling the slope of the v null-clines in both regimes. The τ_ρ time-constant parameters control not only the exponential decays, but also the null-cline slopes in each regime separately.

[0065] The model may be defined to spike when the voltage v reaches a value v_S. Subsequently, the state may be reset at a reset event (which may be one and the same as the spike event):

$$v = \hat{v}_- \qquad (9)$$

$$u = u + \Delta u \qquad (10)$$

where $\hat{v}_-$ and Δu are parameters. The reset voltage $\hat{v}_-$ is typically set to v_-.

[0066] By a principle of momentary coupling, a closed form solution is possible not only for state (and with a single exponential term), but also for the time required to reach a particular state. The close form state solutions are

$$v(t+\Delta t) = \left(v(t) + q_\rho\right)e^{\frac{\Delta t}{\tau_\rho}} - q_\rho \qquad (11)$$

$$u(t+\Delta t) = \left(u(t) + r\right)e^{-\frac{\Delta t}{\tau_u}} - r \qquad (12)$$

[0067] Therefore, the model state may be updated only upon events such as upon an input (pre-synaptic spike) or output (post-synaptic spike). Operations may also be performed at any particular time (whether or not there is input or output).

[0068] Moreover, by the momentary coupling principle, the time of a post-synaptic spike may be anticipated so the time to reach a particular state may be determined in advance without iterative techniques or Numerical Methods (e.g., the Euler numerical method). Given a prior voltage state v_0, the time delay until voltage state v_f is reached is given by

$$\Delta t = \tau_\rho \log \frac{v_f + q_\rho}{v_0 + q_\rho} \qquad (13)$$

[0069] If a spike is defined as occurring at the time the voltage state v reaches v_S, then the closed-form solution for the amount of time, or relative delay, until a spike occurs as measured from the time that the voltage is at a given state v is

$$\Delta t_S = \begin{cases} \tau_+ \log \frac{v_S + q_+}{v + q_+} & \textit{if} \quad v > \hat{v}_+ \\ \infty & \textit{otherwise} \end{cases} \qquad (14)$$

where $\hat{v}_+$ is typically set to parameter v_+, although other variations may be possible.

[0070] The above definitions of the model dynamics depend on whether the model is in the positive or negative regime. As mentioned, the coupling and the regime ρ may be computed upon events. For purposes of state propagation, the regime and coupling (transformation) variables may be defined based on the state at the time of the last (prior) event. For purposes of subsequently anticipating spike output time, the regime and coupling variable may be defined based on the state at the time of the next (current) event.

[0071] There are several possible implementations of the Cold model, and executing the simulation, emulation or model in time. This includes, for example, event-update, step-event update, and step-update modes. An event update is an update where states are updated based on events or "event update" (at particular moments). A step update is an update when the model is updated at intervals (e.g., 1ms). This does not necessarily require iterative methods or Numerical methods. An event-based implementation is also possible at a limited time resolution in a step-based simulator by only updating the model if an event occurs at or between steps or by "step-event" update.

VALUE SYNCHRONIZATION ACROSS NEURAL PROCESSORS

[0072] Aspects of the present disclosure are directed to synchronizing values in a neural network over a spike interface. FIGURE 5 is a high level block diagram illustrating an exemplary system architecture for synchronizing values between neural processors in a neural network. The system architecture 500 comprises neural processors 502 and 522 that may be utilized alone or in combination to emulate a neural system. Further, the neural processors 502 and 522 may be included in the same processing chip or may be provided in separate processing chips. For ease of illustration and explanation, the system architecture 500 is shown as including two neural processors (502 and 522). However, this is merely exemplary, and additional neural processors or processing blocks may be included in the system architecture for processing in the neural network.

Neural processor 502 may comprise a value generator (VG) 504. The value generator 504 may be configured to generate values to be shared with neurons in the system for modeling neuron dynamics. In some aspects, the value may be a neuron parameter, a synaptic weight or delay value, or other value or attribute for use in emulating a neural system. For example, the value may correspond to a neuromodulator value such as a common dopamine value to be applied to neurons across the neural network. In yet another example, the value may correspond to identification information for a neuron or neurons (e.g, 508) that have fired. In some aspects, the value may further include timing information, for example, to indicate a time (τ) at which a particular neuron fires or a timing at which a value is to be applied or consumed by a neuron. There may be one value generator 504, 524 for each processing block 502, 522 (as shown), or there may be multiple value generators 504, 524 for each processing block 502, 522. For example there can be one value generator 504, 524 for each neuron 508, 528, or even one value generator 504, 524 for each neuron type or neuron cluster within each processing block 502, 522.

[0073] The value generator 504 may be configured to perform a value calculation to generate values based, for example, on neural properties such as spikes or other attributes (e.g., synapse weight and/or delay). In some aspects, neurons 508 may send spikes to the value generator 504 to affect the value calculation. Additionally, neurons of remote processors (e.g., 522) in the neural system may also send spikes to the value generator 504 to affect the value calculation. Further, while FIGURE 5 shows only one value generator in a processing block, this is merely exemplary and neural processor 502 (as well as neural processor 522) could be configured with additional value generators. For example, the neural processors 502, 522 could be configured with a value generator for each neuron or neuron type.

[0074] The neural processor 502 may also include value neurons (VNs) 506a, 506b, 506c (collectively value neurons 506). The value neurons 506 may be configured to generate spikes. The spikes are similar to a binary value. That is, they are either on or off. In some aspects, the value neurons 506 generate spikes that correspond to values generated by the value generator 504. That is, the value neurons 506 may produce output spikes encoded with the value generated by the value generator 504 based on a spike protocol. For example, the value neurons 506 may encode the spikes using an inter-spike interval (ISI), binary encoding or other protocol for generating spikes.

[0075] In some aspects, one or more of value neurons 506 may be used to manage a value to be shared with other neurons in the neural network. For example, one or more of the value neurons 506 may monitor a value (e.g., common dopamine value) used by neurons 508. If adjustments are made to the value, the value neurons 506 may be used to update other neurons (e.g., 528) to utilize the value with respect to the change.

[0076] The neural processor 502 may further comprise one or more neurons 508a, 508b (which may be collectively referred to as neurons 508). The neurons 508 may receive spike inputs and consume values to model aspects of neuron behavior or dynamics in a neural network. In turn, the neurons 508 may output spikes to affect other neurons in the neural network. In some aspects, the neurons 508 may also send spikes to the value neurons 506 to adjust the value generator 504. For example, the neurons 508 may send spikes to the value neurons 506 to affect (e.g., delay) value generation. The neurons 508 shown in FIGURE 5 may also represent neuron types, rather than individual neurons.

[0077] The neural processor 502 may be configured to transmit information to and receive information from remote neural processors (e.g., 522) in the neural network via an interface (not shown). In some configurations, the interface may comprise a network of synapses as illustrated in FIGURE 1. In some aspects, the interface may be configured to transmit and receive spikes only. In such configurations, the scalar values generated by the value generator 504 cannot be directly transmitted to the remote neural processors (e.g., 522). However, because spikes may be transmitted via the interface, information regarding the values generated by the value generator 504 may be communicated to remote processors in the form of spikes produced by the value neurons 506. That is, the neural processor 502 may share a value generated by the value generator 504 with a remote neural processor (e.g., 522) by encoding the value into spikes using the value neurons 506 and transmitting the spikes to the remote neural processor 522.

[0078] To receive the transmitted spikes from the neural processor 502, neural processors 522 may comprise proxy neurons (P) 526a, 526b, and 526c (collectively referred to as proxy neurons 526). The proxy neurons 526 may be configured to receive spikes from the value neurons (e.g., 506). The proxy neurons 526 may provide the spikes and/or other properties (e.g., neuron state) to a value generator 524. In doing so, the proxy neurons 526 may, in some aspects, drive the value generator 524 to generate a value on the remote neural processor 522 based on the received spikes.

[0079] The value generator 524, may in turn, perform a value calculation to generate a value based on the received spikes and/or other properties. In some aspects, the value generator 524 may be configured to perform a value calculation to generate a value such that the value is synchronized with a first value generated by value generator 504. Further, in some aspects, the value generator 524 may be configured to generate a value that is the same as that generated by the value generator 504.

[0080] One or more of neurons 528a, 528b, 528c (may be collectively referred to as neurons 528) may consume the value generated by value generator 524 to further model aspects of neuron behavior or dynamics in the neural network.

[0081] In some aspects, neural processor 522 may access a connectivity lookup table to determine routing of the value generated by the value generator 524. The connectivity lookup table may provide source and destination information for the generated values. That is, the connectivity lookup table may identify the neurons that are to consume a particular value.

[0082] In some aspects, a connectivity look up table may be used to determine routing for the values generated via the value generators (e.g., 504, 524). The connectivity lookup table may include source and destination information and may be used to determine which neurons (e.g., 508, 528) are to receive the value generated. For example, when the value generated by the value generator 524 identifies pre-synaptic neurons that have fired, the connectivity lookup table may be used to determine the neurons 528 to receive contribution from the pre-synaptic neurons that fired. In another example, when the value generated by the value generator 524 corresponds to a shared neuromodulator value (e.g., a common dopamine value), the connectivity table may indicate the neurons 528 to consume the generated value.

[0083] Additionally, in some cases, the neurons 508 and 528 may send spikes to the value neurons (506) to adjust a value generated by the value generator (504). In other cases, the neurons 508 and 528 may send spikes to proxy neurons 526 to adjust a value generated by the value generator 524.

[0084] FIGURE 6 is a high level block diagram illustrating an exemplary system architecture for synchronizing values between neural processors in a neural network. As shown in FIGURE 6, neural processor 502 may be configured with additional proxy neurons 616a, 616b, and 616c (collectively referred to as proxy neurons 616). The proxy neurons 616 may be defined between the value neurons (506) and the value generator (504) of the first neural processor 502. In some aspects, the proxy neurons 616 may be utilized to replicate a delay generated when transmitting the spikes from the first neural processor 502 to the second neural processor 522.

