great as possible, and a similarity of $\vec{w}$ and $\vec{c}'$ as small as possible. In the foregoing example, the similarity is measured through vector point multiplication, and the similarity may alternatively be measured in another manner.

Further, if the word vectors are trained by using a gradient descent method, the training the corresponding word vectors according to a designated loss function, a designated negative sample word, and the created word pair may specifically include: traversing the corpus obtained through reading; and according to the designated loss function, the designated negative sample word, and the created word pair, calculating a gradient according to the designated loss function, the designated negative sample word, and the created word pair, and iteratively updating word vectors of traversed current words and context words of the current words according to the gradient.

In an actual application, one or more threads on each worker computer may iteratively update the word vectors of the traversed current words and context words of the current words in a manner of asynchronous calculation without locking. Therefore, the threads on the worker computer may also perform updating in parallel without interfering with each other, thereby further improving training efficiency.

In the present embodiment of the present application, the gradient and the training result may vary with different loss functions and different activation functions. The foregoing example of the loss function is still used to describe calculation during training.

The calculating a gradient according to the designated loss function, the designated negative sample word, and the created word pair, and iteratively updating word vectors of traversed current words and context words of the current words according to the gradient may specifically include:

iteratively updating the word vectors of the traversed current words and context words of the current words according to the following formulas:

$\vec{w}_{i,t+1} = \vec{w}_{i,t} + g\vec{c}_{i,t}, w \in B_{i,k}$ (formula 1) and

$\vec{c}_{i,t+1} = \vec{c}_{i,t} + g\vec{w}_{i,t}, c \in \Gamma(w)$ (formula 2),

where $g=\alpha\left(y\text{-}\sigma\left(\vec{w}\cdot\vec{c}\right)\right)$, $y=\begin{cases}1,\{w,c\}\\0,\{w,c'\}\end{cases}$, w represents a current word, c represents a context word of w, c' represents a negative sample word, $\vec{w}$ represents a word vector of w, $\vec{c}$ represents a word vector of c, $\vec{w}_{i,t}$ and $\vec{c}_{i,t}$ represent a tth

update on an ith worker computer, $B_{i,k}$ represents a kth corpus on the ith worker computer, $\Gamma(w)$ represents a context word set of w, α represents a learning rate, and for example, may be set to 0.025, and σ is a sigmoid function, namely, $\sigma=\frac{1}{1+e^{-x}}$.

Further, calculation of the gradient is described as follows:

if $\nabla\sigma(z)|_z=\frac{1}{\sigma(z)}\cdot\sigma(z)\cdot(1-\sigma(z))=1-\sigma(z)$ and

$\nabla\sigma(-z)|_z=-(1-\sigma(-z))=-\sigma(z)$,

$\nabla L(w,c)|_{\vec{w}}=\left(y-\sigma\left(\vec{w}\cdot\vec{c}\right)\right)\vec{c}$ and

$\nabla L(w,c)|_{\vec{c}}=\left(y-\sigma\left(\vec{w}\cdot\vec{c}\right)\right)\vec{w}$.

In the present embodiment of the present application, when the server updates the word vectors according to the training results, step S208 of updating word vectors of same words according to training results of one or more of the worker computers with respect to the word vectors of the same words may specifically include: obtaining, by the server, the training results of one or more of the worker computers with respect to the word vectors of the same words; and performing average calculation according to the training results and the word vectors of the same words that are stored before training, to obtain vector increment values, and updating the word vectors of the same words according to the vector increment values.

The update process is a model averaging process. The average calculation may be implemented in a plurality of manners, for example, a quantity of occurrences of a word on each worker computer is used as a weight to average the training results of the worker computer; for another example, the training results of the worker computer are averaged directly; and so on. The former manner is used as an example, and the foregoing vector increment values may be obtained through calculation according to the following formula:

$$\Delta\left(\vec{w}\right)=\frac{\sum_{i=0}^{I}\lambda_i(w)\left(\vec{w}_{i,T}-\vec{w}_{srv}\right)}{\sum_{i=0}^{I}\lambda_i(w)} \quad \text{(formula 3) and}$$

$$\Delta\left(\vec{c}\right)=\frac{\sum_{i=0}^{I}\lambda_i(c)\left(\vec{c}_{i,T}-\vec{w}_{srv}\right)}{\sum_{i=0}^{I}\lambda_i(c)} \quad \text{(formula 4),}$$

$\vec{w}_{i,T}$ and $\vec{c}_{i,T}$ represent an iterative update result on an ith worker computer, $\lambda_i(w)$

represents a quantity of occurrences of w on the ith worker computer, and $\vec{w}_{srv}$ represents $\vec{w}$ stored by a respective server before the training.

A word vector before the update and the calculated corresponding vector increment value are added to obtain an updated word vector.

