

NEURAL NETWORK MODEL TRAINING METHODS AND RELATED PRODUCTS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This patent application claims priority to Chinese patent application No. 202010496921.7 entitled “NEURAL NETWORK MODEL TRAINING METHODS AND RELATED PRODUCTS”, filed on June 3, 2020, the entire contents of which are incorporated herein by reference.

TECHNICAL FIELD

[0002] The present disclosure relates to the field of model training, and in particular, to neural network model training methods and related products.

BACKGROUND

[0003] Deep learning is bringing huge development progress for many social fields, and model training is its key link. In the model training process, a large amount of sample data will be read and a large number of mathematical operations will be performed, which is very time-consuming. Although in the industry, breakthroughs are continuously made in benchmark tests on an ImageNet dataset, returning to a general training platform, an efficient distributed model training scheme is still a thorny practical problem. Therefore, more efficient distributed model training schemes need to be studied.

SUMMARY

[0004] Embodiments of the present disclosure disclose neural network model training methods and related products.

[0005] In a first aspect, an embodiment of the present disclosure provides a neural network model training method, including: obtaining, by a first work node, local gradient information of at least one network layer in the neural network model based on a current iteration performed on the neural network model; in a procedure of transmitting local gradient information of a first network layer in the neural network model with at least one second work node, updating, by the first work node, parameters of a second network layer in the neural network model in parallel.

[0006] The neural network model can include several layers, and its distributed parallel training process can be divided into: Forward Pass, Backward Pass, gradient data synchronization (for example, Allreduce Gradients) and parameter update of each layer. In some embodiments, the Forward Pass includes layer-by-layer operations in a forward order, and the Backward Pass includes layer-by-layer operations in a reverse order; the gradient data synchronization mainly occupies network bandwidth resources, and other operations occupy processor computing resources. In the embodiments of the present disclosure, the first work node performs the parameter update and the gradient data synchronization in parallel in order to hide communication overhead, fully explore overlappable parts in the model training process, reduce delay caused by communication, and improve efficiency of model training.

[0007] In some examples of the present disclosure, in the procedure of transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, the first work node updates the parameters of the second network layer in the neural network model in parallel; and a procedure of updating parameters of the neural network model and the procedure of transmitting local gradient information overlap, which can improve the efficiency of model training.

[0008] In some examples, the method further includes: determining, by the first work node, a dependency relationship between multiple operations of the current iteration based on a connection relationship between multiple network layers in the neural network model, where the multiple operations include at least an transmission operation of the local gradient information of at least one network layer in the neural network model, and an parameter update operation; and performing, by the first work node, the multiple operations based on the dependency relationship between the multiple operations.

[0009] In some examples, the dependency relationship between multiple operations of the current iteration can be accurately determined based on the connection relationship between multiple network layers in the neural network model, and multiple operations are sequentially performed based on the dependency relationship between multiple operations

[0010] In some examples, the first work node updates parameters of multiple network layers in the neural network model layer by layer in a reverse order; and/or a network depth of the second network layer is greater than that of the first network layer. Optionally, the first work node transmits local gradient information of multiple network layers in the neural network model with at least one second work node layer by layer in a reverse order; the first

work node calculates the local gradient information of multiple network layers in the neural network model layer by layer in a reverse order (corresponding to the Backward Pass that includes layer-by-layer operations in a reverse order).

[0011] In some examples, in the procedure of transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, updating, by the first work node, the parameters of the second network layer in the neural network model in parallel includes:

[0012] in the procedure of transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, in a case of determining that operations on which an parameter update operation of the second network layer depends have been completed, updating, by the first work node, the parameters of the second network layer in parallel, where the operations on which the parameter update operation depends include transmitting local gradient information of the second network layer with at least one second work node.

[0013] In some examples, it can be guaranteed that the operation of updating the parameters of the second network layer can be successfully performed.

[0014] In some examples, the method further includes: in the procedure of transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, calculating, by the first work node, local gradient information of a third network layer in the neural network model.

[0015] In some examples, in the procedure of transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, the first work node calculates the local gradient information of the third network layer in the neural network model; and a procedure of calculating local gradient information of network layers in the neural network model and a procedure of transmitting the local gradient information overlap (i.e., communication and calculation overlap), which can improve the efficiency of model training.

[0016] In some examples, before performing, by the first work node, the current iteration on the neural network model, the method further includes: performing, by the first work node, at least one inner layer iteration on the neural network model to obtain intermediate fusion gradient information corresponding to at least one inner layer iteration; obtaining, by the first work node, the local gradient information of at least one network layer in the neural network model based on the current iteration performed on the neural network model

includes: obtaining, by the first work node, target fusion gradient information of at least one network layer in the neural network model based on the intermediate fusion gradient information and local gradient information corresponding to the current iteration; the local gradient information of the first network layer transmitted by the first work node with at least one second work node includes target fusion gradient information of the first network layer.

[0017] The first work node performs at least one inner layer iteration on the neural network model to obtain one set of local gradient information. One set of local gradient information can be understood as all local gradient information obtained by the first work node completing the Forward Pass and Backward Pass of each network layer in the neural network model. Target fusion gradient information of one network layer in the neural network model can be understood as gradient information obtained by fusing multiple sets of local gradient information of the network layer obtained by multiple inner layer iterations.

[0018] In some examples, the first work node transmits target fusion gradient information of network layers with at least one second work node, which can reduce the number of gradient information transmission and the total communication traffic.

[0019] In some examples, obtaining, by the first work node, the target fusion gradient information of at least one network layer in the neural network model based on the intermediate fusion gradient information and the local gradient information corresponding to the current iteration includes: performing, by the first work node, accumulation processing on the intermediate fusion gradient information and the local gradient information corresponding to the current iteration to obtain the target fusion gradient information of at least one network layer in the neural network model.

[0020] In some examples, the method further includes: in a procedure of obtaining target fusion gradient information of the third network layer in the neural network model based on the intermediate fusion gradient information and the local gradient information corresponding to the current iteration, transmitting, by the first work node, target fusion gradient information of a fourth network layer in the neural network model with at least one second work node. Optionally, a network depth of the fourth network layer is greater than that of the third network layer.

[0021] In some examples, a procedure of calculating target fusion gradient information of network layers in the neural network model and the procedure of transmitting the target

fusion gradient information of the network layers overlap (i.e., calculation and communication overlap), which can improve the efficiency of model training.

[0022] In some examples, before transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, the method further includes: amplifying, by the first work node, each value in the local gradient information of the first network layer by M times, and converting each amplified value into half precision, where M is a real number greater than 1.

[0023] In some examples, by using low-precision storage for each value in local gradient information, the data amount of the local gradient information can be reduced.

[0024] In some examples, before updating, by the first work node, the parameters of the second network layer in the neural network model in parallel, the method further includes: converting, by the first work node, each value included in obtained local gradient information of the second network layer into single precision, and reducing each converted value by M times to obtain processed gradient information, where M is a real number greater than 1; updating, by the first work node, the parameters of the second network layer in the neural network model in parallel includes: using, by the first work node, the processed gradient information to update the parameters of the second network layer in the neural network model.

[0025] In some examples, before transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, the method further includes: storing, by the first work node, calculated local gradient information of the first network layer into a pre-allocated target storage space based on an offset corresponding to the first network layer, where the target storage space is used for storing local gradient information of multiple network layers in the neural network model, where the local gradient information of the first network layer sent by the first work node is acquired from the target storage space based on the offset corresponding to the first network layer; and/or updating, by the first work node, the local gradient information of the first network layer stored in the target storage space based on received local gradient information of the first network layer from at least one second work node.

[0026] In some examples, the local gradient information of the first network layer can be quickly and accurately acquired from the target storage space based on the offset corresponding to the first network layer and/or the local gradient information of the first network layer stored in the target storage space can be updated.

