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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
02 January 2020 (02.01.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/005599 A1

(51) International Patent Classification:
G06N 3/04 (2006.01) *G06N 5/00* (2006.01)
G06N 3/08 (2006.01)

(21) International Application Number:
PCT/US2019/037408

(22) International Filing Date:
17 June 2019 (17.06.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
201810716540.8 29 June 2018 (29.06.2018) CN

(71) Applicant: **MICROSOFT TECHNOLOGY LICENSING, LLC** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **LIU, Weiqing**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **BIAN, Jiang**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **LIU, Tie-Yan**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) Agent: **MINHAS, Sandip S.** et al.; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

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

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) **Title:** TREND PREDICTION BASED ON NEURAL NETWORK

(57) Abstract: Implementations of the present disclosure relate to trend prediction based on neural network. In some implementations, a method comprises: obtaining time series data associated with an object; computing a vector representation of the time series data with a recurrent neural network; determining, based on the vector representation of the times series data, a probability that the object is associated with a predefined label, wherein the label indicates a class of a trend in which the object varies over time; and generating, based on the probability, a prediction for the trend in which the object varies over time.

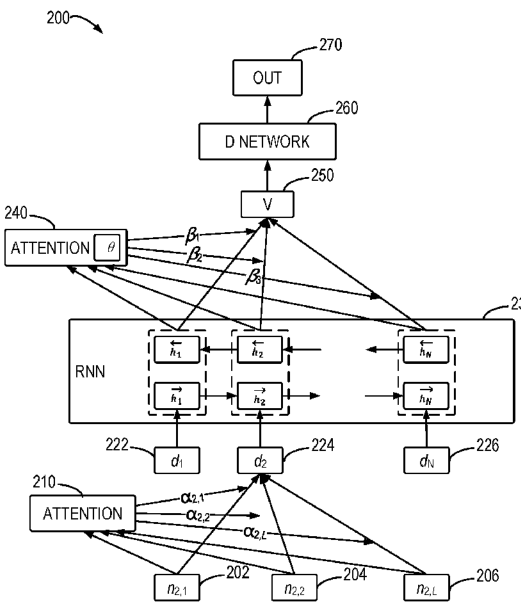

Fig. 2

WO 2020/005599 A1

[Continued on next page]

Declarations under Rule 4.17:

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

Published:

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

TREND PREDICTION BASED ON NEURAL NETWORK

BACKGROUND

[0001] In some trend predicting applications, data associated with one or more objects may be abrupt and time-relevant. For example, news data associated with epidemiology and monitoring data within a large-scale data center may have the above characteristics. Therefore, it becomes quite difficult to predict the object trend based on such data. Furthermore, with the development of the network, information sources are becoming more and more abundant and data are becoming more and more diverse, which makes it difficult to determine information quality and reliability. Hence, it is desirable to provide an improved trend prediction solution.

SUMMARY

[0002] Various implementations of the present disclosure provide a solution for trend prediction based on a neural network. In some implementations, a method comprises: obtaining time series data associated with an object; computing a vector representation of the time series data with a recurrent neural network; determining, based on the vector representation of the times series data, a probability that the object is associated with a predefined label, wherein the label indicates a class of a trend in which the object varies over time; and generating, based on the probability, a prediction for the trend in which the object varies over time.

[0003] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] Fig. 1 is a block diagram illustrating a computing device for implementing various implementations of the present disclosure;

[0005] Figs. 2 illustrates an architecture of a neural network in accordance with some implementations of the present disclosure;

[0006] Fig. 3 is a schematic diagram illustrating a recurrent neural network in accordance with some implementations of the present disclosure;

[0007] Fig. 4 is a flowchart illustrating a method for predicting a trend of an object in accordance with some implementations of the present disclosure; and

[0008] Fig. 5 is a flowchart illustrating a method for training a neural network in accordance with an implementation of the present disclosure.

[0009] Throughout the drawings, the same or similar reference symbols refer to the same or similar elements.

DETAILED DESCRIPTION

[0010] The subject matter described herein will now be discussed with reference to several example implementations. It is to be understood these implementations are discussed only for the purpose of enabling those skilled persons in the art to better understand and thus implement the subject matter described herein, rather than suggesting any limitations on the scope of the subject matter.

[0011] As used herein, the term "includes" and its variants are to be read as open terms that mean "includes, but is not limited to." The term "based on" is to be read as "based at least in part on." The term "one implementation" and "an implementation" are to be read as "at least one implementation." The term "another implementation" is to be read as "at least one other implementation." The terms "first," "second," and the like may refer to different or same objects. Other definitions, explicit and implicit, may be included below.

Example Environment

[0012] Basic principles and several example implementations of the present disclosure will be explained below with reference to the drawings. Fig. 1 illustrates a block diagram of a computing device 100 that can carry out a plurality of implementations of the present disclosure. It should be understood that the computing device 100 shown in Fig. 1 is only exemplary and shall not constitute any restrictions over functions and scopes of the implementations described by the present disclosure. According to Fig. 1, the computing device 100 includes a computing device 100 in the form of a general purpose computing device. Components of the computing device 100 can include, but not limited to, one or more processors or processing units 110, memory 120, storage device 130, one or more communication units 140, one or more input devices 150 and one or more output devices 160.

[0013] In some implementations, the computing device 100 can be implemented as various user terminals or service terminals with computing power. The service terminals can be servers, large-scale computing devices and the like provided by a variety of service providers. The user terminal, for example, is mobile terminal, fixed terminal or portable terminal of any types, including mobile phone, site, unit, device, multimedia computer, multimedia tablet, Internet nodes, communicator, desktop computer, laptop computer, notebook computer, netbook computer, tablet computer, Personal Communication System

(PCS) device, personal navigation device, Personal Digital Assistant (PDA), audio/video player, digital camera/video, positioning device, television receiver, radio broadcast receiver, electronic book device, gaming device or any other combinations thereof consisting of accessories and peripherals of these devices or any other combinations thereof. It can also be predicted that the computing device 100 can support any types of user-specific interfaces (such as “wearable” circuit and the like).

[0014] The processing unit 110 can be a physical or virtual processor and can execute various processing based on the programs stored in the memory 120. In a multi-processor system, a plurality of processing units executes computer-executable instructions in parallel to enhance parallel processing capability of the computing device 100. The processing unit 110 also can be known as central processing unit (CPU), microprocessor, controller and microcontroller.

