

CLAIMS

What is claimed is:

1. A model parameter training method based on federation learning, wherein the method is only based on data from both federation parties, and the federation parties comprises a first terminal and a second terminal, the method comprising the following operations:

when ~~a~~-the first terminal receives encrypted second data sent by ~~a~~-the second terminal, obtaining a loss encryption value and a first gradient encryption value according to the encrypted second data;

randomly generating a random vector with same dimension as the first gradient encryption value, blurring the first gradient encryption value based on the random vector, and sending the blurred first gradient encryption value and the loss encryption value to the second terminal;

when receiving a decrypted first gradient value and a decrypted loss value returned by the second terminal based on the blurred first gradient encryption value and the loss encryption value, detecting whether a model to be trained is in a convergent state according to the decrypted loss value; and

if the model to be trained is in the convergent state, obtaining a second gradient value according to the random vector and the decrypted first gradient value and determining a sample parameter corresponding to the second gradient value as a model parameter of the model to be trained.

2. The model parameter training method based on federation learning of claim 1, wherein the operation of when a first terminal receives encrypted second data sent by a second terminal, obtaining a loss encryption value and a first gradient encryption value according to the encrypted second data comprises:

when the first terminal receives the encrypted second data sent by the second terminal, obtaining first data and a sample label corresponding to the first data;

calculating a loss value based on the first data, the encrypted second data, the sample label, and a preset loss function, and encrypting the loss value through a homomorphic encryption

algorithm to obtain the encrypted loss value which is the loss encryption value; and

obtaining a gradient function according to the preset loss function, calculating a first gradient value according to the gradient function, and encrypting the first gradient value through the homomorphic encryption algorithm to obtain the encrypted first gradient value which is the first gradient encryption value.

3. The model parameter training method based on federation learning of claim 2, further comprising:

calculating an encryption intermediate result according to the encrypted second data and the first data, encrypting the encryption intermediate result with a preset public key, to obtain a double encryption intermediate result;

sending the double encryption intermediate result to the second terminal, to enable the second terminal to calculate a double encryption gradient value based on the double encryption intermediate result; and

when receiving the double encryption gradient value returned by the second terminal, decrypting the double encryption gradient value through a private key corresponding to the preset public key, and sending the decrypted double encryption gradient value to the second terminal, to enable the second terminal to decrypt the decrypted double encryption gradient value to obtain a gradient value of the second terminal.

4. The model parameter training method based on federation learning of claim 2, further comprising:

receiving encryption sample data sent by the second terminal, obtaining a first partial gradient value of the second terminal according to the encryption sample data and the first data, and encrypting the first partial gradient value through the homomorphic encryption algorithm to obtain the encrypted first partial gradient value which is a second gradient encryption value; and

sending the second gradient encryption value to the second terminal, to enable the second terminal to obtain a gradient value of the second terminal based on the second gradient encryption value and a second partial gradient value calculated according to the second data.

5. The model parameter training method based on federation learning of claim 3, wherein after the operation of detecting whether a model to be trained is in a convergent state according to the decrypted loss value, the method further comprises:

5 if the model to be trained is in a non-convergent state, obtaining a second gradient value according to the random vector and the decrypted first gradient value, updating the second gradient value, and updating the sample parameter according to the updated second gradient value; and

generating a gradient value update instruction and sending the gradient value update
10 instruction to the second terminal, to enable the second terminal to update a gradient value of the second terminal according to the gradient value update instruction, and update the sample parameter according to the updated gradient value of the second terminal.

6. The model parameter training method based on federation learning of claim 1, wherein
15 after the operation of obtaining a second gradient value according to the random vector and the decrypted first gradient value and determining a sample parameter corresponding to the second gradient value as a model parameter of the model to be trained, the method further comprises:

after the first terminal determines the model parameter and receives an execution request, sending the execution request to the second terminal, to enable the second terminal, after
20 receiving the execution request, to return a first prediction score to the first terminal according to the model parameter and a variable value of feature variable corresponding to the execution request;

after receiving the first prediction score, calculating a second prediction score according to the determined model parameter and the variable value of the feature variable corresponding to
25 the execution request; and

adding the first prediction score and the second prediction score to obtain a prediction score sum, inputting the prediction score sum into the model to be trained to obtain a model score, and determining whether to execute the execution request according to the model score.

30 7. The model parameter training method based on federation learning of claim 1, wherein

the operation of detecting whether a model to be trained is in a convergent state according to the decrypted loss value comprises:

obtaining a first loss value previously obtained by the first terminal, and recording the decrypted loss value as a second loss value;

5 calculating a difference between the first loss value and the second loss value, and determining whether the difference is less than or equal to a preset threshold;

when the difference is less than or equal to the preset threshold, determining that the model to be trained is in the convergent state; and

10 when the difference is greater than the preset threshold, determining that the model to be trained is in a non-convergent state.

