



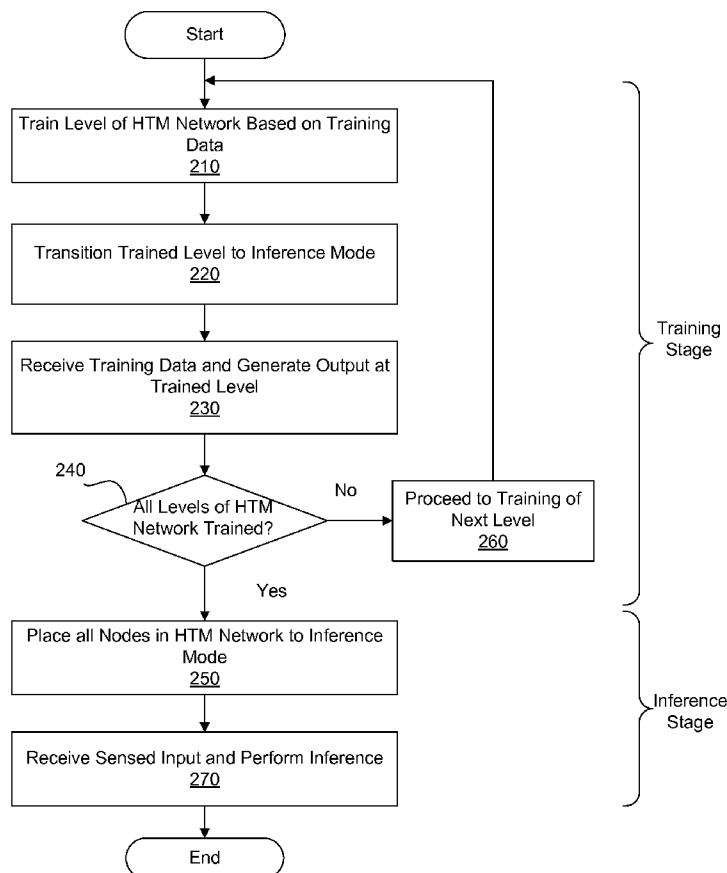
US 20090313193A1

(19) **United States**(12) **Patent Application Publication**
Hawkins et al.(10) **Pub. No.: US 2009/0313193 A1**(43) **Pub. Date: Dec. 17, 2009**(54) **HIERARCHICAL TEMPORAL MEMORY
SYSTEM WITH HIGHER-ORDER
TEMPORAL POOLING CAPABILITY****Publication Classification**(51) **Int. Cl.**
G06F 15/18 (2006.01)
G06N 5/04 (2006.01)
G06N 3/12 (2006.01)(75) **Inventors:** **Jeffrey C. Hawkins**, Atherton, CA
(US); **Dileep George**, Menlo Park,
CA (US); **Charles Curry**, Fremont,
CA (US); **Frank E. Astier**,
Mountain View, CA (US); **Anosh
Raj**, Palo Alto, CA (US); **Robert
G. Jaros**, San Francisco, CA (US)(52) **U.S. Cl. 706/13; 706/52; 706/12**(57) **ABSTRACT**

Correspondence Address:
FENWICK & WEST LLP
**SILICON VALLEY CENTER, 801 CALIFORNIA
STREET**
MOUNTAIN VIEW, CA 94041 (US)

(73) **Assignee:** **NUMENTA, INC.**, Redwood City,
CA (US)(21) **Appl. No.:** **12/483,642**(22) **Filed:** **Jun. 12, 2009****Related U.S. Application Data**(60) **Provisional application No. 61/060,891, filed on Jun.
12, 2008.**

A temporal pooler for a Hierarchical Temporal Memory network is provided. The temporal pooler is capable of storing information about sequences of co-occurrences in a higher-order Markov chain by splitting a co-occurrence into a plurality of sub-occurrences. Each split sub-occurrence may be part of a distinct sequence of co-occurrences. The temporal pooler receives the probability of spatial co-occurrences in training patterns and tallies counts or frequency of transitions from one sub-occurrence to another sub-occurrence in a connectivity matrix. The connectivity matrix is then processed to generate temporal statistics data. The temporal statistics data is provided to an inference engine to perform inference or prediction on input patterns. By storing information related to a higher-order Markov model, the temporal statistics data more accurately reflects long temporal sequences of co-occurrences in the training patterns.



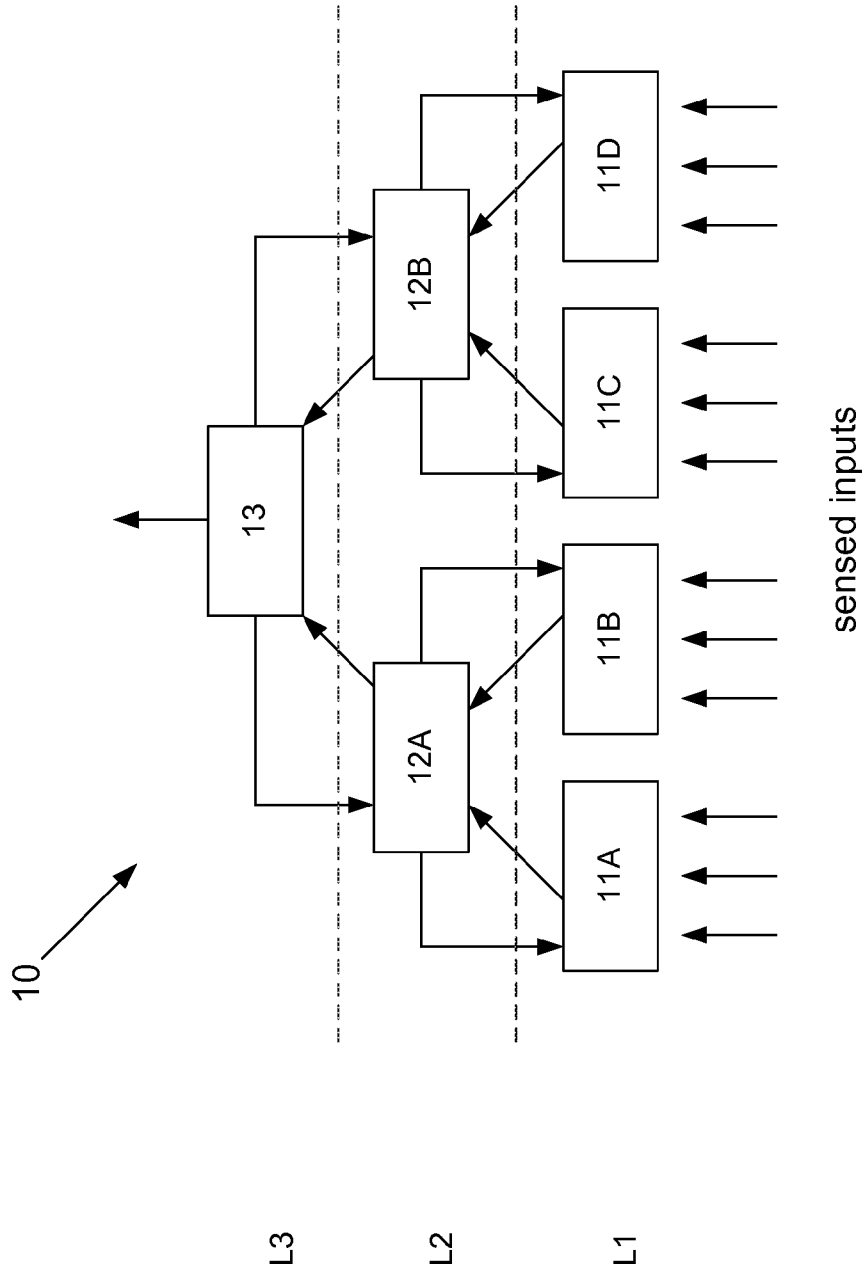


FIG. 1
(PRIOR ART)