[0015] The computing device 100 usually includes a plurality of computer storage media. Such media can be any attainable media accessible by the computing device 100, including but not limited to volatile and non-volatile media, removable and non-removable media. The memory 120 can be a volatile memory (e.g., register, cache, Random Access Memory (RAM)), a non-volatile memory (such as, Read-Only Memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), flash), or any combinations thereof. The memory 120 can include a predicting module 122 configured to execute functions of various implementations described herein. The predicting module 122 can be accessed and operated by the processing unit 110 to perform corresponding functions.

[0016] The storage device 130 can be removable or non-removable medium, and can include machine readable medium, which can be used for storing information and/or data and can be accessed within the computing device 100. The computing device 100 can further include a further removable/non-removable, volatile/non-volatile storage medium. Although not shown in Fig. 1, there can be provided a disk drive for reading from or writing into a removable and non-volatile disk and an optical disk drive for reading from or writing into a removable and non-volatile optical disk. In such cases, each drive can be connected via one or more data medium interfaces to the bus (not shown).

[0017] The communication unit 140 implements communication with another computing device through communication media. Additionally, functions of components of the computing device 100 can be realized by a single computing cluster or a plurality of computing machines, and these computing machines can communicate through communication connections. Therefore, the computing device 100 can be operated in a networked environment using a logic connection to one or more other servers, a Personal Computer (PC) or a further general network node.

[0018] The input device 150 can be one or more various input devices, such as mouse, keyboard, trackball, voice-input device and the like. The output device 160 can be one or more output devices, e.g., display, loudspeaker and printer etc. The computing device 100 also can communicate through the communication unit 140 with one or more external devices (not shown) as required, wherein the external device, e.g., storage device, display device etc., communicates with one or more devices that enable the users to interact with the computing device 100, or with any devices (such as network card, modem and the like) that enable the computing device 100 to communicate with one or more other computing devices. Such communication can be executed via Input/Output (I/O) interface (not shown).

[0019] The computing device 100 can be provided for implementing trend prediction for one or more objects in accordance with implementations of the present disclosure, for example, the object increases, reduces, or remains stable. The computing device 100, when performing the prediction, can receive time series data 170 associated with the one or more objects through the input device 150. The computing device 100 can process the time series data 170 and determine the trend of the object based on the time series data 170. The trend of the object can be provided to the output device 160 as an output 180 for the user and the like.

[0020] Fig. 2 is a schematic diagram illustrating a neural network 200 for predicting a trend of an object in accordance with some implementations of the present disclosure. As shown in Fig. 2, the neural network 200 includes a Recurrent Neural Network (RNN) 230, which can comprise one or more recurrent neural network units, such as Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM) unit and/or the like. At each time step, the recurrent neural network unit can receive the input data and latent state of the previous time step and determine the latent state of this time step based on the input data of this time step and the latent state of the previous time step. In a unidirectional recurrent neural network, the latent state of the last time step usually serves as an output provided to the subsequent network.

[0021] For example, the neural network 200 can obtain the time series data associated with one or more objects. The time series data can be data in the text form, such as news etc., and also can be data in other forms, e.g., numbers. For example, the time series data can include news at different time and from different sources.

[0022] In some implementations, data items such as news can be converted through an embedding layer into corresponding vector representations, which are provided to the neural network 200 to determine a vector representation (e.g., vector V 250 shown in Fig. 2) of the time series data (also known as a sample for training). The vector representation can have the same dimension as the latent state of the recurrent neural network 230. A number of words can be selected from one data item such as news to determine a vector representation of each word through the embedding layer, and these vector representations are averaged to determine the vector representation of the data item. For example, vector representations $n_{2,1}$ 202, $n_{2,2}$ 204 and $n_{2,L}$ 206 of several data items are illustrated as examples in Fig. 2.

[0023] As described above, the recurrent neural network 230 can be used for determining the vector representation of the obtained time series data. For example, the vector representation can be the vector V 250 as shown in Fig. 2. A discriminative network 260 can determine, based on the vector V 250, a probability that an object is associated with a predefined label as an output 270, which is provided to facilitate prediction of the in which the object varies over time. In addition or alternatively, the discriminative network 260 can output a label with the maximum probability as the prediction for the object trend. The label can indicate a class of trend in which the object varies over time, for example, up, down, stable, and/or the like.

[0024] In some implementations, the time series data can be segmented or divided into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network 230 as shown in Fig. 2. For example, the time series data is segmented into these portions based on a predefined time period. Each input of the recurrent neural network 230 can correspond to one time step, such as one day or one week. Therefore, each portion of the time series data can include a plurality of different data items within the respective time step. As shown in Fig. 2, a plurality of different data items $n_{2,1}$ 202, $n_{2,2}$ 204 and $n_{2,L}$ 206 are illustrated as examples when the time step t=2.

[0025] In some implementations, different data items such as different news within each time step or time period show different importance for prediction. For example, some data items can greatly affect the prediction result while some data items have less influence on the prediction result. In order to consider this factor, an attention layer 210 can determine weights of different data items for each time step and weight the respective data items with the weights to determine the weighted data item associated with the time step. The weighted data item can be provided to a respective input of the recurrent neural network 230.

[0026] Equation (1) illustrates a mathematical expression for determining the weighted data item in accordance with some implementations of the present disclosure. The vector representation n_{ti} of the data item i at the time step t is provided to the attention layer 210 to obtain a respective attention value u_{ti}, where sigmoid($\cdot$) represents a sigmoid function and W_n and b_n denote a parameter matrix and a parameter vector, respectively. The attention value u_{ti} is normalized through the *softmax* function to obtain a weight α_{ti}. The weight α_{ti} can weight the vector representation n_{ti} of the respective data item to determine the weighted data item d_t at the time step t. The input vectors over N time steps can be denoted as $D = [d_i], i \in [1, N]$.

$$
\begin{aligned}
u_{ti} &= sigmoid(W_n n_{ti} + b_n) \\
\alpha_{ti} &= \frac{exp(u_{ti})}{\sum_j exp(u_{tj})} \qquad (1) \\
d_t &= \sum_i \alpha_{ti} n_{ti}
\end{aligned}
$$

[0027] As an example, as shown in Fig. 2, when the time step t=2, the attention layer 210 can determine weights $\alpha_{2,1}$, $\alpha_{2,2}$, and $\alpha_{2,L}$ (also known as importance or attention) for the data items $n_{2,1}$ 202, $n_{2,2}$ 204, and $n_{2,L}$ 206. The weights $\alpha_{2,1}$, $\alpha_{2,2}$ and $\alpha_{2,L}$ can weight the respective data items $n_{2,1}$ 202, $n_{2,2}$ 204, and $n_{2,L}$ 206 to determine the weighted data item d_2 224 associated with the time step t=2. The weighted data item d_2 224 can be provided to the respective input of the recurrent neural network 230. In a similar way, respective weighted data items d_1 222, ……, d_N 226 can be obtained for the time steps t=1, ……, t=N.

