(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 738 319**

(51) Int. CI.:

G06N 3/08 (2006.01)
G06N 3/04 (2006.01)
G06N 7/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **12.09.2014 PCT/CN2014/086398**

(87) Fecha y número de publicación internacional: **17.03.2016 WO16037351**

(96) Fecha de presentación y número de la solicitud europea: **12.09.2014 E 14901717 (0)**

(97) Fecha y número de publicación de la concesión europea: **08.05.2019 EP 3192016**

(54) Título: **Sistema informático para entrenar redes neuronales**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
21.01.2020

(73) Titular/es:

MICROSOFT TECHNOLOGY LICENSING, LLC (100.0%)
One Microsoft Way
Redmond, Washington 98052-6399, US

(72) Inventor/es:

LANGFORD, JOHN;
LI, GANG;
SEIDE, FRANK TORSTEN BERND;
DROPPO, JAMES y
YU, DONG

(74) Agente/Representante:

ELZABURU, S.L.P

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Sistema informático para entrenar redes neuronales

Antecedentes

Las redes neuronales profundas son útiles para un rango de problemas de reconocimiento. Por ejemplo, las técnicas de modelado acústico que usan modelos de Markov ocultos de redes neuronales profundas dependientes del contexto (CD-DNN-HMM) para reconocimiento de voz o transcripción de voz a texto supera las técnicas de modelado acústico que usan HMM basado en mezclas Gaussianas convencionales. A diferencia de HMM basado en mezclas Gaussianas, CD-DNN-HMM usa redes neuronales artificiales con múltiples capas ocultas ("redes neuronales profundas") para modelar directamente los estados dependientes del contexto vinculados. Sin embargo, el entrenamiento de CD-DNN-HMM para usarla en reconocimiento de voz consume más tiempo que el entrenamiento de HMM basado en mezcla Gaussiana. La mayor cantidad de tiempo de entrenamiento para las redes neuronales profundas comparado con otros enfoques es un gran obstáculo para el uso de redes neuronales profundas para problemas de reconocimiento, por ejemplo, reconocimiento de voz.

Se han hecho intentos para mejorar el entrenamiento para redes neuronales profundas convencionales mediante el uso de la paralelización, por ejemplo, procesamiento independiente de las expresiones de voz en múltiples servidores. Al final de un lote de cientos de millones de tramas, las estadísticas parciales de los servidores se pueden fusionar, y un modelo actualizado se puede distribuir a los servidores. Sin embargo, el tamaño del modelo actualizado correspondiente a los cientos de millones de tramas a menudo excede la capacidad de los recursos de cálculo disponibles. Por ejemplo, "On parallelizability of stochastic gradient descent for speech DNNS", 2014 IEEE Intertational Conference on Acoustics, Speech and Signal Processing (ICASSP)", describe un método de datos en paralelo para el entrenamiento distribuido de una red neuronal profunda.

Compendio

Esta descripción describe sistemas, métodos, y medios legibles por ordenadores para optimizar matemáticamente soluciones a modelos de cálculo, por ejemplo, entrenamiento de redes neuronales profundas (DNN). En al menos un ejemplo, cada uno de una pluralidad de nodos determina valores de modificación del modelo de cálculo (por ejemplo, valores de gradientes calculados mediante el uso de datos entrenados y el modelo DNN). Los nodos cuantifican los valores de modificación y transmiten los valores cuantificados a los otros nodos. Un módulo de actualización en cada nodo modifica el modelo de cálculo según los valores cuantificados recibidos. Técnicas ejemplares descritas en este documento determinan matrices de gradientes de la DNN, cuantifica las matrices de gradientes mediante el uso de matrices de errores almacenados, actualiza las matrices de errores almacenados, e intercambia las matrices de gradientes cuantificados con otros nodos.

Este Compendio es proporcionado para presentar una selección de conceptos en una forma simplificada que son descritos en mayor profundidad a continuación en la Descripción Detallada. El término "técnicas", por ejemplo, se puede referir a sistema o sistemas, método o métodos, instrucciones legibles por ordenadores, módulo o módulos, algoritmos, lógica de hardware, u operación u operaciones como las permitidas por el contexto descrito anteriormente y a lo largo del documento.

Breve descripción de los dibujos

La descripción detallada es descrita con referencia a las figuras que acompañan. En las figuras, el dígito o dígitos más a la izquierda de un número de referencia identifican la figura en la cual el número de referencia aparece primero. Los mismos números de referencia en diferentes figuras indican elementos similares o idénticos.

La Figura 1 es un diagrama de bloques que representa un entorno ejemplar para implementar un entrenamiento de redes neuronales profundas como se describe en este documento.

La Figura 2 es un diagrama de bloques que ilustra un esquema ejemplar para implementar un motor de entrenamiento que usa un algoritmo para entrenar redes neuronales profundas.

La Figura 3 es un diagrama de bloques que representa un dispositivo informático ejemplar configurado para participar en el entrenamiento de redes neuronales según varios ejemplos descritos en este documento.

La Figura 4 es un diagrama de flujo que ilustra un proceso ejemplar para entrenar redes neuronales profundas.

La Figura 5 es un diagrama de flujo de datos que muestra un proceso ejemplar para intercambiar datos entre nodos para entrenar una red neuronal profunda.

La Figura 6 es un diagrama de flujo que muestra un proceso ejemplar para intercambiar datos entre nodos para entrenar una red neuronal profunda.

Descripción detallada

Visión general

Los ejemplos descritos en este documento proporcionan técnicas y construcciones para mejorar la velocidad del entrenamiento de redes neuronales, por ejemplo, DNN, mediante el intercambio de datos de manera más eficiente mediante el uso de recursos que incluyen, por ejemplo, unidades de procesamiento. Tales recursos pueden ser implementados mediante el uso de programación especializada o hardware programado con instrucciones específicas para implementar las funciones especificadas. Por ejemplo, los recursos pueden tener diferentes modelos de ejecución, como en el caso de las unidades de procesamiento gráficas (GPU) y unidades de procesamiento central (CPU) de ordenadores. Los recursos configurados para el entrenamiento de redes neuronales profundas pueden proporcionar entrenamiento rápido de DNN mediante el uso de arquitecturas de datos paralelos. Esto puede expandir ampliamente los campos de uso en los cuales las DNN pueden servir, y puede permitir una mejora más rápida de los sistemas basados en DNN mediante el aumento del número de ciclos de ajuste del modelo que se pueden ejecutar durante el desarrollo del modelo. En algunos ejemplos, las DNN pueden ser entrenadas rápidamente para convertir voz en texto para personas incapaces de escribir o para convertir texto en voz para el beneficio de los impedidos visuales. En varios ejemplos, las DNN pueden ser entrenadas para facilitar al usuario introducir información o consultas en dispositivos móviles que no tienen teclados o teclados pequeños.

