

CLAIMS

What is claimed is:

1. A model processing method, implemented by a distributed training framework, comprising:

acquiring a first computational graph corresponding to a model to be trained and a parallelization strategy for the model to be trained, wherein the parallelization strategy for the model to be trained comprises at least one of the following: pipeline parallelism, model parallelism, data parallelism, and operator split;

adding, according to the parallelization strategy for the model to be trained, parallelization information into the first computational graph, to obtain a second computational graph;

determining, according to the second computational graph and a computing resource, a distributed computational graph; and

training, according to the distributed computational graph, the model to be trained;

wherein the determining, according to the second computational graph and the computing resource, the distributed computational graph comprises: dividing the computing resource to obtain one or a plurality of virtual devices, wherein the virtual device comprises one or a plurality of physical devices, and obtaining the distributed computational graph based on the parallelization information, the second computational graph and the physical devices, and

wherein the distributed training framework comprises a model and parallel intermediate representation layer, a parallel representation layer in the model and parallel intermediate representation layer is configured to analyze the parallelization strategy for the model to be trained to obtain a corresponding abstraction, wherein the abstraction is configured to unify and express different parallelization strategies, and the distributed training framework implements the parallelization strategy for the model to be trained further based on a graph editing technique of computational graph.

2. The method according to claim 1, wherein the adding, according to the parallelization strategy for the model to be trained, the parallelization information into the first computational graph, to obtain the second computational graph, comprises:

dividing, according to the parallelization strategy for the model to be trained, the first computational graph into a plurality of first subgraphs; and

adding, according to the parallelization strategy for the model to be trained, the parallelization information into each first subgraph in the plurality of first subgraphs, to obtain the second computational graph.

3. The method according to claim 2, wherein the parallelization information comprises the parallelization information between different first subgraphs and the parallelization information in each first subgraph.

4. The method according to claim 3, wherein the parallelization information between different first subgraphs comprises: a parallelization strategy adopted between different first subgraphs.

5. The method according to claim 4, wherein the parallelization information between different first subgraphs further comprises: parameter information of the parallelization strategy adopted between different first subgraphs.

6. The method according to claim 4 or 5, wherein the parallelization strategy adopted between different first subgraphs comprises: the pipeline parallelism and/or the model parallelism.

7. The method according to claim 3, wherein the parallelization information in each first subgraph comprises: a parallelization strategy in each first subgraph.

8. The method according to claim 7, wherein the parallelization information in each first subgraph further comprises: parameter information of the parallelization strategy in each first subgraph.

9. The method according to claim 7 or 8, wherein the parallelization strategy in each first subgraph comprises: the data parallelism and/or the operator split.

10. The method according to claim 2, wherein the obtaining the distributed computational graph based on the parallelization information, the second computational graph and the physical devices, comprises:

converting, according to the parallelization information, the second computational graph to a third computational graph; and

mapping the third computational graph to the physical device, to obtain the distributed computational graph.

11. The method according to claim 10, wherein the dividing the computing resource to obtain one or the plurality of virtual devices, comprises:

dividing, according to the parallelization information, the computing resource, to obtain one or the plurality of virtual devices.

12. The method according to claim 10 or 11, wherein the converting, according to the parallelization information, the second computational graph to the third computational graph, comprises:

converting, according to the parallelization information of each first subgraph in the plurality of first subgraphs, the first subgraph to a distributed second subgraph; and

connecting, according to a connection relationship between the plurality of first subgraphs, the distributed second subgraph corresponding to each first subgraph, to obtain the third computational graph.

13. The method according to claim 12, wherein the mapping the third computational graph to the physical device, comprises:

mapping each second subgraph in the third computational graph to the physical device.

14. The method according to claim 13, wherein the mapping each second subgraph in the third computational graph to the physical device, comprises:

mapping each first subgraph to one virtual device; and

mapping each second subgraph corresponding to the first subgraph to a physical device included in the virtual device corresponding to the first subgraph.

15. The method according to claim 1, wherein the acquiring the parallelization strategy for the model to be trained, comprises:

determining, according to the first computational graph corresponding to the model to be trained, the parallelization strategy for the model to be trained.

16. The method according to claim 1, wherein the acquiring the parallelization strategy for the model to be trained, comprises:

acquiring a user-selected parallelization strategy for the model to be trained.

17. The method according to claim 1, wherein the training, according to the distributed computational graph, the model to be trained, comprises:

converting the distributed computational graph to a distributed computational graph which is recognizable by a training engine; and

inputting the distributed computational graph which is recognizable by the training engine into the training engine, and training, by the training engine, the model to be trained.

18. A model processing apparatus, utilizing a distributed training framework, the apparatus comprising:

an acquiring module, configured for acquiring a first computational graph corresponding to a model to be trained and a parallelization strategy for the model to be trained, wherein the parallelization strategy for the model to be trained comprises at least one of the following: pipeline parallelism, model parallelism, data parallelism, and operator split;

an adding module, configured for adding, according to the parallelization strategy for the model to be trained, parallelization information into the first computational graph, to obtain a second computational graph;

a determining module, configured for determining, according to the second computational graph and a computing resource, a distributed computational graph; and

a training module, configured for training, according to the distributed computational graph, the model to be trained;

wherein the determining module is further configured for dividing the computing resource to obtain one or a plurality of virtual devices, wherein the virtual device comprises one or a plurality of physical devices, and obtaining the distributed computational graph based on the parallelization information, the second computational graph and the physical devices, and

wherein the distributed training framework comprises a model and parallel intermediate representation layer, a parallel representation layer in the model and parallel intermediate representation layer is configured to analyze the parallelization strategy for the model to be trained to obtain a corresponding abstraction, wherein the abstraction is configured to unify and express different parallelization strategies, and the distributed training framework implements the parallelization strategy for the model to be trained further based on a graph editing technique of computational graph.

19. A model processing device, comprising:

a memory;

a processor; and

a computer program;

wherein the computer program is stored in the memory and is configured to be executed by the processor to implement the method according to any one of claims 1-17.

20. A computer-readable storage medium, wherein the computer-readable storage medium stores a computer program, and the computer program, when executed by a processor, implements the method according to any one of claims 1-17.