[0028] In some implementations, the recurrent neural network 230 can be a bidirectional recurrent neural network. Fig. 3 illustrates a recurrent neural network 230 in accordance with some implementations of the present disclosure. The recurrent neural network units 314, 324, and 334 in the recurrent neural network 230 can receive, at each time step, respective input data and the latent state at the previous time step. For example, at the time step t=2, the recurrent neural network unit 324 receives, from the recurrent neural network unit 314, the latent state $\overrightarrow{h_1}$ at the previous time step and respective input data (e.g., weighted data item d_2) and determines the latent state $\overrightarrow{h_2}$ at the time step based on the latent state $\overrightarrow{h_1}$ and the input data (e.g., weighted data item d_2). Likewise, at the time step t=N, the recurrent neural network unit 334 receives, from the recurrent neural network unit at the time step t=N-1, the latent state $\overrightarrow{h_{N-1}}$ and respective input data (e.g., weighted data item d_N) and determines the latent state $\overrightarrow{h_N}$ at the time step t=N based on the latent state $\overrightarrow{h_{N-1}}$ and the input data (e.g., weighted data item d_N).

[0029] In addition, the recurrent neural network units 312, 322 and 332 in the recurrent neural network 230 can receive, at each time step, the respective input data and the latent state at the next time step. For example, at the time step t=1, the recurrent neural network unit 312 receives, from the recurrent neural network unit 322, the latent state $\overleftarrow{h_2}$ at the next time step and respective input data (e.g., weighted data item d_1) and determines the latent state $\overleftarrow{h_1}$ at the time step based on the latent state $\overleftarrow{h_2}$ and the input data (e.g., weighted data item d_1). Similarly, at the time step t=N-1, the recurrent neural network unit can receive, from the recurrent neural network unit 332 at the time step t=N, the latent state $\overleftarrow{h_N}$ and the respective input data (e.g., weighted data item d_{N-1}) and determine, based on the latent state $\overleftarrow{h_N}$ and the input data (e.g., weighted data item d_{N-1}), the latent state $\overleftarrow{h_{N-1}}$ at the time step t=N-1.

[0030] In Fig. 3, at the time step t=1, the recurrent neural network unit 310 is shown to include respective recurrent neural network units 312 and 314; at the time step t=2, the recurrent neural network unit 320 is shown to include respective recurrent neural network units 322 and 324; likewise, at the time step t=N, the recurrent neural network unit 330 is shown to include respective recurrent neural network units 332 and 334. However, it should be understood that the neural network unit at each time step can be the same neural network unit and the neural network unit is shown at each time step for the purpose of illustration only.

[0031] In some implementations, the recurrent neural network unit can be Gated Recurrent Unit (GRU). At the time step t, the GRU computes the state h_t at the time step by linearly interpolating the state h_{t-1} of the previous time step and the current updated state $\tilde{h_t}$ as shown in equation (2):

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (2)$$

where z_t represents an update gate for determining how much historical information should be kept and how much new information should be added and * denotes element-wise multiplication.

[0032] The updated current state $\tilde{h_t}$ can be computed by combining the input vector at the time step and the state at the previous time step as shown in equation (3):

$$\tilde{h}_t = tanh(W_h d_t + r_t * (U_h h_{t-1}) + b_h) \quad (3)$$

where r_t represents a reset gate for controlling how many historical states should be used to update the new state and W_h, U_h, and b_h denote parameter matrices and vectors.

[0033] The update gate z_t and the reset gate r_t can be computed through equation (4):

$$\begin{aligned} r_t &= \sigma(W_r d_t + U_r h_{t-1} + b_r) \\ z_t &= \sigma(W_z d_t + U_z h_{t-1} + b_z) \end{aligned} \quad (4)$$

where $\sigma(\cdot)$ indicates an activation function and W_r, U_r, b_r, W_z, U_z, and b_z denote parameter matrices and vectors.

[0034] At each time step, the latent vector for the time step can be obtained via the GRU. In order to obtain historical and future information for the data, the latent vectors in both directions can be concatenated to build a bidirectional encoded vector h_i as an output of the convolutional neural network 230 according to equation (5):

$$\begin{aligned} \overrightarrow{h_i} &= \overrightarrow{GRU}(d_i), i \in [1, L] \\ \overleftarrow{h_i} &= \overleftarrow{GRU}(d_i), i \in [L, 1] \quad (5) \\ h_i &= [\overrightarrow{h_i}, \overleftarrow{h_i}] \end{aligned}$$

[0035] Although the bidirectional recurrent neural network is introduced above with reference to Figs. 2 and 3, it should be appreciated that implementations of the present disclosure also can be applied in a uni-directional recurrent neural network. For example, the reverse recurrent neural network unit can be omitted and the latent state of each time step is provided for the subsequent neural network layers. Alternatively, the latent state at the last time step is provided for the subsequent neural network. Accordingly, the attention layer 240, which will be subsequently described in details, can be omitted in this implementation.

[0036] In some implementations, as shown in Fig. 2, the attention layer 240 can determine the weights of data items provided by a plurality of outputs of the recurrent neural network 230 and weight the corresponding data items using these weights. For example, more recent data may have a big influence over the prediction result while the less recent data may have a small influence over the prediction result. The data item provided by the output of the recurrent neural network 230 can be the bidirectional encoded vector h_i as mentioned above. The attention layer 240 considers the factor that the importance of the information varies for the trend determination at the respective time step. By means of the attention layer 240, the vector representation (vector V) of the time series data can be calculated via the equation (6):

$$
\begin{aligned}
o_i &= sigmoid(W_h h_i + b_h) \\
\beta_i &= \frac{exp(\theta_i o_i)}{\sum_j exp(\theta_j o_j)} \qquad (6) \\
V &= \sum_i \beta_i h_i
\end{aligned}
$$

where W_h and b_h represent parameter matrix and vector; θ_i denotes parameters at each time step in the *softmax* layer indicating which time step is more important; o_i indicates the latent representation of the bidirectional encoded vector h_i. By combining θ_i and o_i through the *softmax* layer, the attention vector β is acquired to weight the outputs of different time. Accordingly, the attention vector β is employed to weight the bidirectional encoded vector h_i to compute the vector representation V of the time series data for subsequent classification. As the vector representation V is a weighted value of the bidirectional encoded vector h_i, the vector representation V can have the same dimension as the latent state of the recurrent neural network. The dimension can be pre-specified.