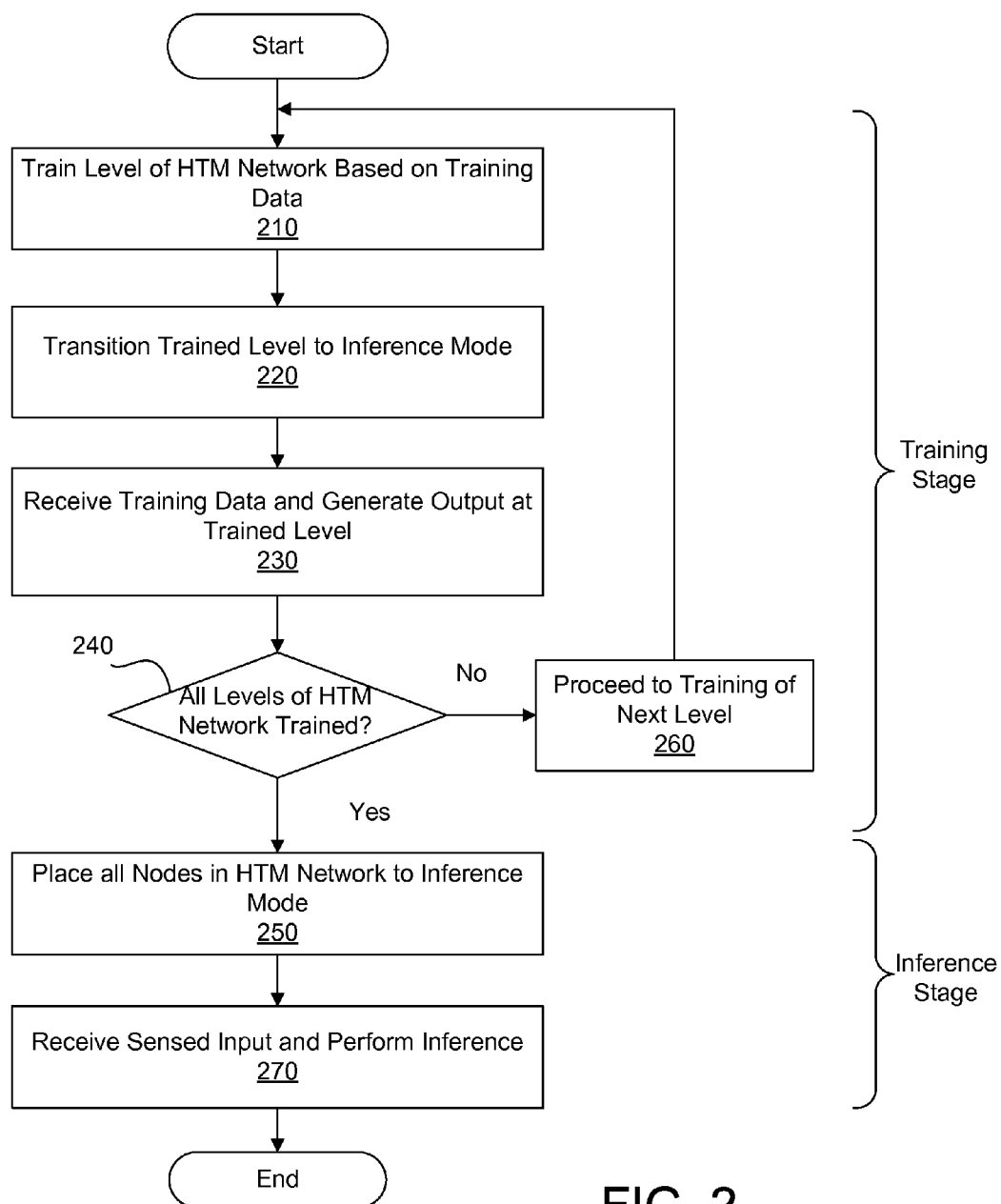


FIG. 2

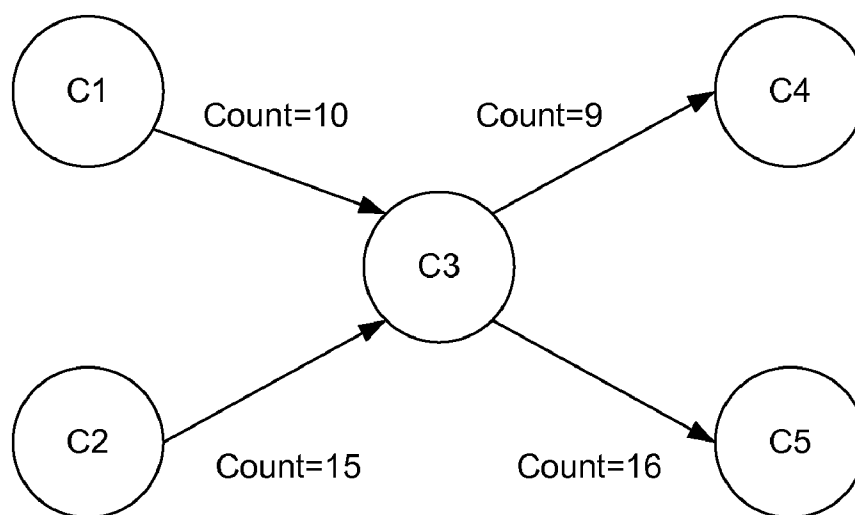


FIG. 3A

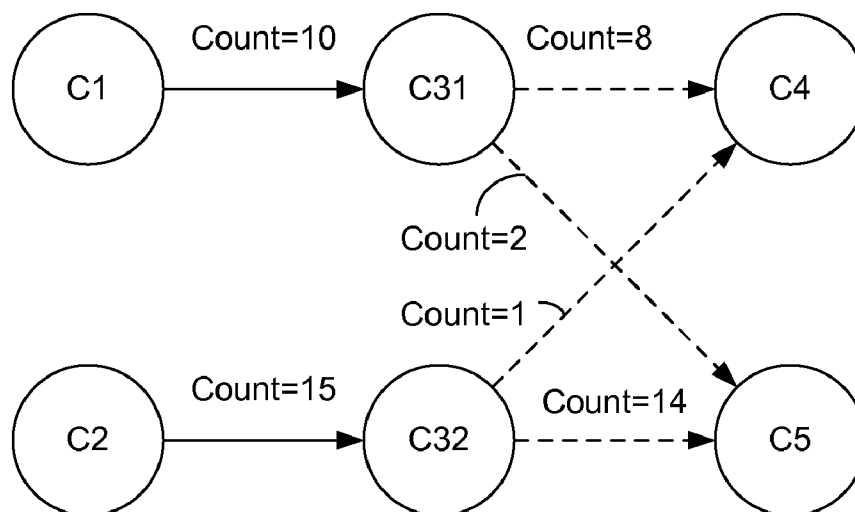


FIG. 3B

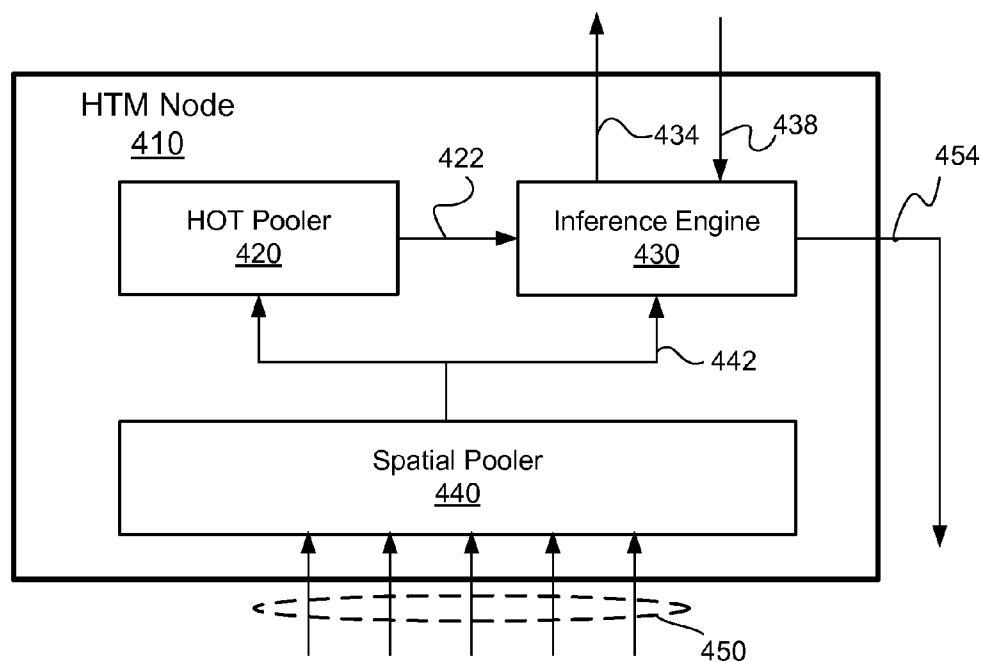


FIG. 4A

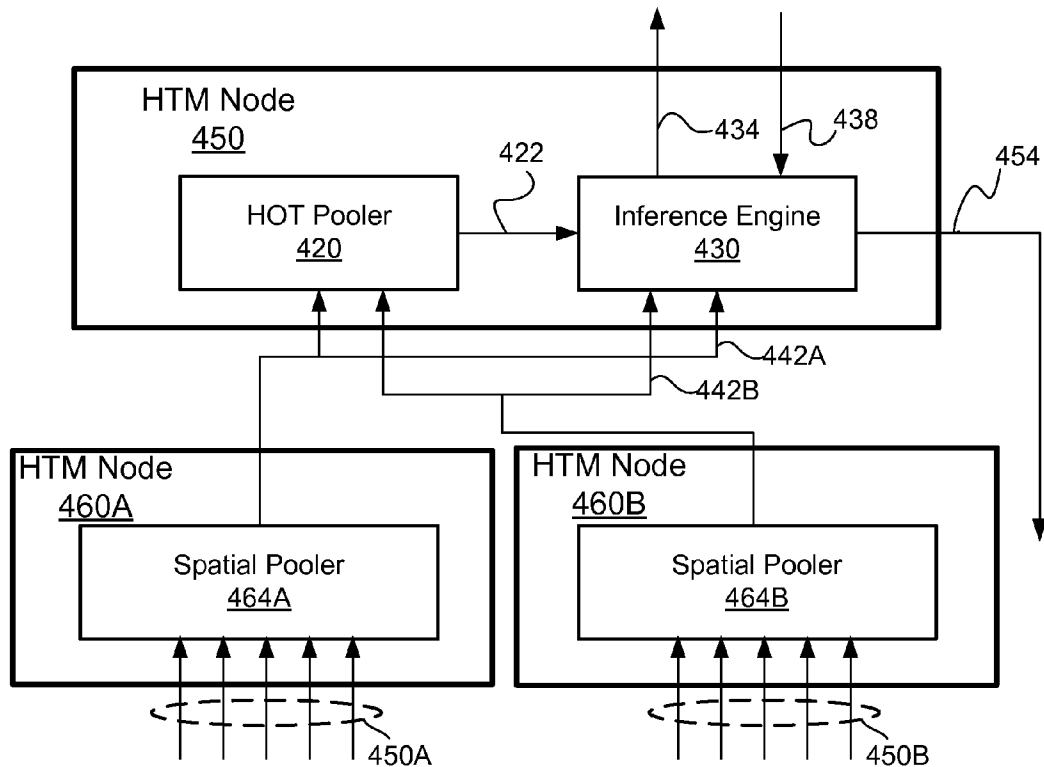


FIG. 4B

		Inbound				
Outbound		C1	C2	C3	C4	C5
	C1	0	0	10	0	0
	C2	0	0	15	0	0
	C3	0	0	0	9	16
	C4	0	0	0	0	0
	C5	0	0	0	0	0

FIG. 5A

		Inbound					520
Outbound		C1	C2	C31	C32	C4	C5
	C1	0	0	10	0	0	0
	C2	0	0	0	15	0	0
	C31	0	0	0	0	8	2
	C32	0	0	0	0	1	14
	C4	0	0	0	0	0	0
	C5	0	0	0	0	0	0

