



(51) International Patent Classification:

G06N 3/04 (2006.01) G06N 3/063 (2006.01)
G06N 3/08 (2006.01)

(21) International Application Number:

PCT/EP2019/081373

(22) International Filing Date:

14 November 2019 (14.11.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/768,927 18 November 2018 (18.11.2018) US

(71) Applicant: **INNATERA NANOSYSTEMS B.V.**
[NL/NL]; Mekelweg 4 (TUD-EWI-CAS), 2628CD DELFT (NL).

(72) Inventors: **KUMAR, Sumeet Susheel**; Innatera Nanosystems B.V., Mekelweg 4 (TUD-EWI-CAS), 2628CD DELFT (NL). **ZAJJO, Amir**; Innatera Nanosystems B.V., Mekelweg 4 (TUD-EWI-CAS), 2628CD DELFT (NL).

(74) Agent: **HOYNG ROKH MONEGIER LLP**; Rembrandt Tower, 31st Floor, Amstelplein 1, 1096 HA Amsterdam (NL).

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

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

(54) Title: SPIKING NEURAL NETWORK

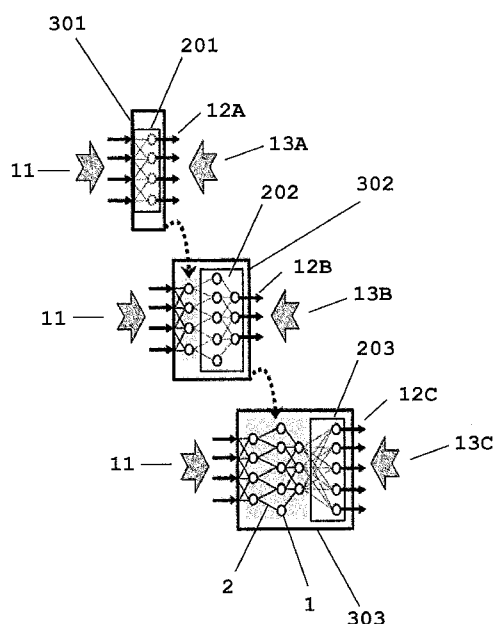


FIG. 2

(57) Abstract: A spiking neural network for classifying input pattern signals, comprising a plurality of spiking neurons implemented in hardware or a combination of hardware and software, and a plurality of synaptic elements interconnecting the spiking neurons to form the network. Each synaptic element is adapted to receive a synaptic input signal and apply a weight to the synaptic input signal to generate a synaptic output signal, the synaptic elements being configurable to adjust the weight applied by each synaptic element, and each of the spiking neurons is adapted to receive one or more of the synaptic output signals from one or more of the synaptic elements, and generate a spatio-temporal spike train output signal in response to the received one or more synaptic output signals.

Published:

- *with international search report (Art. 21(3))*
- *with amended claims (Art. 19(1))*

Spiking neural network

TECHNICAL FIELD

[0001] This disclosure generally relates to composing spiking neural networks. More specifically, but not exclusively, it relates to composing pattern recognizers build out of spiking neurons and a system and method for building spiking neural networks in a compositional way using a unique response method.

BACKGROUND

[0002] Automatic signal recognition (ASR) refers to the recognition of signals through the identification of their constituent features. ASR is utilized in a range of applications, for example to recognize the voice of a speaker and spoken words in a voice/speech recognition system, to recognize arrhythmias in electrocardiograms (ECG), to determine the shape of a gesture in a motion-controlled system, among others. ASR is typically performed by characterizing patterns present in short samples of an input signal, and thus accurate pattern recognition capabilities are fundamental to an effective ASR system.

[0003] Measuring some physical quantities to derive input signals for an ASR system may require the fusion of data from multiple types of sensors. For example, recognizing a hand gesture using a hand-held input device may necessitate the fusion of data from an accelerometer, to measure motion, and data from a gyroscope, to measure orientation. Combining data from both sensors enables the detection of gestures in three-dimensional space.

[0004] In general, data fusion refers to the integration of data gathered from disparate, and potentially heterogeneous sources, in order to decrease the uncertainty in interpretation of data from those individual sources. It is important that during the fusion process, essential features in the disparate input signals are adequately represented in the fused signal.

[0005] Pattern recognition and fusion are typically performed using microprocessors, and/or digital signal processors, both of which implement a stored program architecture. This architecture is inherently inefficient for analytics of streaming data. On a single processor, the extraction and recognition of patterns is carried out sequentially. This is because the pattern extraction and identification are realized in terms of the generic, simplistic instruction sets (e.g. RISC or CISC instruction sets), resulting in lengthy execution sequences for every

pattern in the signal sample. Complex patterns in the input signal necessitate the use of more complex signal processing algorithms, which in a system requiring real-time responses from the pattern recognition engine, further necessitates the use of higher clock frequencies for the processor. In power-constrained devices (such as portable electronic devices, wearable
5 devices, etc.), this is not feasible. Furthermore, due to the sequential execution paradigm, the latency and power dissipation of pattern recognition operations on a microprocessor increase greatly as the complexity and repertoire of patterns increases. The presence of noise in the input signals further increases the complexity of analysis, and adversely impacts performance and efficiency.

[0006] Data fusion is a non-native operation for microprocessors. This means that before
10 input signals or data streams can be fused, their information content must be extracted, and thereafter combined with corresponding content from other streams. Consequently, in microprocessor implementations, each input signal/data source for fusion is processed separately by independent pattern recognition pipelines, followed by a rule-based framework
15 to combine the disparate pattern recognition outcomes. This approach requires multiple invocations of the pattern recognition infrastructure, for each input signal or data stream, leading to increased power dissipation. Furthermore, the limitations of load-store architecture microprocessors and digital signal processors (DSPs) for pattern recognition mean that as pattern complexity increases, the power and latency costs of recognizing these patterns also
20 increase. While the latency cost can be mitigated by increasing clock frequency, this is achieved at the cost of further increased power dissipation. Similarly, the quality of fusion is limited by the complexity of processing and ASR carried out on input signals, the number of signals being fused, and the computational capabilities of the microprocessor or DSP. The sequential nature of processing reduces the throughput of fusion-based ASR, thus, as
25 complexity of patterns in signals increases, the number of input signals that can be fused using the microprocessor or DSP, decreases.

[0007] Artificial neural networks have been proposed as an alternative to a microprocessor
implementation. Spiking neural networks (SNN) are a promising means of realizing ASR for
many different applications. SNNs encode information in the form of one or more precisely
30 timed (voltage) spikes, rather than as integer or real-valued vectors. Computations for pattern classification are effectively performed in the analog and temporal domains. For this reason, SNNs are typically realized in hardware as full-custom mixed signal integrated circuits. This

enables them to perform pattern classification with several orders of magnitude lower energy consumption than their artificial neural network counterparts, in addition to having smaller network sizes.

[0008] SNNs consist of a network of spiking neurons interconnected by synapses that dictate the strength of the connections between the spiking neurons. This strength is represented as a weight, which moderates the effect of the output of a pre-synaptic neuron on the input to a post-synaptic neuron. Typically, these weights are set in a training process that involves exposing the network to a large volume of labelled input data, and gradually adjusting the weights of the synapses until a desired network output is achieved. However, practically, these volumes of labelled data may simply not exist.

[0009] Furthermore, training a large, multi-layer network in a monolithic fashion is time-consuming since the network size results in a complex optimization problem whose solution is computationally expensive, due to the need to back propagate errors over several layers of the network.

[0010] Spiking neural networks typically rely on learning rules based on the relative firing time of the two neurons connected by a synapse, for the purposes of determining error. A common learning rule is referred to as spike timing dependent plasticity or STDP. Although this method enables potentially unsupervised training of an SNN to set the weights of the synapses, it is inherently unstable. This is because STDP targets weight adjustment at individual synapses, and lacks mechanisms to balance synaptic weights at the network level. As a result, the STDP process can be unstable, and yield a divergent synaptic weight configuration. Although it may be possible to mitigate this issue by balancing the magnitude and distribution of excitatory and inhibitory connections formed by the synapses in the network, this process is difficult to implement and necessitates exhaustive searches to yield a stable, convergent weight configuration. Spiking neural networks may also trained with conventional back-propagation methods. However, these techniques are computationally expensive when applied to deep, multi-layer networks due to the complex causal relationship between neurons across different layers of the network.

SUMMARY

[0011] To address the above discussed drawbacks of the prior art, there is proposed, according to an aspect of the disclosure, a spiking neural network for classifying input pattern

signals, comprising a plurality of spiking neurons implemented in hardware or a combination of hardware and software, and a plurality of synaptic elements interconnecting the spiking neurons to form the network. Each synaptic element is adapted to receive a synaptic input signal and apply a weight to the synaptic input signal to generate a synaptic output signal, the synaptic elements being configurable to adjust the weight applied by each synaptic element, and each of the spiking neurons is adapted to receive one or more of the synaptic output signals from one or more of the synaptic elements, and generate a spatio-temporal spike train output signal in response to the received one or more synaptic output signals.

[0012] The spiking neural network comprises a first sub-network comprising a first sub-set of the spiking neurons connected to receive synaptic output signals from a first sub-set of the synaptic elements. The first sub-network is adapted to generate a sub-network output pattern signal from the first sub-set of spiking neurons, in response to a sub-network input pattern signal applied to the first sub-set of synaptic elements, and the weights of the first sub-set of synaptic elements are configured by training the sub-network on a training set of sub-network input pattern signals, so that the sub-network output pattern signal is different for every unique sub-network input pattern signal of the training set.

[0013] In one embodiment, the distance between each different sub-network output pattern signal is larger than a predetermined threshold value, the distance being measured by an output pattern metric. The spiking neurons and synaptic elements may be configured such that respective distances, as measured by an output pattern metric, between two output pattern signals of the sub-network generated in response to two respective different sub-network input pattern signals are maximized for all sub-network input pattern signals of the training set. Each respective distance may be maximized until the output pattern signals meet at least a first minimum sensitivity threshold required for distinguishing between features of the input pattern signals.

[0014] A feature is any characteristic of an input pattern signal that enables identification of that signal, and/or patterns of interest within that signal. Patterns may be composed of one or more features. For example, relevant features in an ECG input signal could be the various time-varying amplitudes that characterize the different stages of a heartbeat, for an audio input signal the relevant features can be the frequency content of the signal within each discrete time interval in which the audio signal is sampled, or for an image input signal the

relevant features can be the fundamental shapes and lines present in the image. Additional examples are provided herein.

[0015] In the spiking neural network, each of the spiking neurons may be configurable to adjust the response of the neuron to the received one or more synaptic output signals. Spikes may be generated by the spiking neurons at one or more firing times, and the sub-set of synaptic elements and/or the sub-set of neurons may be configured such that the union of two sets of firing times of the sub-set of neurons that fire for two different sub-network input pattern signals is minimized for all sub-network input pattern signals of the training set. Each union may be minimized until the output pattern signals meet at least a second minimum sensitivity threshold required for distinguishing between features of the input pattern signals.

[0016] The spiking neural network may comprise a second sub-network comprising a second sub-set of the spiking neurons connected to receive synaptic outputs from a second sub-set of the synaptic elements, wherein the second sub-network is adapted to receive a second sub-network input pattern signal applied to the second sub-set of synaptic elements, and generate a corresponding second sub-network output pattern signal from the second sub-set of neurons, and wherein the configurations of the second sub-set of synaptic elements are adjusted so that the second sub-network output pattern signal is different for every unique feature in the second sub-network input pattern signals, wherein the network comprises a third sub-network comprising a third sub-set of the spiking neurons connected to receive synaptic outputs from a third sub-set of the synaptic elements, wherein the first and second sub-network output pattern signals are input pattern signals of the third sub-network, and wherein the configurations of the third sub-set of synaptic elements are adjusted such that the third sub-network output pattern signal is different for every unique feature in the input pattern signal from both the first and second sub-network and unique combinations of them, such that the features that are present in the input pattern signals from both the first and second sub-network are encoded by the third sub-network.

[0017] The synaptic elements of the third sub-network may be configured such that the input pattern signals from the first and second sub-network are weighted according to importance of the specific input patterns. The network may comprise multiple sub-networks of synaptic elements and spiking neurons, for which the sub-network output pattern signal is different for every unique feature in the sub-network input pattern signals, wherein the network can be divided in multiple layers having a particular sequential order in the network,

and wherein the multiple sub-networks are instantiated in the particular sequential order of the multiple layers each respective sub-network belongs to.

[0018] In the spiking neural network, the synaptic elements may be arranged as a configurable switch matrix.

5 [0019] Configuration information of the neurons, synaptic elements, interconnection topology of the neurons and synaptic elements and/or output configuration of the neural network may be saved on a configuration memory, and the configuration information may be loaded from the configuration memory when the neural network is brought to an initialization state.

10 [0020] A set of neurons may be arranged in one or multiple template networks of neurons, the neurons of a particular template network of neurons forming a sub-network of the neural network, wherein the sub-network output pattern signal is different for every unique feature in the sub-network input pattern signals, and wherein each template network in the neural network was instantiated in a pre-trained manner.

15 [0021] The neural network may be configured to take as input one or multiple sampled input signals and convert the input signals into a representative set of neural network input pattern signals. An output pattern signal of the neural network may be classified in one or multiple output classes.

[0022] According to an aspect of the invention, a method is proposed for classifying input
20 pattern signals using a spiking neural network comprising a plurality of spiking neurons implemented in hardware or a combination of hardware and software, and a plurality of synaptic elements interconnecting the spiking neurons to form the network, wherein each of the synaptic elements is adapted to receive a synaptic input signal and apply a weight to the synaptic input signal to generate a synaptic output signal, the synaptic elements being
25 configurable to adjust the weight applied by each synaptic element, and wherein each of the spiking neurons is adapted to receive one or more of the synaptic output signals from one or more of the synaptic elements, and generate a spatio-temporal spike train output signal in response to the received one or more synaptic output signals. The method comprises the steps of defining a first sub-network of the spiking neural network, the first sub-network
30 comprising a first sub-set of the spiking neurons connected to receive synaptic output signals from a first sub-set of the synaptic elements, configuring the weights of the first sub-set of synaptic elements by training the sub-network on a training set of sub-network input pattern

signals, so that the sub-network output pattern signal is different for every unique sub-network input pattern signal of the training set, applying a sub-network input pattern signal to a first sub-network of the spiking neural network, the first sub-network comprising a first sub-set of the spiking neurons connected to receive synaptic output signals from a first sub-set of the synaptic elements, and receiving a sub-network output pattern signal generated by the first sub-set of spiking neurons in response to the sub-network input pattern signal, wherein the output pattern signal identifies one or more features of the input pattern signal.

[0023] According to another aspect of the invention, a method is proposed for classification of predetermined features in input signals. The method comprises building and training a neural network as described above, submitting at least one sampled input signal to the neural network, and classifying the features in at least one sampled input signal in one or multiple output classes of the neural network.

[0024] According to yet another aspect of the invention, a template library is provided, comprising one or multiple template networks of spiking neurons for use as a sub-network of a spiking neural network or information on a configuration thereof, wherein each template network comprises a set of the spiking neurons connected to receive synaptic outputs from a set of the synaptic elements. Each template network is adapted to receive a template network input pattern signal applied to the set of synaptic elements, and generate a corresponding template network output pattern signal from the set of neurons. The configurations of the set of synaptic elements are adjusted by training the template network on a training set of template network input pattern signals, so that the template network output pattern signal is different for every unique template network input pattern signal of the training set, wherein the training set is used to train the template network to perform a particular task, such that each template network can be used as a sub-network of a spiking neural network in a pre-trained manner to perform the particular task or such that the sub-network of the spiking neural network can be instantiated on the basis of the information on the configuration of the template network in a pre-trained manner to perform the particular task.

[0025] According to yet another aspect of the invention, a method is proposed for composing a spiking neural network, the method comprising obtaining one or multiple template networks or information on the configuration of one or multiple template networks from a template library as described above, and instantiating the one or multiple template networks as a sub-network of the spiking neural network such that the sub-network of the

spiking neural network is able to perform the particular task the template network was pre-trained to perform.

[0026] The invention consists of a method to realize pattern classifiers in a bottom-up fashion, using smaller network stages that are trained separately. Each stage of the pattern classifier is trained for its specific function using a common methodology, applied with different training objectives specific to the function being realized. The training process is aided by a method that accounts for the long-range causal influence of synapses when adapting weights, thus overcoming the instability issues faced with STDP. The three methods may be used independent of the others. The methodology enables the realization of low-latency pattern recognizers, applicable to different signal types, and which can recognize and correlate patterns across multiple signals. The methodology simplifies training and deployment time of pattern recognizers, facilitates reuse of previously trained pattern recognition networks, while concomitantly ensuring that the implementation remains flexible.