8. A model parameter training apparatus based on federation learning, wherein the apparatus is only based on data from both federation parties, and the federation parties comprises a first terminal and a second terminal, the apparatus comprising:

15 a data acquisition module configured to, when ~~a~~the first terminal receives encrypted second data sent by ~~a~~the second terminal, obtain a loss encryption value and a first gradient encryption value according to the encrypted second data;

20 a first sending module configured to randomly generate a random vector with same dimension as the first gradient encryption value, blur the first gradient encryption value based on the random vector, and send the blurred first gradient encryption value and the loss encryption value to the second terminal;

25 a model detection module configured to, when receiving a decrypted first gradient value and a decrypted loss value returned by the second terminal based on the blurred first gradient encryption value and the loss encryption value, detect whether a model to be trained is in a convergent state according to the decrypted loss value; and

a parameter determination module configured to, if the model to be trained is in the convergent state, obtain a second gradient value according to the random vector and the decrypted first gradient value and determine a sample parameter corresponding to the second gradient value as a model parameter of the model to be trained.

30

9. The model parameter training apparatus based on federation learning of claim 8, wherein the data acquisition module includes:

a first acquisition unit configured to, when the first terminal receives the encrypted second data sent by the second terminal, obtain first data and a sample label corresponding to the first data;

a first encryption unit configured to calculate a loss value based on the first data, the encrypted second data, the sample label, and a preset loss function, and encrypt the loss value through a homomorphic encryption algorithm to obtain the encrypted loss value which is the loss encryption value; and

a second encryption unit configured to obtain a gradient function according to the preset loss function, calculate a first gradient value according to the gradient function, and encrypt the first gradient value through the homomorphic encryption algorithm to obtain the encrypted first gradient value which is the first gradient encryption value.

10. The model parameter training apparatus based on federation learning of claim 9, further comprising:

a first encryption module configured to calculate an encryption intermediate result according to the encrypted second data and the first data, encrypt the encryption intermediate result with a preset public key, to obtain a double encryption intermediate result;

a first calculation module configured to send the double encryption intermediate result to the second terminal, to enable the second terminal to calculate a double encryption gradient value based on the double encryption intermediate result; and

a second decryption module configured to, when receiving the double encryption gradient value returned by the second terminal, decrypt the double encryption gradient value through a private key corresponding to the preset public key, and send the decrypted double encryption gradient value to the second terminal, to enable the second terminal to decrypt the decrypted double encryption gradient value to obtain a gradient value of the second terminal.

11. The model parameter training apparatus based on federation learning of claim 9, further comprising:

a second encryption module configured to receive encryption sample data sent by the second terminal, obtain a first partial gradient value of the second terminal according to the encryption sample data and the first data, and encrypt the first partial gradient value through the homomorphic encryption algorithm to obtain the encrypted first partial gradient value which is a second gradient encryption value; and

a second sending module configured to send the second gradient encryption value to the second terminal, to enable the second terminal to obtain a gradient value of the second terminal based on the second gradient encryption value and a second partial gradient value calculated according to the second data.

12. The model parameter training apparatus based on federation learning of claim 10, further comprising:

a parameter updating module configured to, if the model to be trained is in a non-convergent state, obtain a second gradient value according to the random vector and the decrypted first gradient value, update the second gradient value, and update the sample parameter according to the updated second gradient value; and

an instruction sending module configured to generate a gradient value update instruction and send the gradient value update instruction to the second terminal, to enable the second terminal to update a gradient value of the second terminal according to the gradient value update instruction, and update the sample parameter according to the updated gradient value of the second terminal.

13. The model parameter training apparatus based on federation learning of claim 10, further comprising:

a third sending module configured to, after the first terminal determines the model parameter and receives an execution request, send the execution request to the second terminal, to enable the second terminal, after receiving the execution request, to return a first prediction score to the first terminal according to the model parameter and a variable value of feature variable corresponding to the execution request;

a second calculation module configured to, after receiving the first prediction score,

calculate a second prediction score according to the determined model parameter and the variable value of the feature variable corresponding to the execution request; and

5 a score acquisition module configured to add the first prediction score and the second prediction score to obtain a prediction score sum, input the prediction score sum into the model to be trained to obtain a model score, and determine whether to execute the execution request according to the model score.

14. The model parameter training apparatus based on federation learning of claim 8, wherein the model detection module includes:

10 a second acquisition unit configured to obtain a first loss value previously obtained by the first terminal, and record the decrypted loss value as a second loss value;

a difference determination unit configured to calculate a difference between the first loss value and the second loss value, and determine whether the difference is less than or equal to a preset threshold;

15 a first determination unit configured to, when the difference is less than or equal to the preset threshold, determine that the model to be trained is in the convergent state; and

a second determination unit configured to, when the difference is greater than the preset threshold, determine that the model to be trained is in a non-convergent state.