FIG. 5B

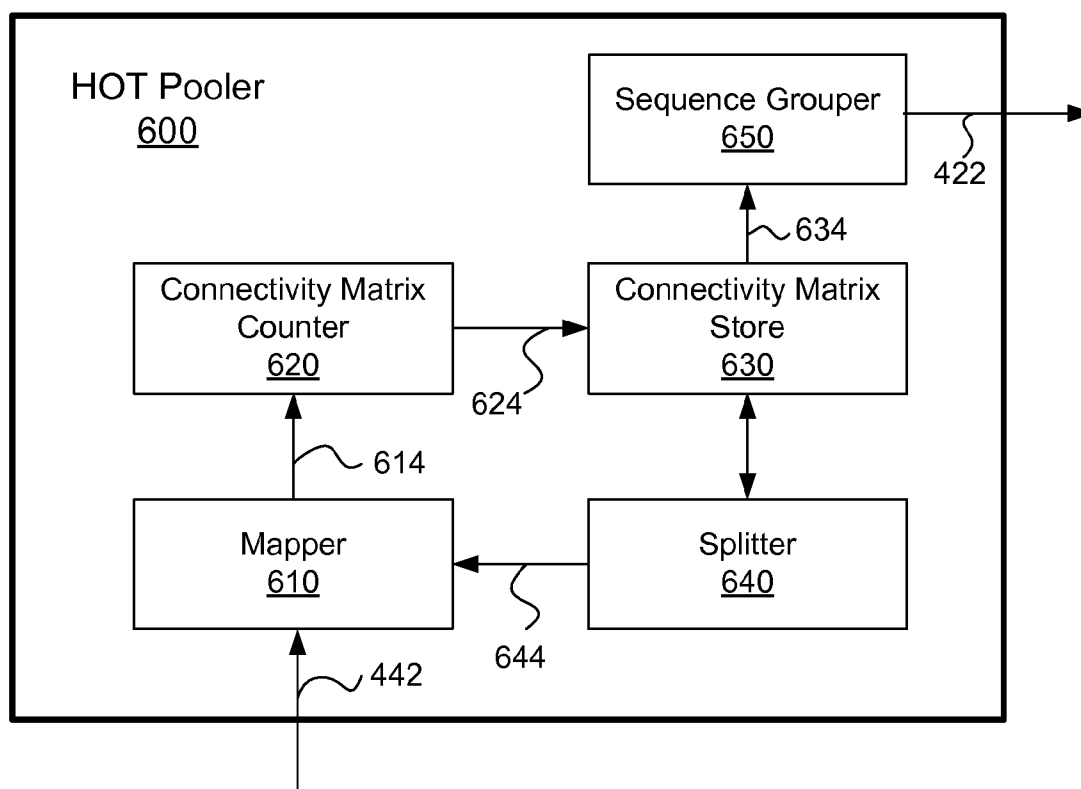


FIG. 6

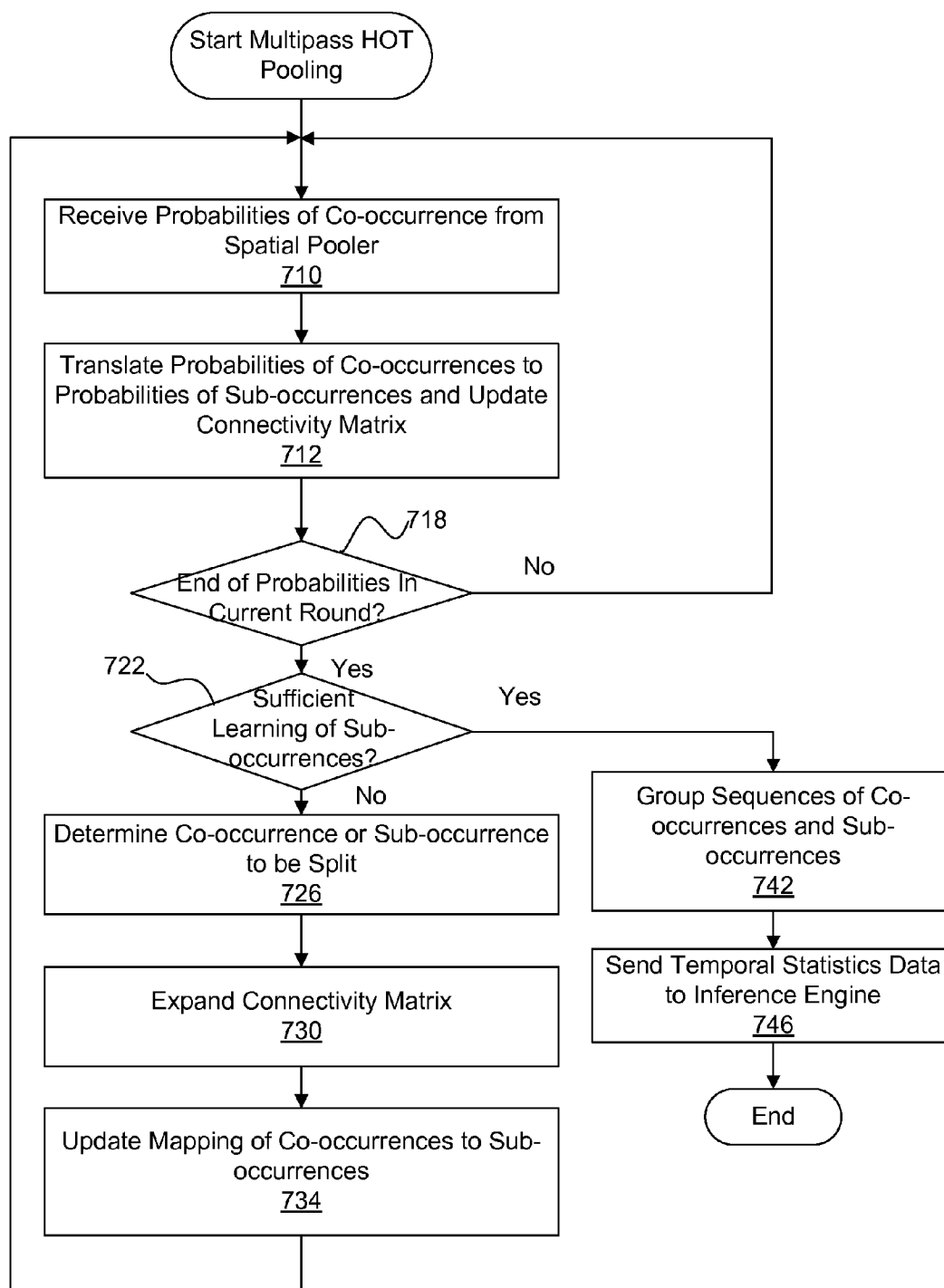


FIG. 7

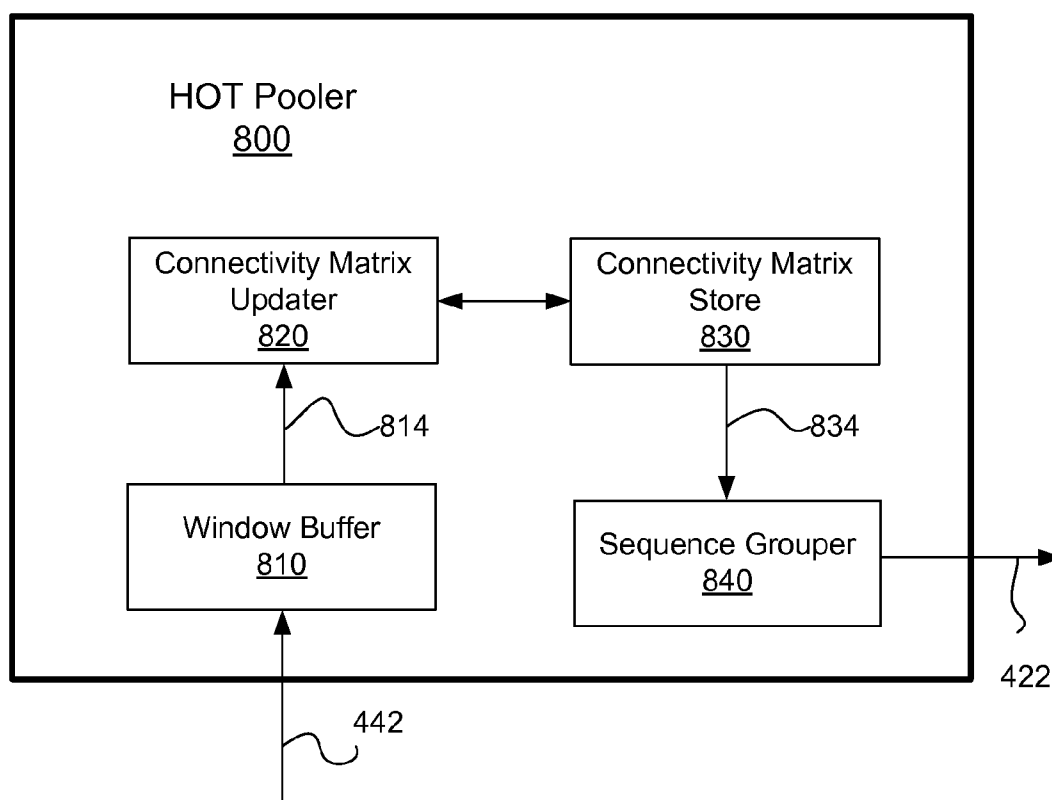


FIG. 8

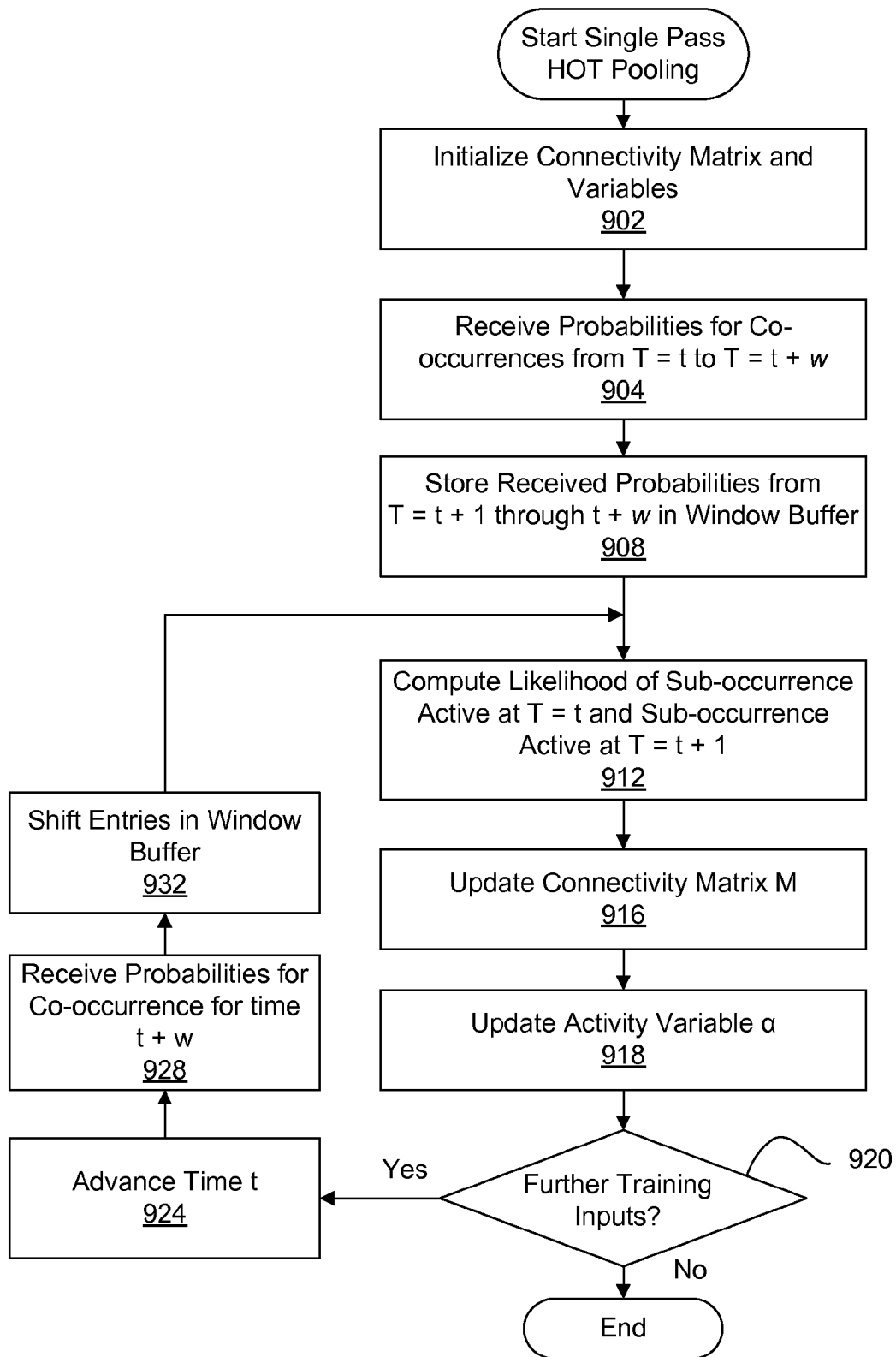


FIG. 9

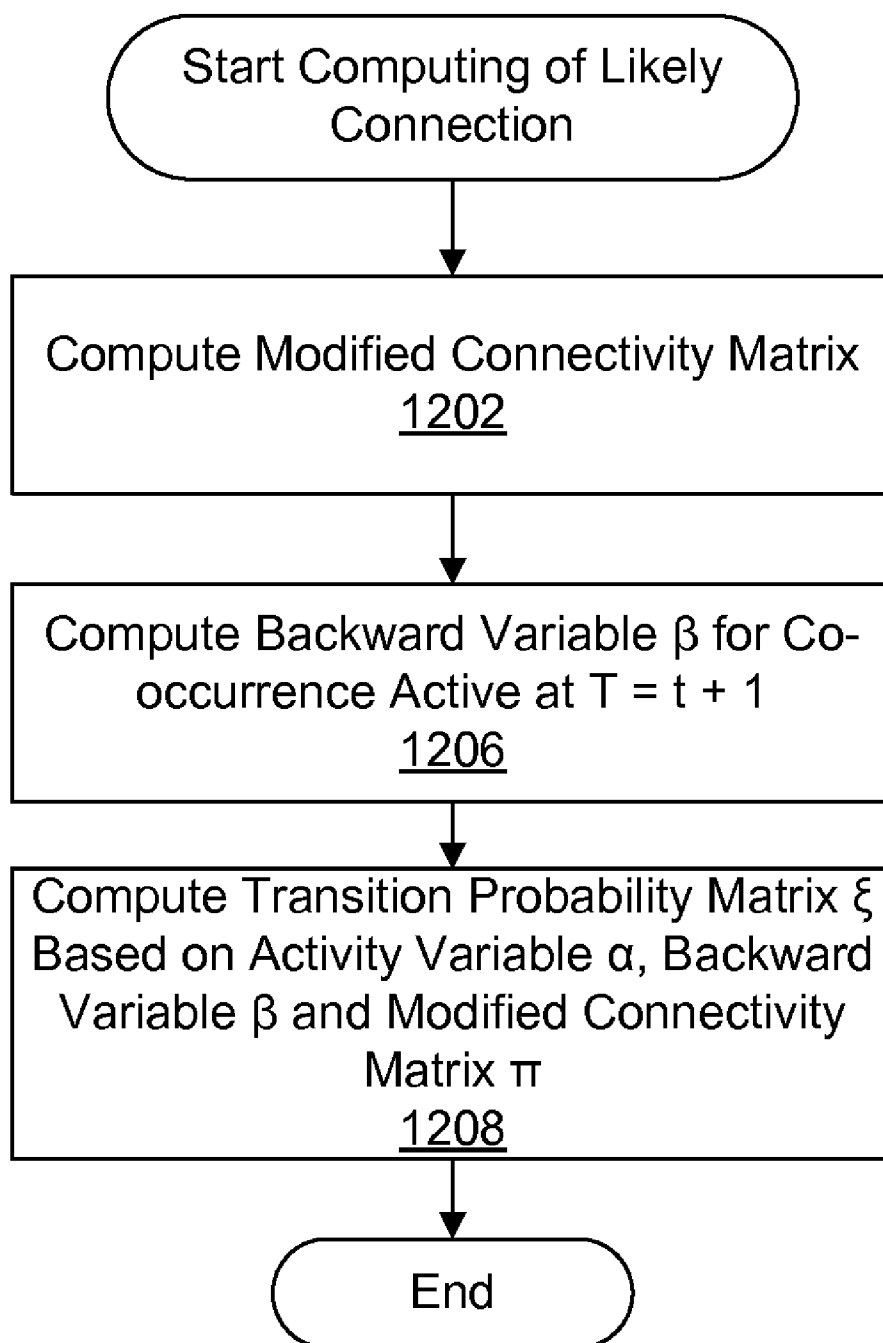
Co-occurrences	C1	C2	C3	C4	C5
Sub-occurrences	C11	C21	C31	C41	C51
	C12	C22	C32	C42	C52
	C13	C23	C33	C43	C53
	C14	C24	C34	C44	C54

FIG. 10

		C1				Inbound			
		C11	C12	C13	C14	C21	C22	...	C54
C1	C11	0	0	0	0	0	0	...	0
	C12	0	0	0	0	0	0	...	0
	C13	0	0	0	0	0	0	...	0
	C14	0	0	0	0	0	0	...	0
Outbound	C21	0	0	0	0	0	0	...	0
	C22	0	0	0	0	0	0	...	0

	C54	0	0	0	0	0	0	...	0

FIG. 11

**FIG. 12**

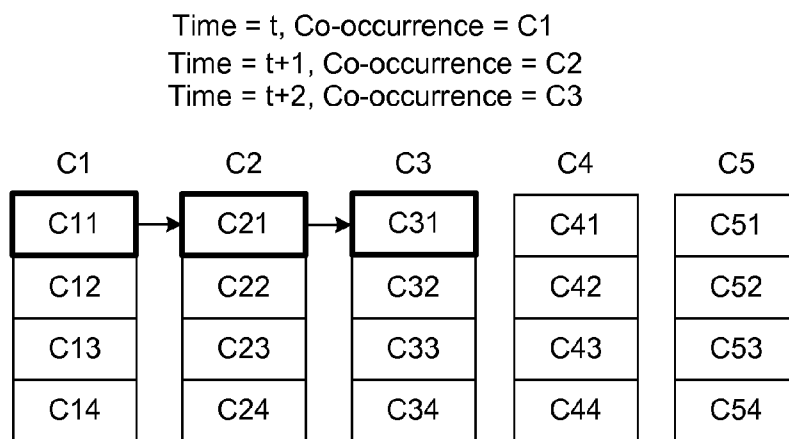


FIG. 13

		Inbound									
		C11	C12	C13	C14	C21	C22	...	C31	...	C54
Outbound	C11					+1		
	C12							
	C13							
	C14							
	C21							...	+1	...	
	C22							

	C54							

FIG. 14

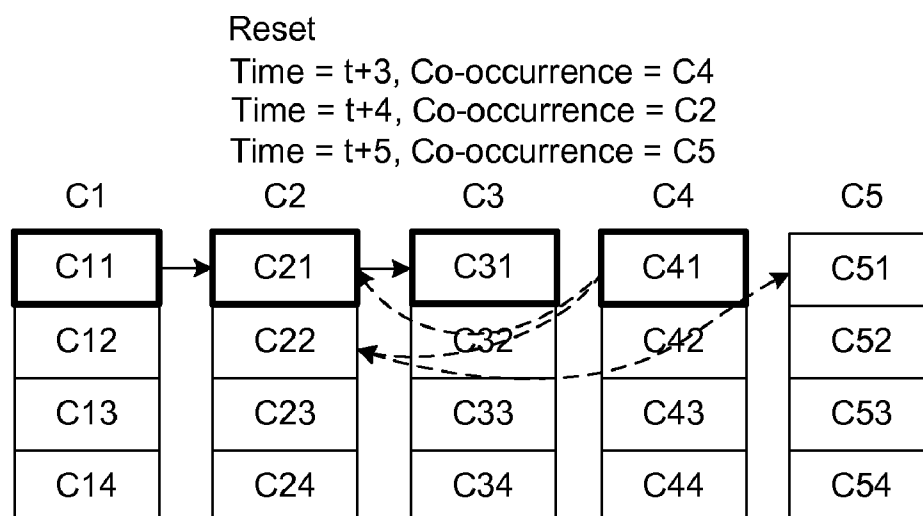


FIG. 15

Time = x, Co-occurrence = C1
 Forward Window: C2 → C3 → C4 → C5

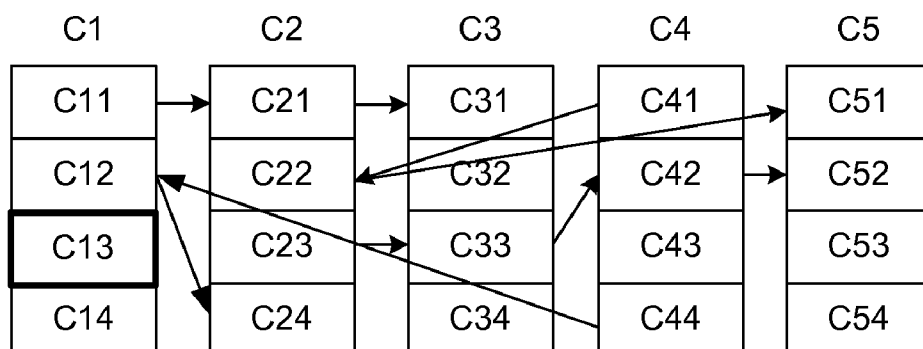


FIG. 16

HIERARCHICAL TEMPORAL MEMORY SYSTEM WITH HIGHER-ORDER TEMPORAL POOLING CAPABILITY

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority under 35 U.S.C. § 119(e) to U.S. Patent Application Ser. No. 61/060,891 entitled “Higher-Order Temporal Pooler” filed on Jun. 12, 2008, which is incorporated by reference herein in its entirety. This application is related to U.S. patent application Ser. No. 12/039,630 entitled “Spatio-Temporal Learning Algorithms in Hierarchical Temporal Networks” filed on Feb. 28, 2008; and U.S. patent application Ser. No. 12/147,348 entitled “Hierarchical Temporal Memory System with Enhanced Inference Capability” filed on Jun. 29, 2007, which are incorporated by reference herein in their entirety.