According to the foregoing description, an embodiment of the present application further provides a schematic principle diagram of a cluster-based word vector processing method in an actual application scenario, as shown in FIG. 3. Further, an embodiment of the present application further provides a detailed schematic flowchart of a cluster-based word vector processing method corresponding to FIG. 3, as shown in FIG. 4.

FIG. 3 exemplarily shows worker computers 0 to 2 and servers 0 to 2. Description is provided mainly for the worker computer 0. The worker computers 1 and 2 are briefly represented, and worker modes of the worker computers 1 and 2 are consistent with a worker mode of the worker computer 0. "wid" and "cid" are identifiers and respectively represent the current word and the context word. "wid list" and "cid list" are lists of identifiers and respectively represent the current word set and the context word set. A brief worker procedure in FIG. 3 includes: each worker computer reads corpuses in a distributed manner, and creates a word pair; each worker computer obtains corresponding word vectors from the server cluster; each worker computer trains the word vectors by using the read corpuses; and the server cluster performs model averaging according to training results of the worker computers.

FIG. 4 shows a more detailed procedure, which mainly includes the following steps:

S402: Each worker computer reads in a distributed manner a corpus or corpuses, creates a word pair $\{w,c\}$, and extracts wid list and cid list from the word pair, as shown by the worker computer 0 in FIG. 3.

S404: The worker computer obtains corresponding word vectors from the server cluster according to wid list and cid list.

S406: The worker computer calculates a gradient according to the word pair, then iteratively updates the word vectors, and specifically performs calculation by using the foregoing formula 1 and formula 2.

S408: After each worker computer performs iterative updating, the server cluster performs model averaging, to integrate iterative update results, and specifically performs calculation by using the foregoing formula 3 and formula 4.

Based on a same idea, an embodiment of the present application further provides an

to the word and the context words of the word specifically includes:

the training module 502 trains the corresponding word vectors according to a designated loss function, a designated negative sample word, and each word pair created by the training module 502.

Optionally, that the training module 502 trains the corresponding word vectors according to the designated loss function, the designated negative sample word, and each word pair created by the training module 502 specifically includes:

the training module 502 traverses the corpus obtained through reading; and

calculates a gradient according to the designated loss function, the designated negative sample word, and each word pair created by the training module 502, and iteratively updates word vectors of traversed current words and context words of the current words according to the gradient.

Optionally, that the training module 502 calculates the gradient according to the designated loss function, the designated negative sample word, and each word pair created by the training module 502, and iteratively updates the word vectors of the traversed current words and context words of the current words according to the gradient specifically includes:

the training module 502 iteratively updates the word vectors of the traversed current words and context words of the current words according to the following formulas:

$\vec{w}_{i,t+1} = \vec{w}_{i,t} + g\vec{c}_{i,t}, w \in B_{i,k}$ and

$\vec{c}_{i,t+1} = \vec{c}_{i,t} + g\vec{w}_{i,t}, c \in \Gamma(w)$,

where $g = \alpha\left(y - \sigma\left(\vec{w} \cdot \vec{c}\right)\right)$, $y = \begin{cases} 1, \{w, c\} \\ 0, \{w, c'\} \end{cases}$, w represents a current word, c represents a context word of w, c' represents a negative sample word, $\vec{w}$ represents a word vector of w, $\vec{c}$ represents a word vector of c, $\vec{w}_{i,t}$ and $\vec{c}_{i,t}$ represent a tth update on an ith worker computer, $B_{i,k}$ represents a kth corpus on the ith worker computer, $\Gamma(w)$ represents a context word set of w, α represents a learning rate, and σ is a sigmoid function.

Optionally, that the training module 502 iteratively updates the word vectors of the traversed current words and context words of the current words specifically includes:

the training module 502 iteratively updates, by using one or more threads on the worker computer, the word vectors of the traversed current words and context words of the current

CLAIMS

What is claimed is:

1. A cluster-based word vector processing method performed at a cluster comprising a plurality of worker computers,

the method comprising operations performed by each of the worker computers:

obtaining a word and context words of the word that are extracted from a corpus;

obtaining word vectors for the word and the context words of the word; and

training the word vectors according to the word and the context words of the word; and

the method comprising updating word vectors of same words according to training results of one or more of the worker computers with respect to the word vectors of the same words.

2. The method according to claim 1, wherein before the obtaining a word and context words of the word that are extracted from a corpus, the method further comprises:

obtaining, by each of the worker computers, a respective portion of the corpus through distributed reading; and

the obtaining a word and context words of the word that are extracted from a corpus comprises:

creating a corresponding word pair according to the respective portion of the corpus obtained through the distributed reading, wherein the word pair comprises a current word and a context word of the current word.