15 BRIEF DESCRIPTION OF DRAWINGS

[0027] Embodiments will now be described, by way of example only, with reference to the accompanying schematic drawings in which corresponding reference symbols indicate corresponding parts, and in which:

[0028] FIG. 1 shows an exemplary neural network, consisting of neurons and synaptic elements;

[0029] FIG. 2 shows the composition of a pattern recognizer through the integration of data convertor, input encoder and pattern classifier stages, each trained separately and instantiated in the subsequent stage;

[0030] FIG. 3A shows schematically how an input encoder transforms a sampled input signal into spatio-temporal spike trains;

[0031] FIG. 3B shows an example of a desired spike response for three input patterns described in the form of spike tuples;

[0032] FIG. 4 shows the principles of Causal Chain STDP for identifying deep causal, anti-causal and non-causal relationships between the post-synaptic neuron and neurons in subsequent network layers;

[0033] FIG. 5A shows the composition of a sensor-fused pattern reconizer through the fusion of two separate spike coded data streams, and subsequent pattern classification using the fused spike train;

[0034] FIG. 5B shows schematically how multiple temporal spike trains are fused into a single fused spike train;

[0035] FIG. 6 shows a system for fusing multiple data streams, and which can perform automatic signal recognition on the fused stream, using a network of spiking neurons;

[0036] FIG. 7A shows an example of how to interface input signal to neurons;

[0037] FIG. 7B shows an example of how to interface input signal to neurons;

[0038] FIG. 8 shows an example of how to generate a reset pulse;

[0039] FIG. 9 shows a method for fusing multiple data streams, and performing automatic signal recognition on the resulting fused stream, using a network of spiking neurons;

[0040] FIG. 10 shows an array of neurons and synaptic elements interconnected through a configurable switch matrix;

[0041] FIG. 11 shows an example of a configured array, illustrating an example mapping of ASR pipeline stages onto neurons and synaptic elements.

[0042] The figures are meant for illustrative purposes only, and do not serve as restriction of the scope or the protection as laid down by the claims.

DESCRIPTION OF EMBODIMENTS

[0043] Hereinafter, certain embodiments will be described in further detail. It should be appreciated, however, that these embodiments may not be construed as limiting the scope of protection for the present disclosure.

[0044] FIG. 1 is a simplified diagram of a neural network 100. The neurons 1 are connected with each other via synaptic elements 2. In order to not clutter the image, only a small number of neurons and synaptic elements are shown (and only some have a reference numeral attached to them). The connecting topology shown in FIG. 1, i.e. the way in which the synaptic elements 2 connect the neurons 1 with each other, is merely an example and many other topologies may be employed. Each synaptic element 2 can transmit a signal to an input of a neuron 1, and each neuron 1 that receives the signal can process the signal and can subsequently generate an output, which is transmitted via further synaptic elements 2 to other neurons 1. Each synaptic element 2 has a certain weight assigned to it, which is applied to

each synaptic input signal that the synaptic element receives and transmits, to produce a weighted synaptic output signal. The weight of a synaptic element is thus a measure of the kind of causal relationship between the two neurons 1 that are connected by the synaptic element 2. The relationship can be causal (positive weight), anti-causal (negative weight) or non-existent (zero weight).

[0045] The neurons 1 and synaptic elements 2 can be implemented in hardware, for example using analog circuit elements or digital hardwired logic circuits. They can also be implemented partly in hardware and partly in software, or entirely in software.

Implementation in hardware is preferred to achieve faster processing, e.g. enabling much faster pattern recognition, and event-driven processing in which blocks of neurons and synaptic elements are only activated when needed.

[0046] The neural network 100 can be a spiking neural network. The neurons 1 are then spiking neurons, which generate a neuron output signal in the form of one or more spikes. The spiking neurons 1 may be configured to fire (i.e. generate an output spike) only when a membrane potential (e.g. an energy potential, or voltage or current level) within the neuron reaches a predetermined threshold value. The membrane potential of the spiking neuron changes as a result of the received input signals, i.e. the synaptic output signals received by the neuron from the synaptic elements are accumulated, integrated, or otherwise processed to alter the membrane potential. When a weight of a synaptic element 2 is positive, a synaptic output signal received from that synaptic element excites the spiking neurons 1 which receive the signal, raising their membrane potentials. When a weight of a synaptic element 2 is negative, a synaptic output signal received from that synaptic element inhibits the spiking neurons 1 which receive the signal, lowering their membrane potentials. When a weight of a synaptic element 2 is zero, a synaptic output signal received from that synaptic element does not have any effect on the spiking neurons 1 which receive the signal.

[0047] When the membrane potential of a spiking neuron 1 reaches the threshold value, the neuron fires, generating a spike at the time of firing, and the membrane potential is reduced as a result of the firing. If the membrane potential subsequently again reaches the threshold value, the neuron will fire again, generating a second spike. Each spiking neuron 1 is thus configured to generate one or more spikes in response to input signals received from the connected synaptic elements 2, the spikes forming a spatio-temporal spike train. Since a spiking neuron 1 only fires when its membrane potential reaches the predetermined threshold

value, the coding and processing of temporal information is incorporated into the neural network 100. In this way, spatio-temporal spike trains are generated in the spiking neural network 100, which are temporal sequences of spikes generated by the spiking neurons 1 of the network 100.

5 [0048] The temporal characteristics of the spike trains encode amplitude and frequency features of the input signal. The temporal characteristics comprise: the latency between onset of stimulus (e.g. an input signal from a synaptic element) and generation of a spike at the output of a neuron; the latency between successive spikes from the same neuron; and the number of spikes fired by the neuron in the time duration for which the input stimulus is
10 applied.

[0049] The synaptic elements 2 can be configurable such that for example the respective weights of the synaptic elements can be varied, for example by training the neural network 100. The neurons 1 can be configurable in the way they respond to a signal from a synaptic element. For example, in the case of spiking neural networks, the neurons 1 can be
15 configured in the way a certain signal increases or decreases the membrane potential, the time it takes for the membrane potential to naturally decay towards a resting potential, the value of the resting potential, the threshold value that triggers a spike of the spiking neuron 1. The configuration of the neurons 1 can for example be kept constant during training, or be variable and set in a training of the neural network 100 on a particular training set.

20 [0050] Input signals 11 are for example multiple, disparate, sampled input signals, or spatio-temporal spike trains. The input can be an analog-to-digital converted value of a signal sample, or the digital value of the sample in the case of for example an analog or digital integrator, or the analog value of the sample in the case of an analog integrator.

[0051] Output signals 12 of the neural network 100 are for example spatio-temporal spike
25 trains, which can be read out from the output neurons 1 and further classified and transformed by an output transformation stage into a set of digital values corresponding to the type of output code selected by the user.

[0052] Operations such as pattern recognition may be accomplished in a number of steps (e.g. data conversion, feature encoding, classification). The present embodiment enables
30 these steps to be realized using standardized sub-networks of synaptic elements and neurons (cells), and configured using a flexible training methodology for each application domain. The objective of the training methodology is to configure each stage or sub-network of the

neural network to produce a unique response for each unique pattern or feature in the input signal.

[0053] FIG. 2 shows the composition of a neural network formed by the integration of a number of constituent stages, in this example including a data converter stage 201, input
5 encoder stage 202, and pattern classifier stage 203. Each stage can be trained separately and instantiated in a network for training of a subsequent stage. The term “instantiated” refers to a process in which a portion of the network forming a sub-network, e.g. a stage or cell of the network, has been previously configured, typically by training, and the configuration data of the sub-network is used to configure that sub-network within the whole network.

10 [0054] The data converter stage 201 receives input pattern signals for processing by the neural network. In this example, the data converter stage 201 is shown as cell 301, the data converter stage 202 is shown as part of cell 302, and the pattern classifier stage 203 is shown as part of cell 303. Each cell 301, 302, 303 comprises a sub-network formed by a sub-set of the synaptic elements 2 and sub-set of the neurons 1 of the network, connected in a topology
15 that realizes the function the cell is intended to implement. Each cell 301, 302, 303 is parameterized, i.e. can be configured to have a different number of synaptic elements 2, neurons 1, input connections, output connections, hidden layer connections (i.e. connections between a layer of neurons not visible to input or output), among others.

[0055] The weights of the synaptic elements 2 in each of stages 201, 202, 203 may be
20 configured by training. In this example, data converter stage 201 is first trained in cell 301. After training the data converter stage 201, the configuration settings of the trained data converter stage 201 and cell 301 may be saved. These saved settings may then be used to configure a cell 302, which includes both a data converter stage 201 and an input encoder stage 203.

25 [0056] Cell 302 may then be trained and the configuration settings of the trained encoder stage 203 and cell 302 may be saved. These saved settings may then be used to configure a cell 303, which includes a data converter stage 201, an input encoder stage 203, and a pattern classifier stage 203. Cell 303 may then be trained and the resulting configuration settings of the trained pattern classifier stage 203 and cell 303 saved.

30 [0057] The training may be accomplished using a unique response algorithm. The unique response algorithm realizes a configuration of the weights of the synaptic elements of each cell, which results in unique features in the input signal being transcribed into a unique

response of the cell's neurons 1. In the case of spiking neurons this can be a unique spiking response. The unique response algorithm is described further below. The uniqueness of a cell's response indicates the cell's ability to separate distinct features in the input pattern signal 11 into distinct output signals 12A, 12B, 12C of the cell. In the case of spiking

5 neurons, these distinct output signals 12A, 12B, 12C can be distinct output spike trains, e.g. a unique set of spatio-temporal spike train outputs from the output neurons of the cell. The uniqueness of a set of spatio-temporal spike train outputs may be determined by the number of spikes and/or the timing of spikes generated by each output neuron of the cell.

Consequently, during training of the network, the uniqueness of each response is tested after
10 the input presented to the network and converted into spikes, and the weights of the synaptic elements of the cell being trained are adapted using the unique response methodology, until the output response 12A, 12B, 12C of each cell 301, 302, 303 meets the uniqueness criterion. Teacher input 13A, 13B, 13C can be used to obtain the desired unique response for each cell 301, 302, 303, as described below.

15 [0058] To realize higher level functions, previously trained cells 301, 302, 303 can be instantiated and interconnected in a bottom-up manner. i.e. lowest level cell 301 can be trained first, then instantiated in a higher-level cell 302 containing the subsequent stage 202. The mid-level cell 302 can then be instantiated in a higher-level cell 303 containing the subsequent stage 203. Following each creation of a higher-level cell, the unique response
20 algorithm is invoked.

[0059] Cells 301, 302, 303 can be trained in this manner for specific functions in different application domains, and the configuration settings of the trained cells saved in a template library from where they can be reused in different networks. For instance, the input encoder stage 202 of a spoken word recognition network could be reused in a hot-word triggering
25 network without any change. Such reuse facilitates a reduction in overall training time, and complexity.

[0060] FIG. 3A shows schematically how an input encoder 22 transforms a sampled input signal 20 into spatio-temporal spike trains 24. Each output neuron of the input encoder 22 forms an output signal 23 which is a temporal spike train. The input encoder 22 is configured
30 by configuration 21, where for example the weights of the synaptic elements of the input encoder are set.

[0061] FIG. 3B shows an example of three desired spike responses 14A, 14B, 14C for three different input patterns to a cell of a spiking neural network. Each response in this example takes the form of a set of one or more spatio-temporal spike trains generated by output neurons N1, N2, N3, N4, N5 of the cell. Each response 14A, 14B, 14C is described for a discrete time bin 15A, 15B, 15C that may be selected for each network. The time bin 15A, 15B, 15C can be at least as large as the time between onset of the first input stimulus for the cell, and the last response from the output neurons of that cell. The time bin can be a multiple of the fall-through latency of a cell, the multiple equal to the typical number of spikes that are generated per output neuron of the cell as a result of an input stimulus.

[0062] During each invocation of the unique response methodology, a cell 301, 302, 303 can be exposed to input data 11 derived from the input pattern to be processed. The input data 11 is derived from an input pattern p that is presented to the neural network for processing.

[0063] The input pattern p may comprise, e.g. an audio signal for a voice recognition application, a still image or video image for an image recognition application, or many other types of pattern suitable for a pattern recognition system. The input data 11 may comprise of a single item of time-varying data (e.g. a single stream of data derived from an audio signal, e.g. an analog or digital signal output by a microphone), multiple items of data (e.g. a set of input data representing individual pixels of a still image output by a CCD array), multiple items of time-varying data (e.g. a set of input data streams representing individual pixels of a video image), or multiple augmented versions generated from a single item of data (e.g. a combination of input signals).

[0064] The weights of the synaptic elements 2 of the cell 301, 302, 303 may be adapted using Causal Chain Spike-timing-dependent plasticity (CC-STDP), a learning rule that accounts for long-range causation when adapting the synaptic weights. In this sense it is different from the conventional Spike-timing-dependent plasticity (STDP) learning rule, which only accounts for short-range causation when adapting weights, making the updating of weights using some training sets quite volatile and chaotic and largely a consequence of localized behaviour. CC-STDP is presented further below. Once a cell is trained, the configuration settings for the trained cell may be stored into a library of template cells from where it may be subsequently recalled and reused.

[0065] The desired spike response 14A, 14B, 14C for each input pattern p can be described as a tuple which specifies the spike response in each time bin where input data 11 derived

from an input pattern p is presented. For example, the tuple may specify the population of the cell's output neurons (A^p) that fired in response to the input data 11, and their precise firing times (B^p). The uniqueness of a response is determined by computing its distance from the cell's responses to other input patterns, using a distance function or metric (e.g. a Victor Purpura or a Van Rossum distance function), which are typically functions of A^p and B^p .

[0066] In a preferred embodiment, for each input pattern p the set A^p can describe a disjoint set containing a single output neuron 1, and the set B^p can describe the time bin corresponding to when that output neuron 1 responds. Thus, in this embodiment, each input pattern p can be identified by the single neuron 1 that responds the earliest. Later responses from the same or other neurons can for example be suppressed using a winner-take-all network, or ignored. For additional robustness, the tuple can be augmented with an additional field (C^p) that describes the strength of the neuron's response (expressed as the number of spike events generated by the single earliest spiking neuron in the period during which input data 11 was applied, C^p).

[0067] In an alternative embodiment, for each pattern p , the set A^p can be a disjoint set containing multiple output neurons 1, and the set B^p can be an identically sized set containing the response times of those neurons. The response of the cell can be steered towards different A^p and B^p values, in order to reach the desired spike tuple, using the Causal Chain STDP rule presented below.

[0068] The B^p values can be reduced by increasing the weights in the synaptic elements contributing to the firing of the corresponding neuron 1, while B^p values can be increased by reducing those weights. This can be achieved by artificially facilitating the causal firing of the output neuron 1, or facilitating anti-causal firing by inducing an early spike concomitant with the application of the input data 11. This causes a positive or negative synaptic weight adjustment respectively of the contributing pathway.

[0069] In FIG. 3B, the desired spike responses 14A, 14B, 14C are shown for three input patterns p . The first input pattern's response was generated during the first time bin 15A, the second pattern's response was generated during the second time bin 15B and the third pattern's response was generated during the third time bin 15C. The first pattern (p_1) has tuple $A^{p_1} = \{N3\}$, $B^{p_1} = \{3\}$ and $C^{p_1} = \{1\}$. The second pattern (p_2) has tuple $A^{p_2} = \{N1, N2, N4\}$, $B^{p_2} = \{2, 4, 5\}$ and $C^{p_2} = \{1, 1, 1\}$. Finally, the third pattern (p_3) has tuple $A^{p_3} = \{N2, N3, N5\}$, $B^{p_3} = \{2, \{2, 3, 4\}, 4\}$ and $C^{p_3} = \{1, 3, 1\}$.

[0070] An object of the unique response method can be to generate a set of unique spatio-temporal spike trains for each unique input pattern. These spatio-temporal spike trains may be composed as a sequence of precisely timed electrical pulses. The uniqueness of the response can be described as explained below.

5 [0071] Let A denote a set of neurons that generate output spike trains. Take p to be a given known input pattern. A^p , a subset of A , denotes the set of neurons that fire for the given input pattern p . Take $A_s = \{A^p | p \in P\}$ to be the set of all sets of neurons that fire for each input pattern p of the set of all known input patterns P .

10 [0072] Furthermore, let F_i^p be the set of precise firing times of a neuron i , for a particular input pattern p . Take $B^p = \{F_1^p, F_2^p, \dots, F_{N_o}^p\}$ the set of sets of precise firing times for all output neurons $1, 2, \dots, N_o$ for the particular input pattern p . Denote $B_s = \{B^p | p \in P\}$ the set of sets containing precise firing times for all input pattern p of the set of all known input patterns P .

15 [0073] In one embodiment, a network's response to input data 11 can be called unique if one or more of the following rules apply, within for example each given sampling window of a time-varying input. As a first rule, for all unique input patterns p, q in P (thus $p \neq q$), it should hold that $\Delta(A^p, A^q) \neq 0$. As a second rule, for all unique input patterns p, q in P , it should hold that $\Delta(B^p, B^q) \neq 0$.

20 [0074] The distances $\Delta(A^p, A^q)$ and $\Delta(B^p, B^q)$ can be calculated together using an appropriate spike distance metric $\Delta(\cdot, \cdot)$ such as Victor-Purpura, Van Rossum, ISI, or SPIKE-distance, among others.

[0075] These rules can be applied as objectives for configuration of the synaptic weights, where they can be restated as follows. A first weight configuration objective can be that for all A^p in A_s , maximize the distance $\Delta(A^p, A^q)$ for each A^q in A_s where q is a particular different input pattern in P . A second weight configuration objective can be that for all B^p in B_s , the distance $\Delta(B^p, B^q)$ for each B^q in B_s is maximized and the union $B^p \cup B^q$ for each B^q in B_s is minimized, where q is a particular different input pattern in P .

30 [0076] Desired spike tuples can be generated based on these weight configuration objectives, and e.g. an input encoder stage 202 can be trained using the CC-STDP as presented below. The object of this exercise is to arrive at a configuration of the input encoder stage 202, that yields unique responses which subsequent stages can use to distinguish between features in the input pattern.

[0077] Although the configuration objectives are intended to yield an optimum, this may be computationally expensive. A simpler approach is to bound the objectives, such that $\Delta(A^p, A^q)$, $\Delta(B^p, B^q)$ are increased and $B^p \cup B^q$ is decreased until the point where the output of the input encoder stage meets at least the minimum sensitivity threshold required for distinguishing between features in the subsequent processing stage. This enables the runtime of the weight configuration training process to be reduced without impacting robustness and quality of encoding.