FIELD OF THE INVENTION

[0002] The present invention is related to machine learning using Hierarchical Temporal Memory (HTM) systems, and more specifically to learning sequences of co-occurrences related to a higher-order Markov model.

BACKGROUND

[0003] Hierarchical Temporal Memory (HTM) networks represent a new approach to machine intelligence. In a HTM network, training data comprising temporal sequences of spatial patterns are presented to a network of nodes. The HTM network then builds a model of the statistical structure inherent to the patterns and sequences in the training data, and thereby learns the underlying ‘causes’ of the temporal sequences of patterns and sequences in the training data. The hierarchical structures of the HTM network allow them to build models of very high dimensional input spaces using reasonable amounts of memory and processing capacity.

[0004] FIG. 1 is a diagram illustrating a hierarchical nature of the HTM network where the HTM network 10 has three levels L1, L2, L3, with level L1 being the lowest level, level L3 being the highest level, and level L2 being an intermediate level between levels L1 and L3. Level L1 has nodes 11A, 11B, 11C and 11D; level L2 has nodes 12A and 12B; and level L3 has node 13. In the example of FIG. 1, the nodes 11A, 11B, 11C, 11D, 12A, 12B, and 13 are hierarchically connected in a tree-like structure such that each node has several children nodes (that is, nodes connected at a lower level) and one parent node (that is, node connected at a higher level). Each node 11A, 11B, 11C, 11D, 12A, 12B, and 13 may have or be associated with a capacity to store and process information. For example, each node 11A, 11B, 11C, 11D, 12A, 12B, and 13 may store input data (for example, sequences of spatial patterns) associated with particular object or a state of an object. Further, each node 11A, 11B, 11C, 11D, 12A, 12B, and 13 may be arranged to (i) propagate information up the HTM hierarchy to any connected parent node and/or (ii) propagate information down the HTM hierarchy to any connected children nodes.

[0005] The HTM training process is a form of unsupervised machine learning. However, in a training stage, indexes attached to the input patterns may be presented to the HTM network to allow the HTM to associate particular categories with the underlying generative causes. Once an HTM network has built a model of a particular input space, it can

operate in an inference stage. In the inference stage, novel sequences of input patterns are presented to the HTM network, and the HTM network will generate a ‘belief vector’ that provides a quantitative measure of the degree of belief or likelihood that the input pattern was generated by the underlying cause associated with each of the indexed categories to which the HTM network was exposed in the training stage.

SUMMARY OF THE INVENTION

[0006] Embodiments provide a temporal pooler in a node of a Hierarchical Temporal Memory (HTM) network for generating temporal statistics data representing higher-order temporal sequences of spatial co-occurrences included in training patterns. The temporal pooler may receive information about spatial co-occurrences. The temporal statistics data is provided to an inference engine to determine the likelihood that the temporal sequences of spatial patterns in input patterns correspond to the temporal sequences of spatial co-occurrences present in the training patterns.

[0007] In one embodiment, the spatial co-occurrence is received from a spatial pooler linked to the temporal pooler. The spatial pooler providing the spatial co-occurrence may be located in the same node as the temporal pooler. Alternatively, the spatial pooler may be located at a node different from the node in which the temporal pooler is located.

[0008] In one embodiment, the temporal pooler splits a spatial co-occurrence into a plurality of sub-occurrences to represent a higher-order Markov model. Each sub-occurrence may be allotted to represent the same spatial co-occurrence in a different temporal sequence.

[0009] In one embodiment, the temporal relationships of the sub-occurrences are stored in a connectivity matrix. The connectivity matrix includes columns and rows representing multiple sub-occurrences split from a co-occurrence. Each entry in the connectivity matrix represents frequency or counts of instances where one sub-occurrence transitioned to another sub-occurrence. In the training stage, the connectivity matrix is updated as the information about spatial co-occurrences is received from the spatial pooler.

[0010] In one embodiment, the temporal pooler is configured to generate the temporal statistics data by receiving a single round of the sequence of the spatial co-occurrences information. The temporal pooler buffers the sequence of the spatial co-occurrence information. When spatial co-occurrence information for a first time and spatial co-occurrence information for a second time are received, the temporal pooler determines which entry in the connectivity matrix is to be updated based on the spatial co-occurrence information at a third time subsequent to the first time and the second time.

[0011] In one embodiment, the temporal pooler determines likelihood of transitions between sub-occurrences by generating a modified connectivity matrix and an activity variable. The modified connectivity matrix is created by adding values to entries of the connectivity matrix to increase the likelihood of otherwise unlikely transitions between sub-occurrences. The activity variable represents likelihood of receiving a co-occurrence at the first time and co-occurrences before the first time based on the temporal relationships represented by the connectivity matrix.

[0012] In one embodiment, the temporal pooler generates the temporal statistics data by receiving multiple rounds of the sequence of the spatial co-occurrence information. The temporal pooler maps co-occurrences in second spatial co-occurrence information from the spatial pooler at a second

time to sub-occurrences based on first spatial co-occurrence information from the spatial pooler at a first time preceding the second time. The temporal pooler expands the connectivity matrix counter by selectively splitting a co-occurrence in the connectivity matrix to a plurality of sub-occurrences.

[0013] The features and advantages described in the specification are not all inclusive and, in particular, many additional features and advantages will be apparent to one of ordinary skill in the art in view of the drawings, specification, and claims. Moreover, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the disclosed subject matter.

BRIEF DESCRIPTION OF DRAWINGS

[0014] FIG. 1 is a conceptual diagram illustrating a Hierarchical Temporal Memory (HTM) network.

[0015] FIG. 2 is a flowchart illustrating a method of operating the HTM network, according to one embodiment of the present invention.

[0016] FIG. 3A is a graph illustrating a Markov chain before splitting a co-occurrence into two sub-occurrences, according to one embodiment of the present invention.

[0017] FIG. 3B is a graph illustrating a Markov chain where a co-occurrence is split into two sub-occurrences, according to one embodiment of the present invention.

[0018] FIG. 4A is a functional block diagram of a HTM node for generating temporal statistics data related to a higher-order Markov model, according to one embodiment of the present invention.

[0019] FIG. 4B is a functional block diagram illustrating a portion of the HTM network, according to one embodiment of the present invention.

[0020] FIG. 5A is a table illustrating a connectivity matrix before splitting a co-occurrence into two sub-occurrences, according to one embodiment of the present invention.

[0021] FIG. 5B is a table illustrating a connectivity matrix after splitting a co-occurrence into two sub-occurrences, according to one embodiment of the present invention.

[0022] FIG. 6 is a higher-order temporal (HOT) pooler, according to one embodiment of the present invention.

[0023] FIG. 7 is a method of operating a HOT pooler, according to one embodiment of the present invention.

[0024] FIG. 8 is a functional block diagram of a HOT pooler, according to one embodiment of the present invention.

[0025] FIG. 9 is a flowchart illustrating a method of operating a HOT pooler, according to one embodiment of the present invention.

[0026] FIG. 10 is a diagram illustrating each co-occurrence in an input pattern split into four sub-occurrences, according to one embodiment of the present invention.

[0027] FIG. 11 is a connectivity matrix with four sub-occurrences allotted to one co-occurrence, according to one embodiment of the present invention.

[0028] FIG. 12 is a flow chart illustrating a method for computing the most likely connection between sub-occurrences, according to one embodiment of the present invention.

[0029] FIG. 13 is a state diagram illustrating connecting of sub-occurrences for the sequence of C1-C2-C3, according to one embodiment of the present invention.

[0030] FIG. 14 is a table illustrating updating of a connectivity matrix after receiving the sequence of C1-C2-C3, according to one embodiment of the present invention.

[0031] FIG. 15 is a state diagram illustrating a process of connecting sub-occurrences for the sequence of C4-C2, according to one embodiment of the present invention.

[0032] FIG. 16 is a state diagram illustrating a process of connecting sub-occurrences for the sequence of C1-C2-C3-C4-C5, according to one embodiment of the present invention.

DETAILED DESCRIPTION

[0033] In the following description of embodiments of the present invention, numerous specific details are set forth in order to provide a more thorough understanding of the present invention. However, note that the present invention may be practiced without one or more of these specific details. In other instances, well-known features have not been described in detail to avoid unnecessarily complicating the description.

[0034] A preferred embodiment of the present invention is now described with reference to the figures where like reference numbers indicate identical or functionally similar elements. Also in the figures, the left most digits of each reference number corresponds to the figure in which the reference number is first used.

[0035] Reference in the specification to “one embodiment” or to “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least one embodiment of the invention. The appearances of the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment.

[0036] Some portions of the detailed description that follows are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps (instructions) leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic or optical signals capable of being stored, transferred, combined, compared and otherwise manipulated. It is convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. Furthermore, it is also convenient at times, to refer to certain arrangements of steps requiring physical manipulations of physical quantities as modules or code devices, without loss of generality.

[0037] However, all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as “processing” or “computing” or “calculating” or “determining” or “displaying” or “determining” or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0038] Certain aspects of the present invention include process steps and instructions described herein in the form of an algorithm. It should be noted that the process steps and instructions of the present invention could be embodied in software, firmware or hardware, and when embodied in software, could be downloaded to reside on and be operated from different platforms used by a variety of operating systems.

[0039] The present invention also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, application specific integrated circuits (ASICs), or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus. Furthermore, the computers referred to in the specification may include a single processor or may be architectures employing multiple processor designs for increased computing capability.

[0040] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may also be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the present invention as described herein, and any references below to specific languages are provided for disclosure of enablement and best mode of the present invention.

[0041] In addition, the language used in the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the claims.

[0042] A Hierarchical Temporal Memory (HTM) network is a hierarchical network of interconnected nodes that individually and collectively (i) learn, over space and time, one or more causes of sensed input data and (ii) determine, dependent on learned causes, likely causes of novel sensed input data. In the HTM network, the nodes are hierarchically arranged so that the number of nodes decreases as level increases.

[0043] In one embodiment, a HTM node includes only the temporal pooler or the spatial pooler. For example, nodes at the first level of the HTM network may consist of HTM nodes having only spatial poolers, and the nodes at the second level of the HTM network may consist of HTM nodes having only temporal poolers. HTM nodes performing other functions may also be placed within the HTM network. Alternatively, a HTM node may include two or more levels of interconnected temporal poolers.

[0044] In one embodiment, one or more nodes of the HTM network receives inputs representing images, videos, audio

signals, sensor signals, data related to network traffic, financial transaction data, communication signals (e.g., emails, text messages and instant messages), documents, insurance records, biometric information, parameters for manufacturing process (e.g., semiconductor fabrication parameters), inventory patterns, energy or power usage patterns, data representing genes, results of scientific experiments or parameters associated with operation of a machine (e.g., vehicle operation) and medical treatment data. The HTM network may process such inputs and produce an output representing, among others, identification of object shown in an image, identification of recognized gestures, classification of digital images as pornographic or non-pornographic, identification of email messages as unsolicited bulk email ('spam') or legitimate email ('non-spam'), prediction of a trend in financial market, prediction of failures in a large-scale power system, identification of a speaker in audio recording, classification of loan applicants as good or bad credit risks, identification of network traffic as malicious or benign, identity of a person appearing in the image, processed natural language processing, weather forecast results, patterns of a person's behavior, control signals for machines (e.g., automatic vehicle navigation), gene expression and protein interactions, analytic information on access to resources on a network, parameters for optimizing a manufacturing process, predicted inventory, predicted energy usage in a building or facility, web analytics (e.g., predicting which link or advertisement that users are likely to click), identification of anomalous patterns in insurance records, prediction on results of experiments, indication of illness that a person is likely to experience, selection of contents that may be of interest to a user, indication on prediction of a person's behavior (e.g., ticket purchase, no-show behavior), prediction on election, predict/detect of adverse events, a string of texts in the image, indication representing topic in text, a summary of text or prediction on reaction to medical treatments,

[0045] A parent node is located at a higher level of the hierarchy compared to a child node. Nodes at different levels may process information differently.

[0046] Information about spatial co-occurrences refers to a value or a set of values indicating likelihood that an input pattern received at a node of the HTM network includes a co-occurrence. In one embodiment, the spatial co-occurrences information is a vector with elements, each element indicating the probability that a co-occurrence is present in the input pattern. In another embodiment, the spatial co-occurrences information indicates a single co-occurrence that is most likely to be present in the input pattern. The co-occurrence most likely to be present in the input pattern is also referred to as "winning co-occurrence" herein.

[0047] Embodiments provide a temporal pooler in a HTM network capable of storing information about sequences of higher-order co-occurrences by splitting a co-occurrence into a plurality of sub-occurrences. Each split sub-occurrence may be part of a distinct sequence of co-occurrences. The temporal pooler receives the probability of spatial co-occurrences in training patterns and tallies frequencies of transitions from one sub-occurrence to another sub-occurrence in a connectivity matrix. The connectivity matrix is then processed to generate temporal statistics data. The temporal statistics data is provided to an inference engine to perform inference or prediction on input patterns. By storing information related to connection of higher-order sequences, the tem-

poral statistics data may more accurately reflect long temporal sequences of co-occurrences in the training patterns.

Overview of HTM Network Operation

[0048] In one embodiment, the HTM network **10** operates in two stages: (i) a training stage and (ii) an inference stage. In the training stage, nodes at different levels of the HTM network **10** are placed in a training mode, and then progressively transitioned to an inference mode as the training at one level is finished. After training all the levels of the HTM network **10**, the inference stage is initiated where all the nodes in the HTM network **10** are placed in an inference mode, and the top node of the HTM network generates an output.

[0049] FIG. 2 is a flowchart illustrating a method of operating the HTM network **10**, according to one embodiment of the present invention. First, a level of the HTM network **10** is trained **210** based on training data. For example, the nodes **11A**, **11B**, **11C** and **11D** in the first level **L1** of the HTM network **10** of FIG. 1 are trained by providing the training data to the nodes **11A**, **11B**, **11C** and **11D**. Then the trained nodes (e.g., the nodes **11A**, **11B**, **11C** and **11D** of the HTM network **10**) are transitioned **220** to an inference mode. After transitioning to the inference mode, the trained nodes receive the same training data and generate **230** intermediate outputs in response to the training data. When a node transitions from the training mode to the inference mode, the node compiles and processes the training data. Depending on the learning mechanism employed, the same training data may be fed more than once to the nodes for training.