Descritas en este documento hay técnicas mejoradas para entrenar redes neuronales, que incluyen redes neuronales profundas referidas en este documento como DNN, para acelerar el entrenamiento de las DNN para usar en la realización de reconocimiento de patrones y análisis de datos, tales como reconocimiento de voz, síntesis de voz, análisis de regresión u otro filtrado de datos, clasificación de imágenes, o reconocimiento de caras. En varios ejemplos, por ejemplo de DNN entrenadas para reconocimiento de voz u otros casos de uso mencionados en este documento, las DNN pueden ser DNN dependientes del contexto o DNN independientes del contexto. Una DNN puede tener al menos dos capas ocultas. Una red neuronal entrenada mediante el uso de técnicas en este documento puede tener una capa oculta, dos capas ocultas, o más de dos capas ocultas. En un ejemplo, por ejemplo, útil con sistemas de reconocimiento de voz, una red neuronal o DNN como se ha descrito en este documento tiene entre cinco y siete capas. Las técnicas descritas en este documento relacionadas con las DNN también aplican a redes neuronales con menos de dos capas a menos que se indique expresamente lo contrario. En algunos ejemplos, tal como reconocimiento de voz, las DNN dependientes del contexto pueden ser usadas junto con Modelos de Markov ocultos (HMM). En tales ejemplos, la combinación de DNN dependientes del contexto y HMM es conocida como DNN-HMM dependientes del contexto (CD-DNN-HMM). Así, las técnicas descritas en este documento para entrenar DNN pueden ser aplicadas para entrenar las CD-DNN-HMM. Las técnicas descritas en este documento pueden incluir el uso de un algoritmo para paralelizar el entrenamiento de las DNN sobre múltiples unidades de procesamiento, por ejemplo, núcleos de un procesador multinúcleo o múltiples unidades de procesamiento gráfico de propósito general (GPGPU). En consecuencia, múltiples capas de DNN pueden ser procesadas en paralelo en las múltiples unidades de procesamiento.

Las redes neuronales tales como DNN son entrenadas comúnmente con descenso de gradiente estocástico (SGD) basado en mini lotes. Los SGD pueden paralelizarse a lo largo de tres dimensiones, parámetros de modelo, capas, y datos (y combinaciones de los mismos). Las tres técnicas de paralelización son implementadas en esquemas anteriores que sufren de muy alto coste de ancho de banda que restringe la aceleración de la paralelización. Por ejemplo, el paralelismo de datos requiere nodos informáticos para intercambiar y fusionar cientos de millones de parámetros de modelos, que puede llevar significativamente más tiempo que el respectivo cálculo paralelizado. Como una consecuencia, solo se pueden obtener como mucho pequeñas aceleraciones en estos esquemas.

Las técnicas en este documento pueden incluir el uso del modelo de separación. En el modelo de separación, la capa de salida de las DNN o cualquier capa de la DNN oculta puede ser procesada en paralelo a través de múltiples unidades de procesamiento.

Las técnicas pueden reducir la cantidad de tiempo usado para entrenar las DNN para un propósito particular, tal como para reconcomiendo de voz. El tiempo de entrenamiento disminuido puede llevar a un aumento en la implementación y el uso de las DNN al realizar transcripción de voz a texto o síntesis de texto a voz.

En algunos ejemplos, se pueden realizar algoritmos para entrenamiento de DNN como se describe en este documento en un dispositivo informático, tal como un teléfono inteligente, una tableta, un ordenador de sobremesa, un servidor, un servidor Blade, un superordenador, etc. Las DNN resultantes pueden ser usadas en tales dispositivos informáticos. Las DNN resultantes pueden ser usadas en dispositivos informáticos que tienen uno o más dispositivos de entrada, tal como teclados físicos, un teclado de software, una pantalla táctil, un panel táctil, micrófono o micrófonos, cámara o cámaras, etc. para proporcionar funciones optimizadas de dispositivo tales como reconocimiento de voz, reconocimiento y búsqueda de imágenes, y síntesis de voz.

Varios ejemplos, escenarios, y ejemplos de técnicas para el entrenamiento de las DNN para análisis de datos según varios ejemplos son presentados en mayor detalle en la descripción de las siguientes figuras.

Entorno ilustrativo

La Figura 1 muestra un entorno 100 ejemplar en el cual los ejemplos de los sistemas de entrenamiento de redes neuronales profundas (DNN) pueden operar o en el cual métodos de optimización matemática tales como métodos de entrenamiento de DNN pueden realizarse. En algunos ejemplos, los varios dispositivos o componentes del entorno 100 incluye dispositivo o dispositivos 102(1)-102(N) informático (individual o colectivamente referidos en este documento con referencia 102) y dispositivos 104(1)-104(K) informático (individual o colectivamente referidos en este documento con referencia 104) que se pueden comunicar entre sí a través de una o más redes 106. En algunos ejemplos, N = K; en otros ejemplos N > K o N < K. En algunos ejemplos, los dispositivos 102 y 104 informáticos se pueden comunicar con dispositivos externos a través de la red o redes 106.

Por ejemplo, la red o redes 106 pueden incluir redes públicas tales como Internet, redes privadas tales como una intranet institucional o personal, o alguna combinación de redes privadas y públicas. La red o redes 106 pueden también incluir cualquier tipo de red cableada o inalámbrica, que incluye pero no se limita a redes de área local (LAN), redes de área ancha (WAN), redes satelitales, redes por cable, redes Wi-Fi, redes WiMAX, redes de comunicaciones móviles (por ejemplo, 3G, 4G, etc.) o cualquier combinación de las mismas. La red o redes 106 pueden usar protocolos de comunicación, que incluyen protocolos basados en paquetes o basados en datagramas tales como el protocolo de internet (IP), el protocolo de control de transmisión (TCP), el protocolo de datagrama de usuario (UDP), otros tipos de protocolos, o combinaciones de los mismos. Además, la red o redes 106 pueden también incluir varios dispositivos para facilitar las comunicaciones de red o formar una base de hardware para las redes, tales como conmutadores, enrutadores, puertas de enlace, puntos de acceso, cortafuegos, estaciones base, repetidores, dispositivos troncales, y similares. La red o redes 106 pueden también incluir dispositivos que facilitan las comunicaciones entre dispositivos 102, 104 informáticos que usan protocolos de bus de varias topologías, por ejemplo, conmutadores de barras cruzadas, conmutadores INFINIBAND, o conmutadores o concentradores de FIBRE CHANNEL.