[0027] In some examples, before transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, the method further includes: storing, by the first work node, the calculated local gradient information of multiple network layers in the neural network model into a pre-allocated target storage space, and determining an offset corresponding to each of multiple network layers through a memory manager, where the target storage space is a continuous storage space; acquiring, by the first work node, local gradient information of at least two of multiple network layers from the target storage space based on the offset corresponding to each of multiple network layers, where at least two network layers include the first network layer. Transmitting the local gradient information of the first network layer in the neural network model with at least one second work node includes: transmitting local gradient information of at least two network layers in the neural network model with at least one second work node.

[0028] It should be understood that the main principle of the implementation manner is: to merge local gradient information of several network layers into a large array, and then initiate one global communication again, so that the efficiency of global communication can be improved, and the number of global communication can be reduced.

[0029] In a second aspect, an embodiment of the present disclosure provides an image predicting method, including: acquiring a to-be-processed image; and performing prediction processing on the to-be-processed image by using the neural network model trained in the first aspect and any example of the first aspect to obtain a prediction result.

[0030] In a third aspect, an embodiment of the present disclosure provides a data processing apparatus, including: a processing module configured to obtain local gradient information of at least one network layer in a neural network model based on a current iteration performed on the neural network model; a transceiving module configured to transmit local gradient information of a first network layer in the neural network model with at least one second work node, where the processing module is further configured to, in a procedure of transmitting, by the transceiving module, the local gradient information of the first network layer in the neural network model with at least one second work node, update parameters of a second network layer in the neural network model in parallel.

[0031] Regarding the technical effects brought from the third aspect or various examples of the third aspect, reference may be made to the introduction to the technical effects of the first aspect or corresponding implementation manners of the first aspect.

[0032] In a fourth aspect, an embodiment of the present disclosure provides a data processing apparatus, including: an acquiring module configured to acquire a to-be-processed image; a processing module configured to perform prediction processing on the to-be-processed image by using the neural network model trained in the first aspect and any example of the first aspect to obtain a prediction result.

[0033] In a fifth aspect, an embodiment of the present disclosure provides an electronic device, including: a processor and a memory, where the memory is used for storing instructions, and the processor is used for executing instructions stored in the memory, so that the processor implements the method in the first aspect and any example of the first aspect.

[0034] In a sixth aspect, an embodiment of the present disclosure provides an electronic device, including: a processor and a memory, where the memory is used for storing instructions, and the processor is used for executing instructions stored in the memory, so that the processor implements the method in the second aspect and any example of the second aspect.

[0035] In a seventh aspect, an embodiment of the present disclosure provides a chip, including a data interface and a processor, where the processor is used to implement the method in the first aspect and any example of the first aspect.

[0036] In an eighth aspect, an embodiment of the present disclosure provides a chip, including a data interface and a processor, where the processor is used to implement the method in the second aspect and any example of the second aspect.

[0037] In a ninth aspect, an embodiment of the present disclosure provides a computer readable storage medium, where a computer program is stored in the computer readable storage medium; the computer program includes program instructions; and when the program instructions are executed by a processor, the processor is caused to implement the method in the first aspect and any example of the first aspect.

[0038] In a tenth aspect, an embodiment of the present disclosure provides a computer readable storage medium, where a computer program is stored in the computer readable storage medium; the computer program includes program instructions; and when the program instructions are executed by a processor, the processor is caused to implement the method in the second aspect and any example of the second aspect.

[0039] In an eleventh aspect, an embodiment of the present disclosure provides a computer program product, including program instructions, where when the program instructions are

executed by a processor, the processor is caused to implement the method in the first aspect and any example of the first aspect.

[0040] In a twelfth aspect, an embodiment of the present disclosure provides a computer program product, including program instructions, where when the program instructions are executed by a processor, the processor is caused to implement the method in the second aspect and any example of the second aspect.

BRIEF DESCRIPTION OF THE DRAWINGS

[0041] In order to more clearly describe the technical solutions in the embodiments or the background technology of the present disclosure, the drawings that need to be used therein will be described below.

[0042] FIG. 1 is an example of a distributed training flowchart provided by an embodiment of the present disclosure.

[0043] FIG. 2 is a flowchart illustrating a neural network model training method provided by an embodiment of the present disclosure.

[0044] FIG. 3 is a schematic diagram illustrating an example where calculation and communication overlap provided by an embodiment of the present disclosure.

[0045] FIG. 4 is a schematic diagram illustrating another example where calculation and communication overlap provided by an embodiment of the present disclosure.

[0046] FIG. 5 is a flowchart illustrating an inner layer iteration method provided by an embodiment of the present disclosure.

[0047] FIG. 6 is a schematic diagram illustrating an example of a communication fusion strategy provided by an embodiment of the present disclosure.

[0048] FIG. 7 is a flowchart illustrating an image predicting method provided by an embodiment of the present disclosure.

[0049] FIG. 8 is a schematic structural diagram illustrating a data processing apparatus provided by an embodiment of the present disclosure.

[0050] FIG. 9 is a schematic structural diagram illustrating another data processing apparatus provided by an embodiment of the present disclosure.

[0051] FIG. 10 is a schematic structural diagram illustrating a server provided by an embodiment of the present disclosure.

[0052] FIG. 11 is a schematic structural diagram illustrating a terminal device provided by an embodiment of the present disclosure.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0053] Terms “first”, “second” and “third” in examples of the specification, claims and drawings in the present disclosure are used to distinguish similar objects, and are not necessarily used to describe specific order or precedence order. Furthermore, terms “comprising”, “having” and any variations thereof are intended to cover non-exclusive inclusion, for example, including a series of steps or units. Methods, systems, products or devices are not necessarily limited to those steps or units expressly listed, but may include other steps or units not expressly listed or inherent to processes, methods, products or devices.

[0054] Efficient distributed model training schemes are intractable practical problems. The present disclosure provides a neural network model training method suitable for a distributed model training scenario, which can improve model training efficiency. The scenario to which the neural network model training method provided by the examples of the present disclosure is applicable will be introduced briefly below.

[0055] Distributed model training scenario: a distributed training system includes multiple work nodes. Each work node has basically the same function. Each work node obtains a trained neural network model by iteratively training a neural network model for many times. In one iteration, each work node uses its training sample to train the neural network model to obtain its local gradient information (INFO for short); then, data synchronization is performed among multiple work nodes, so that each of multiple work nodes obtains local gradient information of all work nodes, and fuses the obtained local gradient information of all work nodes to obtain global gradient information, or each of multiple work nodes obtains fusion gradient information by fusing local gradient information of all other work nodes, and fuses its local gradient information with the fusion gradient information to obtain global gradient information. As an example, each work node sends its calculated local gradient information and/or received local gradient information from at least one other work node to other work nodes, or sends fusion gradient information obtained by fusing its calculated local gradient information with received local gradient information from at least one other work node, for example, to its left or right work node, until each work node obtains local gradient information calculated by all work nodes, fusion gradient information or global gradient information; then, each work node uses global gradient information obtained by fusing the local gradient information calculated by all work nodes to update parameters of the neural network model. Such iteration is performed for several times. Each work node

repeats the previous operations in each iteration, until training cutoff conditions are reached, for example, the neural network model converges or the number of training reaches a preset number. In this distributed model training scenario, in some examples, work nodes use the same neural network model, and the work nodes update parameters of the neural network model synchronously. Different work nodes use different training samples to train the neural network model. That is to say, neural network models used by work nodes are always same. In some examples, multiple work nodes may be multiple processors on the same terminal device or server. For example, 8 GPUs on a server are used as 8 work nodes, that is, one GPU corresponds to one work node. In some examples, one work node or at least two work nodes correspond(s) to one hardware entity, such as a terminal device or a server. For example, 8 notebook computers are used as 8 work nodes, that is, one notebook computer is used as one work node. For another example, 256 GPUs on 32 servers are used as 256 work nodes. For still another example, multiple work nodes included in a distributed training system are respectively multiple virtual machines running in one or more devices (for example, servers).