[0050] Then it is determined **240** if all the levels of the HTM network **10** are trained. If not all the levels of the HTM network are trained, the process proceeds **260** to train the next level, typically a higher level. For example, after training the nodes **11A**, **11B**, **11C** and **11D** of first level **L1**, the process proceeds to train nodes **12A** and **12B** in the second level **L2**. Intermediate outputs from the nodes **11A**, **11B**, **11C** and **11D** function as training data for the nodes **12A** and **12B** in second level **L2**. The same process is repeated until all levels of the nodes are trained based on the training data received from their children nodes.

[0051] In one embodiment, when training the top node **13** of the HTM network **10** after training lower nodes of the HTM network, category information indicating the correct category of the training data is provided to the top node **13**. In response, the top node **13** generates mapping information that maps the outputs from its children nodes (e.g., the nodes **12A** and **12B**) to the correct category. In this way, the top node **13** may output a category identifier corresponding to the sensed inputs in the inference stage.

[0052] If it is determined **240** that all the levels of the HTM network **10** are trained, then all the nodes of the HTM network **10** are placed **250** in the inference mode to start the inference stage. In the inference stage, the nodes **11A**, **11B**, **11C** and **11D** at the lowest hierarchy of the HTM network **10** receive **270** a sensed input. The nodes in the HTM network then generate their outputs using a mechanism disclosed, for example, in U.S. patent application Ser. No. 12/039,630 entitled "Spatio-Temporal Learning Algorithms in Hierarchical Temporal Networks" filed on Feb. 28, 2008; and U.S. patent application Ser. No. 12/147,348 entitled "Hierarchical Temporal Memory System With Enhanced Inference Capability" filed on Jun. 29, 2007, which are incorporated by reference herein in their entirety. In one embodiment, feedback signals from the nodes at higher levels may be provided

to the nodes at lower levels of the HTM network **10**. The top node **13** receives information from the lower nodes and generates a category identifier corresponding to the sensed inputs.

[0053] In one embodiment, at least one of the nodes in the HTM network includes a spatial pooler and a temporal pooler. In a typical HTM node, the spatial pooler collects sensed input or outputs from lower nodes and learns spatial groups of patterns (hereinafter referred to as "spatial co-occurrence"). Specifically, the spatial pooler learns spatial co-occurrences in a set of input patterns received at the HTM node. Spatial pooling is dependent on "spatial" similarities between two or more patterns (a pattern may actually represent a sequence from a lower level). In embodiments where spatial pooling is used, an HTM node may compare a spatial property of a received sensed input pattern with that of a learned sensed input pattern (or "quantization" point). If the two patterns are "similar enough" (that is, the distance between the two patterns are within a predetermined threshold), then the received sensed input pattern may be assigned to the same cause as that of the quantization point. Hence, the spatial pooler generates the same output from the similar input patterns. The similarity needed for "spatial" pooling may vary within and/or among HTM-based systems.

[0054] The outputs from the spatial poolers are provided to the temporal pooler. The temporal pooler learns sequences of co-occurrences that occur at a rate statistically greater than what would be expected by mere chance. The temporal pooler may be classified into a first order temporal pooler and higher-order temporal (HOT) pooler. The first order temporal pooler is capable of implementing and detecting sequences of for which the first order Markov model holds: given the present pattern, future patterns are independent of previous patterns. The first order temporal pooler is disclosed, for example, in U.S. patent application Ser. No. 12/039,630. The HOT pooler is capable of storing models representing a first order Markov chain as well as second or higher-order sequential patterns. Embodiments of the present invention provide HOT poolers for generating a model related to higher-order sequences.

Higher-Order Sequences

[0055] Many data representing real-world events are related to higher-order Markov models. Data related to motions and music, for example, require modeling of sequences in higher-order. FIG. 3A is a graph illustrating a Markov chain showing co-occurrences **C1** through **C5** in a first order graph. The first order Markov model keeps track of the likelihood of one co-occurrence following another co-occurrence but does not keep track of co-occurrences appearing further back in the past. Taking the example of FIG. 3A, the first order Markov model keeps track of transition counts from **C1** to **C3** (count=10), **C2** to **C3** (count=15), **C3** to **C4** (count=9), and **C3** to **C5** (count=16). Based on this first order Markov model, the likelihood of the co-occurrence **C4** or **C5** following **C3** may be predicted or inferred. The first order Markov model, however, does not allow consideration of whether the co-occurrences **C1** or **C2** preceded the co-occurrence **C3** for the purpose of predicting or inferring the co-occurrence to appear after the co-occurrence **C3**. Hence, a lower order Markov Model may be more ambiguous in terms of how the sequences or co-occurrences are arranged compared to a higher order Markov Model.

[0056] FIG. 3B is a graph illustrating a Markov chain showing co-occurrences **C1** through **C5** in a second order model by

splitting or replicating a co-occurrence into two sub-occurrences, according to one embodiment of the present invention. Sub-occurrences are co-occurrences split or replicated from a co-occurrence in a lower order Markov model to create a higher order Markov model of sequences. The sub-occurrences may be created for all co-occurrences or be created for selected co-occurrences. In FIG. 3B, co-occurrences C31 and C32 are sub-occurrences split from the co-occurrence C3 of FIG. 3A.

[0057] In the example of FIG. 3B, the sub-occurrences C31 and C32 are connected by dashed lines representing counts of instances where the co-occurrence C4 or C5 followed the sequences of co-occurrences C1-C3 (represented as C1-C31) or co-occurrences C2-C3 (represented as C2-C32). Specifically, the Markov chain of FIG. 3B indicates eight instances where the co-occurrence C4 followed the sequence of C1-C3, two instances where the co-occurrence C5 followed the sequence of C1-C3, one instance where the co-occurrence C4 followed the sequence of co-occurrences C2-C3, and fourteen instances where the co-occurrence C5 followed the sequence of co-occurrences C2-C3. By using this information, which one of co-occurrences C4 and C5 are more likely to follow after co-occurrence C3 can be determined taking into account which co-occurrence preceded the co-occurrence C3.

[0058] By generalization, the Nth order of Markov model may be implemented by splitting a co-occurrence into a plurality of sub-occurrences. The maximum order of a Markov model is restricted by limited memory space and the number of unique sequences in the training data.

Example HTM Node Detecting Higher-Order Sequences

[0059] FIG. 4A is a functional block diagram of a HTM node 410 for generating temporal statistics data representing a model related to a higher-order Markov chain, according to one embodiment of the present invention. The HTM node 410 may include, among other components, a higher-order temporal (HOT) pooler 420, an inference engine 430, and a spatial pooler 440. The HOT pooler 420 is coupled to the spatial pooler 440 to receive the probabilities $P(e^-_t|C)$ of co-occurrences during the training stage. The probabilities $P(e^-_t|C)$ is also referred to as evidence e^-_t herein. The inference engine 430 is coupled to the spatial pooler 440 and the HOT pooler 420. The inference engine 430 receives the probabilities $P(e^-_t|C)$ of co-occurrences during the inference stage. The inference engine 430 is also coupled to the HOT pooler 420 to receive the temporal statistics data 422.

[0060] The function and operation of the spatial pooler 440 is essentially the same as the spatial spooler described, for example, in U.S. patent application Ser. No. 12/039,630 entitled "Spatio-Temporal Learning Algorithms in Hierarchical Temporal Networks" filed on Feb. 28, 2008, which is incorporated by reference herein in its entirety. The spatial pooler 440 receives input patterns 450 and identifies spatial co-occurrences among the input patterns 50. At each time-step, the spatial pooler 440 outputs a set of probabilities $P(e^-_t|C)$ 442 for a spatial pattern, where $P(e^-_t|C)$ represents the probability of observing e^- (evidence or input patterns from a lower level) at time t over learned spatial co-occurrences C . Thus, for example, at time t , the first entry in $P(e^-_t|C)$ is $P(e^-_t|C_1)$, the second entry is $P(e^-_t|C_2)$, and so forth. In one embodiment, the spatial pooler 440 provides the

set of probabilities $P(e^-_t|C)$ 442 to the HOT pooler 420 in the training mode and to the inference engine 430 in the learning mode.

[0061] The HOT pooler 420 is hardware, software, firmware or a combination thereof responsible for detecting temporal sequences of a higher-order. The HOT pooler 420 maintains a connectivity matrix to tally the probabilities e^-_t of co-occurrences received from the spatial pooler 440, as described below in detail with reference to FIGS. 5A and 5B. In one embodiment, the HOT pooler 420 is a HOT pooler 600, as described below in detail with reference to FIG. 6. In another embodiment, the HOT pooler 420 is a HOT pooler 800, as described below in detail with reference to FIG. 8. The HOT pooler 420 generates the temporal statistics data 422 from the connectivity matrix and provides the temporal statistics data 422 to the inference engine 430.

[0062] The inference engine 430 is hardware, software, firmware or a combination thereof responsible for generating a bottom-up message 434 based on the probabilities e^-_t 442 and the temporal statistics data 422, as described, for example, in U.S. patent application Ser. No. 12/147,348 entitled "Hierarchical Temporal Memory System with Enhanced Inference Capability" filed on Jun. 29, 2007, which is incorporated by reference herein in its entirety. In one embodiment, the inference engine 430 includes a bottom-up message generator and a top-down message generator, as described in U.S. patent application Ser. No. 12/147,348. The inference engine 430 also received a top-down message 438 and generates a feedback message 454 to a child node of the HTM node 410 (if a child node is connected to the HTM node 410).

[0063] In one embodiment, the inference engine 430 generates a bottom-up message representing predictions based on input data received at the HTM node 410. The inference engine 430 may run multiple iterations to generate outputs representing predictions multiple time steps ahead of the current time step. Further, in the inference stage, the bottom-up message may indicate a plurality of top predictions.

[0064] In one embodiment, the function of the inference engine 430 is performed by the HOT pooler 420 and the inference engine 430 is omitted. In this embodiment, the HOT pooler 420 receives the probabilities e^-_t in the inference stage and generates a message to a parent node, as disclosed, for example, in U.S. patent application Ser. No. 12/039,630 entitled "Spatio-Temporal Learning Algorithms in Hierarchical Temporal Networks" filed on Feb. 28, 2008, which is incorporated by reference herein in its entirety.

[0065] FIG. 4B is a block diagram illustrating a portion of the HTM network, according to one embodiment of the present invention. The portion of the HTM network may include, among others, a HTM node 450, a HTM node 460A and a HTM node 460B. The HTM node 450 receives a first set of probabilities 442A and a second set of probabilities from the HTM nodes 460A and 460B. The sets of probabilities 442A and 442B may be concatenated and processed at the HTM nodes 460A and 460B.

[0066] The HTM node 460A and the HTM node 460B may include, among others, spatial poolers 464A and 464B, respectively. The spatial poolers 464A and 464B receives the input patterns 450A and 450B and generates the set of probabilities 442A and 442B in a manner essentially the same as the spatial pooler 440 of FIG. 4A.

[0067] The HTM node 450 may include, among others, a HOT pooler 420 and the inference engine 430. The HTM

node **450** is essentially the same as the HTM node **410** of FIG. **4A** except that the HTM node **450** does not include a spatial pooler. Other than receiving and concatenating the sets of probabilities **442** and **442B** received from the HTM nodes **460A** and **460B**, the function and operation of the HOT pooler **420** and the inference engine **430** of FIG. **4B** are essentially the same as those of HOT pooler **420** and the inference engine **430** of FIG. **4A**.

Connectivity Matrix

[0068] In order to generate the temporal statistics data **422** representing a higher-order Markov sequence of co-occurrences, the HOT pooler **420** maintains a connectivity matrix. The connectivity matrix managed by the HOT pooler **420** is similar to the time adjacency matrix described, for example, in U.S. patent application Ser. No. 12/039,630 entitled "Spatio-Temporal Learning Algorithms in Hierarchical Temporal Networks" filed on Feb. 28, 2008, which is incorporated by reference herein in its entirety, except that the connectivity matrix retains much more detailed information about the sequential order of the occurrences. Rather, the time adjacency matrix retains information on how close in time the co-occurrences (for example, **C1** and **C2**) appeared in a sequence. In embodiments of the present invention, the time adjacency matrix is replaced with a connectivity matrix to accommodate a higher-order sequence.

[0069] FIGS. **5A** and **5B** are tables illustrating connectivity matrixes, according to one embodiment of the present invention. The rows of the connectivity matrices in FIGS. **5A** and **5B** represent preceding co-occurrences or sub-occurrences in a transition (hereinafter referred to as "outbound occurrences"). The columns of the connectivity matrices represent subsequent co-occurrences or sub-occurrences in the transition (hereinafter referred to as "inbound occurrences"). FIG. **5A** is an example of the connectivity matrix corresponding to the Markov chain of FIG. **3A**. For example, the count of 10 in the entry where a row of co-occurrence **C1** intersects with a column of co-occurrence **C3** indicates that 10 instances of transition from co-occurrences **C1** to co-occurrence **C3** were observed. The column and the row may be interchanged so that the columns represent the outbound occurrences and the rows represent the inbound occurrences. Arrangements or mechanisms other than the connectivity matrix may also be used to store the temporal relationship between the co-occurrences.

[0070] When the co-occurrences are split into sub-occurrences, the rows and columns of the connectivity matrix are expanded to accommodate the sub-occurrences. FIG. **5B** is an example of the connectivity matrix corresponding to the Markov chain of FIG. **3B**. In the Markov chain of FIG. **3B**, the co-occurrence of **C3** is split into sub-occurrences **C31** and **C32**. The connectivity matrix of FIG. **5B** mirrors the Markov chain of FIG. **3B** by splitting a row and a column for co-occurrence **C3** in FIG. **5A** into two rows and two columns for sub-occurrences **C31** and **C32**. Taking the example of sequence **C31-C4**, a count of eight is indicated in the fifth column and the third row of the connectivity matrix in FIG. **5B**.