[0078] Once the full set of spike tuples is described or generated for all patterns of interest, the weights of synaptic elements 2 of neuron 1 connections in the cells 301, 302, 303 can be trained such that they exhibit the desired unique response to the presented input pattern. In a supervised learning regime, this training is carried out using a supervised method (for example an autoencoder method, an exhaustive search, a gradient descent, using backpropagation, among others). However, this is computationally expensive. In the following, we describe a semi-supervised training methodology that relies on a new STDP weight facilitation scheme. Semi-supervised can mean that one does not say what the resulting error is, but, for example, what the desired or required spikes are to obtain a unique response to a certain input pattern.

[0079] In conventional STDP, the weight change of a synaptic element is dependent on the relative firing time of two neurons 1 connected to one another by that particular synaptic element 2. However, the resulting weight changes are largely a consequence of localized behaviour. Causal Chain STDP (CC-STDP) instead predicates weight changes on the longer chain of causation of a pre-synaptic neuron, i.e. weights are adjusted only if the firing event contributes to firing of neurons in subsequent layers of the network.

[0080] FIG. 4 shows the principle of the CC-STDP learning rule, which can identify deep causal, anti-causal, and non-causal relationships between the pre-synaptic neuron N1 and post-synaptic neuron N2 of a particular synaptic element 2 with weight $w_{1,2}$ and neurons N3, N4, N5 in subsequent layers of the network.

[0081] The CC-STDP rule can be applied in the semi-supervised training of cells to obtain specific output spike responses. For instance, assume n is the particular output neuron where a spike is desired at a specific time $t_{\text{spike } n}$. During training, by inducing a spike artificially at time $t_{\text{spike } n}$ at neuron n , the nature of the relationship between neurons in the early layers of the network and neuron n can be established. These relationships may be causal (neuron in

early layer causes a spike at the output neuron), anti-causal (neuron in the early layer does not cause a spike at the output neuron, but the two neurons are synchronized in their firing, with the early neuron firing after the output neuron), or non-causal (the two neurons do not have any synchrony in their firing behaviour). CC-STDP can thus enable the identification of causal relationships between neurons in preceding layers of the network and the desired output neurons, and can cause the weights of intervening synaptic elements along this path of causation to be adjusted.

[0082] At neurons where a spike may not be desired, anti-causal relationships can be emulated by inducing a spike concomitant with the presentation of the input sample to the cell, i.e. in the same time interval. This can cause the output neurons to fire before the neurons in all preceding layers, thus yielding an anti-causal relationship, and in turn, causing localized depression of intervening synaptic elements along this path.

[0083] Thus, by inducing spikes at the desired output neuron(s) at a specific time $t_{spike\ k}$ and at other neurons at a time concomitant with the application of the input stimulus, the network is configured to respond with the desired spiking response pattern.

[0084] The weight $w_{i,j}$ of a synaptic element positioned between neuron i and neuron j can for example change according to the following rule:

$$\Delta w_{i,j} = \begin{cases} \delta w_{i,j} \text{ for } \delta w_{i,j} > 0 \text{ if } w_{j,k} \neq 0 \text{ and } t_{spike\ k} > t_{spike\ j} \text{ and } t_{spike\ k} > t_{spike\ i} \\ \delta w_{i,j} \text{ for } \delta w_{i,j} < 0 \\ 0 \text{ if } w_{j,k} = 0 \text{ or } t_{spike\ k} \leq t_{spike\ i} \text{ or } t_{spike\ k} \leq t_{spike\ j} \text{ or } t_{spike\ k} = 0 \end{cases} \quad (1)$$

[0085] Here in equation (1), $\Delta w_{i,j}$ is the effective weight change for the synaptic element between neuron i and j , $w_{j,k}$ is the weight of the synaptic element between neuron j and a neuron k of the subsequent layer. $t_{spike\ i}$, $t_{spike\ j}$ and $t_{spike\ k}$ refer to the precise times at which neuron i , j and k respectively spike. $\delta w_{i,j}$ is the weight change in the synaptic element between neuron i and j , due to their relative time of spiking, and may be computed using the STDP learning rule.

[0086] Thus, according to the learning rule of equation (1), $\Delta w_{i,j} = \delta w_{i,j}$ if the weight change in the synaptic element between neurons i and j is ought to be positive and if the weight between neurons j and k is non-zero and if neuron k spikes later than both neurons i and j . If the weight change in the synaptic element between neurons i and j is ought to be negative then $\Delta w_{i,j} = \delta w_{i,j}$. Lastly, the weight change will be zero if the neuron k does not spike, if the neuron k spikes immediately, or if the neuron k spikes earlier than either neuron i

or j , of if the weight between neuron j and k is zero. In the latter case, there seems to be no causal relationship between neuron i firing and neuron k firing.

[0087] Alternative learning rules may also be used, in which case the spike time parameters in the above equation may be replaced by the fundamental parameter (e.g. spike rate) of the chosen learning rule.

[0088] This principle can be explained with reference to FIG. 4. For example, a spike is induced at a specific time in specific output neurons (for example, in neuron $N3$ at spike time $t_{spike\ N3} = 4$ ms, in neuron $N4$ at spike time $t_{spike\ N4} = 3$ ms, and no induced spike at neuron $N5$). In this way a desired specific spatio-temporal spike output pattern is made. This desired output pattern belongs to a specific spatio-temporal input pattern (for example, neuron $N1$ firing at $t_{spike\ N1} = 1$ ms) of the spiking neural network. For example, due to this input pattern and the specific synaptic weight $w_{1,2}$, the neuron $N2$ fires at $t_{spike\ N2} = 3$ ms. The effective weight change for synaptic element $\Delta w_{1,2}$ can be calculated using for example equation (1) above. Since $w_{2,3} \neq 0$, and if $w_{1,2} > 0$ then the contribution from neuron $N3$ firing at this specific time is $\delta w_{1,2}$. Since $w_{2,4} = 0$, the resulting change from neuron $N4$ firing on the synaptic weight is zero. Since neuron $N5$ does not spike, the resulting change from neuron $N5$ is zero.

[0089] In an embodiment where CC-STDP is used to train the spiking neural network, it can be necessary to be able to induce and/or inhibit spike generation at specific neurons 1 at specific times. As an example, this can be done in the following ways (this list is not exhaustive). To induce a spike, one method would be to drive the corresponding neuron's membrane to a voltage that exceeds the firing threshold, by means of a bias voltage input, thereby inducing the generation of a spike. Another way might be to inject an artificial spike into the neuron's output at the desired time. To prevent a spike, one method would be to drive the corresponding neuron's membrane to the refractory voltage, or lowest possible voltage, preventing the generation of a spike. Alternatively, one could disable spike generation at the neuron by for example disconnecting neuron inputs from the integrator, thereby for example disconnecting the spike generation circuitry from the output. These signals for inducing or disabling spike generation in the neurons are illustrated as the teacher inputs 13A, 13B, 13C, which can be used to obtain the desired unique response for each cell 301, 302, 303.

[0090] FIG. 5A shows the composition of a sensor-fused pattern recognizer through the fusion of two separate data streams 11A, 11B, and subsequent pattern classification using the

fused stream 12C in a pattern classifier stage 204. The output 12D of the pattern classifier stage 204 could for example be a set of spatio-temporal spike trains 12D generated by the output neurons of the pattern classifier stage 204, where each spike train of the set 12D is generated by a single specific output neuron in the pattern classifier stage 204.

5 [0091] Namely, in case a layer has multiple concurrent cells in a stage (e.g. multiple input encoders 202A, 202B forming the input encoder stage in a fusion system), each of these cells 202A, 202B can be configured first before being instantiated in a cell with a subsequent stage 203. All stages of the network may be trained using teacher inputs 13A1, 13A2, 13B1, 13B2, 13C, 13D. In the event that all the concurrent cells perform an identical function on identical
10 data (e.g. input encoding of different receiver antennas in a radar transceiver, or of different microphones pointed in different directions) it might not be necessary to run separate training sequences for each instance. Rather, a single trained cell can be instantiated multiple times. As a result, a trained cell or the configuration thereof can be implemented in a different spiking neural network to perform the same functionality. Therefore, during the composition
15 of a spiking neural network, cells that have been trained already can be implemented, if the same functionality is required, without any new training being necessary. The composition of the spiking neural network will thus take less time, and will be less complex.

[0092] A memory device can be configured to be used as a template library of trained cells. On such a memory device, the software- and/or hardware settings of one or more trained cells
20 of a spiking neural network can be stored. The configuration settings of the synaptic elements 2 of such a cell can be for example the respective weights of the synaptic elements. The configuration settings of the neurons 1 of such a cell can be for example the firing threshold, leakage rate, resting value, refractory level, refractory period, or other parameters of the neuron.

25 [0093] Each cell is trained to perform a particular functionality. When in a particular spiking neural network such a new cell with that particular functionality is necessary, the cell does not have to be trained from scratch, which saves time and enables spiking neural networks to be implemented much more efficiently. Instead, the configuration settings from pre-trained cell designed for that functionality can be obtained from the memory device that
30 is configured as a template library. By instantiating the new cell using the pre-trained cell configuration settings, the new cell will be able to function in the same way as the pre-trained cell.

[0094] The memory device can be any computer-readable storage media. The memory device may be non-transitory storage media. Illustrative computer-readable storage media include, but are not limited to: (i) non-writable storage media (e.g., read-only memory devices within a computer such as CD-ROM disks readable by a CD-ROM drive, ROM chips or any type of solid-state non-volatile semiconductor memory) on which information may be permanently stored; and (ii) writable storage media, e.g., hard disk drive or any type of solid-state random-access semiconductor memory, flash memory, on which alterable information may be stored.

[0095] The template library may also be a collection of hardware spiking neural network cells, which are pre-trained to each perform a particular function. These hardware cells can be directly used in a spiking neural network that requires a cell with that particular function.

[0096] In FIG. 5A, two data converter stages 201A, 201B may be trained with input pattern signals 11A, 11B respectively in cells 301A, 301B, generating spike response outputs 12A1, 12AB. The configuration of cells 301A, 301B may be saved and instantiated in cells 302A, 302B, each also including the subsequent stage, i.e. input encoder stage 202A, 202B. The cells 302A, 302B may be trained with input pattern signals 11A, 11B respectively, which results in the outputs 12A1, 12B1 of the data converter stages 201A, 201B becoming the input data to the input encoder stages 202A, 202B. The configuration of cells 302A, 302B may be similarly saved and instantiated in cell 303, which also includes the subsequent stage, i.e. data fusion stage 203. Cell 303 may be trained using both input pattern signals 11A and 11B, which results in the outputs 12A2 and 12B2 of the input encoder stages 202A and 202B becoming the input data to the data fusion stage 203. The spatio-temporal spike trains of the outputs 12A2 and 12B2 of the input encoder stages 202A and 202B are subsequently co-incident on the one or more neurons 1 of the data fusion stage 203. The output 12C of the data fusion stage 203 is a temporal spike train (if data fusion stage 203 comprises only one output neuron 1) or a set of spatio-temporal spike trains (if data fusion stage 203 comprises multiple output neurons 1), generated in response to joint-stimulation by multiple spike trains (outputs 12A2 and 12B2 from input encoder stages 202A, 202B). The resulting fused set of spike train(s) 12C represents the combination of feature spaces of all input pattern signals 11A, 11B that have been fused. Consequently, the fused spike train encodes all the features that are present in the input pattern signals in the current time window.

[0097] For example, when fusing two sets of input spike trains 12A2, 12B2 containing a feature e and f of the input patterns 11A, 11B respectively, the resulting fused spike train(s) 12C contain the corresponding unique representations of both features e and f . The goal of configuring the data fusion stage 203 is to achieve this unique mapping of the complete feature space of each input pattern signal 11A, 11B to a feature sub-space of the fused output signal 12C.

[0098] Consider two input encoders α and β , which encode features in their respective input pattern signals M_α and M_β , and produce two sets of spike trains P_α and P_β .

[0099] The set of u known features in signal M_α one would like to encode is given as $\Phi(\alpha) = \{\phi_1, \phi_2, \dots, \phi_u\}$, which is thus the feature space of signal M_α . Each feature ϕ_i for i in $\{1, \dots, u\}$ is encoded as a set of spatio-temporal spike trains. $A_\alpha = \{A_1, A_2, \dots, A_u\}$ is the set of sets of output neurons A_i that fire for feature ϕ_i . The set of sets containing precise firing times of neurons for all u known features is $B_\alpha = \{B_1, B_2, \dots, B_u\}$, where each B_i in B_α is the set of precise firing times of output neurons for feature ϕ_i .

[00100] For data fusion to be lossless, the feature space of the fused spike train(s) $\Phi(\gamma)$ can encompass the individual feature spaces of the disparate input signals:

$$\Phi(\gamma) = \Phi(\alpha) \cdot \Phi(\beta) + \Phi(\alpha) + \Phi(\beta). \quad (2)$$

[00101] The spatio-temporal spike trains corresponding to feature space $\Phi(\gamma)$ can adhere to the rule 1 and 2 as stated for input encoding, i.e. such that they can have sufficient uniqueness to enable subsequent processing stages to distinguish between spike-coded features. Thus, the generation of spike tuples can use the unique response methodology described above, wherein each unique combination of features in the input spike-trains yields a unique set of output spatio-temporal spike trains in the fusion block.

[00102] FIG. 5B shows schematically how multiple temporal spike trains 30A, 30B are fused into a single fused spike train 35 in a working example. The temporal spike trains 30A, 30B are each transmitted through synaptic elements 31A, 31B and inbound on a neuron 34. The neuron 34 can have recurrent connections 32 connected to the neuron 34 via one or more synaptic elements 33. The fused spike train 35 represents the combination of feature spaces of both the first temporal spike train 30A and second temporal spike train 30B. Subsequent processing stages can distinguish between the spike-encoded features of the first temporal spike train 30A and second temporal spike train 30B because the representations of these features in the fused spike train 35 are sufficiently unique.

[00103] In a signal recognition system that uses data fusion, data sources may vary in importance based on their impact on the final outcome. Certain features may be primary to the signal recognition process, while others only serve to enhance the quality of the recognition outcome. Consequently, the size of A_γ , which is the set of spike trains

5 representing the fused feature spaces of A_α and A_β can be reduced in size by diminishing the effect of less important features. This may be done in at least three ways.

[00104] First, by introducing anti-causal spikes in the output neurons of the data fusion stage 203, concomitant with the less important features of particular input spike trains. This method can be applicable in the case where none of the input patterns contains a desired feature.

10 [00105] Second, less desired features may also be suppressed by introducing a weight bias in the input synaptic elements of the data fusion stage 203 which receive the outputs from each input encoder stage 202A and 202B, corresponding to the priority of each in the fusion process. Consequently, spike trains from the input encoder stages 202A, 202B can be

15 they are fused. This facilitates the control of the energy transfer between an input signal and the resulting fused representation. Increasing the priority bias of a signal can raise the impact of that signal on the fused representation. Maximizing the bias for a signal causes the fused representation to replicate the input signal. Conversely, minimizing the priority bias weights for a signal, can reduce its impact on the fused representation, whereas completely nullifying the weight eliminates the signal's influence on the fused result altogether. During fusion, the spatial spike distributions A_α and A_β are transformed to a fused spatial distribution A_γ through the transfer function f realized by the configuration of the data fusion stage 203.

Thus, in a priority biased embodiment, the fused distribution takes the form

$$A_\gamma = f\left((A_\alpha, w_\alpha), (A_\beta, w_\beta)\right) \text{ and } B_\gamma = f\left((B_\alpha, w_\alpha), (B_\beta, w_\beta)\right). \quad (3)$$

25 [00106] In equation (3), w_α and w_β are the priority biasing weight sets that control the influence of the spike train sets P_α and P_β on the data fusion stage 203.

[00107] The third way, arguably the simplest, involves not suppressing or diminishing any feature, instead including its influence in the fused representation. The representation corresponding to this feature may simply be ignored in the subsequent classification stage

30 204, i.e. assigned a class that is not of interest. This approach contributes to increased

accuracy as it distinctly classifies the undesired feature combinations, thereby enabling more accurate identification of desired features.

[00108] Next, the application of the above principles in a system for fusion and pattern recognition is described.

5 [00109] FIG. 6 is a simplified block diagram of an embodiment of a system 700 for data fusion and automatic signal recognition, based on a spiking neural network 400 interfaced with a microprocessor or hardware logic system 600.

[00110] A spiking neuron network 400 comprises one or multiple input encoder stages $401_1, \dots, 401_N$; each of the input encoder stages $401_1, \dots, 401_N$ being configurable by configuration data 717. The input encoder stages $401_1, \dots, 401_N$ may receive input pattern signals $711_1, \dots, 711_N$ and convert these input pattern signals into spatio-temporal spike trains $P_1, \dots, P_N$ encoding the feature space of each input pattern signal $711_1, \dots, 711_N$. The set of input pattern signals $711_1, \dots, 711_N$ may be derived, for example, by sampling input pattern data to generate the set of signals, e.g. recording an image using a CCD array image sensor to
 10 generate a set of input pattern signals $711_1, \dots, 711_N$ each representing image data recorded by one or more pixels of the image sensor.

[00111] Each spike train $P_1, \dots, P_N$ may comprise a sequence of precisely timed voltage spikes, generated by one or more neurons of the input encoder stages $401_1, \dots, 401_N$ in response to each input pattern signal $711_1, \dots, 711_N$.