20 15. A model parameter training device based on federation learning, wherein the device is only based on data from both federation parties, and the federation parties comprises a first terminal and a second terminal, the device comprising: a memory, a processor, and a model parameter training program based on federation learning stored on the memory and executable on the processor, the model parameter training program based on federation learning, when
25 executed by the processor, implements the following operations:

when a first terminal receives encrypted second data sent by a second terminal, obtaining a loss encryption value and a first gradient encryption value according to the encrypted second data;

30 randomly generating a random vector with same dimension as the first gradient encryption value, blurring the first gradient encryption value based on the random vector, and sending the

blurred first gradient encryption value and the loss encryption value to the second terminal;

when receiving a decrypted first gradient value and a decrypted loss value returned by the second terminal based on the blurred first gradient encryption value and the loss encryption value, detecting whether a model to be trained is in a convergent state according to the decrypted loss value; and

if the model to be trained is in the convergent state, obtaining a second gradient value according to the random vector and the decrypted first gradient value and determining a sample parameter corresponding to the second gradient value as a model parameter of the model to be trained.

16. The model parameter training device based on federation learning of claim 15, wherein the model parameter training program based on federation learning, when executed by the processor, further implements the following operations:

when the first terminal receives the encrypted second data sent by the second terminal, obtaining first data and a sample label corresponding to the first data;

calculating a loss value based on the first data, the encrypted second data, the sample label, and a preset loss function, and encrypting the loss value through a homomorphic encryption algorithm to obtain the encrypted loss value, which is the loss encryption value; and

obtaining a gradient function according to the preset loss function, calculating a first gradient value according to the gradient function, and encrypting the first gradient value through the homomorphic encryption algorithm to obtain the encrypted first gradient value, which is the first gradient encryption value.

17. The model parameter training device based on federation learning of claim 16, wherein the model parameter training program based on federation learning, when executed by the processor, further implements the following operations:

calculating an encryption intermediate result according to the encrypted second data and the first data, encrypting the encryption intermediate result with a preset public key, to obtain a double encryption intermediate result;

sending the double encryption intermediate result to the second terminal, so that the

second terminal calculates a double encryption gradient value based on the double encryption intermediate result; and

when receiving the double encryption gradient value returned by the second terminal, decrypting the double encryption gradient value through a private key corresponding to the preset public key, and sending the decrypted double encryption gradient value to the second terminal, to enable the second terminal to decrypt the decrypted double encryption gradient value to obtain a gradient value of the second terminal.

18. A storage medium, wherein a model parameter training program based on federation learning is stored on the storage medium, the storage medium is only based on data from both federation parties, the federation parties comprises a first terminal and a second terminal, and the model parameter training program based on federation learning, when executed by a processor, implements the following operations:

when a first terminal receives encrypted second data sent by a second terminal, obtaining a loss encryption value and a first gradient encryption value according to the encrypted second data;

randomly generating a random vector with same dimension as the first gradient encryption value, blurring the first gradient encryption value based on the random vector, and sending the blurred first gradient encryption value and the loss encryption value to the second terminal;

when receiving a decrypted first gradient value and a decrypted loss value returned by the second terminal based on the blurred first gradient encryption value and the loss encryption value, detecting whether a model to be trained is in a convergent state according to the decrypted loss value; and

if the model to be trained is in the convergent state, obtaining a second gradient value according to the random vector and the decrypted first gradient value and determining a sample parameter corresponding to the second gradient value as a model parameter of the model to be trained.

19. The storage medium of claim 18, wherein the model parameter training program based on federation learning, when executed by the processor, further implements the following

operations:

when the first terminal receives the encrypted second data sent by the second terminal, obtaining first data and a sample label corresponding to the first data;

calculating a loss value based on the first data, the encrypted second data, the sample label,
5 and a preset loss function, and encrypting the loss value through a homomorphic encryption algorithm to obtain the encrypted loss value, which is the loss encryption value; and

obtaining a gradient function according to the preset loss function, calculating a first gradient value according to the gradient function, and encrypting the first gradient value through the homomorphic encryption algorithm to obtain the encrypted first gradient value,
10 which is the first gradient encryption value.

20. The storage medium of claim 19, wherein the model parameter training program based on federation learning, when executed by the processor, further implements the following operations:

15 calculating an encryption intermediate result according to the encrypted second data and the first data, encrypting the encryption intermediate result with a preset public key, to obtain a double encryption intermediate result;

sending the double encryption intermediate result to the second terminal, so that the second terminal calculates a double encryption gradient value based on the double encryption
20 intermediate result; and

when receiving the double encryption gradient value returned by the second terminal, decrypting the double encryption gradient value through a private key corresponding to the preset public key, and sending the decrypted double encryption gradient value to the second terminal, to enable the second terminal to decrypt the decrypted double encryption gradient
25 value to obtain a gradient value of the second terminal.