

FEDERATED TRAINING METHOD AND APPARATUS, AND COMPUTER SYSTEM

TECHNICAL FIELD

[0001] One or more embodiments of this specification relate to the field of computer technologies, and in particular, to a federated training method and apparatus, and a computer system.

BACKGROUND

[0002] With the advent of the era of artificial intelligence and big data, machine learning technologies are widely applied to various fields. However, due to data security, privacy protection, and other problems, a conventional centralized machine learning method has many limitations. Therefore, federated learning emerges.

[0003] Federal learning (or referred to as federated training) is a distributed machine learning technology, and allows a plurality of participants to perform model training while data privacy is protected. Each participant locally maintains only a dataset of the participant, and exchanges some model updating parameters by using a federated learning algorithm, to implement joint model training. Federated learning is widely applied to fields such as the medical field, the financial field, and the internet of things filed because of characteristics of data privacy protection and confidentiality of federated learning.

[0004] Therefore, embodiments of this specification disclose a federated training solution, to better meet a requirement in actual applications, for example, improve generalization performance and availability of a model while protecting data privacy.

SUMMARY

[0005] One or more embodiments of this specification describe a federated training method and a corresponding apparatus, to meet a higher requirement for federated learning performance and effect in actual applications.

[0006] According to a first aspect, a federated training method is provided, and involves an intermediate party and n participants. The method is applied to the intermediate party, and any round of iterative training involved in the method includes: correspondingly receiving n embedding representations from the n participants, where each embedding representation is
5 obtained by a corresponding participant by processing a local feature corresponding to a current batch of samples by using a local model; and processing the n embedding representations by using a corresponding deployed first gating network for any first prediction task in m prediction tasks, to obtain n first weights; and processing, by using a corresponding deployed first prediction network, a weighted embedding representation obtained based on the n first weights and the n
10 embedding representations, to obtain a first prediction result, so as to determine a training loss in the current round.

[0007] According to a second aspect, a federated training apparatus is provided, and involves an intermediate party and n participants. The apparatus is integrated into the intermediate party, and the apparatus performs any round of iterative training by using the following modules: an
15 embedding representation receiving unit, configured to correspondingly receive n embedding representations from the n participants, where each embedding representation is obtained by a corresponding participant by processing a local feature corresponding to a current batch of samples by using a local model; a weight prediction unit, configured to process the n embedding representations by using a corresponding deployed first gating network for any first prediction task
20 in m prediction tasks, to obtain n first weights; and a result prediction unit, configured to process, by using a corresponding deployed first prediction network, a weighted embedding representation obtained based on the n first weights and the n embedding representations, to obtain a first prediction result, so as to determine a training loss in the current round.

[0008] According to a third aspect, a computer system is provided, and includes a memory, a
25 processor, and a computer program stored in the memory and capable of running on the processor. The processor is integrated into an intermediate party, and the intermediate party and a plurality of participants cooperatively perform federated training. When the processor executes the program, the following steps in any round of iterative training are implemented: correspondingly receiving
30 n embedding representations from the n participants, where each embedding representation is obtained by a corresponding participant by processing a local feature corresponding to a current batch of samples by using a local model; and processing the n embedding representations by using

a corresponding deployed first gating network for any first prediction task in m prediction tasks, to obtain n first weights; and processing, by using a corresponding deployed first prediction network, a weighted embedding representation obtained based on the n first weights and the n embedding representations, to obtain a first prediction result, so as to determine a training loss in the current round.

[0009] According to a fourth aspect, a computing device is provided, and includes a memory and a processor. The memory stores executable code, and when the processor executes the executable code, the method according to the first aspect is implemented.

[0010] According to the method and the apparatus provided in the embodiments of this specification, joint learning and model parameter sharing of data features of different tasks are implemented by sharing an underlying network structure and setting different gating networks configured to aggregate intermediate results for different tasks, thereby effectively improving a generalization capability of a model.

BRIEF DESCRIPTION OF DRAWINGS

[0011] To describe the technical solutions in the embodiments of the present invention more clearly, the following briefly describes the accompanying drawings needed for describing the embodiments. Clearly, the accompanying drawings in the following descriptions show merely some embodiments of the present invention, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0012] FIG. 1 is a schematic diagram of an implementation architecture of a federated training solution according to an embodiment of this specification;

[0013] FIG. 2 is a schematic diagram of communication and interaction between an intermediate party and n participants involved in a federated training method according to an embodiment of this specification;

[0014] FIG. 3 is a schematic diagram of a functional structure of a federated training apparatus integrated into an intermediate party according to an embodiment of this specification; and

[0015] FIG. 4 is a schematic structural diagram of a computer system according to an embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

[0016] The solutions provided in this specification are described below with reference to the accompanying drawings.