3. The method according to claim 2, wherein the cluster further comprises a plurality of servers, and the obtaining word vectors for the word and the context words of the word comprises:

obtaining a current word set and a context word set through extraction according to the created word pair; and

obtaining word vectors for words included in the current word set and the context word set from a respective server.

4. The method according to claim 2, wherein the training the word vectors according to the word and the context words of the word comprises:

training the word vectors according to a designated loss function, a designated negative sample word, and the created word pair.

5. The method according to claim 4, wherein the training the word vectors according to a designated loss function, a designated negative sample word, and the created word pair comprises:

traversing the corpus obtained through reading; and

calculating a gradient according to the designated loss function, the designated negative sample word, and the created word pair, and iteratively updating word vectors of traversed current words and context words of the current words according to the gradient.

6. The method according to claim 5, wherein the calculating a gradient according to the designated loss function, the designated negative sample word, and the created word pair, and iteratively updating word vectors of traversed current words and context words of the current words according to the gradient comprises:

iteratively updating the word vectors of the traversed current words and context words of the current words according to the following formulas:

$\vec{w}_{i,t+1} = \vec{w}_{i,t} + g\vec{c}_{i,t}, w \in B_{i,k}$ and

$\vec{c}_{i,t+1} = \vec{c}_{i,t} + g\vec{w}_{i,t}, c \in \Gamma(w)$,

wherein $g = \alpha\left(y - \sigma\left(\vec{w} \cdot \vec{c}\right)\right)$, $y = \begin{cases} 1, \{w, c\} \\ 0, \{w, c'\} \end{cases}$, w represents a current word, c represents a context word of w, c' represents a negative sample word, $\vec{w}$ represents a word vector of w, $\vec{c}$ represents a word vector of c, $\vec{w}_{i,t}$ and $\vec{c}_{i,t}$ represent a tth update on an ith worker computer, $B_{i,k}$ represents a kth corpus on the ith worker computer, $\Gamma(w)$ represents a context word set of w, α represents a learning rate, and σ is a sigmoid function.

7. The method according to claim 6, wherein the iteratively updating the word vectors of the traversed current words and context words of the current words comprises:

iteratively updating, by using one or more threads on a respective worker computer, the word vectors of the traversed current words and context words of the current words in a

manner of asynchronous calculation without locking.

8. The method according to claim 3, wherein the updating word vectors of same words according to training results of one or more of the worker computers with respect to the word vectors of the same words comprises:

obtaining, by the respective server, the training results of the one or more of the worker computers with respect to the word vectors of the same words; and

performing average calculation according to the training results and the word vectors of the same words that are stored before the training, to obtain vector increment values, and updating the word vectors of the same words according to the vector increment values.

9. The method according to claim 8, wherein the performing average calculation according to the training results and the word vectors of the same words that are stored before the training, to obtain vector increment values comprises:

obtaining the vector increment values through calculation according to the following formulas:

$$\Delta\left(\vec{w}\right)=\frac{\sum_{i=0}^{I}\lambda_i\left(w\right)\left(\vec{w}_{i,T}-\vec{w}_{srv}\right)}{\sum_{i=0}^{I}\lambda_i\left(w\right)} \text{ and}$$

$$\Delta\left(\vec{c}\right)=\frac{\sum_{i=0}^{I}\lambda_i\left(c\right)\left(\vec{c}_{i,T}-\vec{w}_{srv}\right)}{\sum_{i=0}^{I}\lambda_i\left(c\right)},$$

wherein w represents a current word, c represents a context word of w, $\vec{w}$ represents a word vector of w, $\vec{c}$ represents a word vector of c, $\vec{w}_{i,T}$ and $\vec{c}_{i,T}$ represent an iterative update result on an ith worker computer, $\lambda_i\left(w\right)$ represents a quantity of occurrences of w on the ith worker computer, and $\vec{w}_{srv}$ represents $\vec{w}$ stored by a respective server before the training.

10. A cluster-based word vector processing apparatus at a cluster, wherein the cluster comprises a plurality of worker computers, and the apparatus comprises an integration and update module, and a training module that is located in each of the worker computers;

wherein the training module of each of the worker computers separately performs operations including:

obtaining a word and context words of the word that are extracted from a corpus;

obtaining word vectors for the word and the context words of the word; and

training the word vectors according to the word and the context words of the word; and

wherein the integration and update module updates word vectors of same words according to training results of one or more of the worker computers with respect to the word vectors of the same words.

11. The apparatus according to claim 10, wherein each of the worker computers further comprises a reading module, and before the training module obtains the word and the context words of the word that are extracted from the portion of the corpus, the reading module of each of the worker computers obtains a respective portion of the corpus through distributed reading; and

the training module obtains the word and the context words of the word that are extracted from a corpus by:

creating a corresponding word pair according to the respective portion of the corpus obtained by the reading module of the respective worker computer in which the training module is located, wherein the word pair comprises a current word and a context word of the current word.