[0085] Further, the neural processor 502 may be configured with a delay generator 626. As illustrated in FIGURE 6, the delay generator 626 may be defined within the neural processor 502. However, this is merely exemplary, and the delay generator 626 may be included in other components of the neural processor 502 or may be provided as a separate component. In some aspects, the delay generator 626 may be used to replicate the delay generated when transmitting the spikes from the neural processor 502 to the second neural processor 522. The delay could approximate the delay between the processors 502, 522 or could include some padding so the approximated delay is longer than the actual delay. In some configurations, neural processor 522 may also be configured with a delay generator to replicate the delay generated when transmitting the spikes from the neural processor 522 to the neural processor 502.

[0086] Furthermore, in some configurations, the value neurons 506 of first neural processor 502 may transmit a specific sequence of spikes to reset the second neural processor 522.

[0087] Neurons on the remote neural processor 522 may access the value provided from the first neural processor 502. Thus, the value generated in the neural processor

502 may be deemed synchronized with the value generated in the remote neural processor 522.

[0088] FIGURE 7A is a high level block diagram illustrating an exemplary system for encoding and decoding spikes. As discussed above, value neurons 506 may monitor or manage a value V1 that is to be shared with neurons across the neural network. In some aspects, the value V1 may provide an indication of the neurons that spiked at a particular time. The value V1 may also be a value that is to be shared by neurons across the neural network such as a neuromodulator value (e.g., common dopamine value).

[0089] In the example of FIGURE 7A, the value neurons 506 manage the value V1. When the value V1 is to be shared with a neuron across the neural network, the value neurons 506 may be used to convert the value V1 to spikes for transmission across the inter-block interface 712. In some aspects, the inter-block interface 712 may be configured such that only spikes may be communicated via the interface, and can be, for example, a network of synapses. Further, the inter-block interface 712 may be configured to operate as a spike channel between neural processors.

[0090] In some aspects, the value may be divided into one or more component parts. For example, the value V1 may be divided into its most significant bits and least significant bits. In another example, the value V1 may be divided into a predefined number of portions (e.g., 1/2 of the bits, 1/3 of the bits, etc.)

[0091] The value neurons 506 may generate spikes encoded with the value V1 based on a spike protocol. The spike protocol may employ an encoding scheme such as, for example, absolute latency coding, relative latency coding, rate coding, ISI (inter-spike interval) coding, binary coding and the like.

[0092] In absolute latency coding, the value may be encoded based on the time between spike events for a particular neuron or set of neurons. For example, to encode a value of 8, an 8 ms delay may be included between spike events for the neuron. In some aspects, the value may also be scaled to generate the encoded value. Further, in some aspects, the encoded value may be a function of the absolute latency value.

[0093] In relative latency coding, the value may be encoded according to the interval between spikes for a plurality of neurons. For example, where a neuron N_1

spikes at a time t_1 and neuron N_2 spikes at a time t_2, the value may be represented as the time difference t_2-t_1.

[0094] In rate coding, the value may be represented according to a number of spikes that occur within a particular interval. For example, spikes may be sampled for a 10 ms interval with the encoded value corresponding to the number of spikes that occurred during the 10 ms period. In some aspects, the value may be encoded based on a spike rate for one neuron or a spike rate for multiple neurons.

[0095] The encoding schemes described above are merely exemplary and in some aspects, the spike protocol may employ Inter-Spike Interval (ISI) coding, binary coding, or other encoding schemes for generating spikes encoded with the value V1.

[0096] Connectivity information indicating a particular neuron or neurons that spiked may also be included in the spikes transmitted via value neurons. The connectivity information may be used to route the values encoded and transmitted as spikes to neurons in a remote neural processor (e.g., 522). In some aspects, the connectivity information may include an index identifying one or more neurons that spiked (i.e., source neuron(s)). The connectivity information may further include destination information identifying one or more neurons that are to receive contributions based on the neuron that spiked.

[0097] The proxy neurons 526 receive the spikes sent from the processing block 502. In some aspects, spikes may be received by additional receiver neurons to provide redundancy to recover from spike transmission issues (e.g. spike loss). For example, in some aspects a spike train transmitted via value neuron 506a may be received via multiple proxy neurons (e.g., (526a, 526b, and/or 526c). In a further example, a spike train received via proxy neurons 526 and neurons 528 of neural processor 522.

[0098] The proxy neurons 526 then provide the spikes, which correspond to the first value or a component thereof, to the value generator 524 which decodes the spikes and generates a second value V2. In some aspects, the value generator 524 may be configured to decode spikes encoded based on the spike protocol employed by value neurons 506. Because the spikes may be encoded with timing information, the second value V2 may be generated such that the second value V2 is synchronized with the first value V1. In some aspects, the second value V2 is the same as or equal to the first value V1.

[0099] In some aspects, a connectivity look up table may be used to determine routing for the generated values. The connectivity lookup table may include source and destination information and may be used to determine which neurons of the neural processor 522 are to receive the value generated by the value generator 524. For example, when the value generated by the value generator 524 includes an index which identifies pre-synaptic neuron or neurons that have fired, the connectivity lookup table may be used to determine the neurons 528 (FIGURES 5 and 6) which are to receive contribution from the pre-synaptic neurons that fired. In another example, when the value generated by the value generator 524 corresponds to a shared neuromodulator value (e.g., a common dopamine value), the connectivity table may indicate the neurons 528 which are to consume the generated value.

[00100] FIGURE 7B shows a pair of graphs 750 and 760 illustrating exemplary encoding techniques in accordance with aspects of the present disclosure. Referring to FIGURE 7B, graph 750 illustrates an example of encoding the value based on an inter-spike interval. That is, a spike train may be configured to represent value information according to a number of time steps between spike events for a neuron. As shown in graph 750, trace 755 is provided to correspond to a value based on intervals between spikes 758 for neuron N_1 over a period of time steps. In some aspects, the value encoded increases for each time step without a spike event. For example, in graph 755, there are two time periods before the first spike event for neuron N_1, thus the spike train shown for N_1 may represent a value of 1 at the first time step and a value of 2 at the second time step. At the third, fourth and fifth time steps, the delay increases, so the value increases. At the sixth time step, and thereafter the delay between spikes of neuron N_1 is only one time period, so the encoded value returns to 1.

[00101] On the other hand, graph 760 illustrates a binary encoding approach in which the value 765 may be represented at each time step based on whether a spike event occurred or not. For example, N_0 represents 1, N_1 represents 2, N_3 represents 4, and N_3 represent 8. Thus, at the first time step, a value of 13 (8+4+1) is encoded. At the next time step, a value of 7 (4+2+1) is encoded, and so forth.

[00102] FIGURE 8 illustrates an example implementation 800 of the aforementioned method for converting values to spikes in a neural network using a general-purpose processor 802 in accordance with certain aspects of the present disclosure. Variables (neural signals), synaptic weights, and system parameters associated with a computational network (neural network) may be stored in a memory block 804, while instructions executed at the general-purpose processor 802 may be loaded from a program memory 806. In an aspect of the present disclosure, the instructions loaded into the general-purpose processor 802 may comprise code for converting values to spikes in a neural network. For example in some configurations, the general-purpose processor 802 may comprise code for obtaining a parameter value. Further, in the exemplary configuration, the general-purpose processor 802 may further comprise code for encoding the parameter value based at least in part on a value used by a neuron.

[00103] In another exemplary configuration, the general-purpose processor 802 may comprise code for determining a neuron to receive spikes representing an encoded value. Further, in this exemplary configuration, the general-purpose processor 802 may further comprise code for decoding the spikes to determine a parameter value to be used by the neuron.

[00104] FIGURE 9 illustrates an example implementation 900 of the aforementioned method for converting values to spikes for transmission in a neural network where a memory 902 can be interfaced via an interconnection network 904 with individual (distributed) processing units (neural processors) 9061…906N of a computational network (neural network) in accordance with certain aspects of the present disclosure. Variables (neural signals), synaptic weights, and system parameters associated with the computational network (neural network) may be stored in the memory 902, and may be loaded from the memory 902 via connection(s) of the interconnection network 904 into each processing unit (neural processor) 906. In some aspects, values generated via the processing blocks as well as a connectivity information may also be stored in memory 902 and loaded therefrom for further processing. In an aspect of the present disclosure, the processing unit 906 may be configured to convert values to spikes. For example, in some configurations, the processing unit 906 may be configured to obtain a parameter value. In addition, the processing unit 906 of the exemplary configuration may be further configured to encode the parameter value based at least in part on a value used by a neuron.

[00105] In another exemplary configuration, the processing unit 906 may be configured to determine a neuron to receive spikes representing an encoded value. Further, in this exemplary configuration, the processing unit 906 may be further configured to decode the spikes to determine a parameter value to be used by the neuron.

[00106] FIGURE 10 illustrates an example implementation 1000 of the aforementioned method for converting a value to spikes for transmission in a neural network. As illustrated in FIGURE 10, one memory bank 1002 may be directly interfaced with one processing unit 1004 of a computational network (neural network). Each memory bank 1002 may store variables (neural signals), synaptic weights, and system parameters associated with a corresponding processing unit (neural processor) 1004. In some aspects, values generated via the processing blocks may also be stored in memory 1002 and loaded therefrom for further processing. Further, in some aspects a connectivity information may be stored in memory 1002. In an aspect of the present disclosure, the processing unit 1004 may be configured to convert the values to spikes.