[0017] As described above, federated learning is widely applied to fields such as the medical
5 field, the financial field, and the internet of things field because of characteristics of data privacy protection and confidentiality of federated learning. However, in many application fields, participants in federated learning pay attention to different model tasks (or referred to as prediction tasks). For example, in the field of electronic payment, it is assumed that participants in federated learning are two electronic wallets. A service of a wallet A is mainly to provide collection and
10 payment services for a merchant. Therefore, a model task to which the wallet A pays attention is to detect fraud means such as illegal fund transfer. A service of a wallet B is mainly to provide transfer between individuals and payment services. Therefore, a model task to which the wallet B pays attention is to evaluate credit of a user.

[0018] Therefore, it is proposed to predict a plurality of related tasks in a federated learning
15 framework. Specifically, in the embodiments of this specification, attention is paid to multi-task learning in a vertical federated learning framework. It can be understood that in the vertical federated learning framework, different participants have respective datasets, and these datasets overlap in a sample dimension, but are independent of each other in a feature dimension, that is, different participants have same sample space and different feature space, that is, a plurality of
20 participants hold different feature parts of a same sample.

[0019] In conventional vertical federated learning, a plurality of participants (or referred to as a plurality of data parties) respectively process different feature parts of a same sample by using local models (bottom model), to obtain corresponding intermediate results, and transfer the intermediate results to an intermediate party. After aggregating the intermediate results, the
25 intermediate party continues to perform feedforward processing by using a top model deployed in the intermediate party, to obtain a prediction result for a global task, then calculates a training loss by using the prediction result and a corresponding sample label, and performs back propagation, so that the intermediate party and the plurality of participants can respectively update models deployed in the intermediate party and the plurality of participants.

[0020] To implement multi-task learning in the vertical federated learning framework, it is
30

considered to separately establish a federated learning procedure for each model task, to obtain a plurality of global models for a plurality of prediction tasks. For example, it is assumed that there are two participants and three prediction tasks. In this case, the two participants each deploy three local models for the three prediction tasks, and the intermediate party deploys three top models for the three prediction tasks. However, in this manner, when there is a relatively small amount of labeled data for each task, a generalization capability of a global model trained for each task is relatively weak, and a relatively large amount of communication traffic is generated overall.

[0021] Based on the above-mentioned observations and analysis, the applicant proposes a federated training solution. In this solution, each of a plurality of (denoted as n) participants needs to locally deploy only one local model, and an intermediate party configures m sets of gating networks and prediction networks for a plurality of (denoted as m) prediction tasks. Any j^{th} gating network is configured to determine weights of n intermediate results (or referred to as sample representations) uploaded by the n participants for a sample, and the weights are used to aggregate the n intermediate results. A j^{th} prediction network is configured to process an aggregation result to obtain a prediction result for a j^{th} task, to implement updating of a model of each party based on the prediction result and a corresponding sample label. In FIG. 1, an example in which $n=2$ and $m=2$ is used to show a solution architecture.

[0022] In this way, data of different related tasks can be simultaneously processed while privacy is protected, and joint learning and model parameter sharing of different task data features can be implemented by sharing an underlying network structure, thereby effectively improving a generalization capability of a model and reducing an amount of communication traffic generated during training.

[0023] It should be noted that the above-mentioned federated training solution has a wide range of application scenarios, and can be widely applied to scenarios in which a plurality of related tasks need to be processed and data privacy protection is involved. To facilitate understanding, the following first describes distribution of training data in a plurality of participants with reference to a specific application scenario, and then describes specific implementation steps of the federated training solution.

[0024] Scenario 1: In the smart transportation field, a plurality of participants jointly learn traffic data of a plurality of cities, and execute a traffic flow prediction task and a traffic accident prediction task.

[0025] Specifically, any training sample corresponds to a specific time point or time period, and sample feature parts respectively held by the plurality of participants are traffic data records at the time point or in the time period. For example, each participant can be instantiated as a server cluster that stores the traffic data of the corresponding cities, and the traffic data record can be
5 collected by using a traffic camera, a sensor, a GPS device, or the like.

[0026] Scenario 2: In the e-commerce field, a plurality of e-commerce platforms jointly learn user data on the platforms, to improve more accurate personalized commodity recommendation and advertisement push services.

[0027] Specifically, any training sample corresponds to a user, and the plurality of e-commerce
10 platforms respectively hold historical transaction data and the like of the user on the corresponding platforms. It should be noted that data related to the user in this specification is collected and used after the user confirms the authorization.

[0028] Scenario 3: In the intelligent financial field, different banks and financial organizations have rich customer data. Through multi-task vertical federated learning, the banks and financial
15 organizations can jointly train a model, to improve the accuracy and security of fraud detection tasks, credit scoring tasks, personalized financial services, and other services.

[0029] Specifically, any training sample corresponds to a customer (an individual or an organization), and the different banks or financial organizations respectively hold transaction records, loan records, investment and financial management records, and the like of the customer
20 in the banks or financial organizations.

[0030] Scenario 4: In the electronic payment field, participants are different electronic wallets. The participants may have different service types and service positioning, and have different focuses on risk control.