En algunos ejemplos, la red o redes 106 pueden además incluir dispositivos que permiten la conexión a una red inalámbrica, tales como un punto de acceso inalámbrico (WAP). Ejemplos soportan la conectividad a través de WAP que envían y reciben datos sobre varias frecuencias electromagnéticas (por ejemplo, frecuencias de radio), que incluyen WAP que soportan los estándares 802.11 del Instituto de Ingenieros Eléctricos y de Electrónica (IEEE) (por ejemplo, 802.11g, 802.11n, etc.), otros estándares, por ejemplo, BLUETOOTH, o múltiples o combinaciones de los mismos.

En varios ejemplos, al menos algunos de los dispositivos 102(1)-102(N) o 104(1)-104(K) informáticos pueden operar en un grupo o configuración agrupada para, por ejemplo, compartir recursos, equilibrar la carga, aumentar el rendimiento, o proporcionar soporte o redundancia de fallos. El dispositivo o dispositivos 102, 104 informáticos pueden pertenecer a una variedad de categorías o clases de dispositivos tales como dispositivos tipo cliente o tipo servidor tradicionales, dispositivos tipo ordenador de sobremesa, dispositivos tipo móvil, dispositivos tipo propósito especial, dispositivos tipo incrustado, o dispositivos tipo llevable. Así, aunque se ilustren como, por ejemplo, ordenadores de sobremesa, ordenadores portátiles, ordenadores tabletas, o teléfonos móviles, el dispositivo o dispositivos 102, 104 informáticos pueden incluir una variedad diversa de tipos de dispositivos y no están limitados a un tipo de dispositivo particular. El dispositivo o dispositivos 102 informáticos pueden representar, pero no se limitan a, ordenadores de sobremesa, ordenadores servidores, ordenadores servidores web, ordenadores móviles, ordenadores portátiles, ordenadores tabletas, ordenadores llevables, dispositivos informáticos implantados, dispositivos de telecomunicaciones, ordenadores de automotriz, televisores habilitados para la red, clientes delgados, asistentes de datos personales (PDA), consolas de videojuegos, dispositivos de juegos, estaciones de trabajo, reproductores multimedia, grabadores de video personales (PVR), decodificadores de televisión, cámaras, componentes integrados para incluir en un dispositivo informático, aparatos, dispositivos informáticos de cliente tipo navegación de ordenador, dispositivos de sistemas de navegación basado en satélite que incluye dispositivos de sistemas de posicionamiento global (GPS) y otros dispositivos de sistemas de navegación basados en satélite, dispositivos de telecomunicaciones tales como teléfonos móviles, ordenadores tabletas, dispositivos híbridos teléfono móvil – tableta, asistentes de datos personales (PDA), u otro u otros dispositivos informáticos configurados para participar en el entrenamiento u operación de la DNN como se describe en este documento. En al menos un ejemplo, el dispositivo o dispositivos 102 informáticos incluyen servidores u ordenadores de alto rendimiento configurados para entrenar DNN. En al menos un ejemplo, el dispositivo o dispositivos 104 incluyen portátiles, ordenadores tabletas, teléfonos inteligentes, ordenadores personales de sobremesa, u otro u otros dispositivos informáticos configurados para operar DNN entrenadas, por ejemplo, para proporcionar datos de texto en respuesta a entrada de voz desde un micrófono.

El dispositivo o dispositivos 102, 104 informáticos pueden incluir varios componentes ilustrados en el recuadro 108. El dispositivo o dispositivos 102, 104 informáticos pueden incluir cualquier dispositivo informático que tenga una o más unidades 110 de procesamiento operables conectadas a uno o más medios 112 legibles por un ordenador tal como a través de un bus 114, que en algunos casos puede incluir uno o más de un sistema de bus, un bus de datos, un bus de direcciones, un bus PCI, un bus Mini-PCI, y cualquier variedad de buses locales, periféricos o independientes, o cualquier combinación de los mismos. En al menos un ejemplo, plurales unidades 110 de procesamiento pueden intercambiar datos a través de un bus de interfaz interna (por ejemplo PCIe), en vez de o además de la red 106. Instrucciones ejecutables almacenadas en medios 112 legibles por un ordenador pueden

incluir, por ejemplo, un sistema 116 operativo, un motor 118 de entrenamiento de DNN, un motor 120 de operación de DNN, y otros módulos, programas, o aplicaciones que sean cargables y ejecutables por una unidad o unidades 110 de procesamiento. En un ejemplo no mostrado, una o más de las unidades 110 de procesamiento en uno del dispositivo o dispositivos 102, 104 informáticos pueden ser operables conectadas a medios 112 legibles por un ordenador en un dispositivo 102, 104 informático diferente, por ejemplo, a través de una interfaz 122 de comunicaciones y red 106. Por ejemplo, el código de programa para realizar los pasos de entrenamiento de la DNN en este documento puede ser descargado de un servidor, por ejemplo, dispositivo 102(1) informático, a un cliente, por ejemplo, dispositivo 104(K) informático, por ejemplo, a través de la red 106, y ejecutado por una o más unidades 110 de procesamiento en el dispositivo 104(K) informático. En un ejemplo, el dispositivo o dispositivos 102(1)-102(N) incluyen un motor 118 de entrenamiento de DNN, y uno o unos dispositivos 104(1)-104(K) informáticos incluyen un motor 120 de operación de DNN.

La unidad o unidades 110 de procesamiento pueden ser o incluir uno o más procesadores de núcleo único, procesadores de múltiples núcleos, unidades de procesamiento central (CPU), unidades de procesamiento gráfico (GPU), unidades de procesamiento gráfico de propósito general (GPGPU), o componentes de lógica de hardware tales como aceleradores configurados, por ejemplo, a través de la programación desde módulos o API, para realizar funciones descritas en este documento. Por ejemplo, y sin limitación, tipos ilustrativos de componentes de lógica de hardware que pueden ser usados en o como unidades 110 de procesamiento que incluyen Matrices de Puertas Programables por Campo (FPGA), Circuitos Integrados de específicos de Aplicación (ASIC), Productos Estándares específicos de Aplicación (ASSP), Sistemas de Sistemas en un chip (SOC), Dispositivos de Lógica Programable Compleja (CPLD), y Procesadores de Señales Digitales (DSP). Por ejemplo, un acelerador puede representar un dispositivo híbrido, tal como uno de ALTERA o XILINX que incluyen un núcleo de la CPU incrustado en una estructura FPGA. Estos u otros componentes de lógica de hardware pueden operar de manera independiente, o en algunos casos, pueden ser dirigidos por una CPU.