20 [00112] If multiple neurons are used to encode a single input pattern signal $711_1, \dots, 711_N$, the output from each of the multiple neurons may be a spike train comprising one or more spikes with one or more of the above mentioned temporal characteristics. The aggregated spike trains $P_1, \dots, P_N$ generated by output neurons of the input encoder stages $401_1, \dots, 401_N$ together represent the encoded spatio-temporal spike response for the presented input pattern
 25 signals $711_1, \dots, 711_N$.

[00113] FIG. 7A shows how neurons 1 are interfaced with an input signal 801, such as an output signal from a synaptic element 2. The neuron 1 has a certain configuration which will alter the way the neuron 1 reacts to input signal 1, where the neuron configuration may be set using configuration signal C_N . Depending on the input signal 801 and the neurons
 30 configuration, the neuron 1 will output a certain spatio-temporal spike train S . In FIG. 7B, the input signal 802 is a combination of synaptic output signals from multiple synaptic elements $2_A, 2_B$ each configured in a certain way, where the configuration of each synaptic element

may be set using configuration signals C_{SB} , C_{SB} . The synaptic elements 2_A , 2_B can have as input an input signal 803 or recurrent input from another neuron 804.

[00114] Each neuron 1 can have a set of configuration parameters C that control the precise firing behaviour of that neuron 1. For example, the neuron may be designed with a firing

5 threshold, which represents a threshold of a voltage, energy, or other variable which accumulates in the neuron as a result of receiving inputs, and where the neuron generates an output spike (such as a voltage, current or energy spike) when the accumulated variable meets or exceeds the firing threshold. The neuron may implement an integration function which integrates the inputs to the neuron to determine the adjustment to the accumulated

10 variable. In addition, the neuron may also be designed with: (a) a leakage rate, which represents the rate at which the accumulated variable in the neuron decays over time; (b) a resting value of the accumulated variable, which represents the value to which the accumulated variable will decay over time in the absence of any input signals to the neuron; (c) an integration time constant, which represents the time over which an input signal is

15 integrated to determine any increase or decrease in the accumulated variable in the neuron; (d) a refractory level, which represents the value of the accumulated variable in the neuron immediately after firing of the neuron; (e) a refractory period, which represents the time period required for the accumulated variable in the neuron to rise to the resting value after firing of the neuron. These parameters may be predetermined and/or configurable and/or

20 adjustable for each neuron. By adjusting for example the firing threshold, leakage rate, integration time constant, and refractory period of the neuron to match the energy content of a critical input signal feature, the neuron 1 will generate one or more precisely timed spikes when stimulated with an input signal containing that feature.

[00115] Referring back to FIG. 6, each input encoder stage $401_1, \dots, 401_N$ may comprise

25 multiple spiking neurons 1. The inputs to the input encoder stages $401_1, \dots, 401_N$ may be analog or digital values (e.g. analog-to-digital converted values of input pattern data. When using a neuron with a digital integrator, the digital value of each input signal to the neuron is provided as a direct input to the neuron's integrator. The bit width of the input signal (B_s) and bit width of the integrator (B_i) are matched by one or a combination of the following

30 exemplary methods.

[00116] As a first method, the trailing ($B_s - B_i$) bits of the digital input are discarded, when $B_s > B_i$. As a second method, the digital input is padded with ($B_i - B_s$) trailing '0' bits,

when $B_i > B_s$. As a third method, the digital input is padded with B_n trailing '0' bits, or B_n trailing bits are discarded, where B_n can be a configurable parameter that allows the digital input's amplitude to be shifted within the voltage range of the membrane. This shifting can also be realized when the input signal and integrator bit widths are matched, by using a combination of padding and truncating of the sample value.

[00117] When using a neuron 1 with an analog integrator, the digital value can be provided as pre-synaptic inputs to B_s synaptic elements 2, the post-synaptic output terminals of which are connected to the neurons 1 of the input encoder stage $401_1, \dots, 401_N$. Amplitude shifting can be realized by padding and/or truncating the input signal vector (comprising a string of bits) by B_n bits, as previously described for the digital neuron. In another embodiment, the analog value of a input signal can be provided directly as input to the analog integrator. The analog value can be presented to the integrator for a period of time, for example causing the integrator to accumulate charge. By selecting a suitable threshold voltage, the neuron's exact firing time can be varied in inverse proportion to the amplitude of the analog input signal value (i.e. the higher the amplitude, the sooner the neuron fires). Further, by utilizing multiple neurons 1 with appropriately selected sensitivity thresholds, a complex multi-level analog to spike converter can be realized.

[00118] In such an embodiment, data converter stages $201_1, \dots, 201_N$ may be included, which convert the presented analog voltage value of the input pattern signals $711_1, \dots, 711_N$ into a unique, equivalent spike representation, e.g. where the distance between input values and the corresponding distance between output spike responses are proportional. The input encoder stage $401_1, \dots, 401_N$ then encodes the features identified in the transcoded spike response from the data converter stages $201_1, \dots, 201_N$ into a unique representation.

[00119] Input pattern signals $711_1, \dots, 711_N$ supplied to the system 700 may have varying dynamic ranges: some low, some high. While the absolute dynamic range may not be important in itself, the ability to distinguish between features at different amplitudes can be important. It might consequently be necessary to ensure that the configuration of the input encoder stages $401_1, \dots, 401_N$ results in all known features of interest, and at whichever amplitude they occur, being encoded into a unique spike representation, i.e. the intent may be to obtain unique spike characteristics at more than one amplitude in the range of interest.

[00120] Consequently, the input encoder stages $401_1, \dots, 401_N$ may comprise a group of neurons 1 that connect an arbitrary number of inputs and produce an arbitrary number of

spike train outputs $P_1, \dots, P_N$ that encode the features present in the input pattern signal $711_1, \dots, 711_N$, where an input signal $711_1, \dots, 711_N$ can be defined for example as a single sampled data point of an input pattern signal, or a spatial or temporal or spatio-temporal series of sampled data points of an input pattern signal.

- 5 [00121] Features or patterns to be detected in the input pattern signal can be, for example, the specific content of a single sampled input data point or set of sampled input data points, e.g. the amplitude of an input signal at a specific point in time, the presence of “ON” pixels at a certain location in a binary image frame, intensity of a particular input signal attribute at a specific point in time.
- 10 [00122] Features or patterns to be detected can also be in the spatial sequence of content in a series of sampled data points forming the input signal, e.g. “ON” pixels corresponding to the edge of an object in a binary image.
- [00123] Features or patterns to be detected can also be in the temporal sequence of content across multiple samples of a single data point forming the input signal, or in the temporal
- 15 sequence of content across multiple spatial sequences of sampled data points. For example, a rising sequence of amplitudes, followed by falling sequence of amplitudes, characterizing a triangular wave pattern in an input signal, or displacement of an edge of an object across multiple frames of a video image forming an input signal, corresponding to the movement of an object over time.
- 20 [00124] Referring back to the embodiment shown in FIG. 6, the spike train outputs $P_1, \dots, P_N$ generated by the input encoder stages $401_1, \dots, 401_N$ are first optionally weighted according to a priority biasing scheme in a priority biasing module 402 as explained above and then fused in a data fusion stage 403 depicted in FIG. 6. The resulting spatio-temporal spike trains 712 are then received by classifier stage 404, which classifies the spatio-temporal
- 25 spike trains 712. The classifier stage 404 can identify the strongest spatio-temporal pattern present in the transcoded representation and associates this with for example a specific output neuron.
- [00125] The classifier’s input can be for example either a set of spike trains $P_1, \dots, P_N$ from the input encoder stages, for example in the case of a system with only one set of input
- 30 signals $711_1, \dots, 711_N$, or a set of spike trains 712 from a data fusion stage 403, in the case of a system with multiple sets of input signals $711_1, \dots, 711_N$ which are encoded and thereafter

fused into a representative set of spike trains 712. In FIG. 6 the classifier stage 404 obtains its input 712 from the data fusion stage 403.

[00126] The classifier's output 713 can be a classified set of spatio-temporal spike trains 713 on an output port R, where each spike train 713 can be generated by a single specific output neuron in the classifier 404, or by multiple output neurons. In a classifier embodiment compliant with the unique response principle, spikes can be generated by a separate output neuron for each unique input feature or unique combination of input features in the input pattern signals. In this embodiment, the spike tuple for each pattern of interest consists of a single neuron (i.e. A^p is a unit set for a pattern p) which spikes at a specific time interval, with at least one spike. The output port, consequently, can contain at least as many spike-trains as patterns of interest.

[00127] Restating this in terms of the weight configuration objectives described earlier, take A to be the generic set of the classifier's output neurons that generate output spike trains 713 in response to known input patterns, and let A^p be the subset of A , containing neurons that fire for a given known input pattern p , and let $A_s = \{A^p \mid p \in P\}$ be the superset of all A^p , for all p in the set of known input patterns P , then for a classifier stage 404 for which the above holds: each A^p in A_s will be a unit set, and for each q in P , with $p \neq q$, it holds that $A^p \neq A^q$.

[00128] Next, looking again at FIG. 6, a logic unit 500 comprising an output transformation stage 501 can transform the output 713 from the classifier stage 404 (e.g. a set of spatio-temporal spike trains 713) into for example a digital code 714 which can be stored and retrieved from for example a dictionary 601.

[00129] The output of the classifier stage 404 may thus be a set of spatio-temporal spike trains 713, wherein one or more specific spike trains 713 exhibit specific characteristics, thus for example allowing the input pattern $711_1, \dots, 711_N$ to be identified as belonging to a particular class, or label. The output transformation stage 501 can facilitate this identification process by transforming the spike train characteristics into a digital code 714 which can be stored and retrieved from a dictionary 601.

[00130] Feature information in spike trains may be encoded in several forms, for example: a population code, in which output neurons 1 of the classifier stage 404 fired in the elapsed time period in which the input stimulus was presented; a population temporal code, in which output neurons 1 of the classifier stage 404 fired first within a specific time period since the input stimulus was presented; a population rate code, that describes how many spikes were

generated by output neurons 1 of the classifier 404 in the elapsed time period in which the input stimulus was presented; a temporal code, which describes the precise time at which a specific output neuron 1 of the classifier stage 404 fired; and finally a rate code, which describes the firing rate of a specific output neuron 1 of the classifier stage 404.

5 [00131] The combination of spatial and temporal spiking information can provide a greater degree of detail than a purely spatial classifier 404 output. For instance, consider that an input pattern signal $711_1, \dots, 711_N$ contains two dominant features X and Y, one stronger than the other. The relative strength of the two features could be determined by measuring the firing rate of their respective classification stage outputs 713. Alternatively, the stronger feature
10 could be identified based on the temporal code of the corresponding output neurons 1, specifically by determining which output neuron 1 fired sooner.

[00132] The output transformation stage 501 can contain circuits to analyse the spike activity on the classifier's output port. Circuits are activated for example according to a user-configuration 717, based on the type of analysis the user wishes to carry out on the spiking
15 behaviour of the array's neurons 1: spiking population, precise spike timing, firing rate, and combinations thereof.

[00133] For the code type population, the method of transformation consists of having one register per output neuron of the classifier stage 404, to denote if the neuron spiked in the present time interval following application of the input pattern signal. For the code type
20 population temporal, the method of transformation consists of having one-register per output neuron of the classifier stage 404, to denote which neurons spiked first in the present time interval, following application of the input pattern signal. After the earliest spike event is registered, the other registers are disabled. For the code type population rate, the method of transformation consists of having counters per output neuron of the classifier stage 404, to
25 count the number of spikes generated by the output neuron in the present time interval following application of the input pattern signal. Counters with non-zero spike count values also denote a population code. For the code type temporal, the method of transformation consists of having a time-to-digital converter per output neuron of the classifier stage 404, to precisely measure the time elapsed since application of the input pattern signal until the
30 output neuron spikes. Finally, for the code type rate, the method of transformation consists of having a counter per output neuron of the classifier stage 404, to count the number of spikes

generated by the output neuron in the present time interval following application of the input pattern signal.

[00134] The input to the output transformation stage 501 can be a vector 713 of spikes from all output neurons 1 of the classifier stage 404, and the output 714 from the circuit 501 can be a set of digital values corresponding to the type of output code selected by the user. The inputs 713 to the output transformation stage 501 can be presented in the duration of the time window in which a signal sample is valid. The outputs 714 from the output transformation stage 501 can be valid at the end of the time window, and can be reset before the next time window begins. The classification output 713 can be transformed into a single hash code, or set of hash codes that can be utilized as a unique fingerprint of the patterns in the input pattern signal(s) 711₁, ..., 711_N.

[00135] Next, looking again at FIG. 6, a microprocessor or logic circuit 600 comprising a hash table or dictionary lookup functionality 601 takes the digital code 714, and subsequently the hash-table or dictionary lookup operation can be performed to determine the user-space identifier corresponding to the recognized feature.

[00136] This operation allows the decoding of the output of the classifier stage 404 into the user-space, i.e. user-defined identifier of input patterns. During the configuration phase, the pattern recognizer 400 may be trained with sample features and patterns, and the generated hash codes can be recorded. Sample patterns can be assigned a user-defined identifier which may be stored alongside the generated hash code in a hash-table or dictionary 601, for example. At runtime, when the recognizer 400 is used to recognize a feature in a signal, the generated hash code is used in a hash-table or dictionary lookup operation to determine the user-space identifier corresponding to the recognized feature.

[00137] A reset generator and sequencer 718 shown in FIG. 6 can reset for example the contents of the data fusion stage 403, classifier stage 404 and output transformation stage 501, for example at the start of a new sampling window. This operation removes residual spikes from the network 400, and allows the next sampling window to be evaluated without the influence of previous windows.

[00138] The sampling window can be implemented as an input buffer of depth equal to the window size. As an alternative, a configurable virtual sampling window can be realized through the use of a periodic reset signal that arrests spiking behaviour in the network and resets the output stages. The sampling window in such a method can have the same period as

the reset signal. When configuring the sampling window, the period should be sufficiently long as to fully accommodate the longest known feature, yet small enough that the feature content of each window is minimized, thus increasing the accuracy of feature recognition.

The width of the sampling window can influence the granularity of features the system 700 can detect. In general one could say that the sampling window duration must be larger than the timescale at which the feature presents in the input signal, with a sampling rate greater than or equal to the Nyquist rate. Features at different timescales can be detected through the use of multiple input encoder stages $401_1, \dots, 401_N$, or a single input encoder stage $401_1, \dots, 401_N$. The length of the sampling window can also have an impact on the width of counters used in the output transformation stage 501.

[00139] FIG. 8 shows an example of the generation of a reset pulse. Clock information 904 alters the value of an up-counter 901. Together with information on a sampling period 905, the value of the up-counter 901 is submitted to the comparator 902. The comparator 902 checks whether the up-counter 901 value is still within the sampling period 905. If this is no longer the case, the comparator 902 sends a signal to a pulse generator 903, which sends a reset signal 906 to both the up-counter 901 and to the respective modules within the system 700 as a signal 716 in FIG. 6.

[00140] Similarly, for the reset signal, the period between two consecutive reset pulses could for example follow the following formula:

$$t_{reset} \geq \max(t_{feature}) + \max(t_{fallthrough}) + t_{output} + \max(t_{lookup}). \quad (4)$$

[00141] In equation (4), $t_{feature}$ is the duration of features, $t_{fallthrough}$ is the fall through latency for the spike trains to propagate from the input encoder stage to the output of the classifier stage, t_{output} is the latency of the output transformation stage to generate a hash or set of hashes, and t_{lookup} is the worst-case latency for the hash-table/dictionary lookup operation.

[00142] The duration of the reset pulse must be sufficient to trigger the reset circuits of the appropriate components. During a reset, the state of all neurons 1 in the network 400 is reset i.e. for example membrane voltage brought back to resting, interfacing signals are reset to their initialized state (i.e. the state they had after the system was programmed), synaptic readout circuitry is reset (weight is retained), and inflight synaptic updates are abandoned. In the output stage 500, counters and registers tracking the population and rate of firing in neurons 1 can also be reset. The reset does not affect the input encoder stages $401_1, \dots, 401_N$,

ensuring a continuous encoding of input pattern signals $711_1, \dots, 711_N$ to spikes even during the transition between sampling windows.

[00143] FIG. 9 illustrates a method 1000 to perform the fusion of multiple disparate input pattern signals into a single, representative set of spike trains, and subsequently perform automatic signal recognition upon this spike train in order to detect and identify patterns/features. The method 1000 starts at the starting point 1000S.

[00144] In a first step 1001, new signal samples are encoded as spatio-temporal spike trains by one or multiple input encoder stages $401_1, \dots, 401_N$. The output of the one or multiple output encoders are spatio-temporal spike trains $P_1, \dots, P_N$.

[00145] In a second step 1002, the spike trains $P_1, \dots, P_N$ are weighted according to their significance in the fusion by priority biasing. The output of the priority biasing module 402 are weighted disparate spatio-temporal spike trains.

[00146] In a third step 1003, the disparate spatio-temporal spike trains are unified into a representative set of spike trains 712 encompassing individual feature sub-spaces by a data fusion stage 403. The output of the data fusion stage are fused spatio-temporal spike trains 712.

[00147] In a fourth step 1004, a classifier stage 404 classifies the spatio, temporal and rate features, in the post-fusion spike trains 712. The temporal spike trains are thus categorized.

[00148] In a fifth step 1005, the output transformation stage 501 converts the categorized spike trains 713 into classification code 714.

[00149] In a sixth step 1006, the method might wait until a new sampling window starts.

[00150] In a seventh step, the lookup and output in the hash table or dictionary 601 of the pattern identifier 700 corresponding to the classification code 714 is performed.

[00151] Finally, in an eight step 1008, the data fusion stage 403, classifier stage 404 and output transformation stage 501 are reset.

[00152] The method can then be repeated starting at the first step 1001 again.