[0031] For example, a participant a and a participant b in FIG. 1 can be a wallet A and a wallet
25 B. Local data a of the wallet A is mainly service data generated by providing collection and payment services for a merchant, and local data b of the wallet B is mainly service data generated by providing transfer between individuals and payment services. The two wallets can cooperatively learn service data, to execute a task 1, that is, a fraud detection task, to which the wallet A pays attention and a task 2, that is, a credit scoring task, to which the wallet B pays
30 attention.

[0032] With reference to the four typical scenarios, the distribution of training data in a

plurality of participants is described above by using an example. For a sample label involved in the training data, a single training sample can have a sample label for any quantity of prediction tasks in m prediction tasks. In addition, there may be a plurality of distribution cases of sample labels in the plurality of participants. In a possible case, each participant holds a sample label
5 corresponding to a task to which the participants pays attention. For example, in the example of the scenario 4, the wallet A holds a sample label for the fraud detection task, and the wallet B holds a sample label for the credit scoring task. In another possible case, some participants have only sample features and no label, and sample labels for a multi-task scenario are distributed in other participants.

10 **[0033]** Specific content and distribution of training data in a multi-task federated learning scenario are described above. The specific implementation steps of the above-mentioned federated training solution are described below with reference to more accompanying drawings and embodiments.

[0034] FIG. 2 is a schematic diagram of communication and interaction between an
15 intermediate party and n participants involved in a federated training method according to an embodiment of this specification. It can be understood that the intermediate party is usually independent of the n participants, and this role can be played by an authoritative organization. Each of the intermediate party and the plurality of participants can be implemented as any apparatus, device, server, device cluster, or the like with computing and processing capabilities. In
20 addition, in FIG. 2, a P_i party is used to refer to any i^{th} participant in the n participants.

[0035] The federated training method involves a plurality of rounds of iterative training. As shown in FIG. 2, any round of iterative training includes the following interaction steps.

[0036] Step S21: The P_i party processes a local feature x_i corresponding to a current batch of samples by using a local model w_i , to obtain a corresponding embedding representation e_i .

25 **[0037]** It should be understood that the current batch of samples is a batch of samples sampled for the current round of training. There can be one or more samples, which are usually a plurality of samples, in a batch. There are a plurality of manners of determining the current batch of samples. For example, a specified participant in the plurality of participants performs random sampling on a training sample identifier set for the current round, and then sends an identifier subset obtained
30 through sampling to other parties. Therefore, the P_i party can locate a batch of training samples indicated by the identifier subset. For another example, the intermediate party performs random

sampling on a training sample identifier set for a plurality of times in advance, and sends an established mapping table between a training round and an identifier subset to the plurality of participants. Therefore, the Pi party can directly read a batch of sample identifiers corresponding to the current round from the mapping table.

5 [0038] In addition, a structure of the local model w_i can be independently designed and selected by the Pi party. For example, the local model w_i can be implemented as a deep learning model. Before a first round of iterative training, the Pi party can randomly initialize a parameter in the local model w_i .

[0039] Based on the above, the Pi party can input the local feature x_i to the local model w_i ,
10 to obtain the corresponding embedding representation e_i . It should be understood that if the current batch of samples includes only one sample, the embedding representation e_i is a representation vector of the sample. If the current batch of samples includes a plurality of samples, the embedding representation e_i is a representation matrix including a plurality of embedding representation vectors corresponding to the plurality of samples.

15 [0040] From the above, the PI can obtain the embedding representation e_i corresponding to the current batch of samples.

[0041] Step S22: The Pi party sends the embedding representation e_i to the intermediate party.

[0042] Step S23: The intermediate party processes, for any first prediction task T_j in m prediction tasks by using a locally deployed first gating network w_j^g , n embedding representations
20 $\{e_i\}_{[n]}$ received from the n participants, to obtain n first weights $\{\beta_{j,i}\}_{[n]}$.

[0043] First, it should be noted that "first" in "first prediction task" and "first gating network", and similar words in other parts of this specification are used to distinguish between objects of a same type, and do not have other limitation functions such as ranking.

[0044] The first prediction task T_j can also be referred to as a j^{th} prediction task. A superscript
25 g in the first gating network w_j^g is an initial letter of an English word gate corresponding to "gating", and a subscript j indicates that the network is a j^{th} gating network corresponding to the j^{th} prediction task. For example, the first gating network can be implemented as one or more fully connected networks.

[0045] In this step, fusion processing can be first performed on the n embedding
30 representations $\{e_i\}_{[n]}$ received from the n participants, to obtain a fusion representation e^f . It

should be understood that a subscript $[n]$ in $\{e_i\}_{[n]}$ is used to indicate that a value of i is a natural number from 1 to n , and a superscript f in e^f is an initial letter of an English word fusion. For example, the fusion processing can be concatenation processing, addition processing, averaging processing, or the like.

5 **[0046]** Further, the fusion representation e^f can be input to the first gating network w_j^g , to obtain the n first weights $\{\beta_{j_i}\}_{[n]}$. It can be understood that if the current batch of samples includes only one sample, the first weight is a scalar; or if the current batch of samples includes a plurality of samples, the first weight is a weight vector. In addition, a subscript $[n]$ in $\{\beta_{j_i}\}_{[n]}$ is used to indicate that a value of i is a natural number from 1 to n .