[00107] FIGURE 11 illustrates a method for converting values to spikes for transmission in a neural network in accordance with certain aspects of the present disclosure. In block 1102, the neuron model obtains a parameter value. Furthermore, in block 1104, the neuron model encodes the parameter value based at least in part on a value used by a neuron.

[00108] FIGURE 12 illustrates a method for receiving a parameter value in a neural network in accordance with certain aspects of the present disclosure. In block 1202, the neuron model determines a neuron to receive spikes representing an encoded value. Furthermore, in block 1204, the neuron model decodes the spikes to determine a parameter value to be used by the neuron.

[00109] FIGURE 13 illustrates an example implementation of a neural network 1300 in accordance with certain aspects of the present disclosure. As illustrated in FIGURE 13, the neural network 1300 may have multiple local processing units 1302 that may perform various operations, as described above. Each processing unit 1302 may comprise a local state memory 1304 and a local parameter memory 1306 that store parameters of the neural network. In addition, the processing unit 1302 may have a memory 1308 with local (neuron) model program, a memory 1310 with local learning program, and a local connection memory 1312. Furthermore, as illustrated in FIGURE 13, each local processing unit 1302 may be interfaced with a unit 1314 for configuration processing that may provide configuration for local memories of the local processing unit, and with routing connection processing elements 1316 that provide routing between the local processing units 1302.

[00110] In one configuration, a neuron model is configured for converting a value to spikes for transmission in a neural network. In one aspect, the model includes an obtaining means and/or encoding means, which may be the general-purpose processor 802, program memory 806, memory block 804, memory 902, interconnection network 904, processing units 906, processing unit 1004, local processing units 1302, and or the routing connection processing elements 1316 configured to perform the functions recited. In one aspect, the aforementioned means may be any module or any apparatus configured to perform the functions recited by the aforementioned means.

[00111] In another configuration, a neuron model is configured for receiving a parameter value. In one aspect, the model includes a determining means and/or decoding means, which may be the general-purpose processor 802, program memory 806, memory block 804, memory 902, interconnection network 904, processing units 906, processing unit 1004, local processing units 1302, and or the routing connection processing elements 1316 configured to perform the functions recited. In one aspect, the aforementioned means may be any module or any apparatus configured to perform the functions recited by the aforementioned means.

[00112] According to certain aspects of the present disclosure, each local processing unit 1302 may be configured to determine parameters of the neural network based upon desired one or more functional features of the neural network, and develop the one or more functional features towards the desired functional features as the determined parameters are further adapted, tuned and updated.

[00113] The various operations of methods described above may be performed by any suitable means capable of performing the corresponding functions. The means may include various hardware and/or software component(s) and/or module(s), including, but not limited to, a circuit, an application specific integrated circuit (ASIC), or processor. Generally, where there are operations illustrated in Figures, those operations may have corresponding counterpart means-plus-function components with similar numbering.

[00114] As used herein, the term "determining" encompasses a wide variety of actions. For example, "determining" may include calculating, computing, processing, deriving, investigating, looking up (e.g., looking up in a table, a database or another data structure), ascertaining and the like. Also, "determining" may include receiving (e.g., receiving information), accessing (e.g., accessing data in a memory) and the like. Also, "determining" may include resolving, selecting, choosing, establishing and the like.

[00115] As used herein, a phrase referring to "at least one of" a list of items refers to any combination of those items, including single members. As an example, "at least one of: *a*, *b*, or *c*" is intended to cover: *a, b, c, a-b, a-c, b-c, and a-b-c.*

[00116] The various illustrative logical blocks, modules and circuits described in connection with the present disclosure may be implemented or performed with a general purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array signal (FPGA) or other programmable logic device (PLD), discrete gate or transistor logic, discrete hardware components or any combination thereof designed to perform the functions described herein. A general-purpose processor may be a microprocessor, but in the alternative, the processor may be any commercially available processor, controller, microcontroller or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

[00117] The steps of a method or algorithm described in connection with the present disclosure may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in any form of storage medium that is known in the art. Some examples of storage media that may be used include random access memory (RAM), read only memory (ROM), flash memory, EPROM memory, EEPROM memory, registers, a hard disk, a removable disk, a CD-ROM and so forth. A software module may comprise a single instruction, or many instructions, and may be distributed over several different code segments, among different programs, and across multiple storage media. A storage medium may be coupled to a processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor.

[00118] The methods disclosed herein comprise one or more steps or actions for achieving the described method. The method steps and/or actions may be interchanged with one another without departing from the scope of the claims. In other words, unless a specific order of steps or actions is specified, the order and/or use of specific steps and/or actions may be modified without departing from the scope of the claims.

[00119] The functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in hardware, an example hardware configuration may comprise a processing system in a device. The processing system may be implemented with a bus architecture. The bus may include any number of interconnecting buses and bridges depending on the specific application of the processing system and the overall design constraints. The bus may link together various circuits including a processor, machine-readable media, and a bus interface. The bus interface may be used to connect a network adapter, among other things, to the processing system via the bus. The network adapter may be used to implement signal processing functions. For certain aspects, a user interface (e.g., keypad, display, mouse, joystick, etc.) may also be connected to the bus. The bus may also link various other circuits such as timing sources, peripherals, voltage regulators, power management circuits, and the like, which are well known in the art, and therefore, will not be described any further.

[00120] The processor may be responsible for managing the bus and general processing, including the execution of software stored on the machine-readable media. The processor may be implemented with one or more general-purpose and/or special-purpose processors. Examples include microprocessors, microcontrollers, DSP processors, and other circuitry that can execute software. Software shall be construed broadly to mean instructions, data, or any combination thereof, whether referred to as software, firmware, middleware, microcode, hardware description language, or otherwise. Machine-readable media may include, by way of example, RAM (Random Access Memory), flash memory, ROM (Read Only Memory), PROM (Programmable

Read-Only Memory), EPROM (Erasable Programmable Read-Only Memory), EEPROM (Electrically Erasable Programmable Read-Only Memory), registers, magnetic disks, optical disks, hard drives, or any other suitable storage medium, or any combination thereof. The machine-readable media may be embodied in a computer-program product. The computer-program product may comprise packaging materials.

[00121] In a hardware implementation, the machine-readable media may be part of the processing system separate from the processor. However, as those skilled in the art will readily appreciate, the machine-readable media, or any portion thereof, may be external to the processing system. By way of example, the machine-readable media may include a transmission line, a carrier wave modulated by data, and/or a computer product separate from the device, all which may be accessed by the processor through the bus interface. Alternatively, or in addition, the machine-readable media, or any portion thereof, may be integrated into the processor, such as the case may be with cache and/or general register files.

[00122] The processing system may be configured as a general-purpose processing system with one or more microprocessors providing the processor functionality and external memory providing at least a portion of the machine-readable media, all linked together with other supporting circuitry through an external bus architecture. Alternatively, the processing system may comprise one or more neuromorphic processors for implementing the neuron models and models of neural systems described herein. As another alternative, the processing system may be implemented with an ASIC (Application Specific Integrated Circuit) with the processor, the bus interface, the user interface, supporting circuitry, and at least a portion of the machine-readable media integrated into a single chip, or with one or more FPGAs (Field Programmable Gate Arrays), PLDs (Programmable Logic Devices), controllers, state machines, gated logic, discrete hardware components, or any other suitable circuitry, or any combination of circuits that can perform the various functionality described throughout this disclosure. Those skilled in the art will recognize how best to implement the described functionality for the processing system depending on the particular application and the overall design constraints imposed on the overall system.

[00123] The machine-readable media may comprise a number of software modules. The software modules include instructions that, when executed by the processor, cause the processing system to perform various functions. The software modules may include a transmission module and a receiving module. Each software module may reside in a single storage device or be distributed across multiple storage devices. By way of example, a software module may be loaded into RAM from a hard drive when a triggering event occurs. During execution of the software module, the processor may load some of the instructions into cache to increase access speed. One or more cache lines may then be loaded into a general register file for execution by the processor. When referring to the functionality of a software module below, it will be understood that such functionality is implemented by the processor when executing instructions from that software module.

[00124] If implemented in software, the functions may be stored or transmitted over as one or more instructions or code on a computer-readable medium. Computer-readable media include both computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A storage medium may be any available medium that can be accessed by a computer. By way of example, and not limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to carry or store desired program code in the form of instructions or data structures and that can be accessed by a computer. Also, any connection is properly termed a computer-readable medium. For example, if the software is transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared (IR), radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. Disk and disc, as used herein, include compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk, and Blu-ray® disc where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Thus, in some aspects computer-readable media may comprise non-transitory computer-readable media (e.g., tangible media). In addition, for other aspects computer-readable media may comprise transitory computer- readable media (e.g., a signal). Combinations of the above should also be included within the scope of computer-readable media.

[00125] Thus, certain aspects may comprise a computer program product for performing the operations presented herein. For example, such a computer program product may comprise a computer-readable medium having instructions stored (and/or encoded) thereon, the instructions being executable by one or more processors to perform the operations described herein. For certain aspects, the computer program product may include packaging material.

[00126] Further, it should be appreciated that modules and/or other appropriate means for performing the methods and techniques described herein can be downloaded and/or otherwise obtained by a user terminal and/or base station as applicable. For example, such a device can be coupled to a server to facilitate the transfer of means for performing the methods described herein. Alternatively, various methods described herein can be provided via storage means (e.g., RAM, ROM, a physical storage medium such as a compact disc (CD) or floppy disk, etc.), such that a user terminal and/or base station can obtain the various methods upon coupling or providing the storage means to the device. Moreover, any other suitable technique for providing the methods and techniques described herein to a device can be utilized.

[00127] It is to be understood that the claims are not limited to the precise configuration and components illustrated above. Various modifications, changes and variations may be made in the arrangement, operation and details of the methods and apparatus described above without departing from the scope of the claims.