[0071] In one embodiment, the HOT pooler **420** updates the connectivity matrix as the probabilities e^- , **442** are received from the spatial pooler **440**. After the training of the HTM

node **410**, the HOT pooler **420** compiles and processes the connectivity matrix into the temporal statistics data **422**.

Multi-round Higher-Order Temporal Pooler

[0072] FIG. **6** is a block diagram illustrating a HOT pooler **600**, according to one embodiment of the present invention. The HOT pooler **600** receives the same sequence of input probabilities e^- , **442** for **N** rounds of times. The HOT pooler **600** may include, among other components, a mapper **610**, a connectivity matrix counter **620**, a connectivity matrix store **630**, a splitter **640** and a sequence grouper **650**. The mapper **610** is coupled to the connectivity matrix counter **620** to provide translated probabilities **614**. The connectivity matrix counter **620** is coupled to the connectivity matrix store **630** to provide instructions **624** to update the connectivity matrix. The splitter **640** is coupled to the connectivity matrix store **630** to analyze and split co-occurrences or sub-occurrences. The splitter **640** is also coupled to the mapper **610** to provide mapping update information **644**. The connectivity matrix store **630** is coupled to the sequence grouper **650** to provide the connectivity matrix **634**. In one embodiment, the splitter **640** also merges sub-occurrences, deletes sub-occurrences and modifies the sub-occurrences to find a more accurate and efficient model of the temporal sequences.

[0073] The mapper **610** receives the probabilities of spatial co-occurrences e^- , **442** from the spatial pooler **440**. The mapper **610** includes a buffer for storing probabilities (e.g., e^-_{t-1} and e^-_{t-2}) received at previous time steps. Based on the probabilities of spatial co-occurrences stored in the buffer, the mapper **610** maps co-occurrences in the probabilities of co-occurrences e^- , **442** of to sub-occurrences in the connectivity matrix. The mapper **610** may buffer probabilities of co-occurrences going further back in time in order to select sub-occurrences that encode long distinct sequences. The mapper **610** generates the translated probabilities **614** by mapping the probabilities e^- , **442** to the probabilities of sub-occurrences. The translated probabilities **614** are then sent to the connectivity matrix counter **620**. The translated probabilities **614** are probabilities of co-occurrences mapped to probabilities of sub-occurrences in the connectivity matrix based on the previously received probabilities of co-occurrences. In one embodiment, the mapping is assigned without previous data.

[0074] The connectivity matrix counter **620** receives the translated probabilities **614** from the mapper and sends the instructions **624** to the connectivity matrix store **640** for updating the connectivity matrix. The instructions **624** indicate entries in the connectivity matrix to update and the amount to be incremented in each entry. The connectivity matrix counter **620** may also receive instructions to weaken or delete entries in the connectivity matrix. This may be thought of as "forgetting" and may help generalize to new future data or to improve the efficient utilization of limited resources.

[0075] The connectivity matrix store **630** stores the connectivity matrix representing the temporal sequences of co-occurrences received at the HOT pooler **600**. The connectivity matrix store **630** receives the instructions **624** and updates the entries in the connectivity matrix. After updating the counts based on received co-occurrences, the connectivity matrix in the connectivity matrix store **630** is (i) accessed by the splitter **640** to split co-occurrence(s) or sub-occurrences and updated by another round of the same probabilities of spatial co-occurrences e^- , **442** or (ii) accessed by the sequence grouper **650** to generate the temporal statistics data **422**. In one embodiment, the connectivity matrix also stores additional

values at each row and column, such as time stamps, permanence and other simple statistics.

[0076] After probabilities e^- , 442 for a set of training data are received and processed, the splitter 640 accesses the connectivity matrix store 630 to select any co-occurrences or sub-occurrences to be split into two or more sub-occurrences. Specifically, the splitter 640 determines which occurrence or sub-occurrence to split based on rules that may include, among others, the following: (i) avoid splitting of occurrences or sub-occurrences which appear in the sequence infrequently (i.e., below a threshold number of times), and (ii) avoid splitting of co-occurrences or sub-occurrences which are inbound occurrences predominantly from one outbound occurrence. After determining which co-occurrences or sub-occurrences to split, the splitter 644 expands the connectivity matrix by splitting selected co-occurrences or sub-occurrences. The splitter 644 also sends the mapping update information 644 to the mapper 610 to update the mapping between co-occurrences and the sub-occurrences to prepare the mapper 610 for the next round of the probabilities 442.

[0077] The sequence grouper 650 receives the connectivity matrix 634 from the connectivity matrix store 630 and partitions the co-occurrence, sub-occurrences and their connections into groups representing sequences learned by the HOT pooler 600. The sequence grouper 650 then sends a description of the sequences determined from partitioning to the inference engine 430 in the form of the temporal statistics data 422. The description of the sequences may be a partition of the sub-occurrences and may include overlapping sets of sub-occurrences or skip some sub-occurrences. The sequences may also include information about likely starting and ending points, gathered from the connectivity between sub-occurrences in one sequence to sub-occurrences in other sequences.

[0078] FIG. 7 is a flow chart illustrating a method of operating the HOT pooler 600, according to one embodiment of the present invention. First, the HOT pooler 600 receives 710 probabilities e^- , 442 of co-occurrences from the spatial pooler 440. The HOT pooler 600 then translates the probabilities e^- , 442 of co-occurrences to probabilities of sub-occurrences and updates 712 the connectivity matrix accordingly. Then it is determined 718 whether all inputs in the current round have been processed. If not, the process returns to receive 710 the next probabilities and repeats the process.

[0079] If all inputs in the current round have been used, the process proceeds to determine 722 if valid sub-occurrences were learned and connections of sufficient strength have been formed, given the training data and available capacity limits. If it is determined that more sub-occurrences and better statistics are needed, the splitter 640 determines 726 the co-occurrence or sub-occurrence to be split. Then the connectivity matrix is expanded 730 to include rows and columns of the split co-occurrence or sub-occurrence. The splitter 640 also updates 734 the mapping of co-occurrences to sub-occurrences stored in the mapper 610 based on the splitting of co-occurrences or sub-occurrences.

[0080] Then the process returns to repeat another round of probabilities e^- , 442 that is same as the probabilities e^- , received in the previous round. Specifically, the process returns to receive 710 the set of probabilities e^- , 442 and repeats the updating 712 of the connectivity matrix. In one embodiment, the next round of training data may use data different data compared to the training data in the previous round.

[0081] If it is determined 722 that sufficient sub-occurrences and connectivity counts have been learned, then the process proceeds to group or partition 742 sequences of co-occurrences and sub-occurrences to generate the temporal statistics data 422. Specifically, the sequence grouper 650 partitions the Markov chain represented by the connectivity matrix into one or more groups, for example, as described in U.S. patent application Ser. No. 12/039,630 entitled "Spatio-Temporal Learning Algorithms in Hierarchical Temporal Networks" filed on Feb. 28, 2008, which is incorporated by reference herein in its entirety. Alternatively, the sequence grouper 650 partitions the Markov chain using Agglomerative Hierarchical Clustering method to generate the temporal statistics data 422. Agglomerative Hierarchical Clustering method is well known; and, descriptions thereof are omitted herein for the sake of brevity. In one embodiment, weak connections are eliminated, and likely starting and ending points of a sequence is identified using statistical measures on the connectivity matrix. Sets of connected sub-occurrences may be identified after deleting weak connections and connecting between ending and starting points. The connected sub-occurrences may be marked as sequences.

[0082] After the temporal statistics data 422 is generated, the temporal statistics data 422 are sent 746 to the inference engine 430. Then the process of learning sequences at the HOT node 600 is terminated.

Single-Round Higher-Order Temporal Pooler

[0083] FIG. 8 is a functional block diagram of a HOT pooler 800, according to one embodiment of the present invention. Unlike the HOT pooler 600, the HOT pooler 800 is capable of generating the connectivity matrix representing higher-order Markov chain using a single round of sequences instead of multiple rounds of the same sequences. For this purpose, the HOT pooler 800 allots a predetermined number of sub-occurrences to a co-occurrence and employs a window buffer 810 to store probabilities of co-occurrences for multiple time steps. The order of the Markov model implemented by the HOT pooler 800 is determined by how many states are allotted to the same co-occurrence.

[0084] The HOT pooler 800 may include, among other components, a window buffer 810, a connectivity matrix updater 820, a connectivity matrix store 830, and a sequence grouper 840. The window buffer 810 is coupled to the connectivity matrix updater 820 to provide probabilities 814 of future sub-occurrences that appear after the current co-occurrences. The connectivity matrix updater 820 is coupled to the connectivity matrix store 830 to update the connectivity matrix. The sequence grouper 840 is coupled to the connectivity store 830 to receive the connectivity matrix 834 after the connectivity matrix 834 is updated by the training data.

[0085] The window buffer 810 receives and stores a predetermined number of probabilities received from the spatial pooler 440. Specifically, the window buffer 810 stores the probabilities of co-occurrences across multiple time steps. The probabilities are provided to the connectivity matrix updater 820 to determine the likelihood of transition between a sub-occurrence at time t and a sub-occurrence at time $t+1$ based on the probabilities of co-occurrences across multiple time steps.

[0086] The connectivity matrix updater 820 receives co-occurrences 814 from the window buffer 810 and determines entries in the connectivity matrix to be incremented and the amount to be incremented. In one embodiment, the connec-

tivity matrix updater **820** uses probabilities of future co-occurrences for W length of time (i.e., $e_{t+1}^-, e_{t+2}^-, \dots, e_{t+W}^-$) to update entries of the connectivity matrix corresponding to the transition of sub-occurrence at time t to sub-occurrence at time $t+1$. That is, the processing at the connectivity matrix updater **820** is lagged by the size W of the buffer window. When the lag must be shortened, W can be temporarily decreased. When extra lag is acceptable, W may be increased for more accurate sequence learning.

[0087] The connectivity matrix store **830** stores the connectivity matrix. The HOT pooler **800** receives information about the number of co-occurrences classified by the spatial pooler **440** and initializes the connectivity matrix to include rows and columns for a predetermined number of sub-occurrences. Contrary to the HOT pooler **600** where co-occurrences are selected for splitting into sub-occurrences after each round of training data, the connectivity matrix allots a predetermined number of sub-occurrences to all or any co-occurrences before the training data are received at the HOT pooler **800**. Predetermination may be used with sparse storage of only connected states to efficiently emulate dynamic creation of sub-occurrence when storage memory is limited. Further, splitting processes expressed earlier can be combined with predetermined sub-occurrences if it becomes necessary to increase the capacity from its initial limits.

[0088] In one embodiment, the sequence grouper **840** receives the connectivity matrix **834** from the connectivity matrix store **830** after the connectivity matrix **834** is updated based on the training data. The sequencer grouper **840** partitions the co-occurrences and sub-occurrences into groups of sub-occurrences and their connections. Specifically, the sequence grouper **840** partitions the interconnected matrix into one or more groups, for example, as described in U.S. patent application Ser. No. 12/039,630 entitled "Spatio-Temporal Learning Algorithms in Hierarchical Temporal Networks" filed on Feb. 28, 2008, which is incorporated by reference herein in its entirety. Alternatively, the sequence grouper **840** partitions the Markov chain using Agglomerative Hierarchical Clustering method to generate the temporal statistics data **422**. Agglomerative Hierarchical Clustering method is well known; and, descriptions thereof are omitted herein for the sake of brevity. The sequence grouper **840** then sends information of the sequences determined by partitioning the Markov chain to the inference engine **430** in the form of the temporal statistics data **422**.

[0089] FIG. 9 is a flowchart illustrating a method of operating the HOT pooler **800**, according to one embodiment of the present invention. First, a connectivity matrix M and variables are initialized **902**. In one embodiment, the number of unique co-occurrences C is received from the spatial pooler **440**. If each co-occurrence is split into K sub-occurrences, the connectivity matrix M is initialized to include $(C \times K)$ rows and $(C \times K)$ columns. In one embodiment, all entries in the matrix M are initially set to zero to represent that there is no connection between any sub-occurrences.

[0090] In one embodiment, the initialized variables include activity variable $\alpha_{o,i}$ for each sub-occurrence i (where $\forall i \in [1, C \times K]$). The activity variable $\alpha_{t,i}$ is proportional to the probability that co-occurrence i is active at time t given all previous co-occurrences are active from time $T=0$ to time $T=t$, given the connective relationships as represented by the connectivity matrix M . The activity variable $\alpha_{o,i}$ for each sub-occurrence i (where $\forall i \in [1, C \times K]$) is initialized to $1/S$ where S equals $C \times K$. A connectivity constant C for adding connect-

tivity value to all possible sub-occurrences may also be initialized to a specific value. In another embodiment, initialization of the activity variable $\alpha_{t,i}$ may be non-uniform to capture higher probability of starting in certain sub-occurrences.

[0091] FIG. 10 is a diagram illustrating co-occurrences $C1$ through $C5$ where each co-occurrence is split into four sub-occurrences, according to one embodiment of the present invention. The sub-occurrences $C11$, $C12$, $C13$ and $C14$ are split from the co-occurrence $C1$; the sub-occurrences $C21$, $C22$, $C23$ and $C24$ are split from the co-occurrence $C2$; the sub-occurrences $C31$, $C32$, $C33$ and $C34$ are split from the co-occurrence $C3$; the sub-occurrences $C41$, $C42$, $C43$ and $C44$ are split from the co-occurrence $C4$; and sub-occurrences $C51$, $C52$, $C53$ and $C54$ are split from the co-occurrence $C5$. The number of sub-occurrences is merely illustrative and a co-occurrence may be split into different numbers of sub-occurrences. Over-splitting the co-occurrences simply leads to more distinct sequences. However, such distinct sequences are likely to be classified to the same group during the partitioning process at the sequence grouper **840**. Hence, other than requiring more memory and computational resources, increasing the number of sub-occurrences does not negatively affect the performance of the HTM network. Further, not all co-occurrences must be split into the same number of sub-occurrences. For example, one co-occurrence may be split into four sub-occurrences and another co-occurrence may be split into two sub-occurrences during the initialization process.