[00153] In general, the system and method described herein use a network of spiking neurons 1 as a means of generating unique spatio-temporal spike trains in response to a unique stimulus, where the uniqueness of responses is controllable through the operating parameters of the spiking neurons 1 and interconnection network. The system consists of an array of spiking neurons 1 with configurable parameters, interconnected through synaptic elements 2

with configurable parameters. By virtually partitioning the array into multiple different networks, the system can simultaneously realize multiple functionalities.

[00154] For instance, one network might realize the conversion of multiple, disparate, sampled input signals into a representative set of spatio-temporal spike trains which fully encodes the feature space of the input signal.

[00155] In a second configuration, the invention might realize the conversion of multiple, disparate, sampled input signals into a singular representative set of spatio-temporal spike trains with a feature space that is a superset of the individual feature spaces of all input signals.

[00156] In a third configuration, the invention might realize the classification of spatio-temporal spike trains into one or more output classes through appropriate configuration of the network's uniqueness.

[00157] The configurability of the neurons, synaptic elements and interconnection network of the invention facilitates the simultaneous realization of all three configurations on separate groups of neurons in the same array, in different combinations. The synaptic elements act as a configurable switch matrix.

[00158] A spiking neural network 1100 according to an embodiment is shown in FIG. 10, where neurons $N_0, \dots, N_7$ are arranged in an array of neurons 1101, interconnected through a configurable switch matrix consisting of an array 1102 of synaptic elements 2 configurable via a configurable switch 1103.

[00159] Input signals 1104 may be transmitted to the neurons $N_0, \dots, N_7$ via the array of synaptic elements 1102 in accordance with the weights of the synaptic elements 2. In addition, two neurons $N_0, \dots, N_7$ may be interconnected via the array of synaptic elements 1102 and the configurable switch 1103, in accordance with the weights of the synaptic elements 2 and the connection topology of the configurable switch 1103. Positive weights in the synaptic elements 2 connecting an input signal 1104 to a neuron $N_0, \dots, N_7$, or interconnecting a pair of neurons $N_0, \dots, N_7$, results in excitation of the post-synaptic neuron $N_0, \dots, N_7$. Negative weights in these connecting synaptic elements 2 results in inhibition of the post-synaptic neuron $N_0, \dots, N_7$, and a zero weight in these connecting synaptic elements 2 results in no connection to the post-synaptic neuron $N_0, \dots, N_7$.

[00160] The configuration parameters for neurons $N_0, \dots, N_7$ and synaptic elements 2, interconnection topology, and output stage configuration parameters can be loaded from a

configuration memory every time the spiking neural network 1100 is set to an initialization state. This operation causes neurons N0,...,N7 to be interconnected according to the loaded interconnection topology and synaptic weights, neuron and synaptic element parameters to be initialized according to the relevant loaded configuration parameters, the system parameters to be initialized according to the relevant loaded configuration parameters, and the output dictionary or hash-table to be populated with entries stored in the configuration memory.

[00161] Input signals 1104 are submitted to a subset of the synaptic elements 2, and the network 1100 produces a certain output 1105, that can be send to an output decoder.

[00162] The spiking neural network 1100 may be implemented as shown in FIG. 11, which shows an example of a configured synaptic array 1202, illustrating an example mapping of the described neural network stages onto neurons N0,...,N7 and synaptic elements 2.

[00163] The input encoder stages 1211A, 1211B, data fusion stage 1212, and classifier stage 1213 can be implemented as illustrated in the spiking neural network 1200. The output transformation can be implemented using digital logic circuits, and the hashtable/dictionary lookup operations can be implemented, in one embodiment, as software code executed on the microprocessor accompanying the spiking neural network, and in another embodiment, using digital logic circuits.

[00164] The principle underlying each of the network stages 1211A, 1211B, 1212, 1213 is the same, each stage 1211A, 1211B, 1212, 1213 generates a unique response, according to its configuration parameters, upon being presented with an input stimulus. The objective of the input encoder stages 1211A, 1211B can be to generate a unique response for each unique feature present in the input pattern signals 1204A, 1204B. The objective of the data fusion stage 1212 can be to generate a unique response for each unique combination of input spike trains from the input encoder stages 1211A, 1211B. The objective of the classifier stage 1213 can be to generate a unique classification response for each unique feature or combination of features.

[00165] One or more embodiments may be implemented as a computer program product for use with a computer system. The program(s) of the program product may define functions of the embodiments (including the methods described herein) and can be contained on a variety of computer-readable storage media. The computer-readable storage media may be non-transitory storage media. Illustrative computer-readable storage media include, but are not limited to: (i) non-writable storage media (e.g., read-only memory devices within a computer

such as CD-ROM disks readable by a CD-ROM drive, ROM chips or any type of solid-state non-volatile semiconductor memory) on which information may be permanently stored; and (ii) writable storage media, e.g., hard disk drive or any type of solid-state random-access semiconductor memory, flash memory, on which alterable information may be stored.

- 5 [00166] Two or more of the above embodiments may be combined in any appropriate manner.

CLAIMS

1. A spiking neural network for classifying input pattern signals, comprising a plurality of spiking neurons implemented in hardware or a combination of hardware and software, and a
5 plurality of synaptic elements interconnecting the spiking neurons to form the network,

wherein each synaptic element is adapted to receive a synaptic input signal and apply a weight to the synaptic input signal to generate a synaptic output signal, the synaptic elements being configurable to adjust the weight applied by each synaptic element, and

10 wherein each of the spiking neurons is adapted to receive one or more of the synaptic output signals from one or more of the synaptic elements, and generate a spatio-temporal spike train output signal in response to the received one or more synaptic output signals,

wherein the spiking neural network comprises a first sub-network comprising a first sub-set of the spiking neurons connected to receive synaptic output signals from a first sub-set of the synaptic elements,

15 wherein the first sub-network is adapted to generate a sub-network output pattern signal from the first sub-set of spiking neurons, in response to a sub-network input pattern signal applied to the first sub-set of synaptic elements, and

20 wherein the weights of the first sub-set of synaptic elements are configured by training the sub-network on a training set of sub-network input pattern signals, so that the sub-network output pattern signal is different for every unique sub-network input pattern signal of the training set.

2. The spiking neural network of claim 1, wherein the distance between each different sub-network output pattern signal is larger than a predetermined threshold value, the distance
25 being measured by an output pattern metric.

3. The spiking neural network of any one of the preceding claims, wherein the spiking neurons and synaptic elements are configured such that respective distances, as measured by an output pattern metric, between two output pattern signals of the sub-network generated in
30 response to two respective different sub-network input pattern signals are maximized for all sub-network input pattern signals of the training set.

4. The spiking neural network of claim 3, wherein each respective distance is maximized until the output pattern signals meet at least a first minimum sensitivity threshold required for distinguishing between features of the input pattern signals.

5 5. The spiking neural network of any one of the preceding claims, wherein each of the spiking neurons is configurable to adjust the response of the neuron to the received one or more synaptic output signals.

10 6. The spiking neural network of any one of the preceding claims, wherein spikes are generated by the spiking neurons at one or more firing times, and wherein the sub-set of synaptic elements and/or the sub-set of neurons are configured such that the union of two sets of firing times of the sub-set of neurons that fire for two different sub-network input pattern signals is minimized for all sub-network input pattern signals of the training set.

15 7. The spiking neural network of claim 6, wherein each union is minimized until the output pattern signals meet at least a second minimum sensitivity threshold required for distinguishing between features of the input pattern signals.

20 8. The spiking neural network of any one of the preceding claims, wherein the network comprises a second sub-network comprising a second sub-set of the spiking neurons connected to receive synaptic outputs from a second sub-set of the synaptic elements, wherein the second sub-network is adapted to receive a second sub-network input pattern signal applied to the second sub-set of synaptic elements, and generate a corresponding second sub-network output pattern signal from the second sub-set of neurons, and

25 wherein the configurations of the second sub-set of synaptic elements are adjusted so that the second sub-network output pattern signal is different for every unique feature in the second sub-network input pattern signals,

30 wherein the network comprises a third sub-network comprising a third sub-set of the spiking neurons connected to receive synaptic outputs from a third sub-set of the synaptic elements,

wherein the first and second sub-network output pattern signals are input pattern signals of the third sub-network, and

wherein the configurations of the third sub-set of synaptic elements are adjusted such that the third sub-network output pattern signal is different for every unique feature in the input pattern signal from both the first and second sub-network and unique combinations of them, such that the features that are present in the input pattern signals from both the first and second sub-network are encoded by the third sub-network.

9. The spiking neural network of claim 6, wherein the synaptic elements of the third sub-network are configured such that the input pattern signals from the first and second sub-network are weighted according to importance of the specific input patterns.

10. The spiking neural network of any one of the preceding claims, wherein the network comprises multiple sub-networks of synaptic elements and spiking neurons, for which the sub-network output pattern signal is different for every unique feature in the sub-network input pattern signals,

wherein the network can be divided in multiple layers having a particular sequential order in the network, and

wherein the multiple sub-networks are instantiated in the particular sequential order of the multiple layers each respective sub-network belongs to.

11. The spiking neural network of any one of the preceding claims, wherein the synaptic elements are arranged as a configurable switch matrix.

12. The spiking neural network of any one of the preceding claims, wherein configuration information of the neurons, synaptic elements, interconnection topology of the neurons and synaptic elements and/or output configuration of the neural network is saved on a configuration memory, and

wherein the configuration information is loaded from the configuration memory when the neural network is brought to an initialization state.

13. The spiking neural network of any one of the preceding claims, wherein a set of neurons are arranged in one or multiple template networks of neurons, the neurons of a particular template network of neurons forming a sub-network of the neural network,

wherein the sub-network output pattern signal is different for every unique feature in the sub-network input pattern signals, and

wherein each template network in the neural network was instantiated in a pre-trained manner.

5

14. The spiking neural network of any one of the preceding claims, wherein the neural network is configured to take as input one or multiple sampled input signals and convert the input signals into a representative set of neural network input pattern signals.

10 15. The spiking neural network of any one of the preceding claims, wherein an output pattern signal of the neural network is classified in one or multiple output classes.

16. A method for classifying input pattern signals using a spiking neural network comprising a plurality of spiking neurons implemented in hardware or a combination of
15 hardware and software, and a plurality of synaptic elements interconnecting the spiking neurons to form the network,

wherein each of the synaptic elements is adapted to receive a synaptic input signal and apply a weight to the synaptic input signal to generate a synaptic output signal, the synaptic elements being configurable to adjust the weight applied by each synaptic element, and

20 wherein each of the spiking neurons is adapted to receive one or more of the synaptic output signals from one or more of the synaptic elements, and generate a spatio-temporal spike train output signal in response to the received one or more synaptic output signals,

the method comprising:

25 defining a first sub-network of the spiking neural network, the first sub-network comprising a first sub-set of the spiking neurons connected to receive synaptic output signals from a first sub-set of the synaptic elements;

30 configuring the weights of the first sub-set of synaptic elements by training the sub-network on a training set of sub-network input pattern signals, so that the sub-network output pattern signal is different for every unique sub-network input pattern signal of the training set;

applying a sub-network input pattern signal to a first sub-network of the spiking neural network, the first sub-network comprising a first sub-set of the spiking neurons connected to receive synaptic output signals from a first sub-set of the synaptic elements; and

5 receiving a sub-network output pattern signal generated by the first sub-set of spiking neurons in response to the sub-network input pattern signal, wherein the output pattern signal identifies one or more features of the input pattern signal.

17. A method of classification of predetermined features in input signals, the method comprising:

10 building and training a neural network according to claim 14 and claim 15;
submitting at least one sampled input signal to the neural network; and
classifying the features in at least one sampled input signal in one or multiple output classes of the neural network.

15 18. A template library comprising one or multiple template networks of spiking neurons for use as a sub-network of a spiking neural network or information on a configuration thereof,
wherein each template network comprises a set of the spiking neurons connected to receive synaptic outputs from a set of the synaptic elements, and

20 wherein each template network is adapted to receive a template network input pattern signal applied to the set of synaptic elements, and generate a corresponding template network output pattern signal from the set of neurons, and

25 wherein the configurations of the set of synaptic elements are adjusted by training the template network on a training set of template network input pattern signals, so that the template network output pattern signal is different for every unique template network input pattern signal of the training set,

wherein the training set is used to train the template network to perform a particular task,

30 such that each template network can be used as a sub-network of a spiking neural network in a pre-trained manner to perform the particular task or such that the sub-network of the spiking neural network can be instantiated on the basis of the information on the configuration of the template network in a pre-trained manner to perform the particular task.

19. A method of composing a spiking neural network, the method comprising
obtaining one or multiple template networks or information on the configuration of one
or multiple template networks from a template library according to claim 17, and
instantiating the one or multiple template networks as a sub-network of the spiking
5 neural network such that the sub-network of the spiking neural network is able to perform the
particular task the template network was pre-trained to perform.

AMENDED CLAIMS
received by the International Bureau on 14 April 2020 (14.04.2020)

CLAIMS

1. A spiking neural network for classifying input pattern signals, comprising a plurality of spiking neurons implemented in hardware or a combination of hardware and software, and a plurality of synaptic elements interconnecting the spiking neurons to form the
5 network,
wherein each synaptic element is adapted to receive a synaptic input signal and apply a weight to the synaptic input signal to generate a synaptic output signal, the synaptic elements being configurable to adjust the weight applied by each synaptic element, and
wherein each of the spiking neurons is adapted to receive one or more of the synaptic
10 output signals from one or more of the synaptic elements, and generate a spatio-temporal spike train output signal in response to the received one or more synaptic output signals,
wherein the spiking neural network comprises a first sub-network comprising a first sub-set of the spiking neurons connected to receive synaptic output signals from a first sub-set of the synaptic elements,
15 wherein the first sub-network is adapted to generate a sub-network output pattern signal from the first sub-set of spiking neurons, in response to a sub-network input pattern signal applied to the first sub-set of synaptic elements, and
wherein the weights of the first sub-set of synaptic elements are configured by training the sub-network on a training set of sub-network input pattern signals, so that the
20 sub-network output pattern signal is unique for every unique sub-network input pattern signal of the training set, wherein the degree of uniqueness is controllable through operating parameters of the first sub-set of spiking neurons and synaptic elements.
2. The spiking neural network of claim 1, wherein a respective distance between each
25 unique sub-network output pattern signal is larger than a predetermined threshold value, the distance being measured by an output pattern metric.
3. The spiking neural network of any one of the preceding claims, wherein the spiking
30 neurons and synaptic elements are configured such that respective distances, as measured by an output pattern metric, between two output pattern signals of the sub-network generated in response to two respective different sub-network input pattern signals are maximized for all sub-network input pattern signals of the training set.

4. The spiking neural network of claim 3, wherein each respective distance is maximized until the output pattern signals meet at least a first minimum sensitivity threshold required for distinguishing between features of the input pattern signals.
- 5
5. The spiking neural network of any of the preceding claims, wherein the weights of the first sub-set of synaptic elements are configured by training the sub-network on a training set of sub-network input pattern signals using a semi-supervised training methodology, by inducing spikes at a desired output neuron at a specific time and at other
- 10 neurons at a time concomitant with the application of the sub-network input pattern signals, the sub-network is configured to respond with the desired sub-network output pattern signal.
6. The spiking neural network of claim 5, wherein a desired spike is artificially induced
- 15 at a specific time to obtain a unique response to a certain input pattern.
7. The spiking neural network of claim 5 or 6, wherein by inducing a spike artificially at a neuron n of the sub-network at a desired time, the nature of the relationship between neurons in preceding layers of the sub-network and the neuron n is established as causal,
- 20 anti-causal, or non-causal.
8. The spiking neural network of any one of the preceding claims, wherein the weights of the first sub-set of synaptic elements are configured using a causal-chain spike-timing-dependent plasticity (CC-STDP) learning rule, which enables identification of causal
- 25 relationships between desired output neurons and neurons in preceding layers of the sub-network, and causes the weights of intervening synaptic elements along this path of causation to be adjusted.
9. The spiking neural network of claim 8, wherein using the CC-STDP learning rule,
- 30 the sub-network output pattern signal can be steered towards a different population of the sub-network's output neurons that fired in response to a particular sub-network input pattern signal, and their precise firing times, in order to reach a particular sub-network output pattern signal.

10. The spiking neural network of claim 8 or 9, wherein the CC-STDP learning rule adjusts weights of the first sub-set of synaptic elements on the basis of a firing event only if the firing event contributes to firing of neurons in subsequent layers of the sub-network.

5

11. The spiking neural network of any one of claims 8-10, wherein the training using the CC-STDP learning rule comprises inducing and/or inhibiting spike generation at neurons of the first sub-set of spiking neurons at specific times.

10 12. The spiking neural network of claim 11, wherein inducing spike generation in a neuron comprises driving a membrane of the neuron to a voltage that exceeds the firing threshold of the membrane, by means of a bias voltage input, thereby inducing the generation of a spike, or injecting an artificial spike into the neuron's output at the specific time.

15 13. The spiking neural network of claim 11 or 12, wherein inhibiting spike generation in a neuron comprises driving a membrane of the neuron to its refractory voltage, or lowest possible voltage, preventing the generation of a spike, or disabling spike generation at the neuron.