La unidad o unidades 110 de procesamiento pueden configurarse para ejecutar un sistema 116 operativo que es instalado en el dispositivo 102 informático. En algunos ejemplos, la unidad o unidades 110 de procesamiento pueden ser o incluir unidades de procesamiento de gráficos de propósito general (GPGPU). En otros ejemplos, las unidades 110 de procesamiento pueden ser matrices de puertas programables por campo (FPGA), u otro tipo de procesador personalizable. En varios ejemplos, al menos algunos de los dispositivos 102(1)-102(N) informáticos pueden incluir una pluralidad de unidades 110 de procesamiento de múltiples tipos. Por ejemplo, las unidades 110 de procesamiento en el dispositivo 102(1) informático pueden ser una combinación de una o mas GPGPU y una o más FPGA.

El dispositivo 102 informático puede también incluir una o más interfaces 122 de comunicaciones para habilitar comunicaciones por cable o inalámbricas entre el dispositivo 102 informático y otros dispositivos 102 informáticos implicados en el entrenamiento de la DNN, u otro u otros dispositivos informáticos, sobre la red o redes 106. Tal o tales interfaces 122 de comunicaciones pueden incluir uno o más dispositivos transceptores, por ejemplo, controladores de interfaces de red (NIC) tales como NIC de Ethernet, para enviar y recibir comunicaciones sobre una red. Las unidades 110 de procesamiento pueden intercambiar datos a través de la interfaz 122 de comunicaciones. En un ejemplo, la interfaz 122 de comunicaciones puede ser un transceptor exprés de Interconexión de Componentes Periféricos (PCIe), y la red 106 puede ser un bus de PCIe. En algunos ejemplos, la interfaz 122 de comunicaciones puede incluir, pero no limitarse a, un transceptor para transmisiones celulares, Wi-Fi, banda ultra ancha (UWB), BLUETOOTH, o satélite. La interfaz 122 de comunicaciones puede incluir una interfaz de I/O por cable, tal como una interfaz Ethernet, una interfaz serie, una interfaz de Bus Serie Universal (USB), una interfaz de INFINIBAND, u otras interfaces por cable. Para simplificar, estos y otros componentes son omitidos del dispositivo 102 informático ilustrado.

Mientras que las unidades 110 de procesamiento son descritas como residentes en el dispositivo 102 informático y conectadas por la interfaz 122 de comunicaciones en varios ejemplos, las unidades 110 de procesamiento pueden también residir en diferentes unidades informáticas en algunos ejemplos. En algunos ejemplos, las unidades 110 de procesamiento pueden residir en dispositivos 102 informáticos correspondientes, y pueden intercambiar datos a través de una red 106 a través de la interfaz 122 de comunicaciones. En algunos ejemplos, al menos dos de las unidades 110 de procesamiento pueden residir en diferentes dispositivos 102 informáticos. En tales ejemplos, múltiples unidades 110 de procesamiento en el mismo dispositivo 102 informático pueden usar un bus 114 de interfaz del dispositivo 102 informático para intercambiar datos, mientras que las unidades 110 de procesamiento en diferentes dispositivos 102 informáticos pueden intercambiar los datos a través de la red o redes 106.

Los legibles por ordenador descritos en este documento, por ejemplo medios 112 legibles por ordenador, pueden incluir medios de almacenamiento informático y/o medios de comunicación. Los medios de almacenamiento informático pueden incluir unidades de almacenamiento tangibles tales como memoria volátil, memoria no volátil, u otros medios de almacenamiento informático persistente o auxiliar, medios de almacenamiento informático extraíbles o no extraíbles implementados en cualquier método o tecnología para almacenar información tal como instrucciones legibles por un ordenador, estructuras de datos, módulos de programa, u otros datos. Los medios 112 legibles por un ordenador o memoria 320, Figura 3, pueden ser un ejemplo de medios de almacenamiento informático. Así, los medios 112 legibles por un ordenador o memoria 320 incluyen formas físicas o tangibles de medios incluidos en un dispositivo o componente de hardware que es parte de un dispositivo o externo a un dispositivo, que incluye pero no

se limita a memoria de acceso aleatorio (RAM), memoria de acceso aleatorio estática (SRAM), memoria de acceso aleatorio dinámica (DRAM), memoria de cambio de fase (PRAM), memoria de solo lectura (ROM), memoria de solo lectura programable y borrable (EPROM), memoria de solo lectura programable y borrable de manera electrónica (EEPROM), memoria flash, memoria de solo lectura de disco compacto (CD-ROM), discos versátiles digitales (DVD), tarjetas ópticas u otros medios de almacenamiento óptico, casetes magnéticos, cinta magnética, almacenamiento de disco magnético, tarjetas magnéticas u otros dispositivos o medios de almacenamiento magnético, dispositivos de memoria de estado sólido, matrices de almacenamiento, almacenamiento anexo a la red, redes de área de almacenamiento, almacenamiento de ordenador alojado o cualquier otra memoria de almacenamiento, dispositivo de almacenamiento, o medio de almacenamiento que pueda ser usado para almacenar y mantener información para acceder por un dispositivo informático.

En contraste con los medios de almacenamiento informático, los medios de comunicación pueden representar instrucciones legibles por ordenador, estructuras de datos, módulos de programas, u otros datos en una señal de datos modulada, tal como una onda portadora, u otro mecanismo de transmisión. Como se define en este documento, los medios de almacenamiento informático no incluyen medios de comunicación.

En algunos ejemplos, los medios 112 legibles por un ordenador pueden almacenar instrucciones ejecutables por la unidad o unidades 110 de procesamiento que, como se discutió anteriormente, pueden representar una unidad de procesamiento incorporada en el dispositivo 102 informático. Los medios 112 legibles por un ordenador pueden también almacenar instrucciones ejecutables por unidades de procesamiento externas tales como por una CPU externa o procesador o acelerador externo de cualquier tipo discutido anteriormente. En varios ejemplos al menos una unidad 110 de procesamiento, por ejemplo, una CPU, GPU, o acelerador, es incorporada en el dispositivo 102 informático, mientras que en otros ejemplos al menos una unidad 110 de procesamiento, por ejemplo uno o más de una CPU, GPU, o acelerador, es externa al dispositivo 102 informático.