[0037] The discriminative network 260 can be a classifier of Multi-Layer Perceptron (MLP), which receives the vector V as the input and outputs the classification result. The classification result can include a probability that the object is associated with the predefined label. For example, the label with the maximum probability can be output as the classification result.

[0038] In the training process, the training difficulty varies for different training data. For example, some training data lacks sufficient information and the training of such training data can be quite challenging. In some implementations, some challenging training data can be skipped and the training data which can be easily trained is selected and the challenging training data can be progressively added to the training process. For example, the neural network 200 can be trained using the self-paced learning, which can automatically choose the suitable training samples for different training phases, so as to enhance the model performance.

[0039] The self-paced learning process is introduced below with reference to several mathematical expressions. However, it should be understood that these mathematical expressions are provided as examples only, which enables those skilled in the art to better understand and implement implementations of the present disclosure without limiting the scope of the present disclosure. In some implementations, given a training set $D = (x_i, y_i)_{i=1}^{n}$, where $x_i \in R^m$ denotes all inputs in the i-th sample and y_i represents a corresponding trend label. $L(y_i, g(x_i, w))$ represents the loss function between the label y_i and the output $g(x_i, w)$ of the entire model. An importance weight v_i can be assigned to each sample x_i in the training set.

[0040] In some implementations, the goal of the self-paced learning is to jointly learn the model parameter w and the latent weight $v = [v_1, \ldots, v_n]$ via the equation (7):

$$\min_{w, v \in [0,1]^n} E(w, v, \lambda) = \sum_{i=1}^{n} v_i L(y_i, g(x_i, w)) + f(v; \lambda) \quad (7)$$

where $E(w, v, \lambda)$ represents an objective function, $f(v; \lambda)$ denotes a regularization term for self-paced learning (or known as penalty), which controls the learning solution for penalizing the latent weight variables and λ is a hyper-parameter that controls the pace at which the model learns new samples. For example, the regularization term $f(v; \lambda)$ can be computed using the equation (8):

$$f(v; \lambda) = \frac{1}{2} \lambda \sum_{i=1}^{n} (v_i^2 - 2v_i) \quad (8)$$

[0041] In some implementations, Alternative Convex Search (ACS) can be used to solve equation (7). ACS divides the variables into two disjoint blocks. In each iteration, a block of variables is optimized while the other block is fixed. With the fixed w, the unconstrained solution for the linear regularization term (8) can be computed by the equation (9):

$$v_i^* = argmin_v E(w^*, v, \lambda) = \begin{cases} -\frac{1}{\lambda} l_i + 1 & l_i < \lambda \\ 0 & l_i \geq \lambda \end{cases} \quad (9)$$

where v_i^* denotes the i-th element in the iterated optimal solution, and l_i denotes the loss for each element $L(y_i, g(x_i, w))$. The latent weight for samples that are different to what model has already learned will receive a linear penalty.

[0042] In some implementations, it is required that the self-paced learning inputs the data set D, the linear regularizer f and the self-paced step μ with the expectation of obtaining the model parameter w. In the initialization procedure, v_i^* can be initialized evenly and w^* and v^* can be iteratively updated in accordance with $w^* = argmin_w E(w, v^*, \lambda)$ and $v^* = argmin_v E(w^*, v, \lambda)$. In each iteration, λ, if being too small, can be increased based on the step μ. Finally, w^* can be output as the trained model parameter. As the training progresses and λ increases, samples with larger loss will be gradually incorporated, thereby realizing the effective and efficient learning.

[0043] Fig. 4 is a flowchart illustrating a method 400 for predicting the trend in accordance with some implementations of the present disclosure. The method 400 can be implemented by the computing device 100 or the predicting module 122 shown by Fig. 1. The concepts introduced above with reference to Figs. 2 and 3, alone or in combination, can be applied into the method 400. At 402, time series data associated with an object is obtained.

[0044] At 404, a vector representation of the time series data is computed with a recurrent neural network. In some implementations, computing the vector representation of the time series data includes: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network, a respective portion of the plurality of portions comprising data items from a plurality of sources; and determining the vector representation of the time series data based on relationship of importance between the data items in the respective portion. For example, the vector representation of the time series data can be determined by for the respective portion of the plurality of portions: determining weights of the data items in the respective portion; weighting the data items in the respective portion with the weights to determine the weighted data items associated with the respective portion; and providing the weighted data items to an input of the plurality of inputs associated with the respective portion.

[0045] In some implementations, computing the vector representation of the time series data includes: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network; and determining the vector representation of the time series data based on relationship of importance between the plurality of portions. For example, the recurrent neural network can be a bidirectional recurrent neural network, which also includes a plurality of outputs corresponding to a plurality of inputs. Determining the vector representation of the time series data includes determining weights of a plurality of data items provided by the plurality of outputs; and weighting the plurality of data items with the weights to determine the vector representation of the time series data.

[0046] At 406, a probability that the object is associated with a predefined label is determined based on the vector representation of the time series data. The label indicates a class of a trend in which the object varies over time. In this way, the prediction problem can be transformed into a classification problem.

[0047] At 408, a prediction for the trend in which the object varies over time is generated based on the probability. For example, if the probability associated with a certain label exceeds a predetermined threshold, it is considered that the trend of the object is identical to the label.

[0048] Fig. 5 is a flowchart illustrating a method 500 for training a neural network in accordance with some implementations of the present disclosure. The method 500 can be implemented by the computing device 100 or the predicting module 122 shown by Fig. 1. The concepts introduced above with reference to Figs. 2 and 3, alone or in combination, can be applied into the method 500.

[0049] At 502, the time series data associated with an object and a predefined label for the object are obtained. The label can indicate a class of a trend in which the object varies over time.

[0050] At 504, the recurrent neural network for predicting the trend in which the object varies over time is updated based on the time series data and the label. The updating comprises computing a vector representation of the time series data with the recurrent neural network; and determining, based on the vector representation of the time series data, a probability that the object is associated with the label.

[0051] In some implementations, computing the vector representation of the time series data comprises: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network, a respective portion of the plurality of portions comprising data items from a plurality of sources; and determining the vector representation of the time series data based on relationship of importance between the data items in the respective portion.

[0052] In some implementations, determining, based on the relationship of importance, the vector representation of the time series data comprises: for the respective portion of the plurality of portions: determining weights of the data items in the respective portion; weighting the data items in the respective portion with the weights to determine the weighted data items associated with the respective portion; and providing the weighted data items to an input of the plurality of inputs associated with the respective portion.