[0056] In the above scenario, according to the neural network model training method provided by the examples of the present disclosure, a procedure of updating the parameters of the neural network model by the work nodes and a procedure of synchronizing the gradient data of the work nodes are performed in parallel, which can improve the training efficiency.

[0057] The neural network model training method provided by the examples of the present disclosure will be described below in conjunction with an example of a distributed training flowchart.

[0058] FIG. 1 is an example of a distributed training flowchart provided by an example of the present disclosure. As shown in FIG. 1, GPU 0, GPU 1, GPU 2 and GPU 3 are work nodes in a distributed training system; a neural network model includes several layers; and a parallel training process of GPU 0, GPU 1, GPU 2 and GPU 3 can include: Forward Pass, Backward Pass, gradient data synchronization (for example, gradient protocol communication) and parameter update of each layer. In the Forward Pass, the layers in the neural network model sequentially process an image input to the neural network model to obtain a processing result of the image. Then, a gradient of a last layer in the neural network model can be obtained based on the processing result and specific calculation rules. In the Backward Pass, the gradient of the last layer can be backward passed in a reverse order, and

a gradient of each layer in the neural network model can be calculated sequentially. In the gradient data synchronization, gradient data can be synchronized among multiple work nodes. In the examples of the present disclosure, the purpose of synchronizing the gradient data is to enable each work node to obtain global gradient information obtained by fusing local gradient information calculated by all work nodes. The present disclosure does not limit the way to achieve this purpose. In the parameter update, each work node uses the global gradient information obtained by the gradient data synchronization to update network parameters (for example, a weight) of the neural network model.

[0059] In the example shown in FIG. 1, different work nodes input different training samples to the neural network model for Forward Pass and Backward Pass to obtain their local gradient information. Each work node, after completing one global gradient data synchronization, can obtain global gradient information obtained by fusing local gradient information calculated by all work nodes, or local gradient information calculated by all work nodes. Each work node uses the global gradient information obtained by fusing the local gradient information calculated by all work nodes to update parameters of the neural network model. Each work node can update the parameters of the neural network model in the same way.

[0060] In some examples, gradient data synchronization mainly occupies network bandwidth resources, and other operations occupy GPU computing resources. In order to hide communication overhead, the examples of the present disclosure provide a neural network model training method that makes gradient data synchronization and parameter update overlap (i.e., parallel). The neural network model training method provided by the examples of the present disclosure will be introduced below with reference to the drawings.

[0061] FIG. 2 is a flowchart illustrating a neural network model training method provided by an example of the present disclosure. As shown in FIG. 2, the method includes:

[0062] At 201, a first work node obtains local gradient information of at least one network layer in the neural network model based on a current iteration performed on the neural network model.

[0063] The first work node may be a terminal device, such as a notebook computer, a desktop computer, a tablet computer or a mobile phone; a server; a virtual machine running on a server or a terminal device; or a processor on a terminal device or a server, such as a Graphics Processing Unit (GPU), a Central Processing Unit (CPU) or a Neural-network Processing Unit (NPU). As shown in FIG. 1, each GPU can obtain local gradient information

of each network layer through Backward Pass. In some examples, the Backward Pass is a layer-by-layer operation in a reverse order, and the first work node calculates the local gradient information of each network layer in the neural network model layer by layer in a reverse order with reference to FIG. 1.

[0064] In some examples, before the first work node transmits local gradient information of a first network layer in the neural network model with at least one second work node (step 202 is performed), the following operations may be performed: the first work node amplifies each value in the local gradient information of the first network layer by M times, and converts each amplified value into half precision, where M is a real number greater than 1. In this example, the first work node, before transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, first converts the local gradient information of the first network layer into half-precision float data, such that a storage space occupied thereby will be reduced by half compared to single-precision float data; then, the first work node performs gradient protocol communication; after the protocol communication is completed, the first work node first converts a half-precision gradient obtained by the protocol communication back to single precision, and then updates parameters. In this way, the communication overhead can be reduced by half.

[0065] However, it should be noted that the range of positive numbers that can be represented by half-precision float data is 6.1×10^{-5} to 65504, which is much less than the range that is represented by single-precision float data, and the gradient of the neural network model is often a very small value. Therefore, the first work node first amplifies local gradient information before communication, and then reduces it after the communication, so as to decrease a precision loss in the procedure of transmitting the local gradient information.

[0066] At 202, in the procedure of transmitting local gradient information of a first network layer in the neural network model with at least one second work node, the first work node updates parameters of a second network layer in the neural network model in parallel.

[0067] The first network layer and the second network layer are different. In some examples, each of at least one second work node performs similar operations to the first work node. In some examples, the first work node updates parameters of multiple network layers in the neural network model layer by layer in a reverse order; and/or, a network depth of the second network layer is greater than that of the first network layer. In some examples,

the first work node implements gradient data synchronization through layer-by-layer operations in a reverse order, and likewise, implements parameter update through layer-by-layer operations in a reverse order. For example, the neural network model includes n layers, and the first work node sequentially transmits local gradient information of the n^{th} network layer to the first network layer with at least one second work node (corresponding to implementing the gradient data synchronization through layer-by-layer operations in a reverse order). “Transmit” here means “send” and “receive”. For example, the first work node, while sending local gradient information of the n^{th} network layer calculated by the first work node to at least one second work node, receives local gradient information of the n^{th} network layer from at least one second work node. Then, the first work node sequentially updates parameters of the n^{th} network layer to the first network layer (corresponding to implementing the parameter update through layer-by-layer operations in a reverse order). FIG. 3 is a schematic diagram illustrating an example where calculation and communication overlap provided by an example of the present disclosure. As shown in FIG. 3, 301 represents data stream 1 of implementing gradient data synchronization through layer-by-layer operations in a reverse order, and 302 represents data stream 2 of implementing parameter update through layer-by-layer operations in a reverse order, where the data stream 1 and the data stream 2 are parallel. Each rectangular box in 301 represents an operation of the first work node that transmits (or communicates, synchronizes) local gradient information of one network layer with other work nodes. For example, the n^{th} network layer represents an operation of the first work node that transmits local gradient information of the n^{th} network layer with other work nodes. Each rectangular box in 302 represents an operation of the first work node that updates parameters of one network layer. For example, the n^{th} network layer represents an operation of the first work node that updates parameters of the n^{th} network layer. Arrows represent time axes. n is an integer greater than 1. In FIG. 3, the first work node sequentially transmits local gradient information of the n^{th} network layer, local gradient information of the $(n-1)^{\text{th}}$ network layer, ..., local gradient information of the first network layer with other work nodes. The first work node sequentially updates parameters of the n^{th} network layer, parameters of the $(n-1)^{\text{th}}$ network layer, ..., parameters of the first network layer. in the procedure of transmitting local gradient information of the $(n-i)^{\text{th}}$ network layer with other work nodes, the first work node updates parameters of the $(n-i+1)^{\text{th}}$ network layer in parallel, where i is an integer less than n . Since the first work node implements the gradient data synchronization through layer-by-layer operations in a reverse

order, and likewise, implements the parameter update through layer-by-layer operations in a reverse order, during the gradient data synchronization, the first work node can use the obtained local gradient information of network layers in parallel to implement a part of the parameter update operation. Referring to FIG. 3, since the first work node, before performing the operation of receiving the local gradient information of the $(n-1)^{\text{th}}$ network layer, has received the local gradient information of the n^{th} network layer, during the operation of receiving the local gradient information of the $(n-1)^{\text{th}}$ network layer, the first work node can perform the operation of updating the parameters of the n^{th} network layer in parallel.