20 14. The spiking neural network of any one of claims 8-13, wherein the Causal Chain STDP learning rule changes the weight $w_{i,j}$ of a synaptic element positioned between neuron i and neuron j according to the following rule:

$$\Delta w_{i,j} = \begin{cases} \delta w_{i,j} \text{ for } \delta w_{i,j} > 0 \text{ if } w_{j,k} \neq 0 \text{ and } t_{\text{spike } k} > t_{\text{spike } j} \text{ and } t_{\text{spike } k} > t_{\text{spike } i} \\ \delta w_{i,j} \text{ for } \delta w_{i,j} < 0 \\ 0 \text{ if } w_{j,k} = 0 \text{ or } t_{\text{spike } k} \leq t_{\text{spike } i} \text{ or } t_{\text{spike } k} \leq t_{\text{spike } j} \text{ or } t_{\text{spike } k} = 0 \end{cases}$$

with $\Delta w_{i,j}$ the effective weight change for the synaptic element between neuron i and j ;

25 $w_{j,k}$ the weight of the synaptic element between neuron j and a neuron k of the subsequent layer; $t_{\text{spike } i}$, $t_{\text{spike } j}$ and $t_{\text{spike } k}$ the fundamental parameter of a second learning rule for neuron i , j and k respectively; and $\delta w_{i,j}$ the weight change in the synaptic element between neuron i and j , due to their relative time of spiking, and is computed using the second learning rule.

30

15. The spiking neural network of claim 14, wherein the second learning rule is the STDP learning rule, and $t_{spike\ i}$, $t_{spike\ j}$ and $t_{spike\ k}$ refer to the precise times at which neuron i , j and k respectively spike.
- 5 16. The spiking neural network of any one of the preceding claims, wherein each of the spiking neurons is configurable to adjust the response of the neuron to the received one or more synaptic output signals.
- 10 17. The spiking neural network of any one of the preceding claims, wherein spikes are generated by the spiking neurons at one or more firing times, and wherein the sub-set of synaptic elements and/or the sub-set of neurons are configured such that the union of two sets of firing times of the sub-set of neurons that fire for two unique sub-network input pattern signals is minimized for all sub-network input pattern signals of the training set.
- 15 18. The spiking neural network of claim 17, wherein each union is minimized until the output pattern signals meet at least a second minimum sensitivity threshold required for distinguishing between features of the input pattern signals.
- 20 19. The spiking neural network of any one of the preceding claims, wherein the network comprises a second sub-network comprising a second sub-set of the spiking neurons connected to receive synaptic outputs from a second sub-set of the synaptic elements,
wherein the second sub-network is adapted to receive a second sub-network input pattern signal applied to the second sub-set of synaptic elements, and generate a corresponding second sub-network output pattern signal from the second sub-set of
25 neurons, and
wherein the configurations of the second sub-set of synaptic elements are adjusted so that the second sub-network output pattern signal is unique for every unique feature in the second sub-network input pattern signals,
wherein the network comprises a third sub-network comprising a third sub-set of the
30 spiking neurons connected to receive synaptic outputs from a third sub-set of the synaptic elements,
wherein the first and second sub-network output pattern signals are input pattern signals of the third sub-network, and

wherein the configurations of the third sub-set of synaptic elements are adjusted such that the third sub-network output pattern signal is unique for every unique feature in the input pattern signal from both the first and second sub-network and unique combinations of them, such that the features that are present in the input pattern signals from both the first and second sub-network are encoded by the third sub-network.

20. The spiking neural network of claim 19, wherein the synaptic elements of the third sub-network are configured such that the input pattern signals from the first and second sub-network are weighted according to importance of the specific features in the input pattern signals.

21. The spiking neural network of any one of the preceding claims, wherein the network comprises multiple sub-networks of synaptic elements and spiking neurons, for which the sub-network output pattern signal is unique for every unique feature in the sub-network input pattern signals,

wherein the network can be divided in multiple layers having a particular sequential order in the network, and

wherein the multiple sub-networks are instantiated in the particular sequential order of the multiple layers each respective sub-network belongs to.

20

22. The spiking neural network of any one of the preceding claims, wherein the synaptic elements are arranged as a configurable switch matrix.

23. The spiking neural network of any one of the preceding claims, wherein configuration information of the neurons, synaptic elements, interconnection topology of the neurons and synaptic elements and/or output configuration of the neural network is saved on a configuration memory, and

wherein the configuration information is loaded from the configuration memory when the neural network is brought to an initialization state.

30

24. The spiking neural network of any one of the preceding claims, wherein a set of neurons are arranged in one or multiple template networks of neurons, the neurons of a particular template network of neurons forming a sub-network of the neural network,

wherein the sub-network output pattern signal is unique for every unique feature in the sub-network input pattern signals, and

wherein each template network in the neural network is instantiated as a pre-trained sub-network that was configured according to claim 17.

5

25. The spiking neural network of any one of the preceding claims, wherein the neural network is configured to take as input one or multiple sampled analog or digital input signals and convert the input signals into a representative set of spiking neural network input pattern signals.

10

26. The spiking neural network of any one of the preceding claims, wherein an output pattern signal of the neural network is classified in one or multiple output classes.

27. A method for classifying input pattern signals using a spiking neural network comprising a plurality of spiking neurons implemented in hardware or a combination of hardware and software, and a plurality of synaptic elements interconnecting the spiking neurons to form the network,

15

wherein each of the synaptic elements is adapted to receive a synaptic input signal and apply a weight to the synaptic input signal to generate a synaptic output signal, the synaptic elements being configurable to adjust the weight applied by each synaptic element, and

20

wherein each of the spiking neurons is adapted to receive one or more of the synaptic output signals from one or more of the synaptic elements, and generate a spatio-temporal spike train output signal in response to the received one or more synaptic output signals,

25

the method comprising:

defining a first sub-network of the spiking neural network, the first sub-network comprising a first sub-set of the spiking neurons connected to receive synaptic output signals from a first sub-set of the synaptic elements;

30

configuring the weights of the first sub-set of synaptic elements by training the sub-network on a training set of sub-network input pattern signals, so that the sub-network output pattern signal is unique for every unique sub-network input pattern signal of the training set, wherein the degree of uniqueness is controllable through operating parameters of the first sub-set of spiking neurons and synaptic elements;

applying a sub-network input pattern signal to a first sub-network of the spiking neural network, the first sub-network comprising a first sub-set of the spiking neurons connected to receive synaptic output signals from a first sub-set of the synaptic elements; and

- 5 receiving a sub-network output pattern signal generated by the first sub-set of spiking neurons in response to the sub-network input pattern signal, wherein the output pattern signal identifies one or more features of the input pattern signal.

28. A method of classification of predetermined features in input signals, the method
10 comprising:

building and training a spiking neural network according to claim 27;
submitting at least one sampled input signal to the neural network; and
classifying the features in at least one sampled input signal in one or multiple output
classes of the neural network.

15

29. A template library comprising information on a configuration of one or multiple template networks of spiking neurons for use as a sub-network of a spiking neural network, wherein each template network comprises a set of the spiking neurons implemented in hardware or a combination of hardware and software connected to receive synaptic
20 outputs from a set of the synaptic elements, and

wherein each template network is adapted to receive a template network input pattern signal applied to the set of synaptic elements, and generate a corresponding template network output pattern signal from the set of neurons, and

wherein the configurations of the set of synaptic elements are adjusted by training
25 the template network on a training set of template network input pattern signals, so that the template network output pattern signal is unique for every unique template network input pattern signal of the training set, wherein the degree of uniqueness is controllable through operating parameters of the first sub-set of spiking neurons and synaptic elements,

wherein the training set is used to train the template network to perform a particular
30 task,

such that the sub-network of the spiking neural network can be instantiated on the basis of the information on the configuration of the template network in a pre-trained manner to perform the particular task.

30. A method of composing a spiking neural network, the method comprising
obtaining one or multiple template networks or information on the configuration of
one or multiple template networks from a template library according to claim 29, and
5 instantiating the one or multiple template networks as a sub-network of the spiking
neural network such that the sub-network of the spiking neural network is able to perform
the particular task the template network was pre-trained to perform.