[0092] FIG. 11 is a connectivity matrix for representing the connective relationship between the sub-occurrences $C11$ through $C54$, according to one embodiment of the present invention. Because there are five co-occurrences and each co-occurrence has four sub-occurrences, the connectivity matrix of FIG. 11 includes twenty rows and twenty columns. As in the connectivity matrices of FIGS. 5A and 5B, the rows of the connectivity matrix in FIG. 11 represent outbound occurrences and the columns of connectivity matrix in FIG. 11 represent inbound occurrences. In one embodiment, all entries in the connectivity matrix of FIG. 11 are initially set to zero. As outputs from the spatial pooler **440** are received, the connectivity matrix is updated to reflect the temporal relationships between the sub-occurrences.

[0093] Referring back to FIG. 9, the window buffer **810** stores the probabilities of co-occurrences after initializing **902** the connectivity matrix M and other variables. Specifically, the probabilities for co-occurrences from time t to $t+W$ (i.e., $e_t^-, e_{t+1}^-, \dots, e_{t+W}^-$) are received **904** at the HOT pooler **800** where W represents the size of the buffer window for storing the probabilities of co-occurrences. The window buffer **810** stores **908** the probabilities for co-occurrences from time $t+1$ to $t+W$ (i.e., $e_{t+1}^-, e_{t+2}^-, \dots, e_{t+W}^-$) for reference by the connectivity matrix updater **820** while the probabilities of co-occurrences at time t (i.e., e_t^-) are provided to the connectivity matrix updater **820**. When the sequence of co-occurrences is highly variable, it is generally advantageous to use a shorter buffer window. Conversely, when rapid processing of long sequences is needed, a longer window buffer is generally preferred. In one embodiment, if the number of remaining training co-occurrences is less than the size of the buffer windows, the buffer window is truncated to store a matching number of probabilities for the remaining training data.

[0094] Then the connectivity matrix updater **820** computes **912** the likelihood of sub-occurrences active at time $T=t$ and

at time $T=t+1$ for the sequence of co-occurrences in the spatial pooler **440**. FIG. 12 is a process for computing likely connections from a sub-occurrence at time t to a sub-occurrence at time $t+1$, according to one embodiment of the present invention. In this embodiment, a modified connectivity matrix Π is computed **1202** from the connectivity matrix M . The connectivity matrix M includes a large number of zero entries meaning that transition from a sub-occurrence of the column to a sub-occurrence of the row was not previously observed. In the modified connectivity matrix Π , each entry is incremented by a small value to represent that there is a small possibility that connections exist between sub-occurrences even though such transitions were not previously observed. By computing the likelihood of the transition between sub-occurrences based on the modified connectivity matrix, a single simple procedure may be used to connect new sub-occurrences and strengthening existing connections. The smoothing reduces the number of parameters to control these processes. In another embodiment, the modified connectivity matrix Π is not used, and the following processes of computing backward variable and the probability matrix ξ_t is based on the unmodified connectivity matrix M . Other rules may be also be used to decide whether to create a new connection or randomize creation of connections to increase the number of connections.

[0095] In one embodiment, the modified connectivity matrix Π is computed by the following equation:

$$\Pi = R_M \left(M + \frac{\epsilon}{S} 1_{S \times S} \right) \quad \text{equation (1)}$$

where R_M is a normalizing vector, S represents the total number of sub-occurrences (equals $C \times K$), ϵ is a connectivity constant, and $1_{S \times S}$ is an $S \times S$ matrix of ones. The connectivity constant ϵ represents the likelihood of allocating different sub-occurrences corresponding to the same co-occurrences to different temporal sequences. In general, higher connectivity constant C increases the likelihood of connecting co-occurrences in different sequences to an already allocated sub-occurrence whereas lower connectivity constant C increases the likelihood of allocating co-occurrences appearing different sequences to different sub-occurrences. The normalizing vector R_M is defined as follows:

$$R_M = \quad \text{equation (2)}$$

$$\begin{pmatrix} \left(\epsilon + \sum_{j=1}^S M_{1,j} \right)^{-1} & 0 & 0 & \dots & 0 \\ 0 & \left(\epsilon + \sum_{j=1}^S M_{2,j} \right)^{-1} & 0 & \dots & 0 \\ \vdots & & \ddots & & \\ 0 & \dots & & \left(\epsilon + \sum_{j=1}^S M_{S,j} \right)^{-1} \end{pmatrix}$$

where $M_{i,j}$ represents a value at i column and j row of the connectivity matrix M . The normalize vector R_M functions to make the rows of the modified connectivity matrix Π sum to one, allowing the modified connectivity matrix Π to be treated as a discrete conditional probability distribution.

Other processes may be used to generate modified connectivity matrix to serve the same purpose. For example, the modified connectivity matrix Π' is obtained by adding different random values of small values to each entry of the connectivity matrix M .

[0096] Backward variable $\beta_{t+1,j}^{t+W}$ is then computed **1206**. The backward variable $\beta_{t+1,j}^{t+W}$ represents the probability that evidence e_{t+2}^- through e_{t+W}^- are received from the spatial pooler **440** if the sub-occurrence j is active at time $t+1$. $\beta_{t+1,j}^{t+W}$ is calculated first by setting $\beta_{t+W,j}^{t+W}$ to $1/S$ and then computing back to $\beta_{t+1,j}^{t+W}$. For this purpose, $\beta_{t+W-j,d}^{t+W}$ is computed using the following equation:

$$\beta_{t+W-j,d}^{t+W} = \Pi(\beta_{t+W-j+1,d}^{t+W} \cdot e_{t+W-j+1}^-) \quad \text{equation (3)}$$

where $w=1$ to W , Π is a matrix-vector product, and the dot (\cdot) represents element-wise multiplication of vectors. Then $\beta_{t+W-j,j}^{t+W}$ is computed using the following equation:

$$\beta_{t+W-j,j}^{t+W} = \tilde{\beta}_{t+W-j,j}^{t+W} / \sum_{j=1}^S \tilde{\beta}_{t+W-j,j}^{t+W} \quad \text{equation (4)}$$

By repeating equations (3) and (4) for increased w values, $\beta_{t+W-j,j}^{t+W}$ can be computed.

[0097] In one embodiment, the transition probability matrix ξ_t is then computed **1208** to determine the likelihood of transitions between sub-occurrences. The transition probability matrix ξ_t includes elements indicating the probability of transition from a sub-occurrence at time t to a sub-occurrence at time $t+1$. In order to obtain the transition probability matrix ξ_t , an intermediate matrix N is obtained by computing outer product of the activity variable α_t with the element-wise product $(\beta_{t+1,j}^{t+W} \cdot e_{t+1}^-)$. The intermediate matrix N is a matrix of size $S \times S$. The intermediate matrix N approximates the likelihood of receiving previous co-occurrences up to time t and the co-occurrences in the window buffer (occurrences at time $t+1$ through time $t+W$) if the sub-occurrence of its column was active at time t and sub-occurrence of its row was active at time $t+1$. The transition probability matrix ξ_t is then computed by element-wise multiplying the intermediate matrix N by the modified connectivity matrix Π . The column of the highest valued entry in the transition probability matrix ξ_t represents the most likely sub-occurrence at time t and the row of the highest valued entry in the transition probability matrix ξ_t represents the most likely sub-occurrence at time $t+1$.

[0098] Referring back to FIG. 9, the connectivity matrix M is then updated **916** based on the transition probability matrix ξ_t . Specifically, the transition probability matrix ξ_t is sparsified **1212** by leaving only a predetermined number of entries with the highest values while setting other entries to zeros. By sparsifying the transition probability matrix ξ_t , subsequent processing can be simplified as well as allowing easier interpretation of learned sub-occurrences and easier grouping of sub-occurrences due to the reduced connectivity. After sparsifying the transition probability matrix ξ_t , the transition probability matrix is resealed **1216** to a normalized transition probability matrix $\tilde{\xi}_t$, where the sum of all entries in the normalized transition probability matrix $\tilde{\xi}_t$ equals one. The updated connectivity matrix M is then computed by adding the normalized transition probability matrix $\tilde{\xi}_t$ to the previous connectivity matrix M .

[0099] The updated activity variable α_{t+1} , is computed 918 based on the modified connectivity matrix II. Specifically, the activity variable α_{t+1} , is computed by the following equation:

$$\alpha_{t+1} = (\alpha'_t \Pi)' \cdot e_{t+1}^- \quad \text{equation (5)}$$

where the prime (') indicates transpose function, and the dot (·) indicates element-wise multiplication.

[0100] It is then determined 920 whether there are further training data to the HOT pooler 800. If there is an additional training data from the spatial pooler 440, the process proceeds to advance 924 the time t by incrementing the time. Then probabilities e_{t+w}^- , for co-occurrences at time t+w (with time t advanced) are received 928 at the window mapper 810. The stored probabilities in the window buffer 810 are then shifted 932 to store the newly received probabilities e_{t+w}^- . The co-occurrence e_{t-1}^- is removed from the window buffer 810. Then the process returns to computing 912 the likelihood of sub-occurrences.

[0101] If it is determined 920 that no further training data are available, then the training of the signal pass HOT pooler 800 is terminated and the process ends. After all the training data are exhausted, the connectivity matrix M contains entries indicating connective strengths and temporal relationships between sub-occurrences. That is, each entry of $M_{i,j}$ indicates the strength of a connection from sub-occurrence i to sub-occurrence j.

[0102] In one embodiment, the HOT pooler 800 receives multiple rounds of probabilities e_t^- 442 to refine the connectivity matrix M. In this embodiment, the activation variable α is reset for each rounds of probabilities e_t^- 442 but the connectivity matrix M generated from the previous round(s) of probabilities e_t^- 442 are retained. This embodiment allows even more repetitions of the same data to further differentiate connection strengths, simplifying the removal of relatively weak connections during grouping. Also, additional rounds may include new data, in which case the reset prevents creation of connections across disparate data sets. Additional rounds can also be run after modifying some of the sub-occurrences using forgetting or splitting. In this way, the learned model can be adapted to new data in an on-line manner.

Examples of Processing at Single-Round Higher-Order Temporal Pooler

[0103] The transition between the co-occurrences in the sequence may be one of the following cases: (i) the connectivity matrix does not include entries representing connections of sub-occurrences corresponding to a received sequence and there is at least one unallocated sub-occurrences available; (ii) the connectivity matrix already includes connections of sub-occurrences corresponding to the received sequence; (iii) the connectivity matrix includes connections of sub-occurrences representing a subset of the received sequence; and (iv) the connectivity matrix does not include any connections corresponding to the received sequence and there is no unallocated sub-occurrence remaining to represent the sequence. These cases are described below in detail with reference to FIGS. 13 through 16. In the purpose of following examples, assume the following for the sake of convenience: (i) four sub-occurrences are assigned to each co-occurrence, (ii) only winning occurrence is passed from the spatial pooler 440 to the HOT pooler 800 (i.e., only the most likely co-occurrence in evidence e_t^- has a value of

one while all the other co-occurrences are assigned value of zero), and (iii) the window size W is four.

[0104] FIG. 13 illustrates an example of the first case where the connectivity matrix does not include entries representing connections of sub-occurrences corresponding to a received sequence and there is at least one unallocated sub-occurrences available, according to one embodiment. The sequence received at the HOT pooler 800 includes a co-occurrence C1 at time t, followed by a co-occurrence C2 at time t+1, and a co-occurrence C3 at time t+3. If it is assumed that the co-occurrences C1, C2 and C3 are received at the HOT pooler 800 for the first time, there should be four unassigned sub-occurrences remaining for each co-occurrence. The length of sequence C1-C2-C3 is three co-occurrences; and therefore, the buffer window is truncated to store two co-occurrences.

[0105] When the transition probability matrix ξ_t is computed for transition between co-occurrences C1 and C2, the entries in the transition probability matrix ξ_t representing transitions from one of sub-occurrences C11, C12, C13 and C14 to one of sub-occurrences C21, C22, C23 and C24 all have an equally high value, which means that all of these transitions are equally likely. In one embodiment, one of these entries in the transition probability matrix ξ_t is selected and the value of this entry is set to one other entries are set to zero. This sparsifies the transition probability matrix ξ_t and preserves remaining sub-occurrences for other sequences.

[0106] Similarly, when the transition probability matrix ξ_{t+1} is computed, the entries representing transitions from one of sub-occurrences C21, C22, C23 and C24 to one of sub-occurrences C31, C32, C33 and C34 all have equally high values, which means that all of these transitions are equally likely. Entries in the transition probability matrix ξ_{t+1} corresponding to one of these transitions are selected and set to one while other entries are set to zero. In the example of FIG. 13B, the transition from sub-occurrence C21 to sub-occurrence C31 is selected.

[0107] FIG. 14 is a table illustrating updating entries in the connectivity matrix after receiving the sequence of C1-C2-C3, according to one embodiment. The transition from the sub-occurrence C11 to sub-occurrence C21 and the transition from the sub-occurrence C21 to sub-occurrence C31 are represented by an entry 1310 and an entry 1320, respectively. After receiving the sequences of C1-C2-C3, each of the entries 1310 and 1320 is incremented by one to indicated strengthened connection between the co-occurrences C1, C2 and co-occurrences C2, C3.

[0108] FIG. 15 illustrates an example where a sequence of C4-C2-C5 is received at the HOT pooler 800 as the winning occurrences, according to one embodiment. In this embodiment, the sequence received at the HOT pooler 800 includes a co-occurrence C4 at time t+3 followed by co-occurrences C2 at time t+4 and co-occurrence C5 at time t+5. Assume that the sequence of C1-C2-C3 as illustrated in FIG. 13 was previously received at times t through t+2. The sub-occurrence C21 is already allotted to represent the co-occurrence C2 in the sequence C1-C2-C3. Therefore, the HOT pooler 800 must determine whether to connect any one of sub-occurrence C41 through C44 to the sub-occurrence C21 (already allotted) or one of sub-occurrences C22 through C24 (not allotted). When the transition probability matrix ξ_{t+3} is computed, the entries of the transition probability matrix ξ_{t+3} corresponding to transition from any one of sub-occurrence C41 through C44 to any one of sub-occurrence C22 through C24 will have equally

high values. The entry of the transition probability matrix ξ_{t+3} representing transition from any one of sub-occurrence C41 to sub-occurrence C21 will have lower values because the co-occurrence C5 (not the co-occurrence C2) follows the co-occurrence C2. Therefore, one of the entries corresponding to a transition from any one of sub-occurrences C41 through C44 to sub-occurrences C22 through C24 is selected. The value in the selected entry is increased to one while values for other entries are set to zeroes. In the example of FIG. 15, a transition from sub-occurrence C41 to sub-occurrence C22 is selected.