10 **[0047]** From the above, the intermediate party can obtain a set of weights of the n embedding representations $\{e_i\}_{[n]}$ for the prediction tasks by using m gating networks corresponding to the m prediction tasks.

[0048] Step S24: The intermediate party performs weighted summation processing on the n embedding representations $\{e_i\}_{[n]}$ by using the n first weights $\{\beta_{j_i}\}_{[n]}$, to obtain a first weighted
15 embedding representation e'_j .

[0049] The weighted summation processing in this step can be denoted as the following formula:

$$e'_j = \sum_{i=1}^n \beta_{j_i} * e_i \quad (1)$$

[0050] Therefore, the first weighted embedding representation e'_j corresponding to the first
20 prediction task T_j can be obtained, and then m weighted embedding representations $\{e'_j\}_{[m]}$ corresponding to the m prediction tasks can be obtained. It should be understood that a subscript $[m]$ in $\{e'_j\}_{[m]}$ is used to indicate that a value of j is a natural number from 1 to m .

[0051] Step S25: The intermediate party processes the first weighted embedding representation e'_j by using a locally deployed first prediction network w_j^s , to obtain a first
25 prediction result $\hat{y}_j$.

[0052] By analogy, the intermediate party can obtain m prediction results $\{\hat{y}_j\}_{[m]}$ corresponding to the m prediction tasks. For example, if the m prediction tasks include a fraud

detection task and a credit scoring task in the electronic payment field, the prediction result obtained by the intermediate party can include a predicted fraud probability and a predicted trustworthiness probability that correspond to each sample in the current batch of samples.

[0053] It should be understood that the m prediction results $\{\hat{y}_j\}_{[m]}$ are used in combination with corresponding m sample labels to determine the training loss L in the current round. Further, in a possible implementation, after step S25 is performed, the method can further include the following steps S26, S27, S28, and S29.

[0054] Step S26: The intermediate party determines the training loss L based on the m prediction results $\{\hat{y}_j\}_{[m]}$ and the corresponding m task labels $\{y_j\}_{[m]}$. It should be understood that the m task labels $\{y_j\}_{[m]}$ held by the intermediate party can be provided by the n participants.

[0055] Specifically, the intermediate party determines a corresponding first task loss l_j by using the prediction result $\hat{y}_j$ and a task label y_j corresponding to the first prediction task T_j . It should be noted that a loss function used to calculate the first task loss l_j can be independently configured by the intermediate party, or can be determined by a participant that pays attention to the first prediction task T_j or a participant providing the sample label for the first prediction task T_j .

[0056] From the above, m task losses $\{l_j\}_{[m]}$ corresponding to the m prediction tasks can be obtained. Further, in an embodiment, the intermediate party can perform weighted summation on the m task losses $\{l_j\}_{[m]}$, to obtain the training loss L . This can be denoted as the following formula:

$$L = \sum_{j=1}^m \alpha_j * l_j \quad (2)$$

[0057] Herein, α_j represents a weight configured for the first task loss l_j , and is a hyper-parameter. It should be understood that a value of α_j can be independently set by the intermediate party, or can be determined by the n participants through negotiation.

[0058] In another embodiment, the intermediate party can directly determine a sum value of the m task losses $\{l_j\}_{[m]}$ as the training loss L .

[0059] In another aspect, a regular term can alternatively be introduced into the training loss L , to prevent overfitting of model parameters.

[0060] It should be noted that when the intermediate party does not obtain the task label

provided by the participant, the intermediate party needs to provide the m prediction results $\{\hat{y}_j\}_{[m]}$ to one or more participants that hold the corresponding task label, so that the participant locally calculates the first task loss l_j . Therefore, the intermediate party can determine the training loss L based on the m task losses $\{l_j\}_{[m]}$ fed back by the participant.

5 **[0061]** From the above, the intermediate party can obtain the training loss L of the m prediction tasks corresponding to the current batch of samples.

[0062] Step S27: The intermediate party updates m prediction networks $\{w_j^s\}_{[m]}$ and the m gating networks $\{w_j^g\}_{[m]}$ for the m prediction tasks by using the training loss L , and calculates a back propagation gradient ∇_i to be back propagated to each participant.

10 **[0063]** Step S28: The intermediate party provides the back propagation gradient ∇_i to the corresponding participant P_i .

[0064] Step S29: The participant P_i updates the local model w_i by using the received back propagation gradient.

15 **[0065]** After the training loss L in the current round is calculated, back propagation can be enabled, so that the intermediate party and the n participants respectively complete updating of model parameters in the intermediate party and the participants, to complete iterative updating of each model in the current round.