[0053] In some implementations, computing the vector representation of the time series data comprises: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network; and determining the vector representation of the time series data based on relationship of importance between the plurality of portions.

[0054] In some implementations, the recurrent neural network comprises a bidirectional recurrent neural network and the bidirectional recurrent neural network further comprises a plurality of outputs corresponding to the plurality of inputs, and wherein determining the vector representation of the time series data comprises: determining weights of a plurality of data items provided by the plurality of outputs; and weighting the plurality of data items with the weights to determine the vector representation of the time series data.

[0055] In some implementations, updating the recurrent neural network comprises updating the recurrent neural network through self-paced learning.

[0056] Some example implementations of the present disclosure are listed below.

[0057] In accordance with some implementations, there is provided a computer-implemented method. The method comprises: The method comprises obtaining time series data associated with an object; computing a vector representation of the time series data with a recurrent neural network; determining, based on the vector representation of the times series data, a probability that the object is associated with a predefined label, wherein the label indicates a class of a trend in which the object varies over time; and generating, based on the probability, a prediction for the trend in which the object varies over time.

[0058] In some implementations, computing the vector representation of the time series data comprises: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network, a respective portion of the plurality of portions comprising data items from a plurality of sources; and determining the vector representation of the time series data based on relationship of importance between the data items in the respective portion.

[0059] In some implementations, determining the vector representation of the time series data based on the relationship of importance comprises: for the respective portion of the plurality of portions: determining weights of the data items in the respective portion; weighting the data items in the respective portion with the weights to determine the weighted data items associated with the respective portion; and providing the weighted data items to an input of the plurality of inputs associated with the respective portion.

[0060] In some implementations, computing the vector representation of the time series data comprises: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network; and determining the vector representation of the time series data based on relationship of importance between the plurality of portions.

[0061] In some implementations, the recurrent neural network comprises a bidirectional recurrent neural network and the bidirectional recurrent neural network further comprises a plurality of outputs corresponding to the plurality of inputs, and wherein determining the vector representation of the time series data comprises: determining weights of a plurality of data items provided by the plurality of outputs; and weighting the plurality of data items with the weights to determine the vector representation of the time series data.

[0062] In accordance with some implementations, there is provided a computer-implemented method. The method comprises obtaining time series data associated with an object and a predefined label for the object, the label indicating a class of a trend that the object varies over time; and updating, based on the time series data and the label, a recurrent neural network for predicting the trend in which the object varies over time, the updating comprising: computing a vector representation of the time series data with the recurrent neural network; and determining, based on the vector representation of the time series data, a probability that the object is associated with the label.

[0063] In some implementations, computing the vector representation of the time series data comprises: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network, a respective portion of the plurality of portions comprising data items from a plurality of sources; and determining the vector representation of the time series data based on relationship of importance between the data items in the respective portion.

[0064] In some implementations, determining, based on the relationship of importance, the vector representation of the time series data comprises: for the respective portion of the plurality of portions: determining weights of the data items in the respective portion; weighting the data items in the respective portion with the weights to determine the weighted data items associated with the respective portion; and providing the weighted data items to an input of the plurality of inputs associated with the respective portion.

[0065] In some implementations, computing the vector representation of the time series data comprises: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network; and determining the vector representation of the time series data based on relationship of importance between the plurality of portions.

[0066] In some implementations, the recurrent neural network comprises a bidirectional recurrent neural network and the bidirectional recurrent neural network further comprises a plurality of outputs corresponding to the plurality of inputs, and wherein determining the vector representation of the time series data comprises: determining weights of a plurality of data items provided by the plurality of outputs; and weighting the plurality of data items with the weights to determine the vector representation of the time series data.

[0067] In some implementations, updating the recurrent neural network comprises updating the recurrent neural network through self-paced learning.

[0068] In accordance with some implementations, there is provided a device. The device comprises: a processing unit; and a memory coupled to the processing unit and comprising instructions stored thereon, the instructions, when executed by the processing unit, causing the device to perform acts comprising: obtaining time series data associated with an object; computing a vector representation of the time series data with a recurrent neural network; determining, based on the vector representation of the times series data, a probability that the object is associated with a predefined label, wherein the label indicates a class of a trend in which the object varies over time; and generating, based on the probability, a prediction for the trend in which the object varies over time.

[0069] In some implementations, computing the vector representation of the time series data comprises: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network, a respective portion of the plurality of portions comprising data items from a plurality of sources; and determining the vector representation of the time series data based on relationship of importance between the data items in the respective portion.

[0070] In some implementations, determining the vector representation of the time series data based on the relationship of importance comprises: for the respective portion of the plurality of portions: determining weights of the data items in the respective portion; weighting the data items in the respective portion with the weights to determine the weighted data items associated with the respective portion; and providing the weighted data items to an input of the plurality of inputs associated with the respective portion.

[0071] In some implementations, computing the vector representation of the time series data comprises: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network; and determining the vector representation of the time series data based on relationship of importance between the plurality of portions.

[0072] In some implementations, the recurrent neural network comprises a bidirectional recurrent neural network and the bidirectional recurrent neural network further comprises a plurality of outputs corresponding to the plurality of inputs, and wherein determining the vector representation of the time series data comprises: determining weights of a plurality of data items provided by the plurality of outputs; and weighting the plurality of data items with the weights to determine the vector representation of the time series data.

[0073] In accordance with some implementations, there is provided a device. The device comprises: a processing unit; and a memory coupled to the processing unit and comprising instructions stored thereon, the instructions, when executed by the processing unit, causing the device to perform acts comprising: obtaining time series data associated with an object and a predefined label for the object, the label indicating a class of a trend that the object varies over time; and updating, based on the time series data and the label, a recurrent neural network for predicting the trend in which the object varies over time, the updating comprising: computing a vector representation of the time series data with the recurrent neural network; and determining, based on the vector representation of the time series data, a probability that the object is associated with the label.

[0074] In some implementations, computing the vector representation of the time series data comprises: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network, a respective portion of the plurality of portions comprising data items from a plurality of sources; and determining the vector representation of the time series data based on relationship of importance between the data items in the respective portion.

[0075] In some implementations, determining, based on the relationship of importance, the vector representation of the time series data comprises: for the respective portion of the plurality of portions: determining weights of the data items in the respective portion; weighting the data items in the respective portion with the weights to determine the weighted data items associated with the respective portion; and providing the weighted data items to an input of the plurality of inputs associated with the respective portion.

[0076] In some implementations, computing the vector representation of the time series data comprises: segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network; and determining the vector representation of the time series data based on relationship of importance between the plurality of portions.