[00128] WHAT IS CLAIMED IS:

CLAIMS

1. A method for transmitting values in a neural network, comprising:

obtaining a parameter value; and

encoding the parameter value based at least in part on at least one value used by a neuron, the encoding being based at least in part on at least one spike to be transmitted via a spike channel.

2. The method of claim 1, further comprising encoding based at least in part on an absolute latency code, and/or a relative latency code.

3. The method of claim 1, further comprising encoding based at least in part on a rate code, Inter-Spike Interval encoding, or binary encoding.

4. The method of claim 1, further comprising splitting the parameter value into a plurality of components, each component to be encoded by at least one neuron.

5. A method for receiving parameter values in a neural network, the method comprising:

determining which neuron will receive a spike representing an encoded value; and

decoding at least one spike to determine a parameter value used by the neuron.

6. The method of claim 5, further comprising routing the spike based at least on part on connectivity information.

7. The method of claim 6, in which the connectivity information includes an index for a source neuron.

8. The method of claim 6, in which the connectivity information includes an index for a plurality of source neurons.

9. The method of claim 5, in which the encoded value is represented by a plurality of spikes, each corresponding to a sub component of the encoded value and being decoded to determine the parameter value.

10. The method of claim 5, further comprising receiving the spike via a redundant receiver neuron to recover from spike loss.

11. An apparatus for transmitting values in a neural network, comprising

a memory; and

at least one processor coupled to the memory, the at least one processor being configured:

to obtain a parameter value; and

to encode the parameter value based at least in part on at least one value used by a neuron, the encoding being based at least in part on at least one spike to be transmitted via a spike channel.

12. The apparatus of claim 11, in which the at least one processor is further configured to encode the parameter value based at least in part on an absolute latency code, and/or a relative latency code.

13. The apparatus of claim 11, in which the at least one processor is further configured to encode the parameter value based at least in part on a rate code, Inter-Spike Interval encoding, or binary encoding.

14. The apparatus of claim 11, in which the at least one processor is further configured to split the parameter value into a plurality of components, each component to be encoded by at least one neuron.

15. An apparatus for receiving parameter values in a neural network, comprising:

a memory; and

at least one processor coupled to the memory, the at least one processor being configured:

to determine which neuron will receive a spike representing an encoded value; and

to decode at least one spike to determine a parameter value used by the neuron.

16. The apparatus of claim 15, in which the at least one processor is further configured to route the spike based at least on part on connectivity information.

17. The apparatus of claim 16, in which the connectivity information includes an index for a source neuron.

18. The apparatus of claim 16, in which the connectivity information includes an index for a plurality of source neurons.

19. The apparatus of claim 15, in which the encoded value is represented by a plurality of spikes, each corresponding to a sub component of the encoded value and being decoded to determine the parameter value.

20. The apparatus of claim 15, in which the at least one processor is further configured to receive the spike via a redundant receiver neuron to recover from spike loss.

21. An apparatus for transmitting values in a neural network, comprising

means for obtaining a parameter value; and

means for encoding the parameter value based at least in part on at least one value used by a neuron, the encoding being based at least in part on at least one spike to be transmitted via a spike channel.

22. An apparatus for receiving parameter values in a neural network, comprising:

means for determining which neuron will receive a spike representing an encoded value; and

means for decoding at least one spike to determine a parameter value used by the neuron.

23. A computer program product for transmitting values in a neural network, comprising:

a non-transitory computer readable medium having encoded thereon program code, the program code comprising:

program code to obtain a parameter value; and

program code to encode the parameter value based at least in part on at least one value used by a neuron, the encoding being based at least in part on at least one spike to be transmitted via a spike channel.

24. A computer program product for receiving parameter values in a neural network, comprising:

a non-transitory computer readable medium having encoded thereon program code, the program code comprising:

program code to determine which neuron will receive a spike representing an encoded value; and

program code to decode at least one spike to determine a parameter value used by the neuron.

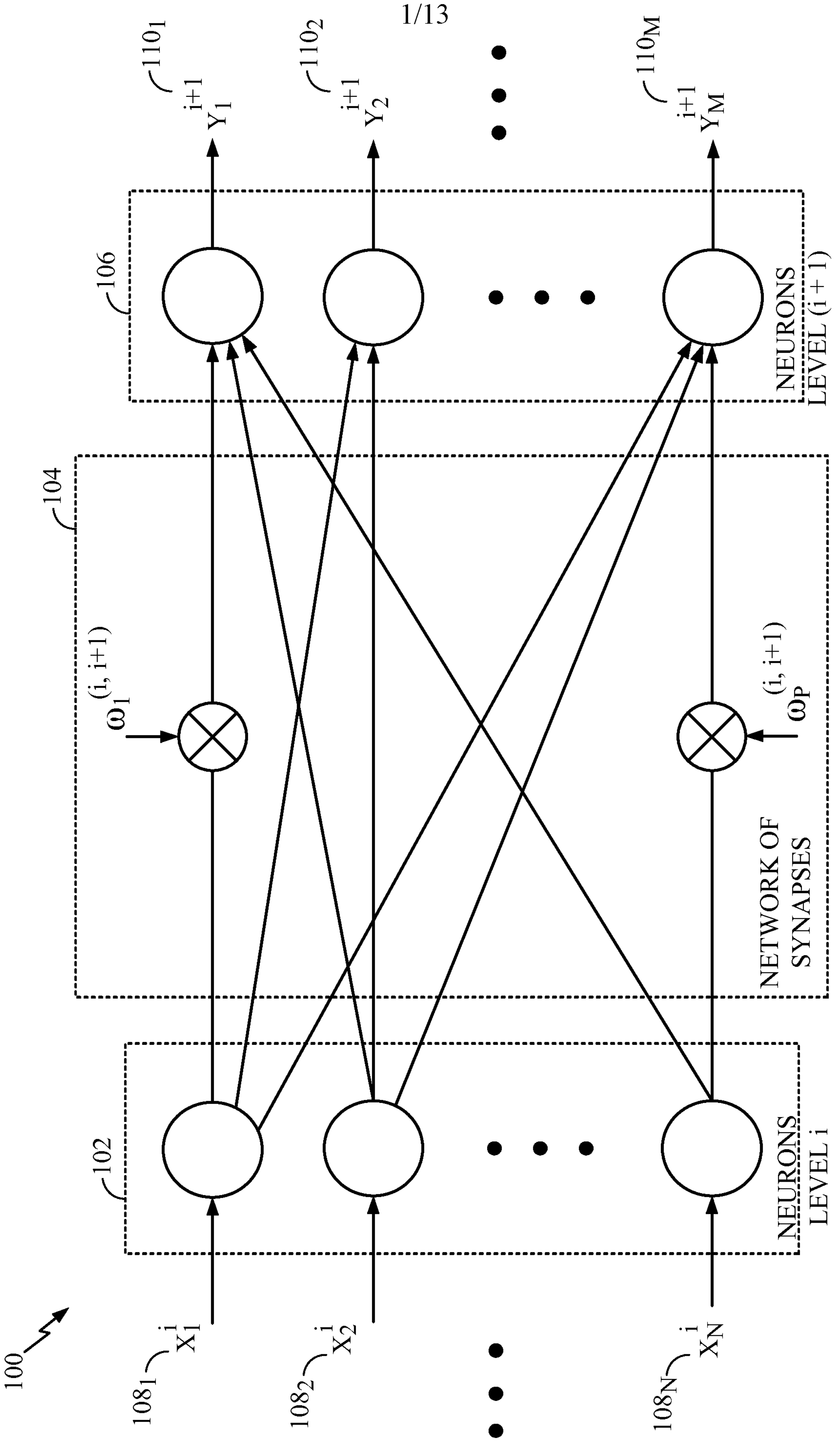

100
108_1
X_1^i
108_2
X_2^i
108_N
X_N^i
102
NEURONS LEVEL i
104
$\omega_1^{(i, i+1)}$
$\omega_P^{(i, i+1)}$
NETWORK OF SYNAPSES
106
NEURONS LEVEL (i + 1)
110_1
Y_1^{i+1}
110_2
Y_2^{i+1}
110_M
Y_M^{i+1}