[0066] In conclusion, according to the multi-task vertical federated training method disclosed in this embodiment of this specification, a model intermediate result instead of raw data is shared
20 between different tasks of participants, thereby implementing privacy protection. In addition, joint learning and model parameter sharing of data features of different tasks are implemented by sharing an underlying network structure and setting different gating networks configured to aggregate intermediate results for different tasks, thereby effectively improving a generalization capability of a model and reducing an amount of communication traffic in a training process.

25 **[0067]** According to an embodiment in another aspect, this embodiment of this specification further discloses a joint training apparatus. FIG. 3 is a schematic diagram of a functional structure of a federated training apparatus integrated into an intermediate party according to an embodiment of this specification. Federated training involves an intermediate party and n participants. The apparatus 300 is integrated into the intermediate party, and performs any round of iterative training
30 by using the following modules:

an embedding representation receiving unit 310, configured to correspondingly receive n embedding representations from the n participants, where each embedding representation is obtained by a corresponding participant by processing a local feature corresponding to a current batch of samples by using a local model; a weight prediction unit 320, configured to process the n embedding representations by using a corresponding deployed first gating network for any first prediction task in m prediction tasks, to obtain n first weights; and a result prediction unit 330, configured to process, by using a corresponding deployed first prediction network, a weighted embedding representation obtained based on the n first weights and the n embedding representations, to obtain a first prediction result, so as to determine a training loss in the current round.

[0068] In an embodiment, the weight prediction unit 320 is specifically configured to: perform fusion processing on the n embedding representations, to obtain a fusion representation; and process the fusion representation by using the first gating network, to obtain the n first weights.

[0069] In a specific embodiment, the fusion processing includes concatenation processing or addition processing.

[0070] In an embodiment, the result prediction unit 330 is specifically configured to: perform weighted summation on the n embedding representations by using the n first weights, to obtain the weighted embedding representation; and process the weighted embedding representation by using the first prediction network, to obtain the first prediction result.

[0071] In an embodiment, the apparatus 300 further includes a loss determining unit 340, configured to determine the training loss based on m prediction results for the m prediction tasks and corresponding m task labels, to update each model in the current round.

[0072] Further, in a specific embodiment, the loss determining unit 340 is specifically configured to: determine a corresponding first task loss by using the prediction result and a task label corresponding to the first prediction task; and perform weighted summation on m task losses corresponding to the m prediction tasks, to obtain the training loss.

[0073] In a specific embodiment, a task label corresponding to the first prediction task is provided by one of the n participants, and a loss function used to calculate the first task loss is determined by the participant.

[0074] In a specific embodiment, the apparatus 300 further includes: a parameter updating unit 350, configured to: update m prediction networks and m gating networks for the m prediction tasks by using the training loss, and calculate a back propagation gradient to be back propagated to each

participant; and a gradient sending unit 360, configured to provide the back propagation gradient to the corresponding participant, so that the corresponding participant updates the local model by using the received back propagation gradient.

[0075] It should be understood that for ease of description, the above apparatus is described by dividing functions into modules. Certainly, during implementation of this specification, functions of the modules can be implemented in same one or more pieces of software and/or hardware.

[0076] The specification also provides a computer-readable non-transitory storage medium, where the storage medium stores a computer program, which, when executed by a processor, can be used to perform one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Herein, the computer-readable non-transitory storage medium or media may include one or more semiconductor-based or other integrated circuits (ICs) (such, as for example, field-programmable gate arrays (FPGAs) or application-specific ICs (ASICs)), hard disk drives (HDDs), hybrid hard drives (HHDs), optical discs, optical disc drives (ODDs), magneto-optical discs, magneto-optical drives, floppy diskettes, floppy disk drives (FDDs), magnetic tapes, solid-state drives (SSDs), RAM-drives, SECURE DIGITAL cards or drives, any other suitable computer-readable non-transitory storage media, or any suitable combination of two or more of these, where appropriate. A computer-readable non-transitory storage medium may be volatile, non-volatile, or a combination of volatile and non-volatile, where appropriate.

[0077] The specification also provides a computer system. FIG. 4 illustrates an example computer system 400. In particular embodiments, one or more computer systems 400 perform one or more steps of one or more methods described or illustrated herein. In particular embodiments, one or more computer systems 400 provide functionality described or illustrated herein. In particular embodiments, software running on one or more computer systems 400 performs one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Particular embodiments include one or more portions of one or more computer systems 400. Herein, reference to a computer system may encompass a computing device, and vice versa, where appropriate. Moreover, reference to a computer system may encompass one or more computer systems, where appropriate.

[0078] This disclosure contemplates any suitable number of computer systems 400. This

disclosure contemplates computer system 400 taking any suitable physical form. As example and not by way of limitation, computer system 400 may be an embedded computer system, a system-on-chip (SOC), a single-board computer system (SBC) (such as, for example, a computer-on-module (COM) or system-on-module (SOM)), a desktop computer system, a laptop or notebook computer system, an interactive kiosk, a mainframe, a mesh of computer systems, a mobile telephone, a personal digital assistant (PDA), a server, a tablet computer system, or a combination of two or more of these. Where appropriate, computer system 400 may include one or more computer systems 400; be unitary or distributed; span multiple locations; span multiple machines; span multiple data centers; or reside in a cloud, which may include one or more cloud components in one or more networks. Where appropriate, one or more computer systems 400 may perform without substantial spatial or temporal limitation one or more steps of one or more methods described or illustrated herein. As an example and not by way of limitation, one or more computer systems 400 may perform in real time or in batch mode one or more steps of one or more methods described or illustrated herein. One or more computer systems 400 may perform at different times or at different locations one or more steps of one or more methods described or illustrated herein, where appropriate.