[0068] In some examples, the first work node determines a dependency relationship between multiple operations of the current iteration based on a connection relationship between multiple network layers in the neural network model, where multiple operations include at least an transmission operation of the local gradient information of at least one network layer in the neural network model, and an parameter update operation, and the first work node performs multiple operations based on the dependency relationship between multiple operations. That is to say, the first work node can determine the dependency relationship between multiple operations of the current iteration according to a precedence relationship of network layers to which multiple operations of the current iteration belong, i.e., the specific timing of performing each operation is driven by the dependency relationship. Exemplarily, the first work node implements gradient data synchronization through layer-by-layer operations in a reverse order, and likewise, implements parameter update through layer-by-layer operations in a reverse order. The local gradient information transmission operation of any network layer in the neural network model depends on that the local gradient information transmission operation of each network layer after the any network layer is completed, and the parameter update operation of any network layer in the neural network model depends on that the local gradient information transmission operation of the any network layer is completed. For example, the first work node, after completing the local gradient information transmission operation of the n^{th} network layer in the neural network model, can perform the local gradient information transmission operation of the $(n-1)^{\text{th}}$ network layer and the parameter update operation of the n^{th} network layer.

[0069] In some examples, the implementation of step 202 is as follows: in the procedure of transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, in a case of determining that operations on which

an parameter update operation of the second network layer depends have been completed, the first work node updates the parameters of the second network layer in parallel to the transmission of the local gradient information of the first network layer, where the operations on which the parameter update operation depends include transmission of local gradient information of the second network layer with at least one second work node. In some examples, each operation to be performed by the first work node is bound to an event, and an event that each operation needs to wait for is determined according to the dependency relationship between operations; each data stream, after waiting for completion of an associated event of current operation through a lightweight blocking interface (for example, `cudaStreamWaitEvent`), starts the current operation.

[0070] In an example, before the first work node updates the parameters of the second network layer in the neural network model, the following operations may be performed: the first work node converts each value included in obtained local gradient information of the second network layer into single precision, and reduces each converted value by M times to obtain processed gradient information, where M is a real number greater than 1; updating, by the first work node, the parameters of the second network layer in the neural network model in parallel may include: using, by the first work node, the processed gradient information to update the parameters of the second network layer in the neural network model.

[0071] In the examples of the present disclosure, in the procedure of transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, the first work node updates the parameters of the second network layer in the neural network model in parallel; and a procedure of updating parameters of the neural network model and the procedure of transmitting local gradient information overlap (i.e., parameter update and calculation overlap), which can improve the efficiency of model training.

[0072] To further hide the communication overhead, the first work node can further make gradient data synchronization and Backward Pass overlap. An example of making gradient data synchronization and Backward Pass overlap will be described below with reference to the drawings.

[0073] In an example, on the basis that the method in FIG. 1 is executed, the following operations may be performed: in the procedure of transmitting the local gradient information of the first network layer in the neural network model with at least one second work node,

the first work node calculates local gradient information of a third network layer in the neural network model. A network depth of the third network layer is less than that of the first network layer. In some examples, the Backward Pass includes layer-by-layer operations in a reverse order. The first work node implements gradient data synchronization through the layer-by-layer operations in a reverse order, and a procedure of implementing the Backward Pass by the first work node may overlap with a procedure of implementing gradient data synchronization, that is, the Backward Pass and the gradient data synchronization are implemented in parallel.

[0074] FIG. 4 is a schematic diagram illustrating another example where calculation and communication overlap provided by an example of the present disclosure. As shown in FIG. 4, 401 represents data stream 3 of implementing Backward Pass through layer-by-layer operations in a reverse order, 301 represents data stream 1 of implementing gradient data synchronization through layer-by-layer operations in a reverse order, and 302 represents data stream 2 of implementing parameter update through layer-by-layer operations in a reverse order, where the data stream 1, the data stream 2 and the data stream 3 are parallel. Each rectangular box in 401 represents an operation of the first work node that calculates local gradient information of one network layer (corresponding to a backward operation). For example, the n^{th} network layer represents an operation of the first work node that calculates local gradient information of the n^{th} network layer. Each rectangular box in 301 represents an operation of the first work node that transmits local gradient information of one network layer with other work nodes. For example, the n^{th} network layer represents an operation of the first work node that transmits local gradient information of the n^{th} network layer with other work nodes. Each rectangular box in 302 represents an operation of the first work node that updates parameters of one network layer. For example, the n^{th} network layer represents an operation of the first work node that updates parameters of the n^{th} network layer. N is an integer greater than 1. In FIG. 4, the first work node sequentially calculates local gradient information of the n^{th} network layer, local gradient information of the $(n-1)^{\text{th}}$ network layer, ..., local gradient information of the first network layer. The first work node sequentially transmits local gradient information of the n^{th} network layer, local gradient information of the $(n-1)^{\text{th}}$ network layer, ..., local gradient information of the first network layer with other work nodes. The first work node sequentially updates parameters of the n^{th} network layer, parameters of the $(n-1)^{\text{th}}$ network layer, ..., parameters of the first network layer. In a procedure of receiving local gradient information of the $(n-i)^{\text{th}}$ network layer, the

first work node updates parameters of the $(n-i+1)^{\text{th}}$ network layer and calculates local gradient information of the $(n-i-1)^{\text{th}}$ network layer in parallel, where i is an integer less than $(n-1)$.

[0075] In this example, in the procedure of transmitting the local gradient information of the first network layer in the neural network model with at least one second work node, the first work node calculates the local gradient information of the third network layer in the neural network model; and a procedure of calculating local gradient information of network layers in the neural network model and the procedure of transmitting the local gradient information overlap, which can improve the efficiency of model training.

[0076] The above examples describe a scheme where calculation and communication overlap. The essence of the scheme is to hide a communication time with a parameter update time and/or a Backward Pass time, but when a calculation time of the neural network model is less than its communication time, the communication overhead cannot be sufficiently hidden. Therefore, it is necessary to study a communication reduction scheme to further reduce the communication overhead.

[0077] The examples of the present disclosure introduce an inner layer iteration strategy. Each inner layer iteration performs one complete Forward Pass and Backward Pass, and accumulates local gradient information, but does not perform gradient data synchronization and parameter update, that is, does not synchronize gradient data of work nodes and does not update parameters of the neural network model. Multiple inner layer iterations correspond to one global communication, where protocol communication is performed on local gradient information and parameter values are updated in a final inner layer iteration. In some examples, the global communication operation may overlap with Backward Pass of the final inner layer iteration. The inner layer iteration strategy is essentially to increase a batch size of each iteration, which is equivalent to reducing the total communication traffic in the overall training process. An inner layer iteration method provided by the examples of the present disclosure will be described below with reference to the drawings.

[0078] FIG. 5 is a flowchart illustrating an inner layer iteration method provided by an example of the present disclosure. As shown in FIG. 5, the inner layer iteration method includes:

[0079] At 501, a first work node inputs a training sample into a neural network model for Forward Pass to obtain a processing result.

[0080] At 502, the first work node uses the processing result and the neural network model for Backward Pass to obtain local gradient information of at least one network layer in the neural network model.

[0081] Steps 502 and 501 can be understood as an implementation manner in which the first work node performs one inner layer iteration on the neural network model to obtain local gradient information of at least one network layer in the neural network model. In some examples, step 502 may be replaced by: the first work node uses the processing result and the neural network model for Backward Pass to obtain local gradient information of each network layer in the neural network model. For example, the first work node implements the Backward Pass through layer-by-layer operations in a reverse order to obtain the local gradient information of each network layer in the neural network model.

[0082] At 503, the first work node obtains target fusion gradient information of at least one network layer in the neural network model based on intermediate fusion gradient information and local gradient information corresponding to the current iteration (i.e., a current inner layer iteration).