[0077] In some implementations, the recurrent neural network comprises a bidirectional recurrent neural network and the bidirectional recurrent neural network further comprises a plurality of outputs corresponding to the plurality of inputs, and wherein determining the vector representation of the time series data comprises: determining weights of a plurality of data items provided by the plurality of outputs; and weighting the plurality of data items with the weights to determine the vector representation of the time series data.

[0078] In some implementations, updating the recurrent neural network comprises updating the recurrent neural network through self-paced learning.

[0079] In accordance with some implementations, there is provided a computer program product comprising instructions stored tangibly on a computer-readable medium, the instructions, when executed by a machine, causing the machine to execute the method described above.

[0080] The functionality described herein can be performed, at least in part, by one or more hardware logic components. For example, and without limitation, illustrative types of hardware logic components that can be used include Field-Programmable Gate Arrays (FPGAs), Application-specific Integrated Circuits (ASICs), Application-specific Standard Products (ASSPs), System-on-a-chip systems (SOCs), Complex Programmable Logic Devices (CPLDs), and the like. Additionally, the functions described above herein can be at least partially executed by a Graphical Processing Unit (GPU).

[0081] Program code for carrying out methods of the present disclosure may be written in any combination of one or more programming languages. These program codes may be provided to a processor or controller of a general purpose computer, special purpose computer, or other programmable data processing apparatus, such that the program codes, when executed by the processor or controller, cause the functions/operations specified in the flowcharts and/or block diagrams to be implemented. The program code may execute entirely on a machine, partly on the machine, as a stand-alone software package, partly on the machine and partly on a remote machine or entirely on the remote machine or server.

[0082] In the context of this disclosure, a machine readable medium may be any tangible medium that may contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device. The machine readable medium may be a machine readable signal medium or a machine readable storage medium. A machine readable medium may include but not limited to an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples of the machine readable storage medium would include an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing.

[0083] Further, although operations are depicted in a particular order, it should be understood that the operations are required to be executed in the shown particular order or in a sequential order, or all shown operations are required to be executed to achieve the expected results. In certain circumstances, multitasking and parallel processing may be advantageous. Likewise, while several specific implementation details are contained in the above discussions, these should not be construed as limitations on the scope of the subject matter described herein. Certain features that are described in the context of separate implementations may also be implemented in combination in a single implementation. Conversely, various features that are described in the context of a single implementation may also be implemented in multiple implementations separately or in any suitable sub-combination.

[0084] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter specified in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

CLAIMS

1. A computer-implemented method, comprising:

obtaining time series data associated with an object;

computing a vector representation of the time series data with a recurrent neural network;

determining, based on the vector representation of the times series data, a probability that the object is associated with a predefined label, wherein the label indicates a class of a trend in which the object varies over time; and

generating, based on the probability, a prediction for the trend in which the object varies over time.

2. The method of claim 1, wherein computing the vector representation of the time series data comprises:

segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network, a respective portion of the plurality of portions comprising data items from a plurality of sources; and

determining the vector representation of the time series data based on relationship of importance between the data items in the respective portion.

3. The method of claim 2, wherein determining the vector representation of the time series data based on the relationship of importance comprises:

for the respective portion of the plurality of portions:

determining weights of the data items in the respective portion;

weighting the data items in the respective portion with the weights to determine the weighted data items associated with the respective portion; and

providing the weighted data items to an input of the plurality of inputs associated with the respective portion.

4. The method of claim 1, wherein computing the vector representation of the time series data comprises:

segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network; and

determining the vector representation of the time series data based on relationship of importance between the plurality of portions.

5. The method of claim 4, wherein the recurrent neural network comprises a bidirectional recurrent neural network and the bidirectional recurrent neural network further comprises a plurality of outputs corresponding to the plurality of inputs, and wherein

determining the vector representation of the time series data comprises:

determining weights of a plurality of data items provided by the plurality of outputs; and

weighting the plurality of data items with the weights to determine the vector representation of the time series data.

6. A computer-implemented method, comprising:

obtaining time series data associated with an object and a predefined label for the object, the label indicating a class of a trend that the object varies over time; and

updating, based on the time series data and the label, a recurrent neural network for predicting the trend in which the object varies over time, the updating comprising:

computing a vector representation of the time series data with the recurrent neural network; and

determining, based on the vector representation of the time series data, a probability that the object is associated with the label.

7. The method of claim 6, wherein computing the vector representation of the time series data comprises:

segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network, a respective portion of the plurality of portions comprising data items from a plurality of sources; and

determining the vector representation of the time series data based on relationship of importance between the data items in the respective portion.

8. The method of claim 7, wherein determining, based on the relationship of importance, the vector representation of the time series data comprises:

for the respective portion of the plurality of portions:

determining weights of the data items in the respective portion;

weighting the data items in the respective portion with the weights to determine the weighted data items associated with the respective portion; and

providing the weighted data items to an input of the plurality of inputs associated with the respective portion.

9. The method of claim 6, wherein computing the vector representation of the time series data comprises:

segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network; and

determining the vector representation of the time series data based on relationship of importance between the plurality of portions.

10. The method of claim 9, wherein the recurrent neural network comprises a bidirectional recurrent neural network and the bidirectional recurrent neural network further comprises a plurality of outputs corresponding to the plurality of inputs, and wherein determining the vector representation of the time series data comprises:

determining weights of a plurality of data items provided by the plurality of outputs; and

weighting the plurality of data items with the weights to determine the vector representation of the time series data.

11. The method of claim 6, wherein updating the recurrent neural network comprises updating the recurrent neural network through self-paced learning.

12. A device, comprising:

a processing unit; and

a memory coupled to the processing unit and comprising instructions stored thereon, the instructions, when executed by the processing unit, causing the device to perform acts comprising:

obtaining time series data associated with an object and a predefined label for the object, the label indicating a class of a trend in which the object varies over time; and

updating, based on the time series data and the label, a recurrent neural network for predicting the trend in which the object varies over time, the updating comprising:

computing a vector representation of the time series data with the recurrent neural network; and

determining, based on the vector representation of the time series data, a probability that the object is associated with the label.

13. The device of claim 12, wherein computing the vector representation of the time series data comprises:

segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network, a respective portion of the plurality of portions comprising data items from a plurality of sources; and

determining the vector representation of the time series data based on relationship of importance between the data items in the respective portion.

14. The device of claim 12, wherein computing the vector representation of the time series data comprises:

segmenting the time series data into a plurality of portions corresponding to a plurality of inputs of the recurrent neural network; and

determining the vector representation of the time series data based on relationship of importance between the plurality of portions.