[0079] In particular embodiments, computer system 400 includes a processor 402, memory 404, storage 406, an input/output (I/O) interface 408, a communication interface 410, and a bus 412. Although this disclosure describes and illustrates a particular computer system having a particular number of particular components in a particular arrangement, this disclosure contemplates any suitable computer system having any suitable number of any suitable components in any suitable arrangement.

[0080] In particular embodiments, processor 402 includes hardware for executing instructions, such as those making up a computer program. As an example and not by way of limitation, to execute instructions, processor 402 may retrieve (or fetch) the instructions from an internal register, an internal cache, memory 404, or storage 406; decode and execute them; and then write one or more results to an internal register, an internal cache, memory 404, or storage 406. In particular embodiments, processor 402 may include one or more internal caches for data, instructions, or addresses. This disclosure contemplates processor 402 including any suitable number of any suitable internal caches, where appropriate. As an example and not by way of limitation, processor 402 may include one or more instruction caches, one or more data caches, and one or more

translation lookaside buffers (TLB s). Instructions in the instruction caches may be copies of instructions in memory 404 or storage 406, and the instruction caches may speed up retrieval of those instructions by processor 402. Data in the data caches may be copies of data of 40 in memory 404 or storage 406 for instructions executing at processor 402 to operate on; the results of previous
5 instructions executed at processor 402 for access by subsequent instructions executing at processor 402 or for writing to memory 404 or storage 406; or other suitable data. The data caches may speed up read or write operations by processor 402. The TLBs may speed up virtual-address translation for processor 402. In particular embodiments, processor 402 may include one or more internal registers for data, instructions, or addresses. This disclosure contemplates processor 402 including
10 any suitable number of any suitable internal registers, where appropriate. Where appropriate, processor 402 may include one or more arithmetic logic units (ALUs); be a multi-core processor; or include one or more processors 402. Although this disclosure describes and illustrates a particular processor, this disclosure contemplates any suitable processor.

[0081] In particular embodiments, memory 404 includes main memory for storing instructions
15 for processor 402 to execute or data for processor 402 to operate on. As an example and not by way of limitation, computer system 400 may load instructions from storage 406 or another source (such as, for example, another computer system 400) to memory 404. Processor 402 may then load the instructions from memory 404 to an internal register or internal cache. To execute the instructions, processor 402 may retrieve the instructions from the internal register or internal cache
20 and decode them. During or after execution of the instructions, processor 402 may write one or more results (which may be intermediate or final results) to the internal register or internal cache. Processor 402 may then write one or more of those results to memory 404. In particular embodiments, processor 402 executes only instructions in one or more internal registers or internal caches or in memory 404 (as opposed to storage 406 or elsewhere) and operates only on data in
25 one or more internal registers or internal caches or in memory 404 (as opposed to storage 406 or elsewhere). One or more memory buses (which may each include an address bus and a data bus) may couple processor 402 to memory 404. Bus 412 may include one or more memory buses, as described below. In particular embodiments, one or more memory management units (MMUs) reside between processor 402 and memory 404 and facilitate accesses to memory 404 requested
30 by processor 402. In particular embodiments, memory 404 includes random access memory (RAM). This RAM may be volatile memory, where appropriate. Where appropriate, this RAM

may be dynamic RAM (DRAM) or static RAM (SRAM). Moreover, where appropriate, this RAM may be single-ported or multi-ported RAM. This disclosure contemplates any suitable RAM. Memory 404 may include one or more memories 404, where appropriate. Although this disclosure describes and illustrates particular memory, this disclosure contemplates any suitable memory.

5 **[0082]** In particular embodiments, storage 406 includes mass storage for data or instructions. As an example and not by way of limitation, storage 406 may include a hard disk drive (HDD), a floppy disk drive, flash memory, an optical disc, a magneto-optical disc, magnetic tape, or a Universal Serial Bus (USB) drive or a combination of two or more of these. Storage 406 may include removable or non-removable (or fixed) media, where appropriate. Storage 406 may be
10 internal or external to computer system 400, where appropriate. In particular embodiments, storage 406 is non-volatile, solid-state memory. In particular embodiments, storage 406 includes read-only memory (ROM). Where appropriate, this ROM may be mask-programmed ROM, programmable ROM (PROM), erasable PROM (EPROM), electrically erasable PROM (EEPROM), electrically alterable ROM (EAROM), or flash memory or a combination of two or more of these. This
15 disclosure contemplates mass storage 406 taking any suitable physical form. Storage 406 may include one or more storage control units facilitating communication between processor 402 and storage 406, where appropriate. Where appropriate, storage 406 may include one or more storages 406. Although this disclosure describes and illustrates particular storage, this disclosure contemplates any suitable storage.