FIG. 1

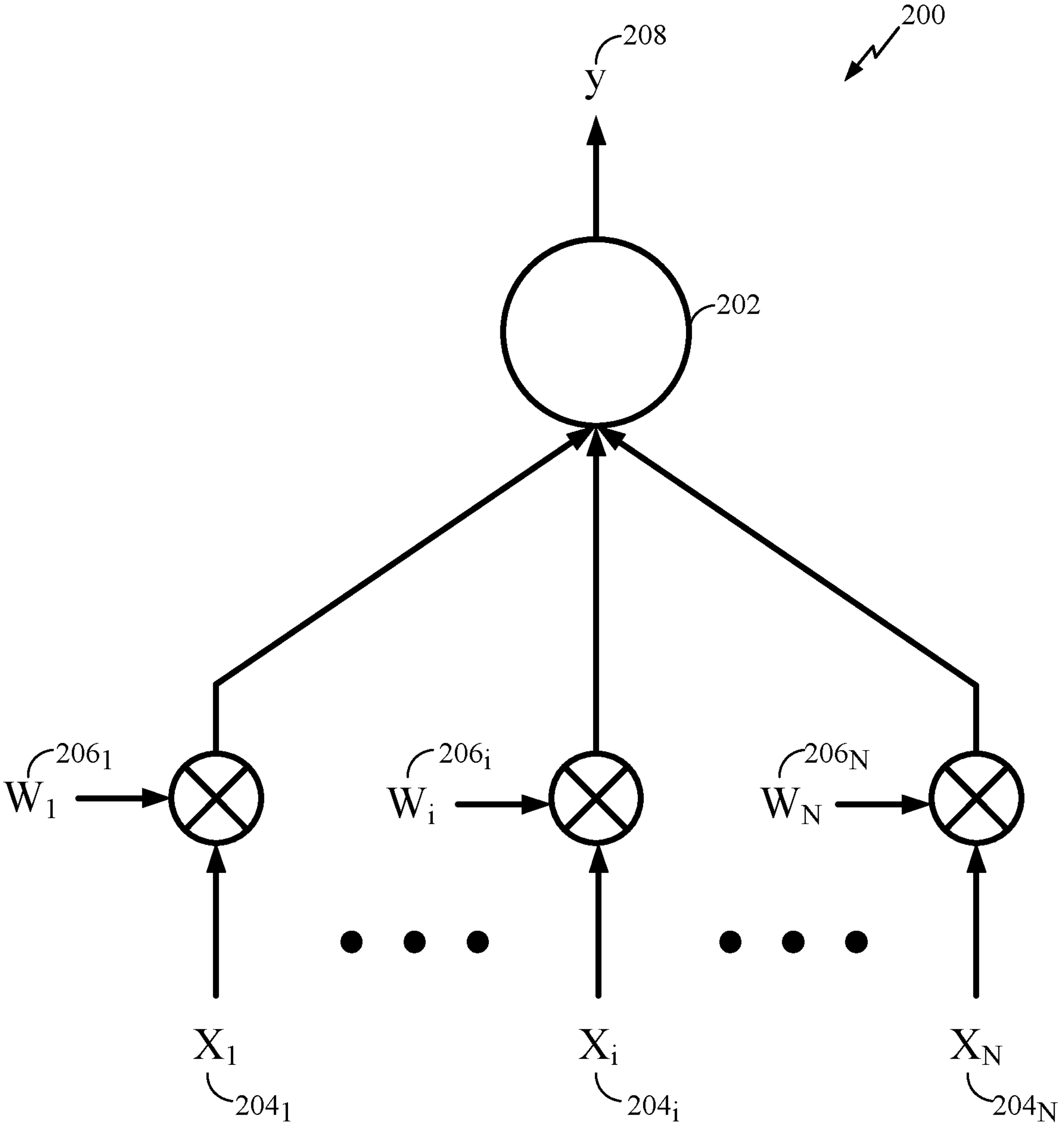


FIG. 2

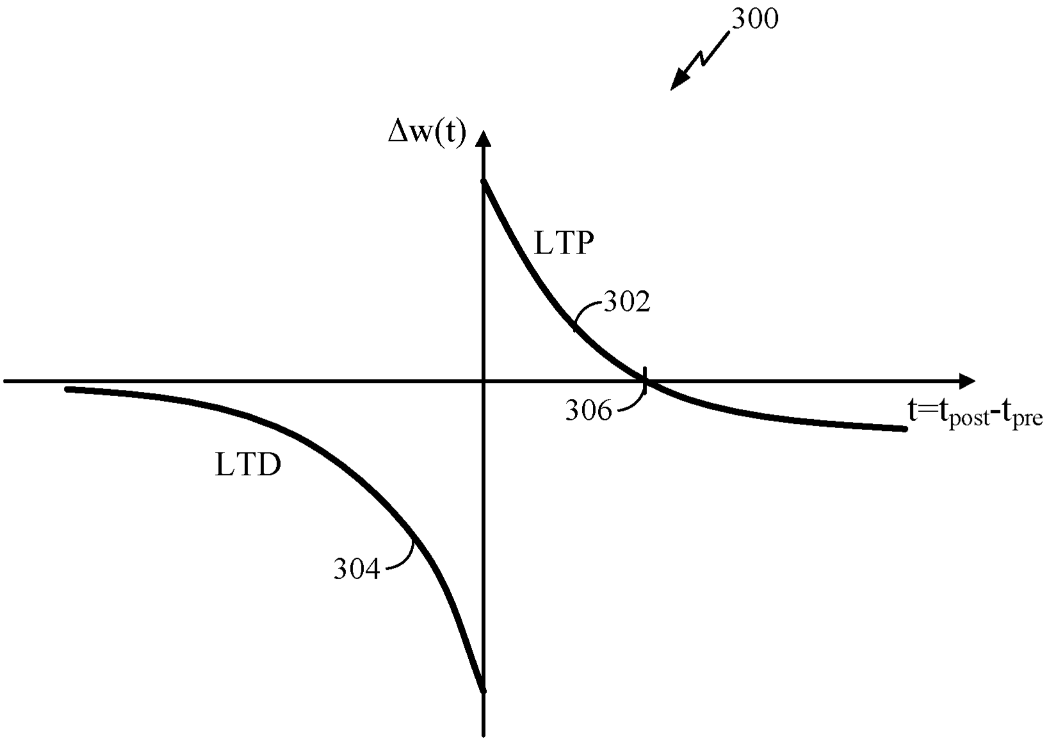


FIG. 3

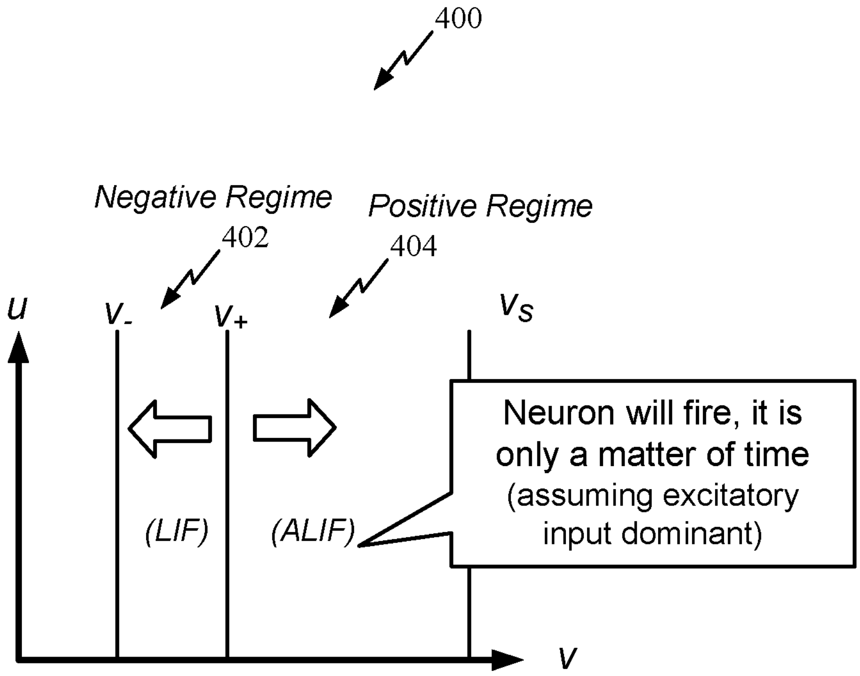

400
Negative Regime
402
Positive Regime
404
u
v-
v+
vS
(LIF)
(ALIF)
Neuron will fire, it is only a matter of time (assuming excitatory input dominant)
v