Los medios 112 legibles por un ordenador del dispositivo 102 informático pueden almacenar un sistema 116 operativo. En algunos ejemplos, el sistema 116 operativo no es usado (comúnmente referido como una configuración “metal desnudo”). En varios ejemplos, el sistema 116 operativo puede incluir componentes que permiten o dirigen al dispositivo 102 informático para recibir datos a través de varias entradas (por ejemplo, controles de usuario, interfaces de redes o comunicaciones, o dispositivos de memoria), y procesa los datos mediante el uso de la unidad o unidades 110 de procesamiento para generar la salida. El sistema 116 operativo puede además incluir uno o más componentes que presentan la salida (por ejemplo, presenta una imagen en un elemento de presentación electrónica, almacena datos en memoria, transmite datos a otro dispositivo electrónico, etc.). El sistema 116 operativo puede permitir que un usuario interactúe con módulos del motor 118 de entrenamiento mediante el uso de una interfaz (no mostrada). De manera adicional, el sistema 116 operativo puede incluir componentes que realicen varias funciones generalmente asociadas con un sistema operativo, por ejemplo, gestión del almacenamiento y gestión del dispositivo interno.

Componentes ilustrativos

La Figura 2 es un diagrama de bloques que ilustra una técnica 200 ejemplar para implementar un motor 202 de entrenamiento, tal como un motor 118 de entrenamiento, que usa algoritmos para entrenar una red neuronal profunda (DNN) 204 (o una pluralidad de DNN, e igualmente a lo largo), y para implementar un motor 206 de análisis de datos, tal como un motor 120 de operación de la DNN, para operar la DNN 208 entrenada. El motor 202 de entrenamiento puede estar implementado mediante el uso de un dispositivo 210 informático, que, en algunos casos, puede incluir uno o unos dispositivos 102 informáticos. El motor 206 de análisis de datos puede estar implementado mediante un dispositivo informático tal como uno o unos dispositivos 104 informáticos. Para claridad, un dispositivo informático separado que implementa un motor 206 de análisis de datos no es mostrado en la Figura 2. En al menos un ejemplo, el dispositivo 210 informático implementa tanto el motor 202 de entrenamiento como el motor 206 de análisis de datos. El dispositivo 210 puede incluir una o más unidades 212(1)-212(N) de procesamiento, que pueden representar unidades 110(1)-110(N) de procesamiento como se discutieron anteriormente con referencia a la Figura 1. Las unidades 212(1)-212(N) de procesamiento son referidas de manera individual o colectiva en este documento con referencia 212. En algunos ejemplos, las unidades 212 de procesamiento pueden ser unidades 212 de procesamiento como se discutieron anteriormente con referencia a la Figura 1, por ejemplo, GPGPU. Las unidades 212 de procesamiento puede intercambiar datos a través de un bus 114 o una red 106, ambos en la Figura 1. Las unidades 212 de procesamiento pueden llevar a cabo instrucciones del bloque 214 de entrenamiento de DNN que incluye DNN 204, motor 202 de entrenamiento, datos 216 de entrenamiento, y mini lotes 218 de datos 216 de entrenamiento. Los mini lotes 218 se discutirán a continuación.

El entrenamiento de la DNN puede ser realizado por múltiples nodos en paralelo para reducir el tiempo requerido para el entrenamiento. A lo largo de esta descripción, el término “nodo” se refiere a un dispositivo o parte de un dispositivo configurado como parte de tal disposición de entrenamiento de DNN paralelo. En al menos un ejemplo, el motor 202 de entrenamiento se ejecuta en cada uno de una pluralidad de dispositivos 210 informáticos, y cada dispositivo 210 informático tiene exactamente una unidad 212 de procesamiento de núcleo único. Cada uno de tales dispositivos 210 informáticos es un nodo en este ejemplo. En algunos ejemplos, el motor 202 de entrenamiento se ejecuta en un único dispositivo 210 informático que tiene una pluralidad de unidades 212 de procesamiento de múltiples núcleos. En tales ejemplos, cada núcleo de las unidades 212 de procesamiento de múltiples núcleos

representa un nodo. Otras combinaciones, y puntos entre estos extremos, pueden también usarse. Por ejemplo, un acelerador individual (por ejemplo, una FPGA) puede incluir uno o más nodos. En otros ejemplos, múltiples núcleos de una unidad 212 de procesamiento pueden configurarse para operar juntos como un único nodo.

El motor 202 de entrenamiento puede usar un algoritmo 220 para entrenar la DNN 204 para realizar análisis de datos, tal como para usar en reconocimiento de voz. La DNN 204 puede ser un perceptrón multicapa (MLP). Como tal, la DNN 204 puede incluir una capa 222(1) de entrada inferior y una capa 222(L) superior (entero L > 1), así como múltiples capas ocultas, tal como las múltiples capas 222(2)-222(3). Las capas 222(1)-222(L) son referidas de manera individual o colectiva en este documento con referencia 222. En algunos ejemplos que usan DNN dependientes del contexto, la DNN 204 puede incluir un total de ocho capas (N = 8). En varios ejemplos, la DNN 204 pueden ser DNN dependientes del contexto o DNN independientes del contexto. Los datos 216 de entrenamiento pueden ser usados por el algoritmo 220 como datos de entrenamiento para entrenar la DNN 204. Los datos 216 de entrenamiento pueden incluir un cuerpo de voz que incluye datos de audio de una colección de voz de muestra de humanos. Por ejemplo, el cuerpo de voz puede incluir voces en inglés de Norte América recogidas de hablantes de inglés de Norte América en los Estados Unidos y Canadá. Sin embargo, en otros ejemplos, los datos 216 de entrenamiento pueden incluir voz de muestra en otras lenguas respectivas (por ejemplo, chino, japonés, francés, etc.), dependiendo del idioma deseado de la voz a ser reconocida, u otros tipos de datos de entrenamiento para diferentes aplicaciones como reconocimiento de escritura o clasificación de imágenes. Los datos 216 de entrenamiento pueden también incluir información sobre el reconocimiento correcto o respuestas de clasificación para el cuerpo. Mediante el uso de esta información, se pueden detectar errores en el procesamiento del cuerpo por la DNN 204. Esta información puede ser usada, por ejemplo, en el cálculo del gradiente con respecto a los parámetros del modelo del valor D del criterio de entropía cruzada de, por ejemplo, la Ecuación (1) a continuación. En varios ejemplos, esta información es usada en el cálculo de un valor de un criterio tal como una función de error cuadrático medio (“coste cuadrático”). Los datos 216 de entrenamiento pueden también incluir un conjunto de prueba de un segundo cuerpo y datos de clasificación correctos para ese segundo cuerpo. El rendimiento de la DNN 204 puede ser evaluado en el conjunto de prueba para ajustar el entrenamiento de forma que la DNN 204 desempeña con eficacia más allá de los límites del cuerpo de entrenamiento.