20 **[0083]** In particular embodiments, I/O interface 408 includes hardware, software, or both, providing one or more interfaces for communication between computer system 400 and one or more I/O devices. Computer system 400 may include one or more of these I/O devices, where appropriate. One or more of these I/O devices may enable communication between a person and computer system 400. As an example and not by way of limitation, an I/O device may include a
25 keyboard, keypad, microphone, monitor, mouse, printer, scanner, speaker, still camera, stylus, tablet, touch screen, trackball, video camera, another suitable I/O device or a combination of two or more of these. An I/O device may include one or more sensors. This disclosure contemplates any suitable I/O devices and any suitable I/O interfaces 408 for them. Where appropriate, I/O interface 408 may include one or more device or software drivers enabling processor 402 to drive
30 one or more of these I/O devices. I/O interface 408 may include one or more I/O interfaces 408, where appropriate. Although this disclosure describes and illustrates a particular I/O interface, this

disclosure contemplates any suitable I/O interface.

[0084] In particular embodiments, communication interface 410 includes hardware, software, or both providing one or more interfaces for communication (such as, for example, packet-based communication) between computer system 400 and one or more other computer systems 400 or one or more networks. As an example and not by way of limitation, communication interface 410 may include a network interface controller (NIC) or network adapter for communicating with an Ethernet or other wire-based network or a wireless NIC (WNIC) or wireless adapter for communicating with a wireless network, such as a WI-FI network. This disclosure contemplates any suitable network and any suitable communication interface 410 for it. As an example and not by way of limitation, computer system 400 may communicate with an ad hoc network, a personal area network (PAN), a local area network (LAN), a wide area network (WAN), a metropolitan area network (MAN), or one or more portions of the Internet or a combination of two or more of these. One or more portions of one or more of these networks may be wired or wireless. As an example, computer system 400 may communicate with a wireless PAN (WPAN) (such as, for example, a BLUETOOTH WPAN), a WI-FI network, a WI-MAX network, a cellular telephone network (such as, for example, a Global System for Mobile Communications (GSM) network), or other suitable wireless network or a combination of two or more of these. Computer system 400 may include any suitable communication interface 410 for any of these networks, where appropriate. Communication interface 410 may include one or more communication interfaces 410, where appropriate. Although this disclosure describes and illustrates a particular communication interface, this disclosure contemplates any suitable communication interface.

[0085] In particular embodiments, bus 412 includes hardware, software, or both coupling components of computer system 400 to each other. As an example and not by way of limitation, bus 412 may include an Accelerated Graphics Port (AGP) or other graphics bus, an Enhanced Industry Standard Architecture (EISA) bus, a front-side bus (FSB), a HYPERTRANSPORT (HT) interconnect, an Industry Standard Architecture (ISA) bus, an INFINIBAND interconnect, a low-pin-count (LPC) bus, a memory bus, a Micro Channel Architecture (MCA) bus, a Peripheral Component Interconnect (PCI) bus, a PCI-Express (PCIe) bus, a serial advanced technology attachment (SATA) bus, a Video Electronics Standards Association local (VLB) bus, or another suitable bus or a combination of two or more of these. Bus 412 may include one or more buses 412, where appropriate. Although this disclosure describes and illustrates a particular bus, this

disclosure contemplates any suitable bus or interconnect.

[0086] A person skilled in the art should be aware that in the above-mentioned one or more examples, the functions described in this application can be implemented by hardware, software, firmware, or any combination thereof. When being implemented by software, these functions can
5 be stored in a computer-readable medium or transmitted as one or more instructions or code on a computer-readable medium.

[0087] In the above-mentioned specific implementations, the objectives, technical solutions, and beneficial effects of the present invention are further described in detail. It should be understood that the above-mentioned descriptions are merely specific implementations of the
10 present invention, but are not intended to limit the protection scope of the present invention. Any modification, equivalent replacement, improvement, or the like made based on the technical solutions of the present invention shall fall within the protection scope of the present invention.

CLAIMS

What is claimed is:

1. A federated training method, involving an intermediate party and n participants, wherein the method is applied to the intermediate party, and any round of iterative training involved in the method comprises:

correspondingly receiving n embedding representations from the n participants, wherein each embedding representation is obtained by a corresponding participant by processing a local feature corresponding to a current batch of samples by using a local model; and

processing the n embedding representations by using a corresponding deployed first gating network for any first prediction task in m prediction tasks, to obtain n first weights; and processing, by using a corresponding deployed first prediction network, a weighted embedding representation obtained based on the n first weights and the n embedding representations, to obtain a first prediction result, so as to determine a training loss in the current round.

2. The method according to claim 1, wherein the processing the n embedding representations by using a corresponding deployed first gating network, to obtain n first weights comprises:

performing fusion processing on the n embedding representations, to obtain a fusion representation; and

processing the fusion representation by using the first gating network, to obtain the n first weights.