FIG. 4

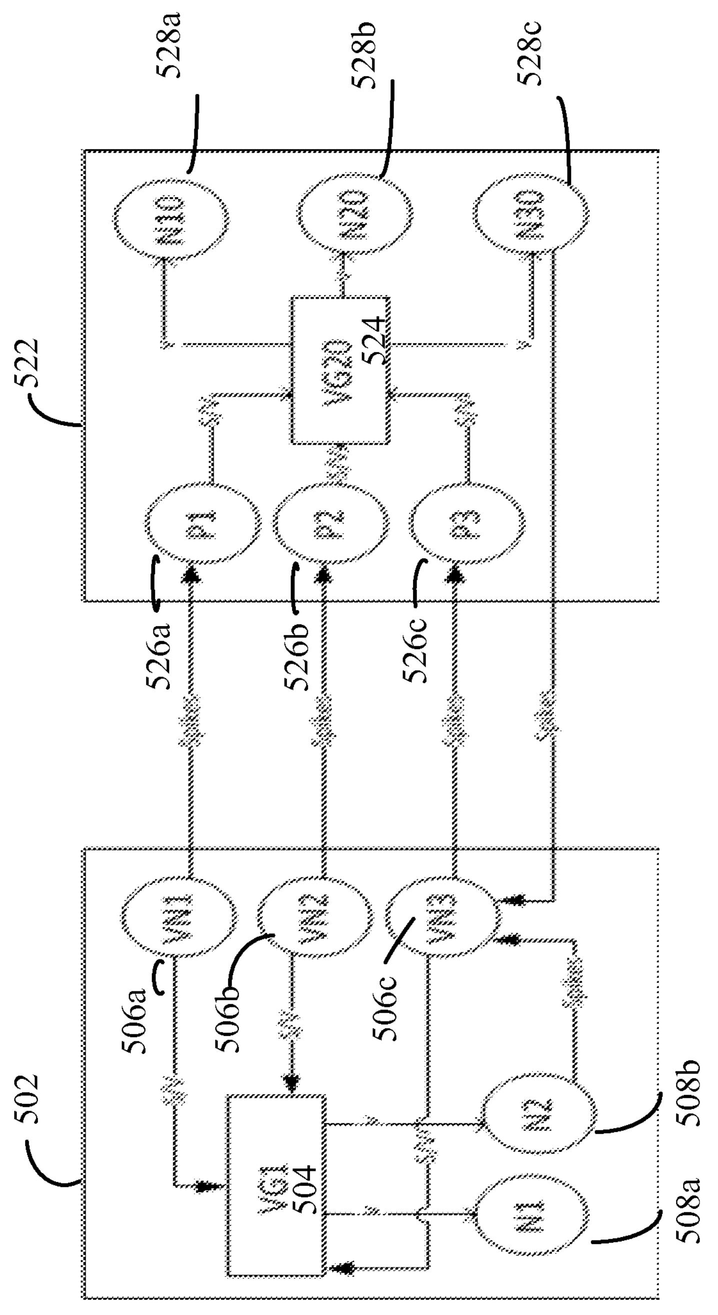
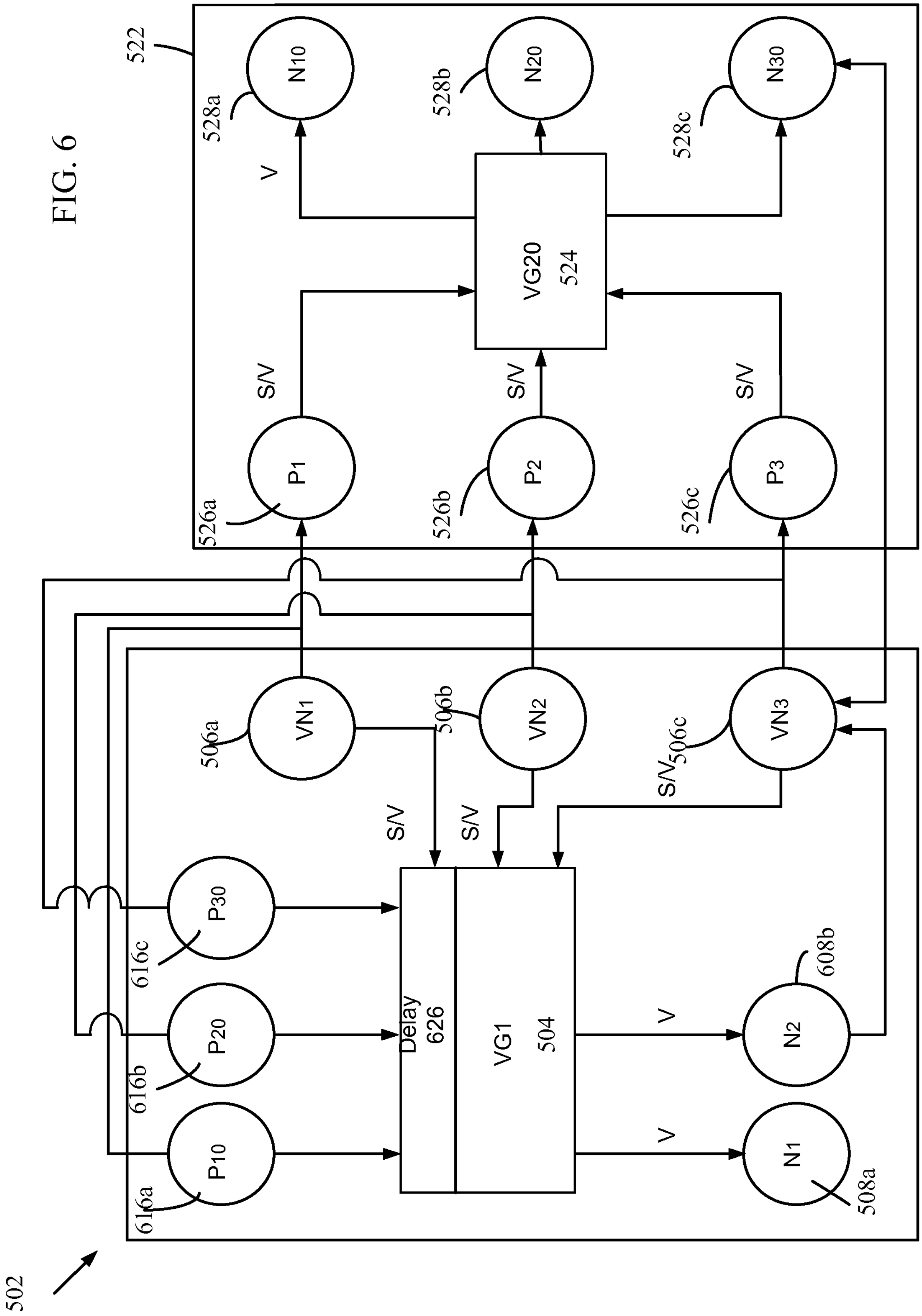

502
VG1
504
506a
506b
506c
VN1
VN2
VN3
N1
N2
508a
508b
522
VG20
524
P1
P2
P3
526a
526b
526c
N10
N20
N30
528a
528b
528c

FIG. 5

FIG. 6


522
N10
N20
N30
528a
528b
528c
V
VG20
524
S/V
S/V
S/V
P1
P2
P3
526a
526b
526c
VN1
VN2
VN3
506a
506b
506c
S/V
S/V
S/V
P30
P20
P10
616c
616b
616a
Delay
626
VG1
504
V
V
N2
N1
608b
508a
502

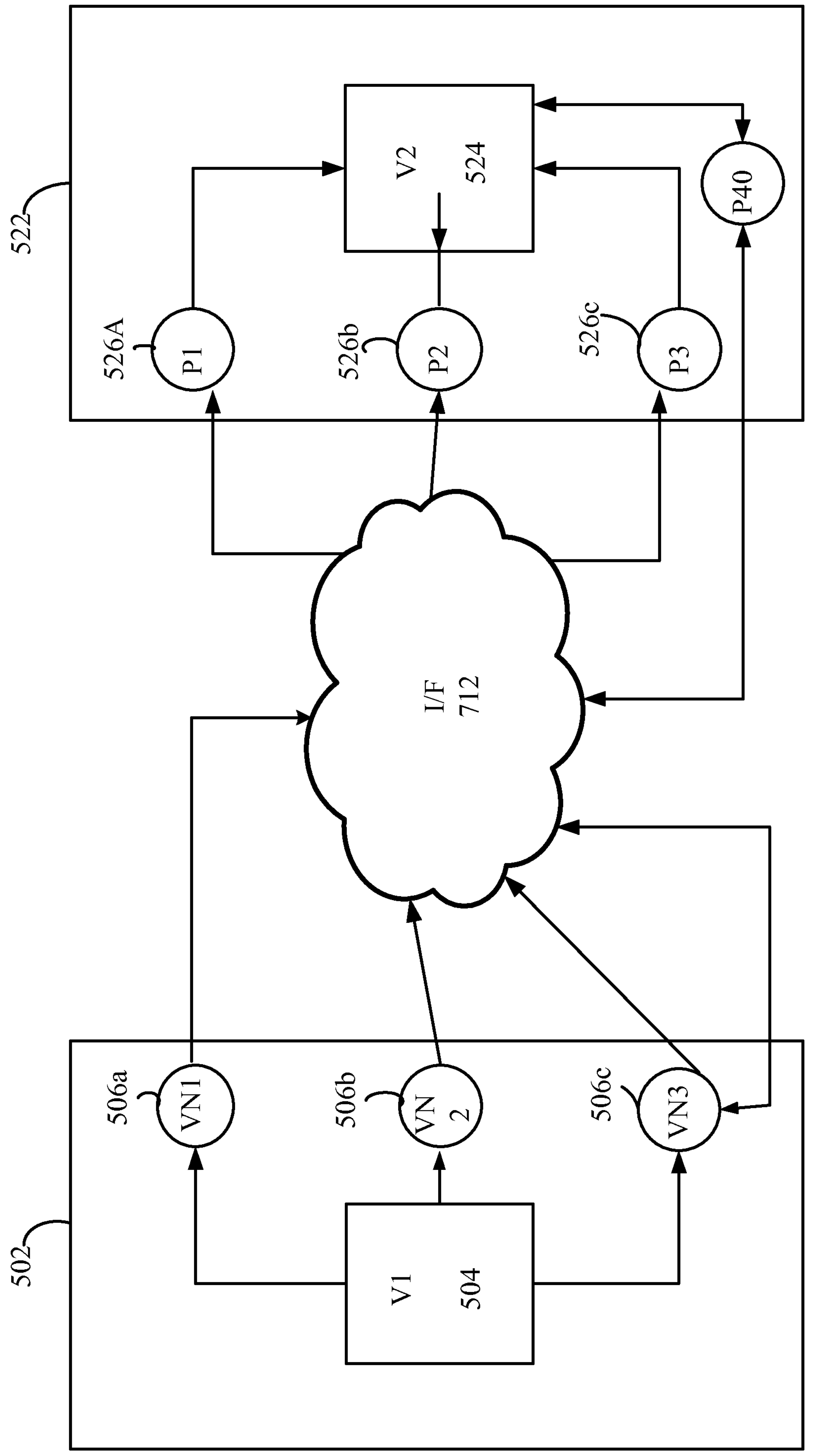

522
526A
P1
526b
P2
526c
P3
V2
524
P40
I/F
712
502
506a
VN1
506b
VN
2
506c
VN3
V1
504