[0109] In the second case where the connectivity matrix already includes connections corresponding to the received sequence, the connectivity matrix updater 820 increases counts in the entries corresponding to the connections between sub-occurrences representing the received sequence. For example, if another sequence of C1-C2-C3 is received, the counts in entries 1310 and 1320 are again incremented to indicate stronger connections between the co-occurrences.

[0110] In the third case where the connectivity matrix includes connections that represent a subset of the received sequence, either new connections may be created or preexisting connections may be strengthened. Whether to create new connections or to strengthen preexisting connections may be decided based on unallocated sub-occurrences remaining and the connectivity constant ϵ .

[0111] In the fourth case where the connectivity matrix does not include any connections that correspond to the received sequence and there is no unallocated sub-occurrence remaining to represent the new sequence, an allocated sub-occurrence is selected for connection in a manner that least affects other important connections in the connectivity matrix. FIG. 16 illustrates receiving a new sequence of C1-C2-C3-C4-C5 after all the sub-occurrences corresponding to the co-occurrence C2 are allotted, according to one embodiment of the present invention. In order to determine which one of the sub-occurrence C21, C22, C23 or C24 are to be connected to sub-occurrence C13, the transition probability matrix ξ_t is computed, as described above in detail with reference to FIG. 12. The row of the entry in the transition probability matrix ξ_t with the highest value indicates which one of the sub-occurrences C21, C22, C23 and C24 is to be connected to sub-occurrence C13.

Alternative Embodiments

[0112] In one embodiment, the connectivity matrix M is updated as additional training patterns become available. After generating a connectivity matrix M based on a first set of training patterns, a second set of training patterns may be fed to the HTM network to update the connectivity matrix M. When training the HTM network with the second set of training patterns, it is advantageous to use the connectivity matrix generated by the first set of training patterns because previous connections established by the first set of training patterns may still be applicable. The activation variable α may be reset when training the HTM network with the second set of training patterns when there is unlikely to be any connections between the first set of training patterns and the second training patterns.

[0113] In one embodiment, a transition from one sub-occurrence to the same sub-occurrence is prevented. In cases where sequences can have very flexible timings, durations or speeds, it is undesirable to allow connections from a sub-occurrence to itself, or even to other sub-occurrences of the

same occurrence. Such connections model the duration an occurrence is active for, and if this can change wildly then it is wasted modeling. To prevent the recursive connection, the term

$$\left(M + \frac{\epsilon}{S} 1_{S \times S}\right)$$

is replaced with

$$\left(M + \frac{\epsilon}{S-1} (1_{S \times S} - I)\right)$$

in equation (1) where I represents an identity matrix of size $S \times S$. The replaced term prevents incrementing of values in diagonal entries ($i=j$) of the modified connectivity matrix, preventing any recursive connections. The remaining process associated with equations (2) through (5) remains unchanged.

[0114] In one embodiment, entries of the connectivity matrix are corrected if values of the entries drop to zero despite the presences of corresponding co-occurrences in the sequence of co-occurrences. The entries may drop to zero despite the presence of corresponding co-occurrences due to a loss of numerical precision or the constraints imposed on the learning. Without correcting such entries, correct connections may not be made. To rectify this problem, the connectivity matrix may be checked to determine if entries corresponding to a co-occurrence present in the sequence dropped to zero. The entries for some or all of such sub-occurrences may be increased to a threshold value to revive the connection.

[0115] In one embodiment, the input patterns to the HTM network include natural breaks. These breaks may be leveraged to increase likelihood of correctly learning the temporal relationship between the sub-occurrences. In order to take advantage of the breaks, the following modification is made to the process: (i) add a beginning sub-occurrence and an ending sub-occurrence, and do not connect the added sub-occurrences to any other sub-occurrences initially; (ii) change the initialization time of activity variable α at time 0 (before training stage) to time 1 and the activity variable of all other sub-occurrences to zero; (iii) truncate the training data at the break; (iv) append an input to this range of data that will activate only the end sub-occurrences ($e_{t,end}^- = 1, e_{t,\dots}^- = 0$); (v) perform training based on the truncated data set; and (vi) repeat the process for the remaining sets of data truncated by the breaks.

[0116] In one embodiment, the temporal pooler updates the temporal statistics data while performing the inference. The data for inference received during the inference stage may be processed at the HTM network so that the temporal pooler updates the connectivity matrix M after the inference data is received at the HTM network. The temporal pooler may then group sequences based on the updated connectivity matrix to update the temporal statistics data.

[0117] In one embodiment, the entries in the connectivity matrix corresponding to sub-occurrences that do not appear frequently during the training stage are reduced or set to zero. For example, the entries may be set to zero when the sub-occurrences corresponding to the entries are not active for a predetermined amount of time. Alternatively, the entries may be gradually decreased as the sub-occurrence is not active in a next sequence. By setting the values or reducing the values

of the entries, unlearning or forgetting of temporal sequences may be implemented. Similar mechanism may be used for input patterns received during the inference stage. If a co-occurrence or sub-occurrence is not activated for a predetermined amount of time, the connectivity matrix may be updated to reduce corresponding entries or set the values to zero. The updated connectivity matrix may then be processed to group or partition sequences and the connections thereof. This embodiment is advantageous to accommodate cases where the input data change and the Markov model created from previous input data becomes inapplicable.

[0118] In one embodiment, two or more connectivity matrices are generated and maintained by the HOT temporal pooler. Each connectivity matrix may represent transition for different time increments. For example, one connectivity matrix may tally transition counts for every time steps while another connectivity matrix may tally transition counts for every other transitions. Different groups of co-occurrences may be generated from each connectivity matrix. Alternatively, the transition counts from multiple connectivity matrices may be combined to generate groups of co-occurrences.

[0119] While particular embodiments and applications of the present invention have been illustrated and described herein, it is to be understood that the invention is not limited to the precise construction and components disclosed herein and that various modifications, changes, and variations may be made in the arrangement, operation, and details of the methods and apparatuses of the present invention without departing from the spirit and scope of the invention as it is defined in the appended claims.

What is claimed is:

1. A node in a computer-implemented Hierarchical Temporal Memory network for generating an output representing an object or a state of an object corresponding to input patterns, the node comprising:

a temporal pooler configured to generate temporal statistics data representing a higher-order Markov chain of temporal sequences of the spatial co-occurrences based on temporal relationships of the spatial co-occurrences of the first input patterns in the training stage; and

an inference engine configured to generate an output based on the temporal statistics data responsive to receiving a sequence of spatial co-occurrence information about second input patterns in an inference stage subsequent to the training stage, the output representing probabilities of the sequence of the spatial co-occurrence information corresponding to temporal sequences of the temporal statistics data.

2. The node of claim 1, wherein the temporal pooler is further configured to split a spatial co-occurrence into a plurality of sub-occurrences, each sub-occurrence assigned to represent the spatial co-occurrence in a different temporal sequence.

3. The node of claim 2, wherein the temporal pooler is further configured to store temporal relationships of the sub-occurrences in a connectivity matrix, each entry in the connectivity matrix representing frequency or transition counts from a first sub-occurrence to a second sub-occurrence, the first sub-occurrence split from a first spatial co-occurrence, and the second sub-occurrence split from a second spatial co-occurrence.

4. The node of claim 2, wherein the temporal pooler is configured to generate the temporal statistics data by receiving a single-round of the sequence of the spatial co-occurrence information.

5. The node of claim 2, wherein the temporal pooler comprises:

a window buffer for storing the sequence of the spatial co-occurrence information for a predetermined time period;

a connectivity matrix updater configured to determine transition from a first sub-occurrence at a first time to a second sub-occurrence at a second time subsequent to the first time based on the spatial co-occurrence information at a third time subsequent to the second time received from the window buffer;

a connectivity matrix store configured to store counts or frequency of the transition from the first occurrence to the second sub-occurrence; and

a sequence grouper configured to group sequences of co-occurrences based on the connectivity matrix.

6. The node of claim 5, wherein the connectivity matrix updater is configured to determine the transition based on a modified connectivity matrix generated by adding values to entries of the connectivity matrix to represent added likelihood of transitions between sub-occurrences not appearing in the connectivity matrix.

7. The node of claim 5, wherein the connectivity matrix updater is configured to determine the transition based on an activity variable representing probability of receiving a co-occurrence at the first time and previous co-occurrences before the first time according to the temporal relationships represented by the connectivity matrix.

8. The node of claim 1, wherein the temporal pooler is configured to generate the temporal statistics data by receiving a plurality rounds of the sequence of the spatial co-occurrence information.

9. The node of claim 7, wherein the temporal pooler comprises:

a mapper configured to map spatial co-occurrences of the first input patterns at a second time to sub-occurrences based on spatial co-occurrences of the first input patterns at a first time preceding the second time;

a connectivity matrix counter for adding counts to a connectivity matrix, the connectivity matrix configured to store counts or frequency of the transition from a first co-occurrence or a first sub-occurrence to a second co-occurrence or a second sub-occurrence;

a splitter configured to expand the connectivity matrix counter by selectively splitting a co-occurrence in the connectivity matrix, the splitter configured to update the mapper based on the splitting of the co-occurrence; and

a sequence grouper configured to group sequences of co-occurrences based on the connectivity matrix.

10. A computer-implemented method of determining an object or a state of an object represented by input patterns in a node of a Hierarchical Temporal Memory network, comprising:

generating information about spatial co-occurrences of first input patterns received by the node based on spatial similarity of the first input patterns in a training stage, generating temporal statistics data representing a higher-order Markov chain of temporal sequences of the spatial co-occurrences based on temporal relationships of the spatial co-occurrences in the training stage; and

generating an output based on the temporal statistics data responsive to receiving a sequence of spatial co-occurrence information about second input patterns, the output representing probabilities of the sequence of the spatial co-occurrence information corresponding to temporal sequences of the temporal statistics data in an inference stage subsequent to the training stage.

11. The method of claim **10**, further comprising splitting a spatial co-occurrence into a plurality of sub-occurrences, each sub-occurrence assigned to represent the spatial co-occurrence in a different temporal sequence of the higher-order Markov chain.

12. The method of claim **11**, further comprising storing temporal relationships of the sub-occurrences in a connectivity matrix, each entry in the connectivity matrix representing frequency or transition counts from a first sub-occurrence to a second sub-occurrence, the first sub-occurrence split from a first spatial co-occurrence, and the second sub-occurrence split from a second spatial co-occurrence.

13. The method of claim **11**, further comprising generating the temporal statistics data by receiving a single round of the sequence of the spatial co-occurrence information.

14. The method of claim **11**, further comprising:

storing the sequence of the spatial co-occurrence information for a predetermined time period;

determining transition from a first sub-occurrence at a first time to a second sub-occurrence at a second time subsequent to the first time based on the spatial co-occurrence information at a third time subsequent to the second time received from the window buffer;

storing configured to store counts or frequency of the transition from the first occurrence to the second sub-occurrence; and

grouping sequences of co-occurrences based on the connectivity matrix.

15. The method of claim **14**, wherein determining the transition comprises:

generating a modified connectivity matrix by adding values to entries of the connectivity matrix to represent added likelihood of transitions between sub-occurrences not appearing in the connectivity matrix;

computing an activity variable representing probability of receiving a co-occurrence at the first time and previous co-occurrences before the first time according to the temporal relationships represented by the connectivity matrix; and

determining the transition based on the modified connectivity matrix and the activity variable.

16. The method of claim **10**, wherein the temporal statistics data is generated by receiving a plurality rounds of probabilities of co-occurrences.

17. The method of claim **16**, further comprising:

mapping spatial co-occurrences of the first input patterns at a second time to sub-occurrences based on spatial co-occurrences of the first input patterns at a first time preceding the second time;

adding counts to a connectivity matrix, the connectivity matrix configured to store counts or frequency of the transition from a first co-occurrence or a first sub-occurrence to a second co-occurrence or a second sub-occurrence;

expanding the connectivity matrix counter by selectively splitting a co-occurrence in the connectivity matrix, the splitter configured to update the mapper based on the splitting of the co-occurrence; and

grouping sequences of co-occurrences based on the connectivity matrix.

18. A computer program storage medium storing computer instructions to operate a Hierarchical Temporal Memory network on a computer, the computer instructions when executed cause a processor in the computer to:

generate information about spatial co-occurrences of first input patterns received by a node in the Hierarchical Temporal Memory system based on spatial similarity of the first input patterns in a training stage,

generate temporal statistics data representing a higher-order Markov chain of temporal sequences of the spatial co-occurrences based on temporal relationships of the spatial co-occurrences in the training stage; and

generate an output based on the temporal statistics data responsive to receiving a sequence of spatial co-occurrence information about second input patterns, the output representing probabilities of the sequence of the spatial co-occurrence information corresponding to temporal sequences of the temporal statistics data in an inference stage subsequent to the training stage.

19. The computer program storage medium of claim **18**, further comprising instructions to split a spatial co-occurrence into a plurality of sub-occurrences, each sub-occurrence assigned to represent the spatial co-occurrence in a different temporal sequence of the higher-order Markov chain.

20. The computer program storage medium of claim **18**, further comprising instructions to:

store the sequence of the spatial co-occurrence information for a predetermined time period;

determine transition from a first sub-occurrence at a first time to a second sub-occurrence at a second time subsequent to the first time based on the spatial co-occurrence information at a third time subsequent to the second time received from the window buffer;

store configured to store counts or frequency of the transition from the first occurrence to the second sub-occurrence; and

group sequences of co-occurrences based on the connectivity matrix.

* * * * *