3. The method according to claim 2, wherein the fusion processing comprises concatenation processing or addition processing.

4. The method according to claim 1, wherein the processing, by using a corresponding deployed first prediction network, a weighted embedding representation obtained based on the n first weights and the n embedding representations, to obtain a first prediction result comprises:

performing weighted summation on the n embedding representations by using the n first weights, to obtain the weighted embedding representation; and

processing the weighted embedding representation by using the first prediction network, to obtain the first prediction result.

5. The method according to claim 1, wherein after the obtaining a first prediction result, the method further comprises:

determining the training loss based on m prediction results for the m prediction tasks and corresponding m task labels, to update each model in the current round.

6. The method according to claim 5, wherein the determining the training loss comprises:

5 determining a corresponding first task loss by using the prediction result and a task label corresponding to the first prediction task; and
performing weighted summation on m task losses corresponding to the m prediction tasks, to obtain the training loss.

7. The method according to claim 5, wherein a task label corresponding to the first prediction task is provided by one of the n participants, and a loss function used to calculate the first task loss
10 is determined by the participant.

8. The method according to claim 5, wherein after the determining the training loss, the method further comprises:

updating m prediction networks and m gating networks for the m prediction tasks by using the training loss, and calculating a back propagation gradient to be back propagated to each
15 participant; and

providing the back propagation gradient to the corresponding participant, so that the corresponding participant updates the local model by using the received back propagation gradient.

9. A federated training apparatus, involving an intermediate party and n participants, wherein the apparatus is integrated into the intermediate party, and the apparatus performs any round of
20 iterative training by using the following modules:

an embedding representation receiving unit, configured to correspondingly receive n embedding representations from the n participants, wherein each embedding representation is obtained by a corresponding participant by processing a local feature corresponding to a current batch of samples by using a local model;

25 a weight prediction unit, configured to process the n embedding representations by using a corresponding deployed first gating network for any first prediction task in m prediction tasks, to obtain n first weights; and

a result prediction unit, configured to process, by using a corresponding deployed first prediction network, a weighted embedding representation obtained based on the n first weights
30 and the n embedding representations, to obtain a first prediction result, so as to determine a training loss in the current round.

10. A computer system, comprising a memory, a processor, and a computer program stored in the memory and capable of running on the processor, wherein the processor is integrated into an intermediate party, the intermediate party and a plurality of participants cooperatively perform federated training, and when the processor executes the program, the following steps in any round of iterative training are implemented:

correspondingly receiving n embedding representations from the n participants, wherein each embedding representation is obtained by a corresponding participant by processing a local feature corresponding to a current batch of samples by using a local model; and

processing the n embedding representations by using a corresponding deployed first gating network for any first prediction task in m prediction tasks, to obtain n first weights; and processing, by using a corresponding deployed first prediction network, a weighted embedding representation obtained based on the n first weights and the n embedding representations, to obtain a first prediction result, so as to determine a training loss in the current round.

11. The computer system according to claim 10, wherein the processing the n embedding representations by using a corresponding deployed first gating network, to obtain n first weights comprises:

performing fusion processing on the n embedding representations, to obtain a fusion representation; and

processing the fusion representation by using the first gating network, to obtain the n first weights.

12. The computer system according to claim 11, wherein the fusion processing comprises concatenation processing or addition processing.

13. The computer system according to claim 10, wherein the processing, by using a corresponding deployed first prediction network, a weighted embedding representation obtained based on the n first weights and the n embedding representations, to obtain a first prediction result comprises:

performing weighted summation on the n embedding representations by using the n first weights, to obtain the weighted embedding representation; and

processing the weighted embedding representation by using the first prediction network, to obtain the first prediction result.

14. The computer system according to claim 10, wherein after the first prediction result is

obtained, when the processor executes the program, the following step is further implemented:

determining the training loss based on m prediction results for the m prediction tasks and corresponding m task labels, to update each model in the current round.

5 15. The computer system according to claim 14, wherein the determining the training loss comprises:

determining a corresponding first task loss by using the prediction result and a task label corresponding to the first prediction task; and

performing weighted summation on m task losses corresponding to the m prediction tasks, to obtain the training loss.

10 16. The computer system according to claim 14, wherein a task label corresponding to the first prediction task is provided by one of the n participants, and a loss function used to calculate the first task loss is determined by the participant.

17. The computer system according to claim 14, wherein after the training loss is determined, when the processor executes the program, the following steps are further implemented:

15 updating m prediction networks and m gating networks for the m prediction tasks by using the training loss, and calculating a back propagation gradient to be back propagated to each participant; and

providing the back propagation gradient to the corresponding participant, so that the corresponding participant updates the local model by using the received back propagation gradient.

20 18. A computer-readable nonvolatile storage medium, wherein the storage medium stores a computer program, and when the computer program is executed by a processor, the method according to any one of claims 1 to 8 is implemented.