[0083] In some examples, the intermediate fusion gradient information may be intermediate fusion gradient information corresponding to at least one inner layer iteration obtained by the first work node performing at least one inner layer iteration on the neural network model. Exemplarily, the intermediate fusion gradient information may be local gradient information of each network layer in the neural network model obtained by the first work node performing one inner layer iteration, or be obtained by fusing at least two sets of local gradient information obtained by the first work node performing at least two inner layer iterations. It should be understood that when the first work node performs step 503 for the first time, intermediate fusion gradient information does not exist, and the implementation manner of step 503 may include using the local gradient information of at least one network layer in the neural network model obtained in step 502 as intermediate fusion gradient information, and storing the intermediate fusion gradient information; when the first work node performs step 503 for the second time, the implementation manner of step 503 may include obtaining new intermediate fusion gradient information (corresponding to updating intermediate fusion gradient) based on current intermediate fusion gradient information and local gradient information corresponding to the current inner layer iteration (i.e., gradient information obtained by performing step 502 for the second time); and so on, after the first work node performs step 503 for the K^{th} time, target

fusion gradient information of at least one network layer in the neural network model is obtained, where K is an integer greater than 1. It can be understood that the first work node performs step 503 for the first time to obtain initial intermediate fusion gradient (corresponding to gradient information obtained by performing step 502 for the first time), and each subsequent performance of step 503 is to use current intermediate fusion gradient information and local gradient information corresponding to the current iteration (i.e., the current inner layer iteration) to obtain new intermediate fusion gradient information.

[0084] In some examples, the first work node performs one inner layer iteration to obtain one set of local gradient parameters. Each set of local gradient parameters includes local gradient information of each network layer in the neural network model; fusing, by the first work node, at least two sets of local gradient information obtained by performing at least two inner layer iterations may be: fusing local gradient information of each network layer respectively included in at least two sets of local gradient information to obtain an intermediate fusion gradient of each network layer. For example, the first work node fuses local gradient information of the first network layer respectively included in at least two sets of local gradient information to obtain an intermediate fusion gradient of the first network layer. Exemplarily, fusing, by the first work node, local gradient information of the first network layer respectively included in at least two sets of local gradient information may be sequentially fusing corresponding parameters of the first network layer respectively included in two sets of local gradient information. For example, a value of a parameter of the first network layer included in the first set of local gradient information is a, a value of the parameter of the first network layer included in the second set of local gradient information is b, and a value of the parameter of the first network layer included in the third set of local gradient information is c. Taking this parameter as an example, fusing, by the first work node, local gradient information of the first network layer respectively included in the three sets of local gradient information may be: first calculating $(a+b)$, and then calculating $((a+b)+c)$. In this example, corresponding value of this parameter in intermediate fusion gradient information of the first network layer is $((a+b)+c)$.

[0085] In some examples, an implementation manner of step 503 may be: performing, by the first work node, accumulation processing on intermediate fusion gradient information and local gradient information corresponding to the current iteration to obtain target fusion gradient information of at least one network layer in the neural network model. Gradients in the intermediate fusion gradient information correspond to gradients in the local gradient

information corresponding to the current iteration one to one. Performing, by the first work node, the accumulation processing on the intermediate fusion gradient information and the local gradient information corresponding to the current iteration to obtain the target fusion gradient information of at least one network layer in the neural network model may be performing accumulation processing on parameters that correspond to each other one to one in the intermediate fusion gradient information and the local gradient information corresponding to the current iteration. For example, a value of a parameter in intermediate fusion gradient information is d, and corresponding value of this parameter in local gradient information corresponding to the current iteration is e. Accumulation processing is performed on d and e to obtain (d+e). It should be understood that target fusion gradient information of any network layer in the neural network model can be obtained by fusing multiple sets of local gradient information of any network layer obtained by the first work node performing multiple inner layer iterations.

[0086] At 504, the first work node determines whether an inner layer iteration threshold is reached.

[0087] If the first work node determines that the inner layer iteration threshold is reached, step 505 is performed. If the first work node determines that the inner layer iteration threshold is not reached, step 501 is performed. The inner layer iteration threshold may be 3, 5, 10, 20, etc., which is not limited in the present disclosure. In practical applications, the first work node can correspondingly set the inner layer iteration threshold according to actual requirements. The greater the inner layer iteration threshold is, the fewer the number of the first work node performing global communication is.

[0088] At 505, the first work node performs a global communication operation to obtain global gradient information.

[0089] In some examples, the global gradient information may be gradient information obtained by fusing local gradient information calculated by all work nodes. Exemplarily, the global gradient information may be gradient information obtained by accumulating corresponding gradients in local gradient information calculated by all work nodes. For example, local gradient information calculated by each work node corresponds to one vector, and a vector corresponding to global gradient information obtained by fusing local gradient information calculated by all work nodes may be obtained by accumulating elements at the same position in vectors corresponding to local gradient information calculated by work

nodes. In some examples, after the first work node obtains global gradient information, each work node in the distributed training system obtains global gradient information.

[0090] At 506, the first work node uses the global gradient information to update the neural network model.

[0091] It should be understood that each work node in the distributed training system uses the global gradient information to update the neural network model, so that each work node will obtain the same updated neural network model. Steps 501 to 506 describe a procedure of the first work node implementing one parameter update operation. In practical applications, the first work node may execute the method in FIG. 5 for many times to obtain a converged neural network model.

[0092] In some examples, the following operations may be performed: in a procedure of obtaining target fusion gradient information of the third network layer in the neural network model based on intermediate fusion gradient information and local gradient information corresponding to the current iteration, the first work node transmits target fusion gradient information of a fourth network layer in the neural network model in parallel with at least one second work node. Optionally, a network depth of the fourth network layer is greater than that of the third network layer. The first work node can perform a final inner layer iteration through layer-by-layer operations in a reverse order. Therefore, the first work node can sequentially obtain target fusion gradient information of the highest network layer (with a maximum network depth) to target fusion gradient information of the lowest network layer (with a minimum network depth) in the neural network model. It should be understood that in a procedure of calculating target fusion gradient information of a network layer, the first work node may transmit the calculated target fusion gradient information of some network layers to other work nodes. That is, the global communication operation may overlap with Backward Pass of the final inner layer iteration.

[0093] In this implementation manner, the procedure of calculating target fusion gradient information of network layers in the neural network model and a procedure of transmitting the target fusion gradient information of the network layers overlap (i.e., calculation and communication overlap), which can improve the efficiency of model training.

[0094] In the examples of the present disclosure, the first work node transmits target fusion gradient information of network layers with at least one second work node, which can reduce the number of gradient information transmission and the total communication traffic.

[0095] In order to further improve communication efficiency, the examples of the present disclosure further provide a communication fusion strategy, that is, gradients of several network layers are merged into a large array, and one global communication is initiated again. The communication fusion strategy can be applied to the above examples to improve the communication efficiency.

[0096] For most of operators in common neural network models, the number of gradient parameters is quite small, usually, a small constant multiple of the number of feature maps, and the communication traffic is on the order of KBytes or even Byte. According to relevant researches on underlying communication, when a data transmission amount is small, a network bandwidth cannot be fully utilized. In order to obtain a larger communication traffic and thereby improve the communication efficiency, a communication fusion strategy is introduced.

[0097] There are several points that need to focus on in this strategy. On one hand, the scale of communication fusion (also known as gradient fusion) needs to be reasonably configured. If the scale of fusion is too small, the communication efficiency will be low; if the scale of fusion is too large, the initiation timing of communication operations will be delayed. Therefore, when the communication fusion strategy is implemented, the scale of fusion is allowed to be configured, for example, the most suitable scale of fusion is debugged for each neural network model and platform (such as the distributed training system) by dry-running. On the other hand, under the original scheme of communication fusion, multiple discretely stored small arrays are merged into a continuously stored large array before communication, and then the continuously stored large array is disassembled back after communication, so that memory copies are introduced twice, which will generate additional overhead.