15. The device of claim 12, wherein updating the recurrent neural network comprises updating the recurrent neural network through self-paced learning.

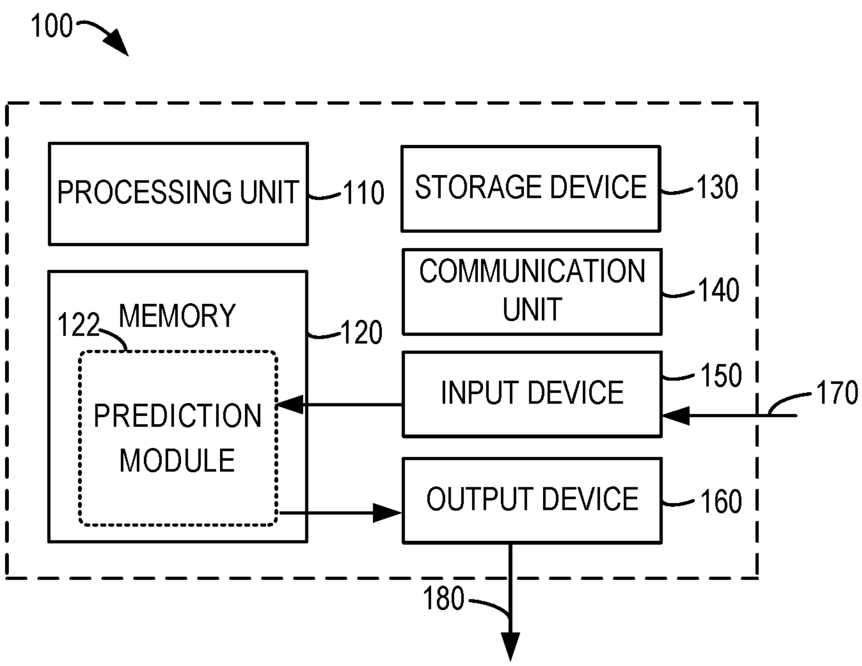
100
PROCESSING UNIT
110
MEMORY
122
PREDICTION MODULE
120
STORAGE DEVICE
130
COMMUNICATION UNIT
140
INPUT DEVICE
150
170
OUTPUT DEVICE
160
180

Fig. 1

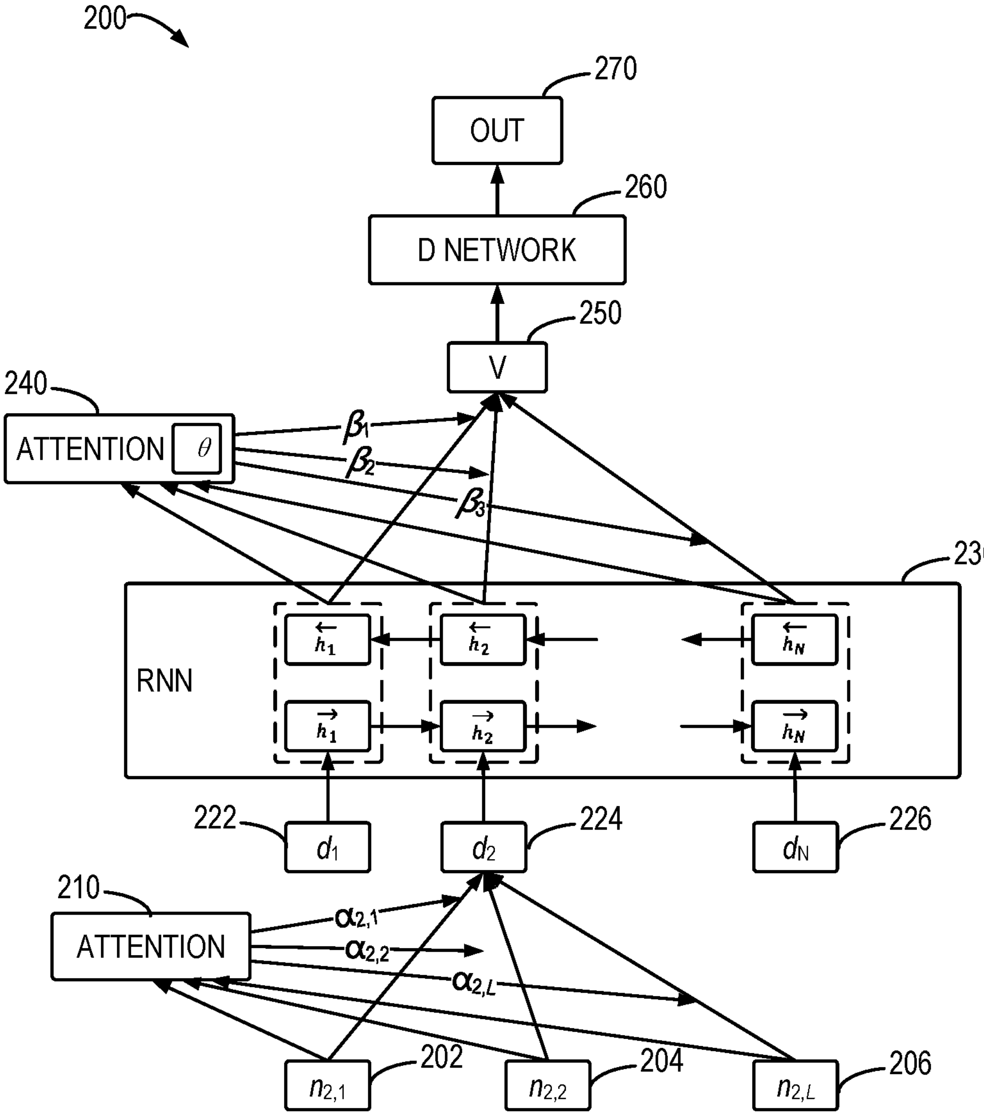
200
270
OUT
260
D NETWORK
250
V
240
ATTENTION
θ
β1
β2
β3
230
RNN
222
224
226
d1
d2
dN
210
ATTENTION
α2,1
α2,2
α2,L
n2,1
n2,2
n2,L
202
204
206