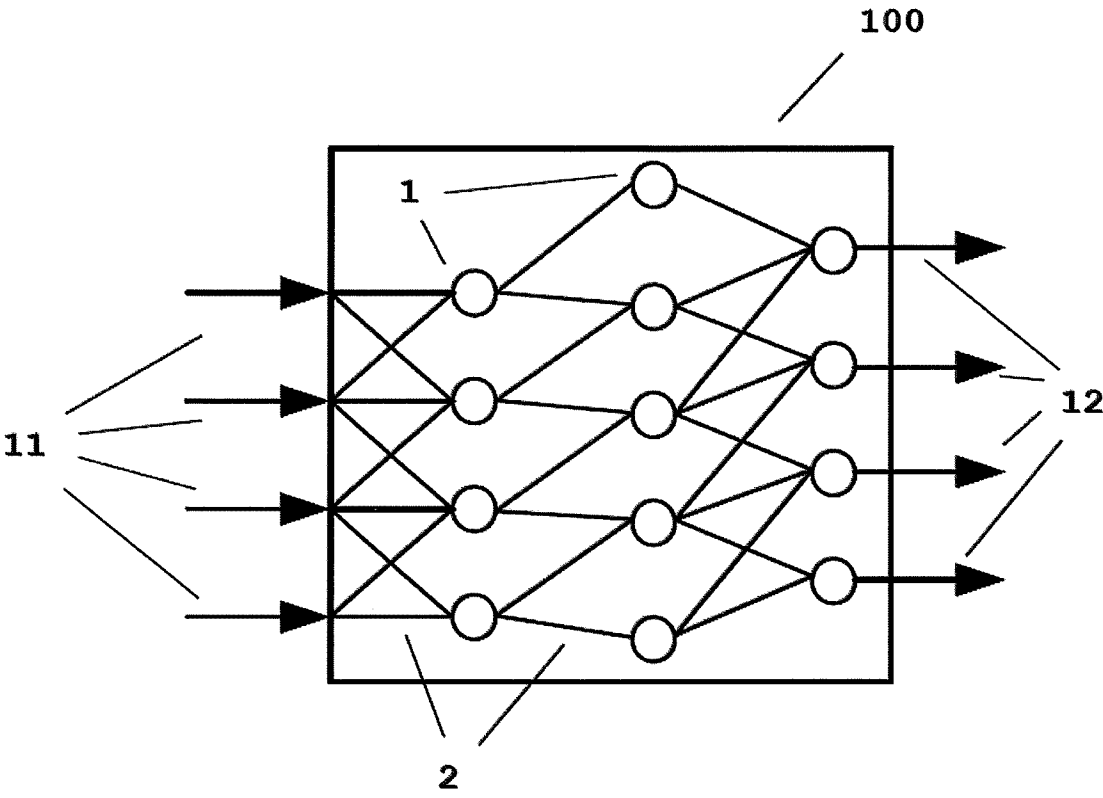


FIG. 1

2/13

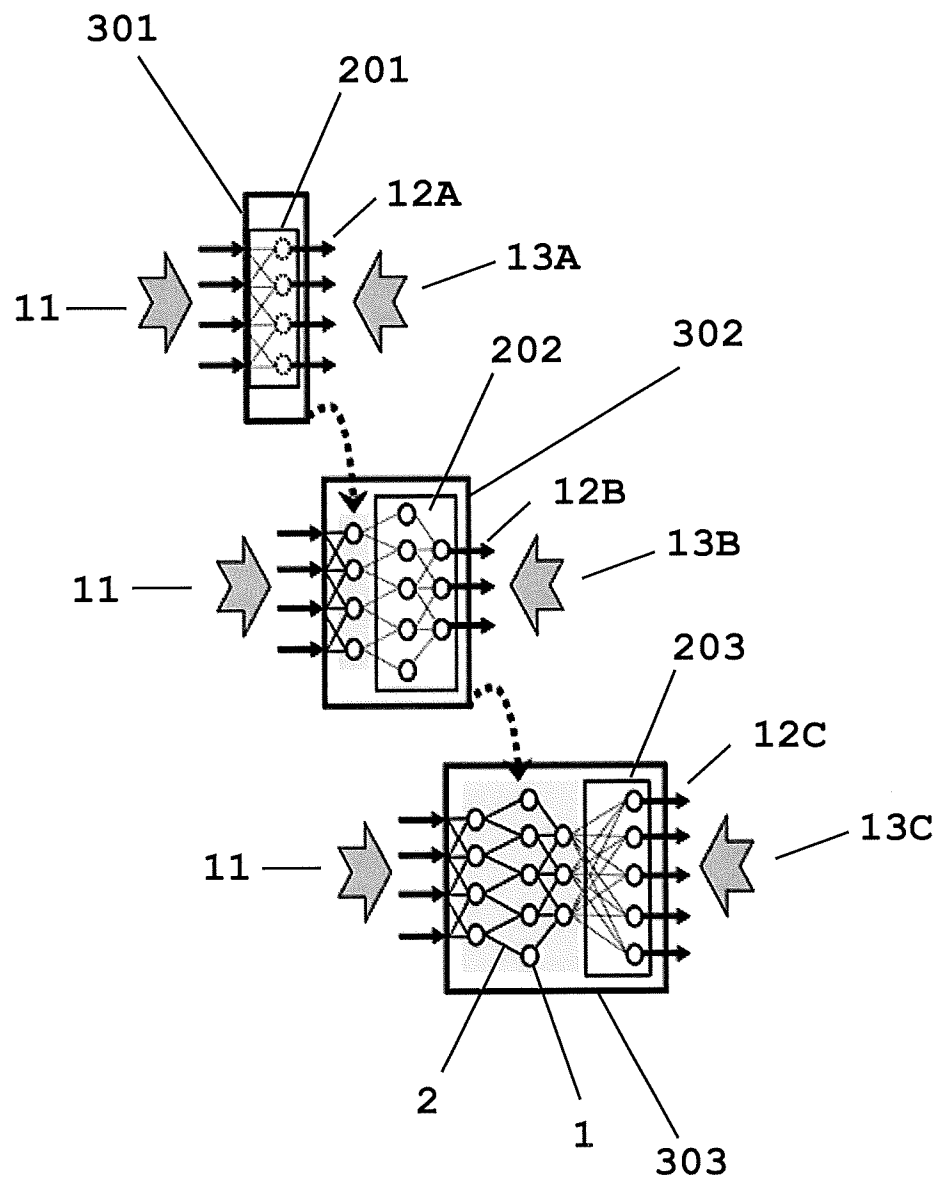


FIG. 2

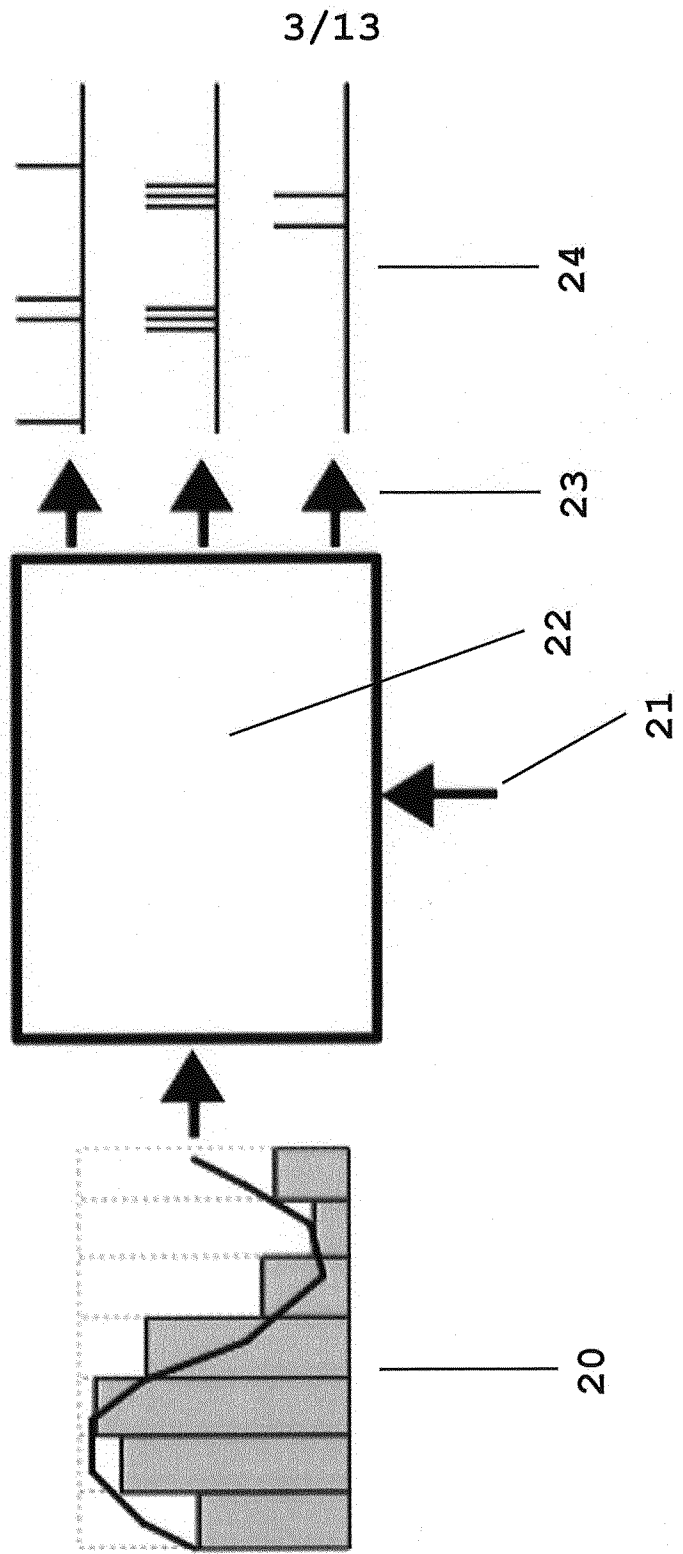


FIG. 3A

4/13

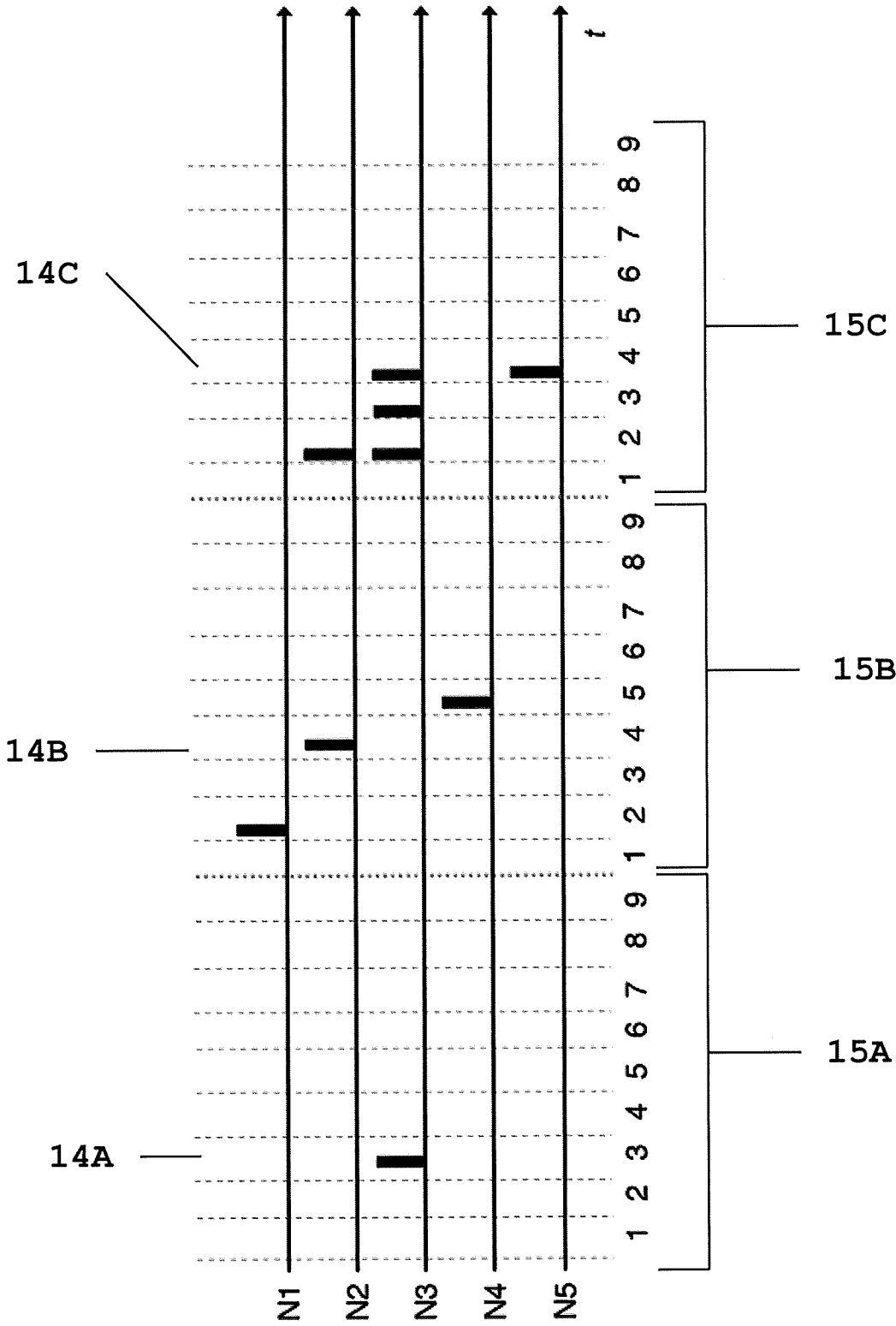


FIG. 3B

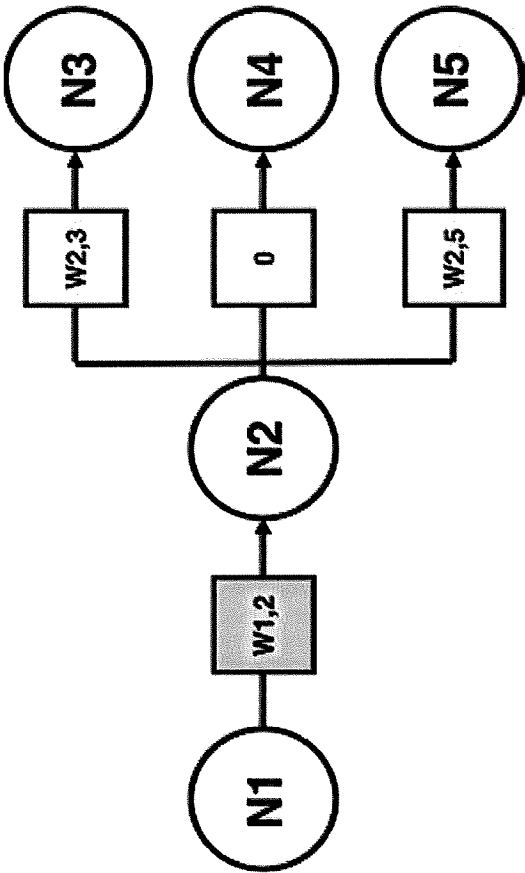


FIG. 4

6/13

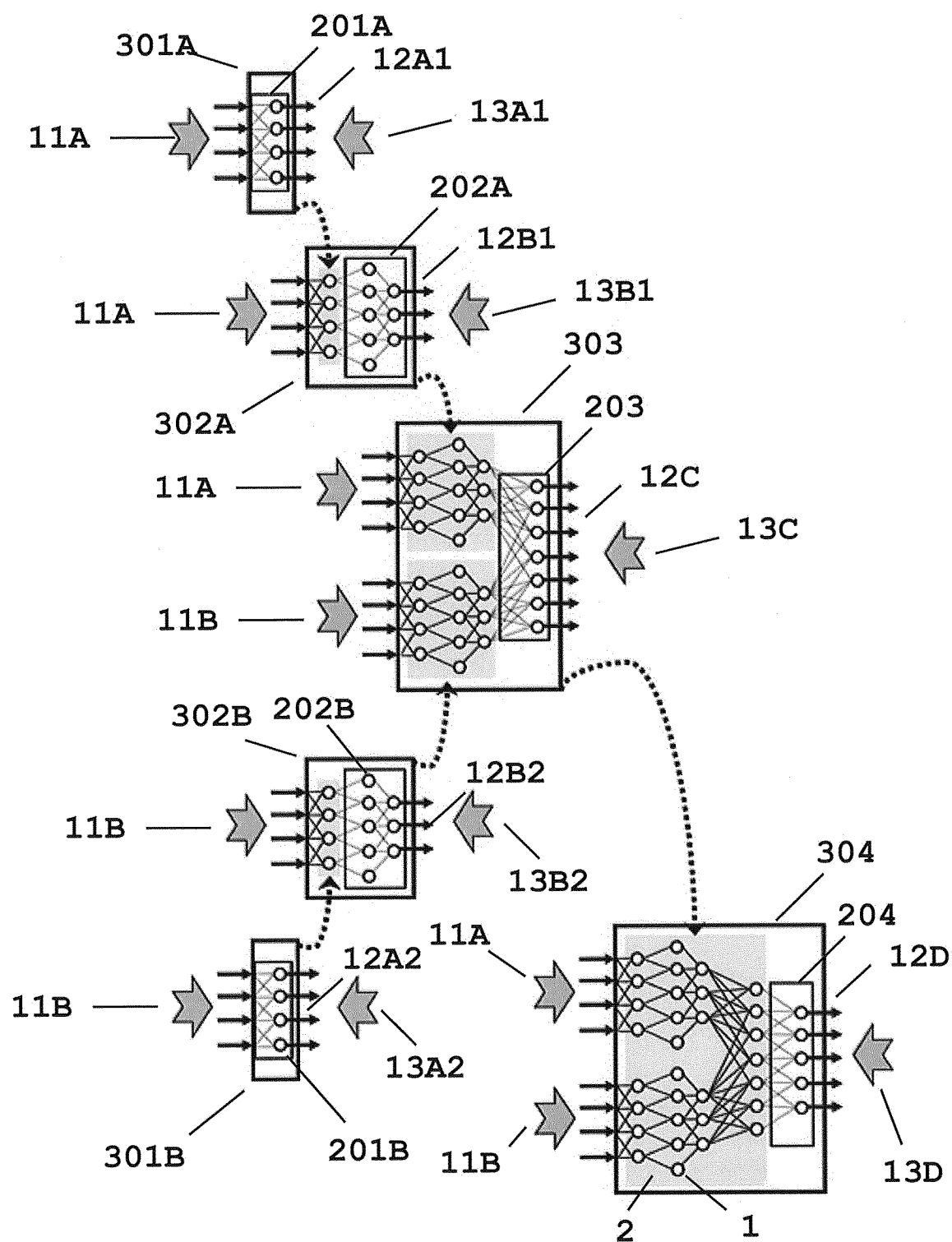


FIG. 5A

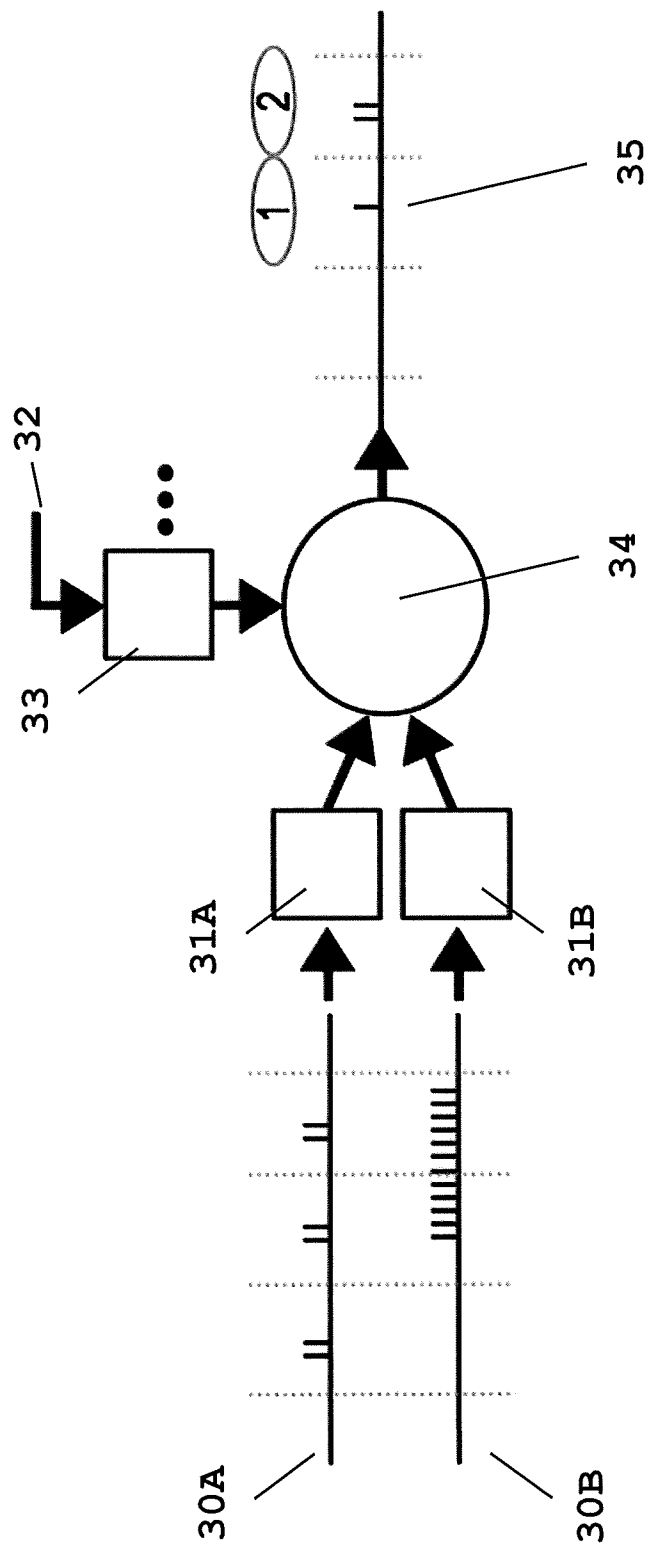
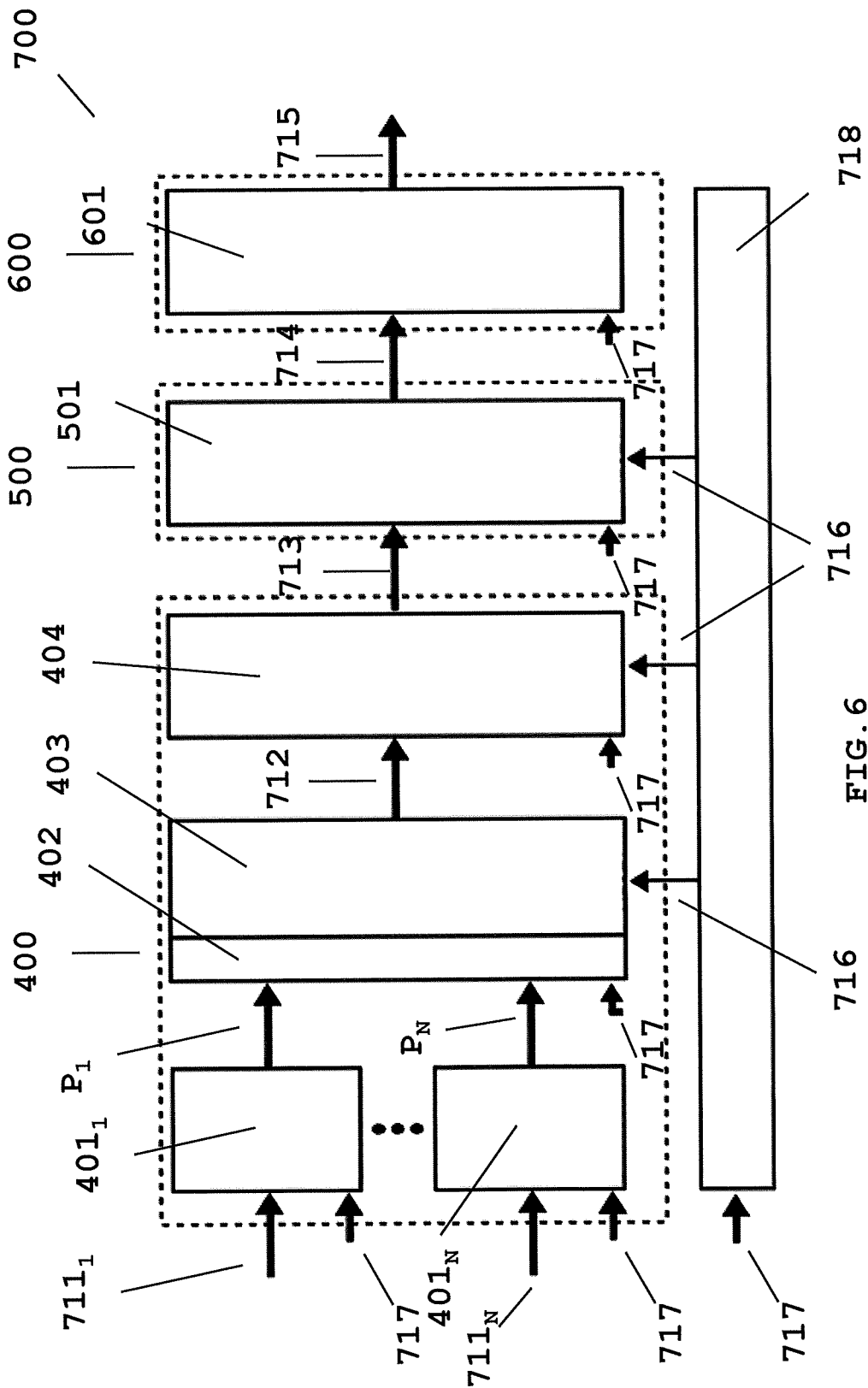