[0098] In some examples, before step 201 is performed, the following operations may be performed: the first work node stores the calculated local gradient information of the first network layer into a pre-allocated target storage space based on an offset corresponding to the first network layer, where the target storage space is used for storing local gradient information of multiple network layers in the neural network model;

[0099] where the local gradient information of the first network layer sent by the first work node is acquired from the target storage space based on the offset corresponding to the first network layer; and/or the first work node updates the local gradient information of the first

network layer stored in the target storage space based on received local gradient information of the first network layer from at least one second work node.

[0100] In this example, the first work node opens up a unified continuous memory space (corresponding to the target storage space) for all parameter gradients (corresponding to gradient information) of the neural network model in advance, and then a parameter gradient of each network layer is pointed to corresponding offset through a memory manager, thereby avoiding additional memory copies during communication.

[0101] In some examples, before step 201 is performed, the following operations may be performed: the first work node stores the calculated local gradient information of multiple network layers in the neural network model into a pre-allocated target storage space, and determines an offset corresponding to each of multiple network layers through a memory manager, where the target storage space is a continuous storage space; the first work node acquires local gradient information of at least two of multiple network layers from the target storage space based on the offset corresponding to each of multiple network layers, where at least two network layers include the first network layer. Step 201 may be replaced by: transmitting local gradient information of at least two network layers in the neural network model with at least one second work node.

[0102] FIG. 6 is a schematic diagram illustrating an example of a communication fusion strategy provided by an example of the present disclosure. As shown in FIG. 6, 601 represents network layers in a neural network model, where L1 represents the first network layer, and L_n represents the nth network layer; 602 represents local gradient information of network layers, where gradient m, gradient (m - 1), ...gradient 1 each represent one gradient or a gradient of one network layer; 603 represents local gradient information of network layers after being merged, where gradient group k, gradient group (k-1), ...gradient group 1 each include at least two gradients or gradients of at least two network layers. In the examples of the present disclosure, network layers and gradients in the neural network model are not in a one-to-one correspondence. Some network layers may have multiple gradients. Some network layers may have no gradient. In some examples, if each rectangular box (for example, gradient m) in 602 represents a gradient of one network layer, each time the first work node transmits the gradient of one network layer to other work nodes, it needs to transmit for m times, and each time the first work node transmits one gradient group (such as gradient group k) to other work nodes, it needs to transmit for k times, where k is less than m. In some examples, if each rectangular box (for example, gradient m) in 602

represents a gradient of one parameter vector, each time the first work node transmits one gradient group (such as gradient group k) to other work nodes, it needs to transmit for k times. It should be understood that the first work node can merge local gradient information of several network layers into a large array, and then initiate one global communication again, so that global communication information can be reduced.

[0103] The above examples describe the neural network model training method. Below, examples of using the trained neural network model to realize prediction tasks will be introduced.

[0104] FIG. 7 is a flowchart illustrating an image predicting method provided by an example of the present disclosure. As shown in FIG. 7, the method includes:

[0105] At 701, an image processing apparatus acquires a to-be-processed image.

[0106] The image processing apparatus may be the first work node, or other work nodes, and may also be an apparatus that does not participate in the training of a neural network model, such as a terminal device or a server.

[0107] In some examples, the image processing apparatus is a server, and the image processing apparatus acquiring the to-be-processed image may be that the server receives a to-be-processed image from a terminal device or acquires a to-be-processed image from other devices according to an instruction input by a user.

[0108] In some examples, the image processing apparatus is a server, and the image processing apparatus acquiring the to-be-processed image may be that the server acquires a to-be-processed image uploaded by a user or acquires a to-be-processed image from other devices according to an instruction input by a user.

[0109] At 702, prediction processing is performed on the to-be-processed image using a trained neural network model to obtain a prediction result.

[0110] The neural network model may be trained using the method in the above examples. It should be understood that FIG. 7 is an example of applying a neural network model. The neural network model trained using the training method in the above examples can process different prediction tasks, such as text identification, image identification and image classification.

[0111] In some examples, the image processing apparatus is a server, and after performing step 702, the image processing apparatus may send the prediction result to a terminal device, such as a mobile phone or a personal computer.

[0112] In some examples, the image processing apparatus is a terminal device, and after performing step 702, the image processing apparatus may output the prediction result, for example, display the prediction result through a display screen.

[0113] In the examples of the present disclosure, the prediction processing is performed on the to-be-processed image using the trained neural network model to obtain the prediction result, which can efficiently realize different image prediction tasks.

[0114] The above examples describe the neural network model training method implemented by the first work node. Below, functions of each module of the first work node will be introduced with reference to the drawings.

[0115] FIG. 8 is a schematic structural diagram illustrating a data processing apparatus provided by an example of the present disclosure. The data processing apparatus in FIG. 8 may be the first work node in the above examples. As shown in FIG. 8, the data processing apparatus may include:

[0116] a processing module 801 configured to obtain local gradient information of at least one network layer in a neural network model based on a current iteration performed on the neural network model;

[0117] a transceiving module 802 configured to transmit local gradient information of a first network layer in the neural network model with at least one second work node,

[0118] where the processing module 801 is further configured to, in the procedure of transmitting, by the transceiving module 802, the local gradient information of the first network layer in the neural network model with at least one second work node, update parameters of a second network layer in the neural network model in parallel.

[0119] In some examples, the processing module 801 may be a processor such as a CPU, a GPU, or an NPU, and the transceiving module 802 may be a transceiver having a data transceiving function.

[0120] In some examples, the processing module 801 is further configured to determine a dependency relationship between multiple operations of the current iteration based on a connection relationship between multiple network layers in the neural network model, where the multiple operations include at least an transmission operation of the local gradient information of at least one network layer in the neural network model, and an parameter update operation; and perform the multiple operations based on the dependency relationship between multiple operations.

[0121] In some examples, the first work node updates parameters of multiple network layers in the neural network model layer by layer in a reverse order; and/or a network depth of the second network layer is greater than that of the first network layer.

[0122] In some examples, the processing module 801 is specifically configured to, in the procedure of transmitting, by the transceiving module, the local gradient information of the first network layer in the neural network model with at least one second work node, in a case of determining that operations on which a parameter update operation of the second network layer depends have been completed, update the parameters of the second network layer in parallel, where the operations on which the parameter update operation depends include transmitting local gradient information of the second network layer with at least one second work node.

[0123] In some examples, the processing module 801 is further configured to, in the procedure of transmitting, by the transceiving module, the local gradient information of the first network layer in the neural network model with at least one second work node, calculate local gradient information of a third network layer in the neural network model.

[0124] In some examples, the processing module 801 is further configured to perform at least one inner layer iteration on the neural network model to obtain intermediate fusion gradient information corresponding to at least one inner layer iteration;

[0125] the processing module 801 is specifically configured to obtain target fusion gradient information of at least one network layer in the neural network model based on the intermediate fusion gradient information and local gradient information corresponding to the current iteration; the local gradient information of the first network layer transmitted by the first work node with at least one second work node includes target fusion gradient information of the first network layer.

[0126] In some examples, the processing module 801 is specifically configured to perform accumulation processing on the intermediate fusion gradient information and the local gradient information corresponding to the current iteration to obtain the target fusion gradient information of at least one network layer in the neural network model.

[0127] In some examples, the transceiving module 802 is further configured to, in a procedure of obtaining, by the processing module, target fusion gradient information of the third network layer in the neural network model based on the intermediate fusion gradient information and the local gradient information corresponding to the current iteration,

transmit target fusion gradient information of a fourth network layer in the neural network model in parallel with at least one second work node.

[0128] In some examples, the processing module 801 is further configured to amplify each value in the local gradient information of the first network layer by M times, and convert each amplified value into half precision, where M is a real number greater than 1.

[0129] In some examples, the processing module 801 is further configured to convert each value included in obtained local gradient information of the second network layer into single precision, and reduce each converted value by M times to obtain processed gradient information, where M is a real number greater than 1;

[0130] the processing module 801 is specifically configured to use the processed gradient information to update the parameters of the second network layer in the neural network model.

