

CLAIMS

What is claimed is:

1. A model training method, applied to a first processor, wherein the first processor is communicatively connected to a second processor, the first processor comprises a computing core and a first memory, the second processor comprises a second memory, and the first processor and the second processor are processors of different types; and the method comprises:

obtaining, by the computing core, a first target embedding from the second memory;

training, by the computing core, a to-be-trained model based on the first target embedding, to obtain an updated to-be-trained model and a second target embedding, wherein the second target embedding is an updated first target embedding; and

writing, by the computing core, the second target embedding to the first memory.

2. The method according to claim 1, wherein the training, by the computing core, a to-be-trained model based on the first target embedding comprises:

performing, by the computing core, model training of an M^{th} batch on the to-be-trained model based on the first target embedding, wherein M is a positive integer greater than 1; and

after the writing, by the computing core, the second target embedding to the first memory, the method further comprises:

obtaining, by the computing core, the second target embedding from the first memory; and

performing, by the computing core, model training of an $(M + 1)^{\text{th}}$ batch on the updated to-be-trained model based on the second target embedding.

3. The method according to claim 2, wherein the method further comprises:

after the performing, by the computing core, model training of an $(M + 1)^{\text{th}}$ batch on the updated to-be-trained model based on the second target embedding, obtaining a third target embedding, wherein the third target embedding is an updated second target embedding; and

writing, by the computing core, the third target embedding to the first memory.

4. The method according to any one of claims 1 to 3, wherein the obtaining, by the computing core, a first target embedding from the second memory comprises:

when the first target embedding is an embedding required for model training of the M^{th} batch and the first target embedding is not stored in the first memory, obtaining, by the first memory, the first target embedding from the second memory; and

obtaining, by the computing core, the first target embedding from the first memory.

5. The method according to claim 4, wherein the obtaining, by the first memory, the first target embedding from the second memory comprises:

in a process during which the computing core performs model training of an $(M - 1)^{\text{th}}$ batch

on the to-be-trained model, obtaining, by the first memory, the first target embedding from the second memory.

6. The method according to claim 4 or 5, wherein the method further comprises:

before the obtaining, by the computing core, the first target embedding from the first memory, obtaining, by the computing core, a first vector and a target vector that are from the second processor, wherein the first vector indicates embeddings required for model training of the M^{th} batch, each element in the first vector indicates one embedding, and different elements indicate different embeddings; and the target vector indicates permutations of the embeddings indicated by the elements in the first vector when model training of the M^{th} batch is performed, and the embedding indicated by each element appears at least once in the permutations; and

the obtaining, by the computing core, the first target embedding from the first memory comprises:

obtaining, by the computing core, based on the first vector, the embeddings required for model training of the M^{th} batch from the first memory; and

determining, by the computing core, based on the target vector and the embeddings required for model training of the M^{th} batch, the permutations of the embeddings required for model training of the M^{th} batch.

7. The method according to any one of claims 1 to 6, wherein before the training, by the computing core, a to-be-trained model based on the first target embedding, the method further comprises:

obtaining, by the computing core, the third target embedding from a third memory of a third processor, wherein the third target embedding and the first target embedding are embeddings required for model training of the M^{th} batch, and the first processor and the third processor are processors of a same type; and

the training, by the computing core, a to-be-trained model based on the first target embedding comprises:

processing, by the computing core, the first target embedding and the third target embedding by using the to-be-trained model, to obtain a target output, and determining a first gradient based on the target output; and

updating, by the computing core, the to-be-trained model and the first target embedding based on the first gradient.

8. The method according to claim 7, wherein the determining a first gradient based on the target output comprises: determining the first gradient and a second gradient based on the target output; and

after the determining the first gradient and a second gradient based on the target output, the

method further comprises:

transferring the second gradient to a computing core of the third processor, so that the computing core of the third processor updates the third target embedding based on the second gradient.

5 9. The method according to any one of claims 1 to 8, wherein the first processor is a graphics processing unit GPU, an embedded neural network processing unit NPU, or a tensor processing unit TPU, the first memory is a cache, and the second processor is a central processing unit CPU.