FIG. 5B



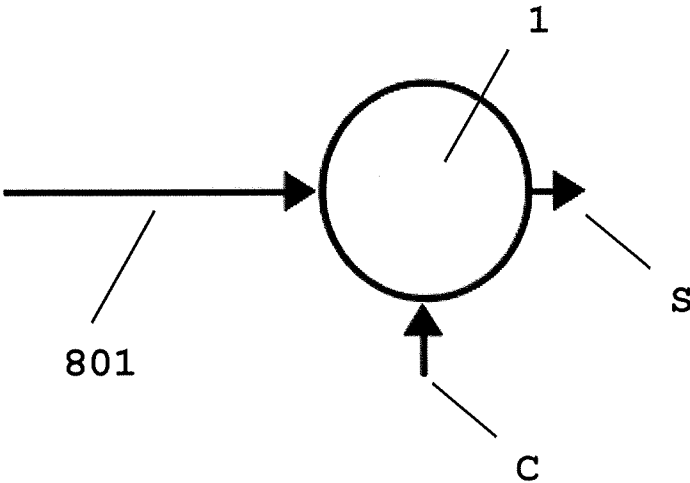


FIG. 7A

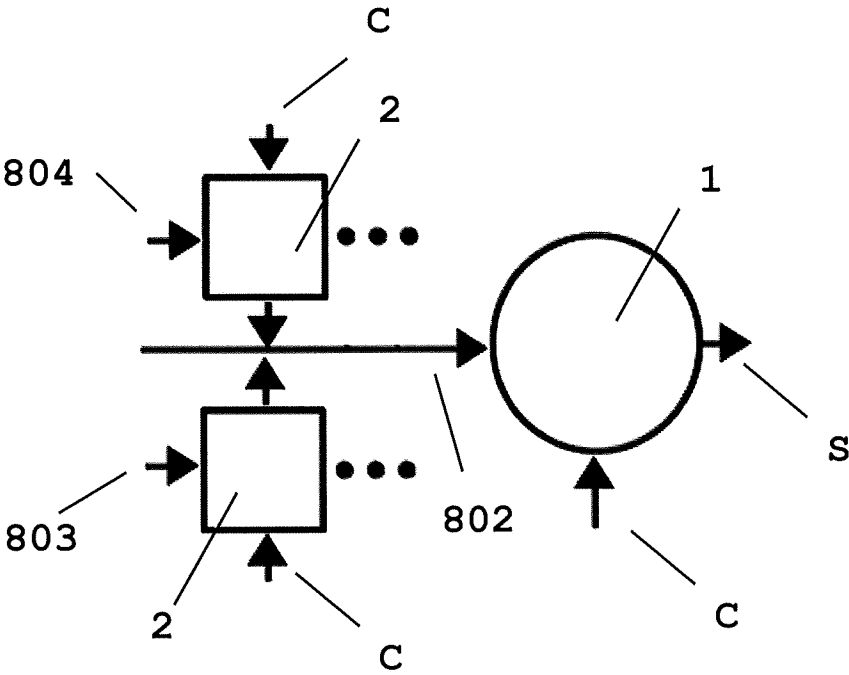


FIG. 7B

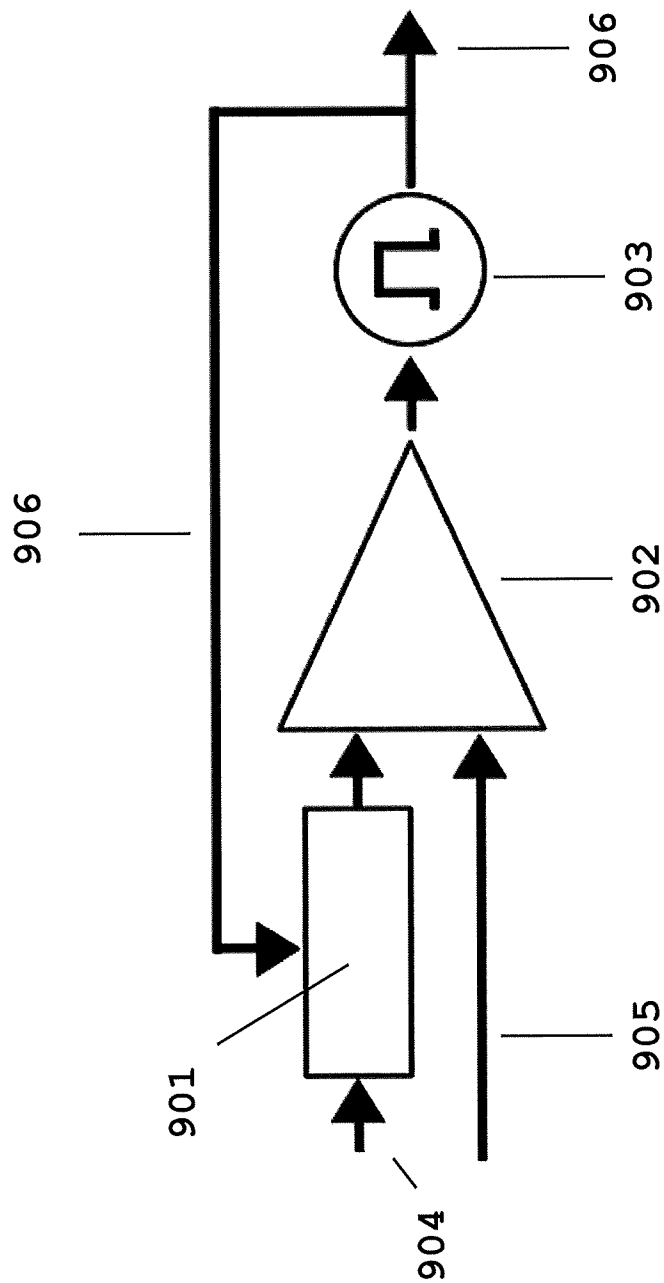


FIG. 8

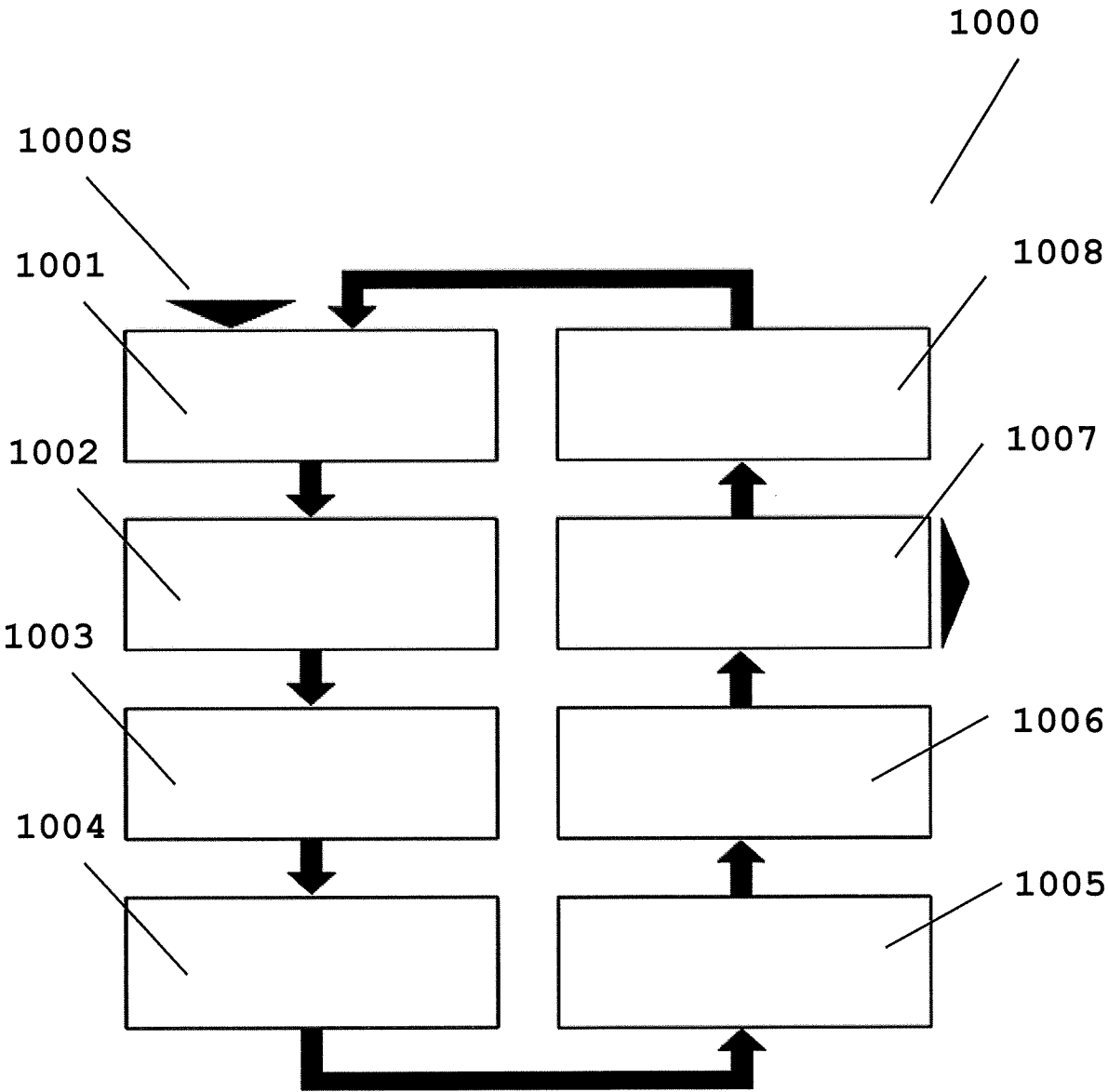


FIG. 9

12/13

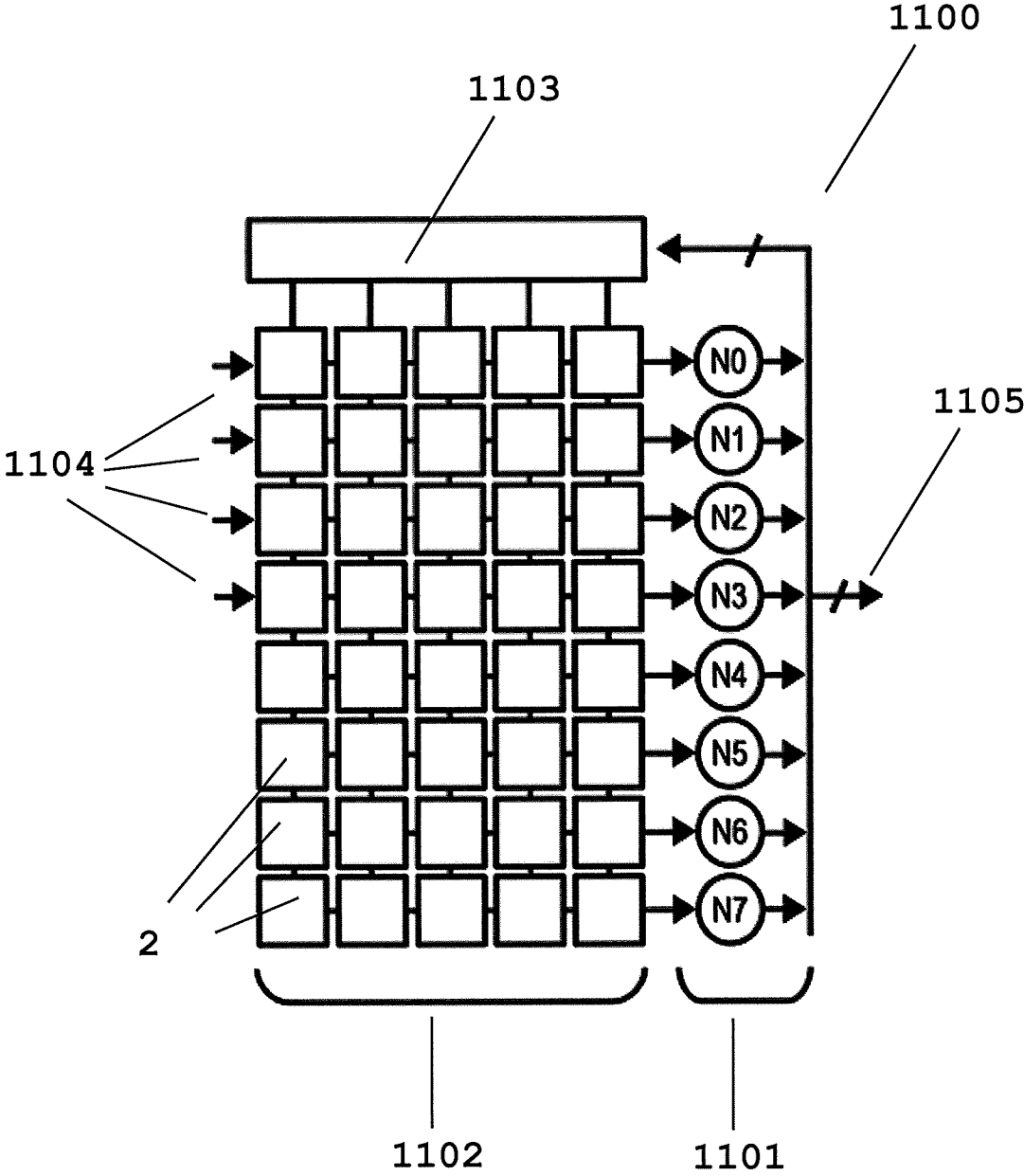


FIG. 10

13/13

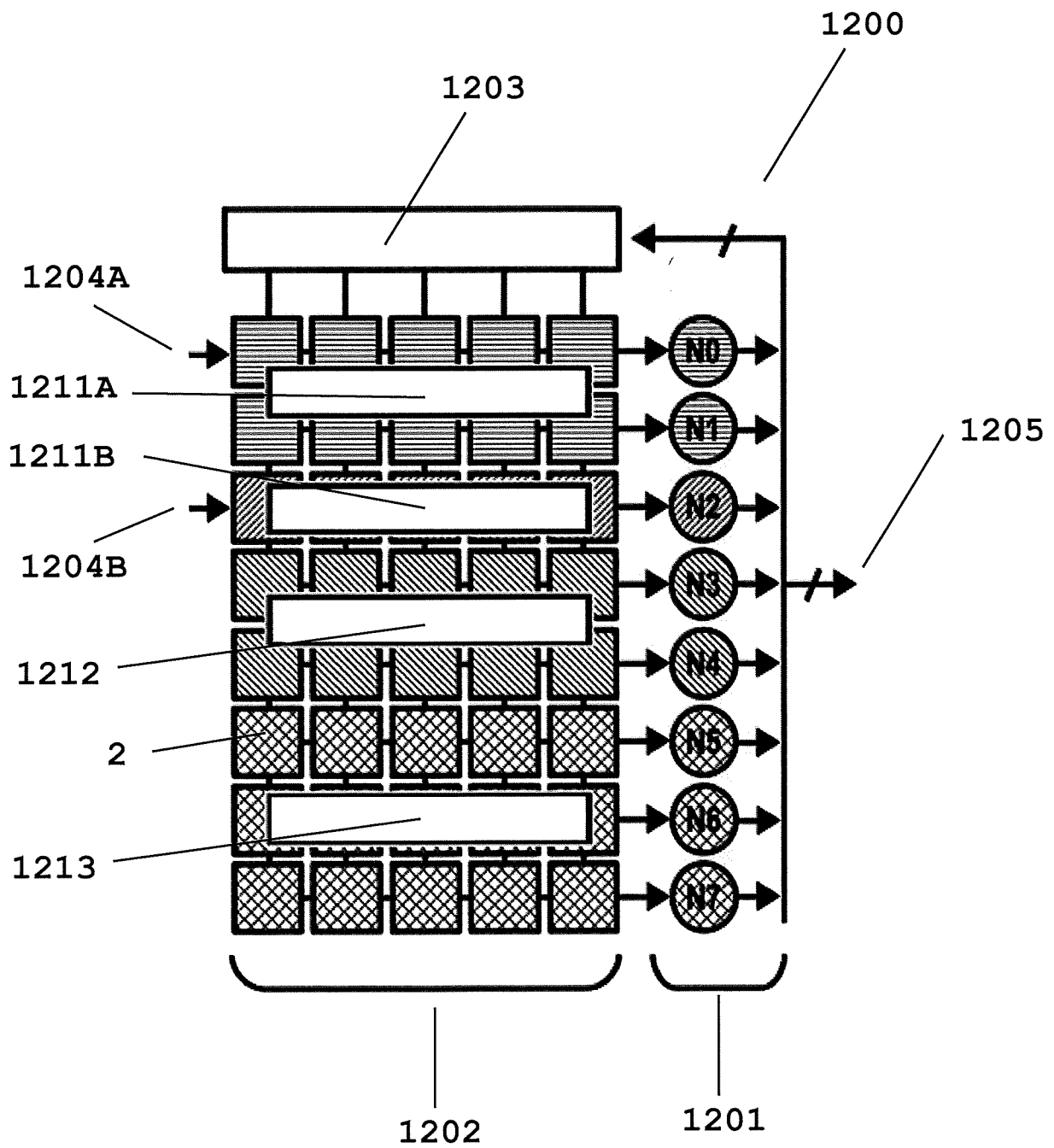


FIG. 11

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2019/081373

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06N3/04 G06N3/08 G06N3/063
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	US 2017/236027 A1 (VAN DER MADE PETER AJ [US] ET AL) 17 August 2017 (2017-08-17) abstract paragraph [0002] - paragraph [0012] paragraph [0024] - paragraph [0056]; figures 1-8	1-19
A	US 2017/316310 A1 (CHO YOUNGKWAN [US] ET AL) 2 November 2017 (2017-11-02) abstract paragraph [0002] - paragraph [0017] paragraph [0077] - paragraph [0087]; figures 3, 4 ----- -/--	2-4



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 February 2020

Date of mailing of the international search report

14/02/2020

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Hasnas, Sergiu

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2019/081373

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WANG JINLING ET AL: "SpikeTemp: An Enhanced Rank-Order-Based Learning Approach for Spiking Neural Networks With Adaptive Structure", IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE, PISCATAWAY, NJ, USA, vol. 28, no. 1, 1 January 2017 (2017-01-01), pages 30-43, XP011637569, ISSN: 2162-237X, DOI: 10.1109/TNNLS.2015.2501322 [retrieved on 2016-12-26] abstract page 32 - page 34; figures 1-3 -----	9
A	US 2012/259804 A1 (BREZZO BERNARD V [US] ET AL) 11 October 2012 (2012-10-11) abstract paragraph [0005] paragraph [0030] - paragraph [0046]; figures 1-3 -----	11,12

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2019/081373

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2017236027	A1	17-08-2017	NONE

US 2017316310	A1	02-11-2017	CN 108027896 A 11-05-2018
		EP 3271872 A1 24-01-2018	
		US 2017316310 A1 02-11-2017	
		WO 2016149689 A1 22-09-2016	

US 2012259804	A1	11-10-2012	US 2012259804 A1 11-10-2012
		US 2012317062 A1 13-12-2012	
		US 2016247063 A1 25-08-2016	
		US 2016292569 A1 06-10-2016	
		US 2016358067 A1 08-12-2016	