[0131] In some examples, the processing module 801 is further configured to store calculated local gradient information of the first network layer into a pre-allocated target storage space based on an offset corresponding to the first network layer, where the target storage space is used for storing local gradient information of multiple network layers in the neural network model;

[0132] where the local gradient information of the first network layer sent by the transceiving module 802 is acquired from the target storage space based on the offset corresponding to the first network layer; and/or the processing module 801 is further configured to update the local gradient information of the first network layer stored in the target storage space based on received local gradient information of the first network layer from at least one second work node.

[0133] In some examples, the processing module 801 is further configured to store the calculated local gradient information of multiple network layers in the neural network model into a pre-allocated target storage space, and determine an offset corresponding to each of multiple network layers through a memory manager, where the target storage space is a continuous storage space; the first work node acquires local gradient information of at least two of multiple network layers from the target storage space based on the offset corresponding to each of multiple network layers, where at least two network layers include the first network layer. The transceiving module is specifically configured to transmit local gradient information of at least two network layers in the neural network model with at least one second work node.

[0134] FIG. 9 is a schematic structural diagram illustrating another data processing apparatus provided by an example of the present disclosure. As shown in FIG. 9, the data processing apparatus includes:

[0135] an acquiring module 901 configured to acquire a to-be-processed image;

[0136] a processing module 902 configured to perform prediction processing on the to-be-processed image by using the trained neural network model to obtain a prediction result.

[0137] It should be understood that the division of units of the data processing apparatus is only a division of logical functions, and the units of the data processing apparatus may be fully or partially integrated into one physical entity during actual implementation, or be physically separated. For example, the units may be separately established processing elements, or be integrated into the same chip for implementation. In addition, the units may be stored in a storage element of a controller in the form of program codes, and functions of the units are called and executed by a processing element of a processor. In addition, the units may be integrated together, or be implemented independently. The processing element here may be an integrated circuit chip with signal processing capability. In the implementation process, each method step or unit may be completed by an integrated logic circuit of hardware in processor elements or an instruction in the form of software. The processing element may be a general-purpose processor, such as a central processing unit (CPU), or be one or more integrated circuits configured to implement the above method, such as one or more application-specific integrated circuits (ASIC), or one or more digital signal processor (DSP), or one or more field-programmable gate array (FPGA).

[0138] FIG. 10 is a schematic structural diagram illustrating a server provided by an example of the present disclosure. The server 1000 may vary greatly due to different configurations or performances, and include one or more central processing units (CPUs) 1022 (for example, one or more processors) and a memory 1032, one or more storage media 1030 of storing an application 1042 or data 1044 (for example, one or more mass storage devices), one or more acceleration devices (for example, GPU or NPU) 1024. The memory 1032 and the storage media 1030 may be temporary storage or persistent storage. The application stored in the storage media 1030 may include one or more modules (not shown in the drawing), and each module may include a series of instruction operations on the server. Further, the central processing unit 1022 may be configured to communicate with the storage media 1030 to perform a series of instruction operations in the storage media 1030 on the server 1000. The acceleration devices 1024 may execute tasks assigned by the central

processing units 1022, such as image processing tasks. The server 1000 may be the data processing apparatus provided in the examples of the present disclosure.

[0139] The server 1000 may further include one or more power supplies 1026, one or more wired or wireless network interfaces 1050, one or more input/output interfaces 1058, and/or, one or more operating systems 1041, such as Windows Server™, Mac OS X™, Unix™, Linux™ and FreeBSD™.

[0140] The steps performed by the data processing apparatus in the examples may be based on the server structure shown in FIG. 10. Specifically, the acceleration devices 1024 can implement the functions of the processing module 801 in FIG. 8, and the wired or wireless network interfaces 1050 can implement the functions of the transceiving module 802 in FIG. 8. Specifically, the acceleration devices 1024 can implement the functions of the processing module 902 in FIG. 9, and the wired or wireless network interfaces 1050 or the input/output interfaces 1058 can implement the functions of the acquiring module in FIG. 9.

[0141] FIG. 11 is a schematic structural diagram illustrating a terminal device provided by an example of the present disclosure. As shown in FIG. 11, the terminal device 110 includes a processor 1101, a memory 1102 and a communication interface 1103. The processor 1101, the memory 1102 and the communication interface 1103 are connected to each other through a bus 1104. The terminal device in FIG. 11 may be the data processing apparatus in the examples.

[0142] The memory 1102 includes, but is not limited to, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read only memory (EPROM), or a compact disc read-only memory (CDROM). The memory 1102 is used for related instructions and data. The communication interface 1103 is used to receive and transmit data.

[0143] The processor 1101 may include one or more CPUs and one or more GPUs. In the case where the processor 1101 includes one CPU, the CPU may be a single-core CPU or a multi-core CPU. The steps performed by the data processing apparatus in the examples may be based on the structure of the terminal device shown in FIG. 11. Specifically, the processor 1101 can implement the functions of the processing module 801 in FIG. 8, and the communication interface 1103 can implement the functions of the transceiving module in FIG. 8. Specifically, the processor 1101 can implement the functions of the processing module 902 in FIG. 9, and the communication interface 1103 can implement the functions of the acquiring module in FIG. 9.

[0144] The examples of the present disclosure provide a computer readable storage medium. The computer readable storage medium stores a computer program. The computer program is executed by a processor to implement the neural network model training method provided in the above examples.

[0145] The examples of the present disclosure provide a computer readable storage medium. The computer readable storage medium stores a computer program. The computer program is executed by a processor to implement the image predicting method provided in the above examples.

[0146] The examples of the present disclosure provide a computer program product including instructions, which, when running on a computer, cause the computer to implement the neural network model training method provided in the above examples.

[0147] The examples of the present disclosure provide a computer program product including instructions, which, when running on a computer, cause the computer to implement the image predicting method provided in the above examples.

[0148] The above are only specific examples of the present disclosure, but the protection scope of the disclosure is not limited thereto. Any person skilled in the art can easily think of various equivalent modifications or replacements within the technical scope in this disclosure. These modifications or replacements shall be included within the protection scope of the disclosure. Therefore, the protection scope of the present disclosure shall be subject to the protection scope of the claims.

CLAIMS

1. A neural network model training method, comprising:

obtaining, by a first work node, local gradient information of at least one network layer in a neural network model based on a current iteration performed on the neural network model;

in a procedure of transmitting local gradient information of a first network layer in the neural network model with at least one second work node, updating, by the first work node, parameters of a second network layer in the neural network model in parallel.

2. The method of claim 1, further comprising:

determining, by the first work node, a dependency relationship between multiple operations of the current iteration based on a connection relationship between multiple network layers in the neural network model, wherein the multiple operations comprise at least an transmission operation of the local gradient information of the at least one network layer in the neural network model, and an parameter update operation,

performing, by the first work node, the multiple operations based on the dependency relationship between the multiple operations.

3. The method of claim 1 or 2, wherein the first work node updates parameters of the multiple network layers in the neural network model layer by layer in a reverse order; and/or a network depth of the second network layer is greater than that of the first network layer.

4. The method of any of claims 1 to 3, wherein in the procedure of transmitting the local gradient information of the first network layer in the neural network model with the at least one second work node, updating, by the first work node, the parameters of the second network layer in the neural network model in parallel comprises:

in the procedure of transmitting the local gradient information of the first network layer in the neural network model with the at least one second work node, in a case of determining that operations on which an parameter update operation of the second network layer depends have been completed, updating, by the first work node, the parameters of the second network layer in parallel, wherein the operations on which the parameter update operation depends comprise transmitting local gradient information of the second network layer with the at least one second work node.