10 10. A model training method, applied to a second processor, wherein the second processor is communicatively connected to a first processor, the first processor comprises a computing core and a first memory, the second processor comprises a second memory, and the first processor and the second processor are processors of different types; and the method comprises:

obtaining, by the second processor, embedding information required by the computing core for performing model training of an M^{th} batch and embedding information stored in the first memory, wherein M is a positive integer greater than 1; and

15 writing, by the second processor, a first target embedding to the first memory based on the embedding information required by the computing core for performing model training of the M^{th} batch and the embedding information stored in the first memory, wherein the first target embedding is an embedding that is required for model training of the M^{th} batch and that is not stored in the first memory.

20 11. The method according to claim 10, wherein the writing a first target embedding to the first memory comprises:

in a process during which the computing core performs model training of an $(M - 1)^{\text{th}}$ batch on a to-be-trained model, writing the first target embedding to the first memory.

25 12. The method according to claim 10 or 11, wherein before the writing a first target embedding to the first memory, the method further comprises:

when available storage space of the first memory is less than storage space required for storing the first target embedding, deleting, by the second processor, at least one embedding from the first memory, and writing the at least one embedding to the second memory, wherein the at least one embedding is an embedding that is not required for model training of the M^{th} batch.

30 13. The method according to any one of claims 10 to 12, wherein the method further comprises:

before the writing a first target embedding to the first memory, obtaining, by the second processor, a first vector and a target vector, wherein the first vector indicates embeddings required for model training of the M^{th} batch, each element in the first vector indicates one embedding, and
35 different elements indicate different embeddings; and the target vector indicates permutations of

the embeddings indicated by the elements in the first vector when model training of the M^{th} batch is performed, and the embedding indicated by each element appears at least once in the permutations; and

transferring, by the second processor, the first vector and the target vector to the first processor.

5 ~~14. The method according to any one of claims 10 to 13, wherein the first processor is a graphics processing unit GPU, an embedded neural network processing unit NPU, or a tensor processing unit TPU, the first memory is a cache, and the second processor is a central processing unit CPU.~~

10 ~~15. A model training apparatus, applied to a first processor, wherein the first processor is communicatively connected to a second processor, the first processor comprises a computing core and a first memory, the second processor comprises a second memory, the first processor and the second processor are processors of different types, and the computing core comprises:~~

~~an obtaining unit, configured to obtain a first target embedding from the second memory;~~

15 ~~a model training unit, configured to train a to-be-trained model based on the first target embedding, to obtain an updated to-be-trained model and a second target embedding, wherein the second target embedding is an updated first target embedding; and~~

~~a data writing unit, configured to write the second target embedding to the first memory.~~

20 ~~16. The apparatus according to claim 15, wherein the model training unit is configured to perform model training of an M^{th} batch on the to-be-trained model based on the first target embedding, and M is a positive integer greater than 1;~~

~~the obtaining unit is configured to obtain the second target embedding from the first memory after the computing core writes the second target embedding to the first memory; and~~

~~the model training unit is further configured to:~~

25 ~~perform model training of an $(M + 1)^{\text{th}}$ batch on the updated to-be-trained model based on the second target embedding.~~

~~17. The apparatus according to claim 16, wherein the model training unit is further configured to:~~

30 ~~after performing model training of the $(M + 1)^{\text{th}}$ batch on the updated to-be-trained model based on the second target embedding, obtain a third target embedding, wherein the third target embedding is an updated second target embedding; and~~

~~the data writing unit is further configured to write the third target embedding to the first memory.~~

35 ~~18. The apparatus according to any one of claims 15 to 17, wherein when the first target embedding is an embedding required for model training of the M^{th} batch and the first target embedding is not stored in the first memory, the first memory obtains the first target embedding~~

from the second memory; and

the obtaining unit is specifically configured to:

obtain the first target embedding from the first memory.

19. The apparatus according to claim 18, wherein that the first memory obtains the first target embedding from the second memory comprises:

in a process during which the computing core performs model training of an $(M-1)^{\text{th}}$ batch on the to-be-trained model, the first memory obtains the first target embedding from the second memory.