Fig. 2

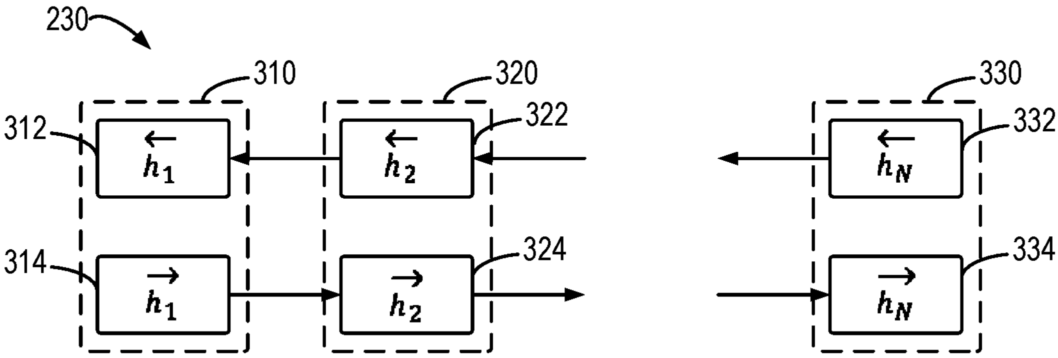
230
310
320
330
312
322
332
314
324
334
$\overleftarrow{h_1}$
$\overleftarrow{h_2}$
$\overleftarrow{h_N}$
$\overrightarrow{h_1}$
$\overrightarrow{h_2}$
$\overrightarrow{h_N}$

Fig. 3

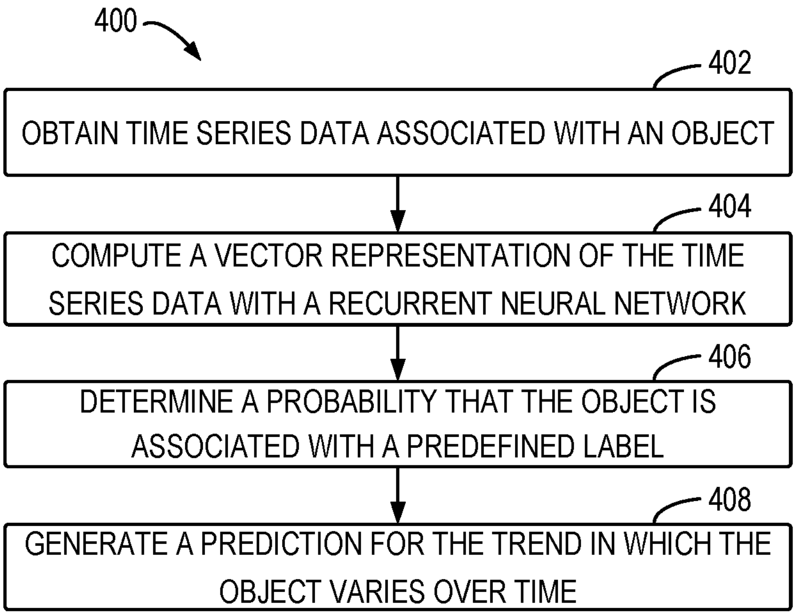
400
402
OBTAIN TIME SERIES DATA ASSOCIATED WITH AN OBJECT
404
COMPUTE A VECTOR REPRESENTATION OF THE TIME SERIES DATA WITH A RECURRENT NEURAL NETWORK
406
DETERMINE A PROBABILITY THAT THE OBJECT IS ASSOCIATED WITH A PREDEFINED LABEL
408
GENERATE A PREDICTION FOR THE TREND IN WHICH THE OBJECT VARIES OVER TIME

Fig. 4

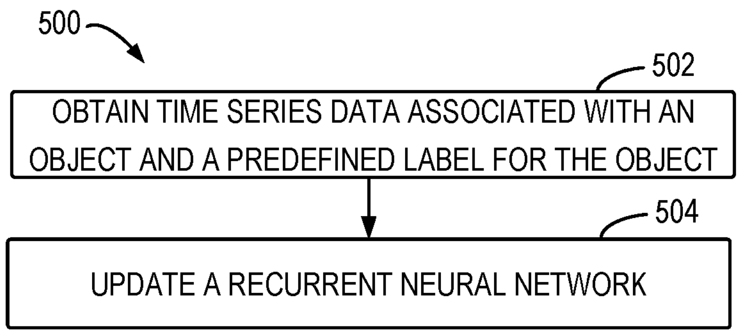
500
502
OBTAIN TIME SERIES DATA ASSOCIATED WITH AN OBJECT AND A PREDEFINED LABEL FOR THE OBJECT
504
UPDATE A RECURRENT NEURAL NETWORK

Fig. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/037408

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06N3/04 G06N3/08
ADD. G06N5/00

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

B. FIELDS SEARCHED

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

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ZINIU HU ET AL: "Listening to Chaotic Whispers : A Deep Learning Framework for News-oriented Stock Trend Prediction", WEB SEARCH AND DATA MINING, 6 December 2017 (2017-12-06), pages 261-269, XP055621582, 2 Penn Plaza, Suite 701New YorkNY10121-0701USA DOI: 10.1145/3159652.3159690 ISBN: 978-1-4503-5581-0 the whole document ----- -/--	1-15

[X] Further documents are listed in the continuation of Box C.

[ ] See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 September 2019	07/10/2019

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Moro Pérez, Gonzalo

2

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	AKITA RYO ET AL: "Deep learning for stock prediction using numerical and textual information", 2016 IEEE/ACIS 15TH INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION SCIENCE (ICIS), IEEE, 26 June 2016 (2016-06-26), pages 1-6, XP032948521, DOI: 10.1109/ICIS.2016.7550882 [retrieved on 2016-08-23] the whole document -----	1-15
X	YOSHIHARA AKIRA ET AL: "Predicting Stock Market Trends by Recurrent Deep Neural Networks", 1 December 2014 (2014-12-01), INTERNATIONAL CONFERENCE ON COMPUTER ANALYSIS OF IMAGES AND PATTERNS. CAIP 2017: COMPUTER ANALYSIS OF IMAGES AND PATTERNS; [LECTURE NOTES IN COMPUTER SCIENCE; LECT.NOTES COMPUTER], SPRINGER, BERLIN, HEIDELBERG, PAGE(S) 759 - 769, XP047311129, ISBN: 978-3-642-17318-9 the whole document -----	1-15
A	LU JIANG ET AL: "Self-Paced Curriculum Learning", PROCEEDINGS / EIGHTEENTH NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (AAAI-2002), FOURTEENTH INNOVATIVE APPLICATIONS OF ARTIFICIAL INTELLIGENCE CONFERENCE (IAAI-2002) : [JULY 28 - AUGUST 1, 2002, EDMONTON, ALBERTA, CANADA], 25 January 2015 (2015-01-25), pages 2694-2700, XP055475953, US ISBN: 978-0-262-51129-2 the whole document -----	1-15