12. The apparatus according to claim 11, wherein the cluster further comprises a plurality of servers, and the training module obtains the word vectors for the word and the context words of the word by:

obtaining a current word set and a context word set through extraction according to each word pair created by the training module; and

obtaining word vectors for words included in the current word set and the context word set from a respective server.

13. The apparatus according to claim 11, wherein the training module trains the word vectors according to the word and the context words of the word by:

training the word vectors according to a designated loss function, a designated negative sample word, and the word pair created by the training module.

14. The apparatus according to claim 13, wherein the training module trains the word vectors according to the designated loss function, the designated negative sample word, and

the word pair created by the training module by:

traversing the corpus obtained through reading; and

calculating a gradient according to the designated loss function, the designated negative sample word, and the word pair created by the training module, and iteratively updating word vectors of traversed current words and context words of the current words according to the gradient.

15. The apparatus according to claim 14, wherein the training module calculates the gradient according to the designated loss function, the designated negative sample word, and the word pair created by the training module, and iteratively updates the word vectors of the traversed current words and context words of the current words according to the gradient by:

iteratively updating the word vectors of the traversed current words and context words of the current words according to the following formulas:

$\vec{w}_{i,t+1} = \vec{w}_{i,t} + g\vec{c}_{i,t}, w \in B_{i,k}$ and

$\vec{c}_{i,t+1} = \vec{c}_{i,t} + g\vec{w}_{i,t}, c \in \Gamma(w)$,

wherein $g = \alpha\left(y - \sigma\left(\vec{w} \cdot \vec{c}\right)\right)$, $y = \begin{cases} 1, \{w, c\} \\ 0, \{w, c'\} \end{cases}$, w represents a current word, c represents a context word of w, c' represents a negative sample word, $\vec{w}$ represents a word vector of w, $\vec{c}$ represents a word vector of c, $\vec{w}_{i,t}$ and $\vec{c}_{i,t}$ represent a tth update on an ith worker computer, $B_{i,k}$ represents a kth corpus on the ith worker computer, $\Gamma(w)$ represents a context word set of w, α represents a learning rate, and σ is a sigmoid function.

16. The apparatus according to claim 15, wherein the training module iteratively updates the word vectors of the traversed current words and context words of the current words by:

iteratively updating, by using one or more threads on a respective worker computer, the word vectors of the traversed current words and context words of the current words in a manner of asynchronous calculation without locking.

17. The apparatus according to claim 12, wherein the integration and update module is located in a respective server, and the integration and update module updates the word

vectors of the same words according to training results of one or more of worker computers with respect to the word vectors of the same words by:

obtaining the training results of the one or more of the worker computers with respect to of the word vectors of the same words; and

performing average calculation according to the training results and the word vectors of the same words that are stored before the training, to obtain vector increment values, and updating the word vectors of the same words according to the vector increment values.

18. The apparatus according to claim 17, wherein the integration and update module performs average calculation according to the training results and the word vectors of the same words that are stored before the training, to obtain the vector increment values by:

obtaining the vector increment values through calculation according to the following formulas:

$$\Delta\left(\vec{w}\right)=\frac{\sum_{i=0}^{I}\lambda_i\left(w\right)\left(\vec{w}_{i,T}-\vec{w}_{srv}\right)}{\sum_{i=0}^{I}\lambda_i\left(w\right)} \text{ and}$$

$$\Delta\left(\vec{c}\right)=\frac{\sum_{i=0}^{I}\lambda_i\left(c\right)\left(\vec{c}_{i,T}-\vec{w}_{srv}\right)}{\sum_{i=0}^{I}\lambda_i\left(c\right)},$$

wherein w represents a current word, c represents a context word of w, $\vec{w}$ represents a word vector of w, $\vec{c}$ represents a word vector of c, $\vec{w}_{i,T}$ and $\vec{c}_{i,T}$ represent an iterative update result on an ith worker computer, $\lambda_i\left(w\right)$ represents a quantity of occurrences of w on the ith worker computer, and $\vec{w}_{srv}$ represents $\vec{w}$ stored by a respective server before the training.

19. A cluster-based word vector processing device, wherein the device belongs to a cluster and comprises:

at least one processor; and

a memory in communication connection with the at least one processor, wherein

the memory stores an instruction executable by the at least one processor to cause the at least one processor to:

obtain a word and context words of the word that are extracted from a corpus;

obtain word vectors for the word and the context words of the word;

train the word vectors according to the word and the context words of the word; and

update word vectors of same words according to one or more respective training results of the word vectors of the same words.