20. The apparatus according to claim 18 or 19, wherein the obtaining unit is further configured to: before the computing core obtains the first target embedding from the first memory, obtain a first vector and a target vector that are from the second processor, wherein the first vector indicates embeddings required for model training of the M^{th} batch, each element in the first vector indicates one embedding, and different elements indicate different embeddings; and the target vector indicates permutations of the embeddings indicated by the elements in the first vector when model training of the M^{th} batch is performed, and the embedding indicated by each element appears at least once in the permutations;

obtain, based on the first vector, the embeddings required for model training of the M^{th} batch from the first memory; and

determine, based on the target vector and the embeddings required for model training of the M^{th} batch, the permutations of the embeddings required for model training of the M^{th} batch.

21. The apparatus according to any one of claims 15 to 20, wherein the obtaining unit is further configured to:

before the computing core trains the to-be-trained model based on the first target embedding, obtain the third target embedding from a third memory of a third processor, wherein the third target embedding and the first target embedding are embeddings required for model training of the M^{th} batch, and the first processor and the third processor are processors of a same type; and

the model training unit is specifically configured to:

process the first target embedding and the third target embedding by using the to-be-trained model, to obtain a target output, and determine a first gradient based on the target output; and

update the to-be-trained model and the first target embedding based on the first gradient.

22. The apparatus according to claim 21, wherein the model training unit is specifically configured to:

determine the first gradient and a second gradient based on the target output; and

the computing core further comprises a second data transfer unit, configured to: after the first gradient and the second gradient are determined based on the target output, transfer the second

gradient to a computing core of the third processor, so that the computing core of the third processor updates the third target embedding based on the second gradient.

23. The apparatus according to any one of claims 15 to 22, wherein the first processor is a graphics processing unit GPU, an embedded neural network processing unit NPU, or a tensor processing unit TPU, the first memory is a cache, and the second processor is a central processing unit CPU.

24. A model training apparatus, applied to a second processor, wherein the second processor is communicatively connected to a first processor, the first processor comprises a computing core and a first memory, the second processor comprises a second memory, and the first processor and the second processor are processors of different types; and the apparatus comprises:

an obtaining unit, configured to obtain embedding information required by the computing core for performing model training of an M^{th} batch and embedding information stored in the first memory, wherein M is a positive integer greater than 1; and

a data writing unit, configured to write a first target embedding to the first memory based on the embedding information required by the computing core for performing model training of the M^{th} batch and the embedding information stored in the first memory, wherein the first target embedding is an embedding that is required for model training of the M^{th} batch and that is not stored in the first memory.

25. The apparatus according to claim 24, wherein the data writing unit is configured to write the first target embedding to the first memory in a process during which the computing core performs model training of an $(M-1)^{\text{th}}$ batch on a to-be-trained model.

26. The apparatus according to claim 24 or 25, wherein the data writing unit is further configured to: before writing the first target embedding to the first memory, delete at least one embedding from the first memory and write the at least one embedding to the second memory when available storage space of the first memory is less than storage space required for storing the first target embedding, wherein the at least one embedding is an embedding that is not required for model training of the M^{th} batch.

27. The apparatus according to any one of claims 24 to 26, wherein the obtaining unit is further configured to obtain a first vector and a target vector before the data writing unit writes the first target embedding to the first memory, wherein the first vector indicates embeddings required for model training of the M^{th} batch, each element in the first vector indicates one embedding, and different elements indicate different embeddings; and the target vector indicates permutations of the embeddings indicated by the elements in the first vector when model training of the M^{th} batch is performed, and the embedding indicated by each element appears at least once in the permutations; and

the data writing unit is further configured to transfer the first vector and the target vector to the first processor.

28. The apparatus according to any one of claims 24 to 27, wherein the first processor is a graphics processing unit GPU, an embedded neural network processing unit NPU, or a tensor processing unit TPU, the first memory is a cache, and the second processor is a central processing unit CPU.

2914. A model training apparatus, wherein the apparatus comprises a memory and a processor, code is stored in the memory, and the processor is configured to obtain the code and perform the method according to any one of claims 1 to 1413.

30. A computer storage medium, wherein one or more instructions are stored in the computer storage medium, and when the one or more instructions are executed by one or more computers, the one or more computers are enabled to perform the method according to any one of claims 1 to 14.

3115. A computer program product, wherein the computer program product comprises code, and when the code is executed, steps of the method according to any one of claims 1 to 1413 are implemented.