Los cálculos realizados por el algoritmo 220 pueden paralelizarse sobre las unidades 212 de procesamiento. Por ejemplo, durante la propagación hacia atrás, un cálculo en los datos de entrada realizado por la unidad 212(1) de procesamiento puede producir un primer resultado de cálculo. El primer resultado de cálculo puede ser canalizarse a la unidad 212(2) de procesamiento para un cálculo adicional para generar un segundo resultado de cálculo. De manera concurrente con la generación del segundo resultado de cálculo, la unidad 212(1) de procesamiento puede procesar datos de entrada adicionales para generar un tercer resultado de cálculo. En al menos algunos ejemplos, de manera concurrente con la generación del segundo resultado de cálculo, la unidad 212(1) de procesamiento puede transferir al menos parte del primer resultado de cálculo a otra unidad 212 de procesamiento. Tales cálculos concurrentes por las unidades 212 de procesamiento u otros ejemplos de nodos pueden resultar en una canalización de cálculos que entrenan la DNN 204, y, en consecuencia, en una reducción de tiempo de cálculo debido al paralelismo resultante de cálculo. Cálculos concurrentes y comunicación por las unidades 212 de procesamiento u otros ejemplos de nodos pueden resultar en tiempo de retraso reducido de espera de datos para llegar al nodo y, en consecuencia, a una reducción del tiempo de cálculo general.

En varios ejemplos, los cálculos realizados por el algoritmo 220 pueden ser mejorados mediante el uso de una o más técnicas, tales como selección 224 de lote, cuantificación 226, separación 228 del modelo, intercambio 230, y paralelización 232 de transferencia de datos. Dado que los datos 216 de entrenamiento son procesados por el algoritmo como mini lotes 218 de muestras de entrada, como se discutió anteriormente, la selección 224 de lotes puede incluir configurar el tamaño de los lotes o mini lotes de muestras de entrada para equilibrar la precisión de cálculo y la eficiencia de ejecución según los criterios seleccionados. En un ejemplo, el tamaño puede ser seleccionado para maximizar tanto la precisión del cálculo como la eficiencia de ejecución del algoritmo 220. En un ejemplo, el tamaño puede ser seleccionado para maximizar la eficiencia de ejecución del algoritmo 220 mientras proporciona un nivel seleccionado de precisión del cálculo. La selección 224 de lotes puede ser realizada como parte del algoritmo 220 o como un módulo de código separado del algoritmo 220, como se muestra. En al menos un ejemplo, los gradientes son calculados para sub mini lotes como se describe a continuación.

Además, las capas 222(1)-222(L) en la DNN 204 pueden tener tamaños que varían debido a las diferencias en el número de unidades en varias capas de la DNN 204. Por ejemplo, una capa más grande en la DNN 204 puede tener un tamaño que es diez veces mayor que la de una o más de las capas más pequeñas. En consecuencia, puede ser más eficiente dedicar un procesador de núcleo múltiple particular para procesar la capa más grande, mientras se procesan dos o más de las capas más pequeñas en otro procesador de núcleo múltiple. Tal agrupación puede reducir los retrasos de ida y vuelta y mejorar la eficiencia.

La cuantificación 226 es reducir la cantidad de información a ser enviada entre los nodos mediante la reducción de la precisión con la cual los valores de los datos son representados. El algoritmo 220 puede transferir, por ejemplo, valores de gradientes de un modelo de red neuronal entre nodos. Los valores de gradientes pueden ser valores de punto flotante del IEEE 754 de 32 bits (valores “flotantes” del lenguaje C), valores (“dobles” de C) de 64 bits, o valores de otras profundidades de bit. La cuantificación 226 puede incluir transmitir representaciones (por ejemplo, aproximaciones) de los valores de gradientes desde un nodo, las representaciones que usan menos bits que los

valores de gradientes, por ejemplo, menos de 32 bits, por ejemplo, 1 bit. La cuantificación 226 también incluye la operación inversa de “reconstrucción”, esto es, interpretar valores cuantificados recibidos en un nodo como valores de 32 bits específicos u otros valores que tienen mayor precisión o profundidad de bit que los valores de gradientes, por ejemplo, más de un bit. La cuantificación 226 incluye hacer seguimiento de “valores de errores”, valores que representan la diferencia entre los valores de gradientes y sus representaciones cuantificadas, y determinar valores de cuantificación basados parcialmente en los valores de errores. Esto permite ventajosamente mantener la precisión del proceso de entrenamiento mediante la dispersión del error de cuantificación sobre valores de gradientes sucesivos. La cuantificación 226 es discutida a continuación, por ejemplo, con referencia a las Ecuaciones (10), (11), (12), (13), (14), y (15). En un ejemplo discutido a continuación con referencia a la ecuación (10), los valores de cuantificación son determinados mediante la suma del error de un mini lote anterior a un valor de gradiente de un mini lote actual antes de la cuantificación.

La separación 228 del modelo es el procesamiento de partes del modelo de cálculo de la DNN 204 mediante múltiples, unidades de procesamiento respectivas, tales como una pluralidad de los procesadores de las unidades 212 de procesamiento. La separación 228 del modelo también es referida en este documento como “modelo de paralelismo”.

El intercambio 230 es la transmisión de valores de gradientes entre nodos. Esto permite actualizaciones del modelo para ser calculado de manera efectiva en una forma de datos paralelos sobre un gran número de nodos. Esto a su vez reduce el tiempo transcurrido requerido para entrenar la DNN 204. En varios ejemplos, el intercambio 230 es realizado en cooperación con la cuantificación 226 para intercambiar valores de gradientes cuantificados entre los nodos. El intercambio 230 puede incluir particionar los valores de gradientes y realizar una distribución totalmente reducida para proporcionar actualizaciones de los valores de gradientes a todos los nodos. El intercambio 230 es discutido a continuación con referencia a las Figuras 5 y 6.

Una iteración de cálculo del algoritmo 220 puede ejecutar los siguientes pasos: propagación hacia delante de datos de entrada, propagación hacia atrás del error, y actualización del modelo. La paralelización 232 de la transferencia de datos puede incluir paralelizar el flujo de los datos de salida desde una iteración de cálculo del algoritmo 220 con otros pasos en la iteración de cálculo. Por ejemplo, matrices de gradientes pueden ser transferidas de manera concurrente con el cálculo. En casos en los cuales el tiempo de flujo es más corto que el tiempo de cálculo, tal paralelización puede reducir o eliminar retraso de tiempo en realizar los cálculos debido al intercambio de datos entre las unidades de procesamiento durante la ejecución del algoritmo 220. En al menos un ejemplo los pasos de la propagación hacia adelante, la propagación hacia atrás, y la actualización del modelo son realizados en ese orden.