FIG. 7A

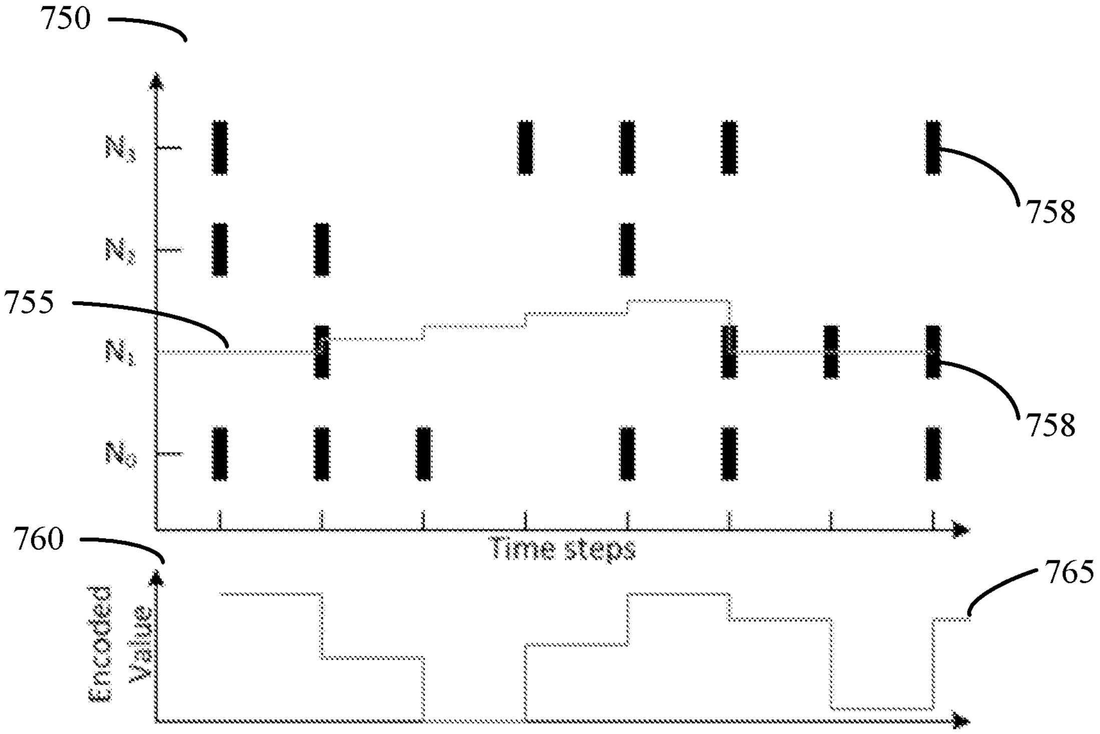


FIG. 7B

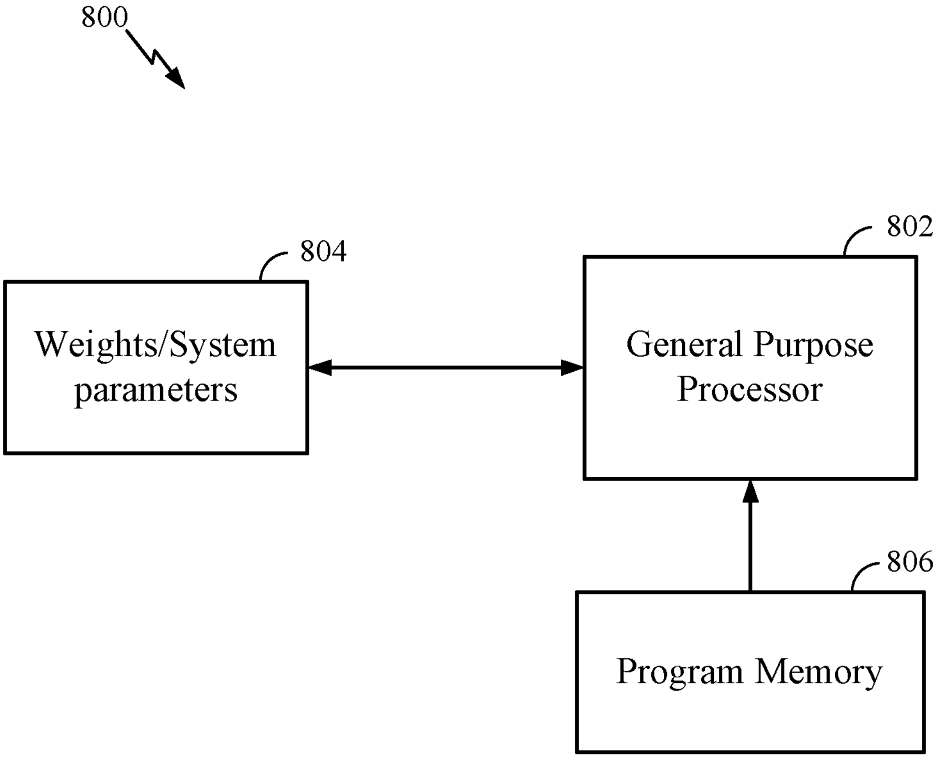

800
804
Weights/System parameters
802
General Purpose Processor
806
Program Memory

FIG. 8

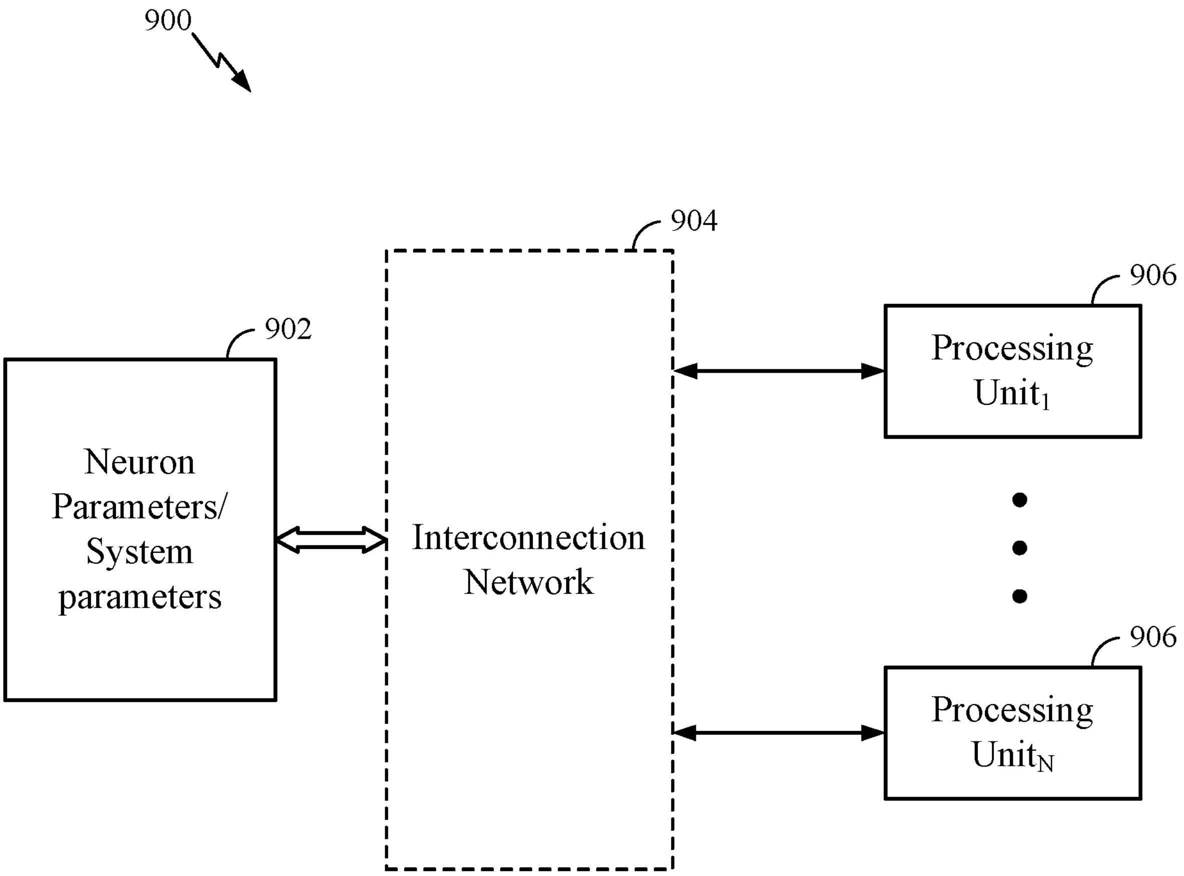

900
904
906
902
Neuron Parameters/ System parameters
Interconnection Network
Processing Unit$_1$
906
Processing Unit$_N$

FIG. 9

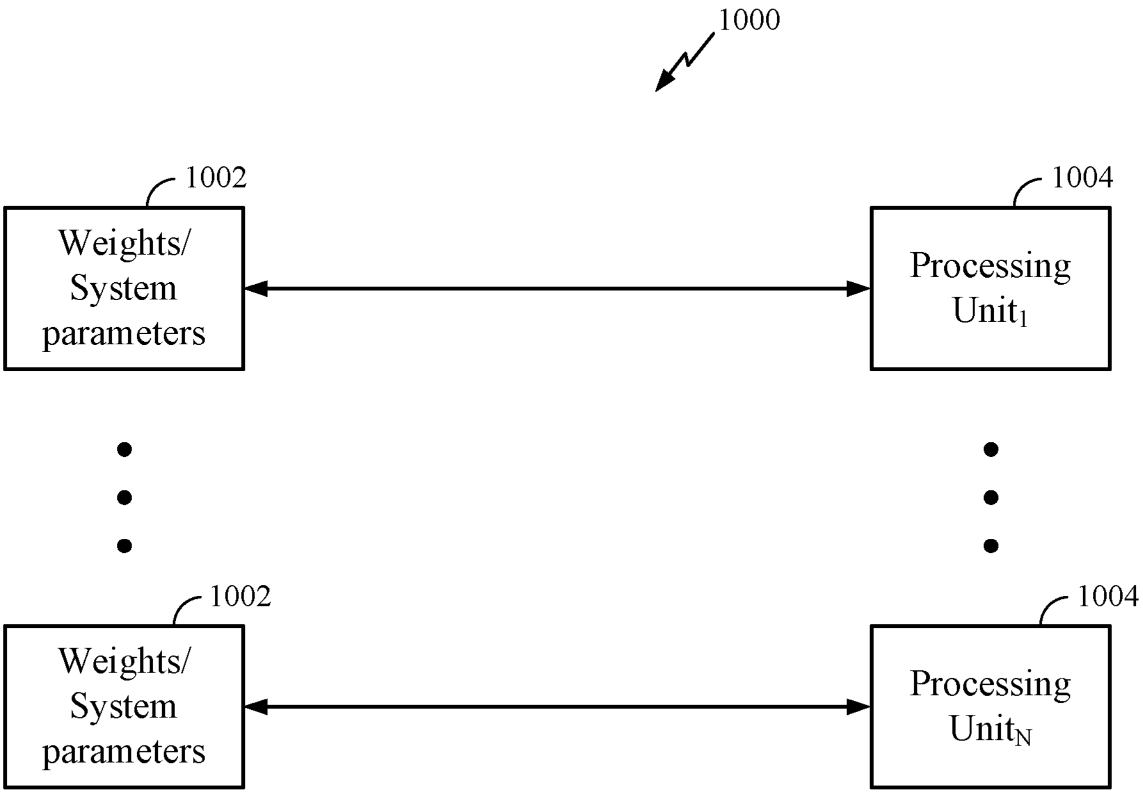

1000
1002
Weights/ System parameters
1004
Processing Unit$_1$
1002
Weights/ System parameters
1004
Processing Unit$_N$