5. The method of any of claims 1 to 4, further comprising:

in the procedure of transmitting the local gradient information of the first network layer in the neural network model with the at least one second work node, calculating, by the first work node, local gradient information of a third network layer in the neural network model.

6. The method of any of claims 1 to 5, wherein before performing, by the first work node, the current iteration on the neural network model, the method further comprises:

performing, by the first work node, at least one inner layer iteration on the neural network model to obtain intermediate fusion gradient information corresponding to the at least one inner layer iteration;

obtaining, by the first work node, the local gradient information of the at least one network layer in the neural network model based on the current iteration performed on the neural network model comprises: obtaining, by the first work node, target fusion gradient information of the at least one network layer in the neural network model based on the intermediate fusion gradient information and local gradient information corresponding to the current iteration; the local gradient information of the first network layer transmitted by the first work node with the at least one second work node comprises target fusion gradient information of the first network layer.

7. The method of claim 6, wherein obtaining, by the first work node, the target fusion gradient information of the at least one network layer in the neural network model based on the intermediate fusion gradient information and the local gradient information corresponding to the current iteration comprises:

performing, by the first work node, accumulation processing on the intermediate fusion gradient information and the local gradient information corresponding to the current iteration to obtain the target fusion gradient information of the at least one network layer in the neural network model.

8. The method of claim 6 or 7, further comprising:

in a procedure of obtaining target fusion gradient information of the third network layer in the neural network model based on the intermediate fusion gradient information and the local gradient information corresponding to the current iteration, transmitting, by the first work node, target fusion gradient information of a fourth network layer in the neural network model in parallel with the at least one second work node.

9. The method of any of claims 1 to 8, wherein before transmitting the local gradient information of the first network layer in the neural network model with the at least one second work node, the method further comprises:

amplifying, by the first work node, each value in the local gradient information of the first network layer by M times, and converting each amplified value into half precision, wherein M is a real number greater than 1.

10. The method of any of claims 1 to 9, wherein before updating, by the first work node, the parameters of the second network layer in the neural network model in parallel, the method further comprises:

converting, by the first work node, each value comprised in obtained local gradient information of the second network layer into single precision, and reducing each converted value by M times to obtain processed gradient information, wherein M is a real number greater than 1;

updating, by the first work node, the parameters of the second network layer in the neural network model in parallel comprises:

using, by the first work node, the processed gradient information to update the parameters of the second network layer in the neural network model.

11. The method of any of claims 1 to 10, wherein before transmitting the local gradient information of the first network layer in the neural network model with the at least one second work node, the method further comprises:

storing, by the first work node, calculated local gradient information of the first network layer into a pre-allocated target storage space based on an offset corresponding to the first network layer, wherein the target storage space is used for storing local gradient information of the multiple network layers in the neural network model, wherein the local gradient information of the first network layer sent by the first work node is acquired from the target storage space based on the offset corresponding to the first network layer; and/or

updating, by the first work node, the local gradient information of the first network layer stored in the target storage space based on received local gradient information of the first network layer from the at least one second work node.

12. An image predicting method, comprising:

acquiring a to-be-processed image;

performing prediction processing on the to-be-processed image by using a neural network model trained according to any of claims 1 to 11 to obtain a prediction result.

13. A data processing apparatus, comprising:

a processing module configured to obtain local gradient information of at least one network layer in a neural network model based on a current iteration performed on the neural network model;

a transceiving module configured to transmit local gradient information of a first network layer in the neural network model with at least one second work node;

wherein the processing module is further configured to, in the procedure of transmitting, by the transceiving module, the local gradient information of the first network layer in the neural network model with the at least one second work node, update parameters of a second network layer in the neural network model in parallel.

14. The data processing apparatus of claim 13, wherein,

the processing module is further configured to determine a dependency relationship between multiple operations of the current iteration based on a connection relationship between multiple network layers in the neural network model, wherein the multiple operations comprise at least an transmission operation of the local gradient information of the at least one network layer in the neural network model, and an parameter update operation; and perform the multiple operations based on the dependency relationship between the multiple operations.

15. The data processing apparatus of claim 13 or 14, wherein the processing module updates parameters of the multiple network layers in the neural network model layer by layer in a reverse order; and/or

a network depth of the second network layer is greater than that of the first network layer.

16. The data processing apparatus of any of claims 13 to 15, wherein,

the processing module is specifically configured to, in the procedure of transmitting, by the transceiving module, the local gradient information of the first network layer in the neural network model with the at least one second work node, in a case of determining that operations on which an parameter update operation of the second network layer depends have been completed, update the parameters of the second network layer in parallel, wherein the operations on which the parameter update operation depends comprise transmitting local gradient information of the second network layer with the at least one second work node.

17. The data processing apparatus of any of claims 13 to 16, wherein,

the processing module is further configured to, in the procedure of transmitting, by the transceiving module, the local gradient information of the first network layer in the neural

network model with the at least one second work node, calculate local gradient information of a third network layer in the neural network model.

18. The data processing apparatus of any of claims 13 to 17, wherein,

the processing module is further configured to perform at least one inner layer iteration on the neural network model to obtain intermediate fusion gradient information corresponding to the at least one inner layer iteration;

the processing module is specifically configured to obtain target fusion gradient information of the at least one network layer in the neural network model based on the intermediate fusion gradient information and local gradient information corresponding to the current iteration; the local gradient information of the first network layer transmitted by the first work node with the at least one second work node comprises target fusion gradient information of the first network layer.

19. The data processing apparatus of claim 18, wherein,

the processing module is specifically configured to perform accumulation processing on the intermediate fusion gradient information and the local gradient information corresponding to the current iteration to obtain the target fusion gradient information of the at least one network layer in the neural network model.

20. The data processing apparatus of claim 18 or 19, wherein,

the transceiving module is further configured to, in a procedure of obtaining, by the processing module, target fusion gradient information of the third network layer in the neural network model based on the intermediate fusion gradient information and the local gradient information corresponding to the current iteration, transmit target fusion gradient information of a fourth network layer in the neural network model in parallel with the at least one second work node.

21. The data processing apparatus of any of claims 13 to 20, wherein,

the processing module is further configured to amplify each value in the local gradient information of the first network layer by M times, and convert each amplified value into half precision, wherein M is a real number greater than 1.

22. The data processing apparatus of any of claims 13 to 21, wherein,

the processing module is further configured to convert each value comprised in obtained local gradient information of the second network layer into single precision, and reduce each converted value by M times to obtain processed gradient information, wherein M is a real number greater than 1;

the processing module is specifically configured to use the processed gradient information to update the parameters of the second network layer in the neural network model.

23. The data processing apparatus of any of claims 13 to 22, wherein,

the processing module is further configured to store calculated local gradient information of the first network layer into a pre-allocated target storage space based on an offset corresponding to the first network layer, wherein the target storage space is used for storing local gradient information of the multiple network layers in the neural network model;

wherein the local gradient information of the first network layer sent by the transceiving module is acquired from the target storage space based on the offset corresponding to the first network layer; and/or the processing module is further configured to update the local gradient information of the first network layer stored in the target storage space based on received local gradient information of the first network layer from the at least one second work node.

24. A data processing apparatus, comprising:

an acquiring module configured to acquire a to-be-processed image;

a processing module configured to perform prediction processing on the to-be-processed image by using a neural network model trained according to any of claims 1 to 11 to obtain a prediction result.

25. A computer readable storage medium, wherein a computer program is stored in the computer readable storage medium; the computer program comprises program instructions; and when the program instructions are executed by a processor of a mobile device, the processor is caused to implement a method as claimed in any of claims 1 to 12.

26. An electronic device, comprising: a memory and a processor, wherein the memory is used for storing instructions, and the processor is used for executing instructions stored in the memory, so that the processor implements a method as claimed in any of claims 1 to 12.

27. A computer program, wherein when the computer program is executed by a processor, the processor is caused to implement a method as claimed in any of claims 1 to 12.