Así, mediante el uso del algoritmo 220 y los datos 216 de entrenamiento, el motor 202 de entrenamiento puede producir DNN 208 entrenadas a partir de la DNN 204. A su vez, el motor 206 de análisis de datos puede usar la DNN 208 entrenada para producir datos 234 de salida a partir de los datos 236 de entrada. En algunos ejemplos, el motor 206 de análisis de datos puede ser un motor de voz a texto que usa la DNN 208 entrenada en la forma de una DNN-HMM dependiente del contexto entrenada. El motor de voz a texto puede usar la DNN-HMM dependiente del contexto entrenada para producir datos 234 de salida en la forma de texto de salida a partir de los datos 236 de entrada que está en la forma de voz de entrada. El motor 206 de análisis de datos puede ser ejecutado en el dispositivo 210 de cálculo o un dispositivo que sea similar al dispositivo 210 de cálculo. Además, el motor 206 de análisis de datos puede recibir datos 236 de entrada en vivo desde un micrófono y componentes de procesamiento de audio del dispositivo 210 de cálculo, que puede ser, por ejemplo, un dispositivo 104(5) de cálculo de un teléfono inteligente, Figura 1. En varios ejemplos el motor 206 de análisis de datos puede recibir datos 236 de entrada desde un archivo o flujo multimedia, por ejemplo con el propósito de indexar audio del contenido hablado en el archivo/flujo multimedia. En algunos ejemplos, el motor 206 de análisis de datos puede también ser un motor de texto a voz que usa las DNN dependientes del contexto para sintetizar voz de salida (datos 234 de salida) en base al texto de entrada (datos 236 de entrada), o un motor de reconocimiento de escritura.

En algunos ejemplos, el algoritmo 220, mejorado con una o más de las técnicas descritas en este documento, por ejemplo, las técnicas 224-232, puede implementarse para producir una DNN 208 independiente del contexto entrenada bajo otros escenarios que exhiban características similares. De este modo, las formas independientes del contexto de la DNN 204 pueden ser entrenadas con datos de entrenamiento apropiados para una variedad de propósitos de análisis de datos. Las características pueden incluir un conjunto más grande de datos de entrenamiento que resulte en tiempo de procesamiento prolongado (por ejemplo, muestras más grandes de 50 millones, 1,3 mil millones, etc.), las estructuras de las DNN en las cuales la salida de cada red de las DNN excede un umbral (por ejemplo, mayor que dos mil, cuatro mil, etc. salidas desde una DNN), etc. Los propósitos del análisis de datos pueden incluir usar DNN independientes del contexto entrenadas para actividades tales como reconocimiento de imágenes, reconocimiento de escritura, visión de ordenador, seguimiento de video, etc.

La Figura 3 es un diagrama ilustrativo que muestra componentes ejemplares de un dispositivo 300 informático, que puede representar el dispositivo o dispositivos 102, 104, 210 informáticos. El dispositivo 300 informático puede implementar el motor 302 de entrenamiento, tal como el motor 118, 202 de entrenamiento para entrenar la DNN 304. El dispositivo 300 informático puede ser usado para determinar soluciones a uno o más problemas de optimización matemática, por ejemplo, problemas de minimización matemática. Por ejemplo, el entrenamiento de DNN mediante un proceso de descenso de gradiente estocástico (SGD) ejemplar puede implicar minimizar matemáticamente, por

ejemplo, una entropía cruzada D (Ecuación (1)). El dispositivo 300 informático puede configurarse para incluir u operar como uno o más nodos. En varios ejemplos la DNN 304 puede ser una DNN dependiente del contexto o una DNN independiente del contexto.

El dispositivo 300 informático puede incluir una o más unidades 306 de procesamiento, que pueden representar la unidad o unidades 110, 212 de procesamiento. La unidad o unidades 306 de procesamiento pueden incluir, por ejemplo, tipos de unidades de procesamiento descritas anteriormente tales como unidades de procesamiento de tipo CPU –o CPGPU–. En varios ejemplos, el dispositivo 300 informático puede ser un servidor, un ordenador de sobremesa, cualquier tipo de dispositivo electrónico, o cualquier tipo de dispositivo mencionado anteriormente, o una combinación de ellos, que sea capaz de alojar una o más unidades 306 de procesamiento para procesar datos.

El dispositivo 300 informático puede también incluir una interfaz 308 de comunicaciones, que puede representar la interfaz 122 de comunicaciones. Por ejemplo, la interfaz 308 de comunicaciones puede incluir un dispositivo transceptor tal como una NIC para enviar y recibir comunicaciones sobre una red, por ejemplo, como se discutió anteriormente. Como tal, el dispositivo 300 informático puede tener capacidades de red. Por ejemplo, el dispositivo 300 informático puede intercambiar datos con otros dispositivos informáticos (por ejemplo, portátiles, ordenadores, servidores, etc.) a través de una o más redes 106, tales como Internet.

El dispositivo 300 informático puede además incluir uno o más interfaces 310 de entrada/salida (I/O) para permitir al dispositivo 300 informático comunicarse con dispositivos de entrada/salida (no mostrados) tales como dispositivos de entrada de usuario que incluye dispositivos de entrada periféricos (por ejemplo, un teclado, un teclado numérico, un ratón, un lápiz, un controlador de juegos, un dispositivo de entrada de voz como un micrófono, dispositivo de reconocimiento de voz, un dispositivo de entrada táctil, un dispositivo de entrada de gestos tal como una pantalla táctil, y similares) y dispositivos de salida que incluyen dispositivos de salida periféricos (por ejemplo, un elemento de presentación visual, una impresora, altavoces de audio, una salida háptica, y similares). El dispositivo 300 informático puede comunicarse a través de la interfaz 310 I/O con cualquier otro dispositivo adecuado u otro método de interacción electrónico/software. Tales comunicaciones pueden ser usadas, por ejemplo, en dispositivos 300 informáticos que implementan un motor 206 de análisis de datos. Los datos 236 de entrada pueden ser recibidos a través de la interfaz o interfaces 310 de I/O, por ejemplo, desde un usuario o un sistema informático tal como un sistema de monitorización, y los datos 234 de salida pueden ser proporcionados a través de la interfaz o interfaces 310 de I/O, por ejemplo, a un usuario o a un sistema informático tal como un sistema de reporte.