FIG. 10

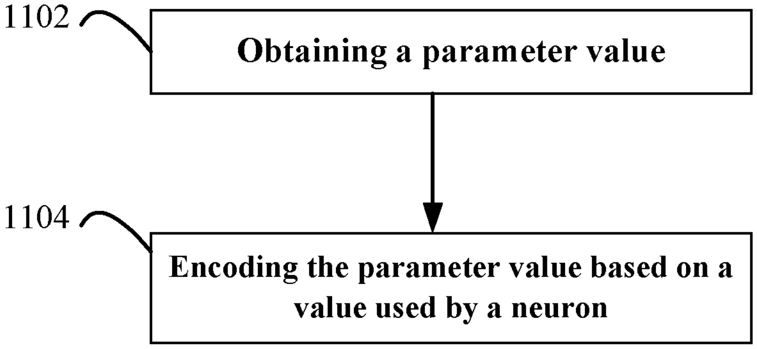

1102
Obtaining a parameter value
1104
Encoding the parameter value based on a value used by a neuron

FIG. 11

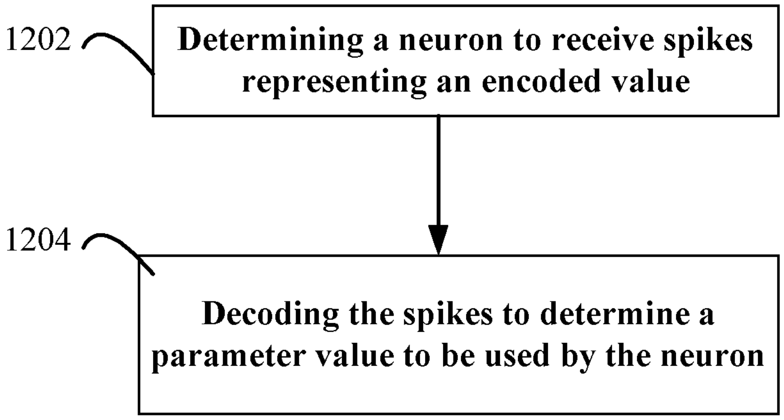

1202
Determining a neuron to receive spikes representing an encoded value
1204
Decoding the spikes to determine a parameter value to be used by the neuron

FIG. 12

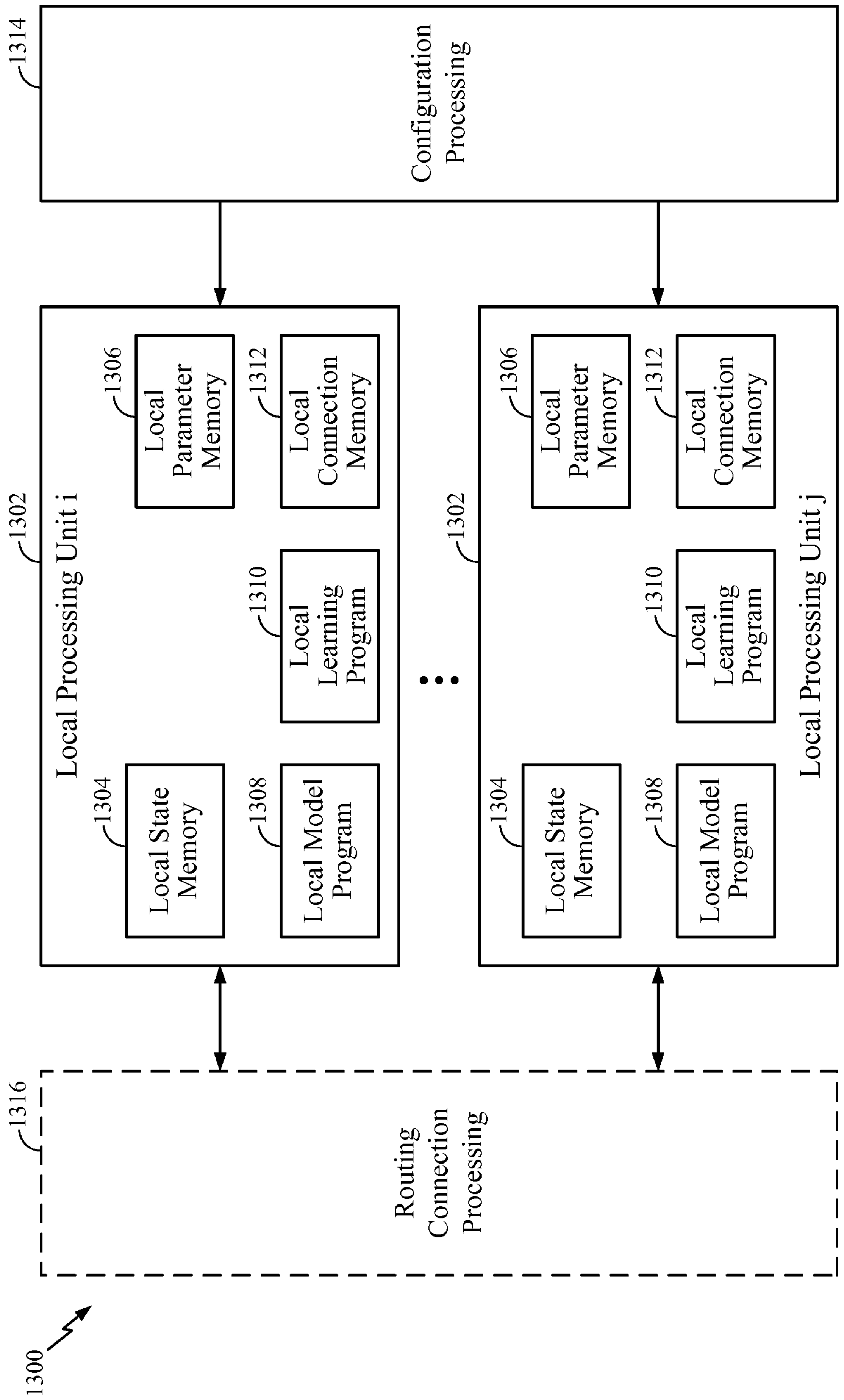

1314
Configuration Processing
1302
Local Processing Unit i
1306
Local Parameter Memory
1312
Local Connection Memory
1310
Local Learning Program
1304
Local State Memory
1308
Local Model Program
1302
Local Processing Unit j
1306
Local Parameter Memory
1312
Local Connection Memory
1310
Local Learning Program
1304
Local State Memory
1308
Local Model Program
1316
Routing Connection Processing
1300

FIG. 13

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/012464

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06N3/04
ADD.

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

B. FIELDS SEARCHED

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

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	THORPE S ET AL: "Spike-based strategies for rapid processing", NEURAL NETWORKS, ELSEVIER SCIENCE PUBLISHERS, BARKING, GB, vol. 14, no. 6-7, 9 July 2001 (2001-07-09) , pages 715-725, XP004310074, ISSN: 0893-6080, DOI: 10.1016/S0893-6080(01)00083-1 page 715 - page 724, right-hand column, paragraph 3 -----	1-24
X	US 2011/235698 A1 (PETRE CSABA [US] ET AL) 29 September 2011 (2011-09-29)	1,2,11, 12,21,23
A	abstract; claims 1,15,26; figures 1,2 paragraph [0005] - paragraph [0050] ----- -/--	3-10, 13-20, 22,24

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search 6 July 2015	Date of mailing of the international search report 14/07/2015
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Volkmer, Markus

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	TUFFY ET AL: "Inter-neuron communication strategies for spiking neural networks", NEUROCOMPUTING, ELSEVIER SCIENCE PUBLISHERS, AMSTERDAM, NL, vol. 71, no. 1-3, 3 November 2007 (2007-11-03), pages 30-44, XP022327387, ISSN: 0925-2312 page 30 - page 42, left-hand column, paragraph 2 -----	1-24
A	US 5 216 752 A (TAM DAVID C [US]) 1 June 1993 (1993-06-01) the whole document -----	1-24
A	BHUMBRA G S ET AL: "Spike coding from the perspective of a neurone", COGNITIVE PROCESSING ; INTERNATIONAL QUARTERLY OF COGNITIVE SCIENCE, SPRINGER, BERLIN, DE, vol. 6, no. 3, 12 August 2005 (2005-08-12), pages 157-176, XP019358632, ISSN: 1612-4790, DOI: 10.1007/S10339-005-0006-X page 157 - page 171, left-hand column, paragraph 2 -----	1-24

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2015/012464

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011235698 A1	29-09-2011	US 2011235698 A1 US 2011235914 A1 US 2013251278 A1	29-09-2011 29-09-2011 26-09-2013
US 5216752 A	01-06-1993	NONE	