El dispositivo 300 informático puede también incluir uno o más medios 312 legibles por un ordenador, que pueden representar medios 112 legibles por un ordenador. Los medios 312 legibles por un ordenador pueden incluir un sistema operativo, por ejemplo, sistema 116 operativo (omitido por claridad). En el ejemplo ilustrado, los medios 312 legibles por un ordenador incluyen un almacenamiento 314 de datos. En algunos ejemplos, el almacenamiento 314 de datos incluye almacenamiento de datos, estructurados o no estructurados, tales como base de datos o almacén de datos. En algunos ejemplos, el almacenamiento 314 de datos incluye un cuerpo o una base de datos relacional con una o más tablas, matrices, índices, procedimientos almacenados, etc. para permitir el acceso a datos que incluye uno o más tablas de lenguaje de marcas de hipertexto (HTML), tablas de marcos de descripción de recursos (RDF), tablas de lenguaje de ontología web (OWL), o tablas de lenguaje de marcas extensible (XML), por ejemplo. El almacenamiento 314 de datos puede almacenar datos para las operaciones de los procesos, aplicaciones, componentes, o módulos almacenados en los medios 312 legibles por un ordenador o ejecutados por una o unas unidades de procesamiento o uno o unos aceleradores 306. En al menos un ejemplo, el almacenamiento de datos puede almacenar datos 316 de entrenamiento, una DNN 304 u otro modelo matemático, datos usados para entrenar la DNN 304 tal como variables temporales, una DNN 318 entrenada, o cualquier combinación de ellos. Algunos o todos los datos referenciados anteriormente pueden ser almacenados en memorias 320 separadas incorporadas en una o más unidades 306 de procesamiento, tal como una memoria en tarjeta incorporada a un procesador tipo CPU, un procesador tipo GPU, un acelerador tipo FPGA, un acelerador tipo DSP, u otro acelerador. La memoria 320 puede incluir, por ejemplo, una memoria cache de CPU o GPU.

En al menos un ejemplo, un sistema incluye uno o más medios 312 legibles por un ordenador que tienen en ellos una pluralidad de módulos y un modelo de cálculo de un problema de optimización. Por ejemplo, los medios 312 legibles por un ordenador del dispositivo 300 informático pueden almacenar los módulos del motor 302 de entrenamiento de la DNN. El modelo de cálculo puede incluir, por ejemplo un modelo de red neuronal tal como DNN 304. El sistema puede incluir una pluralidad de nodos, por ejemplo, dispositivo o dispositivos 300 informáticos o nodos que se ejecutan en ellos. Un nodo puede incluir al menos una unidad 306 de procesamiento (por ejemplo, un procesador o un núcleo de un procesador) operables acoplados a al menos uno de los medios 312 legibles por un ordenador. Las unidades 306 de procesamiento pueden ser adaptadas para intercomunicarse y para ejecutar módulos de la pluralidad de módulos. Por ejemplo, el dispositivo 300 informático puede incluir uno o más nodos y comunicarse con el nodo o nodos de otro u otros dispositivos 300 informáticos a través de la red 106. Los nodos de los dispositivos 300 informáticos en el sistema pueden cooperar como se describe en este documento para determinar los valores de modificación para un problema de optimización, por ejemplo, gradientes para entrenamiento de redes neuronales.

Los módulos almacenados en los medios 312 legibles por un ordenador del motor 302 de entrenamiento de DNN pueden incluir uno o más módulos o API, que son ilustrados como un módulo 322 de selección de lotes, un módulo 324 de cuantificación, un módulo 326 de determinación de actualización, un módulo 328 de actualización, un módulo 330 de transferencia, y un módulo 332 de agregación. Los módulos pueden también incluir un módulo de modelo de separación, por ejemplo, que implementa la separación 228 del modelo, y un módulo de paralelización de transferencia de datos, por ejemplo, que implementa la paralelización 232 de transferencia de datos (ambos omitidos por claridad). El número de módulos puede variar más alto o bajo, y se pueden usar módulos de varios tipos en varias combinaciones. Por ejemplo, la funcionalidad descrita asociada con los módulos ilustrados puede combinarse para ser realizada por un menor número de módulos o API o pueden ser separados y realizados por un mayor número de módulos o API. Por ejemplo, el módulo 324 de cuantificación y el módulo 330 de transferencia pueden combinarse en un único módulo que realiza ambas funciones. En varios ejemplos, la unidad o unidades 306 de procesamiento pueden acceder al módulo o módulos en los medios 312 legibles por un ordenador a través de un bus 334, que puede representar el bus 114, Figura 1. La interfaz 308 de comunicaciones y la interfaz 310 de I/O pueden también comunicarse con la unidad o unidades 306 de procesamiento a través del bus 334.

En un ejemplo, los módulos incluyen el módulo 326 de determinación de la actualización configurado para determinar los valores de modificación del modelo de cálculo. El módulo 324 de cuantificación puede configurarse para cuantificar los valores de modificación determinados mediante el uso, por ejemplo, mediante la incorporación, de valores de errores almacenados y para actualizar los valores de errores almacenados mediante el uso de los valores de modificación determinados y los valores de modificación cuantificados. Esto puede ser, por ejemplo, como se describió anteriormente con referencia a la cuantificación 226. El módulo 330 de transferencia puede configurarse para transmitir al menos algunos de los valores de modificación cuantificados a al menos otro de los nodos, por ejemplo, unidades 306 de procesamiento. Esto puede ser, por ejemplo, como se describió anteriormente con referencia al intercambio 320. El módulo 330 de transferencia puede también configurarse para recibir valores de modificación desde otros nodos, y el módulo 332 de agregación puede configurarse para agregar datos en el almacenamiento 314 de datos con datos recibidos desde otros nodos. La operación del modulo 330 de transferencia en un ejemplo es discutida a continuación con referencia a las Figuras 5 y 6. El módulo 328 de actualización puede configurarse para modificar el modelo de cálculo almacenado según los valores de modificación cuantificados recibidos. El módulo 328 de actualización puede configurarse para modificar el modelo de cálculo almacenado según los valores de modificación recibidos agregados proporcionados por el módulo 332 de agregación. Estas funciones son discutidas más en profundidad a continuación.

El módulo 326 de determinación de la actualización y módulo 328 de actualización pueden usar el algoritmo 220 para entrenar la DNN 304 en base a los datos 316 de entrenamiento, que puede ser un cuerpo de voz. En casos en los que la DNN 304 es entrenada para propósitos de análisis de voz, la DNN 304 puede ser una DNN dependiente del contexto que es usada junto con una HMM. En algunos ejemplos, la DNN puede ser una DNN independiente del contexto. La DNN 304 puede ser un MLP que modela la probabilidad posterior $P_{s|o}(s|o)$ de una clase s (por ejemplo, un trifonema o senón), dado un vector de observación o (por ejemplo, datos de audio), y una pila de (L + 1) capas de modelos de logaritmo lineal. Las primeras L capas, ℓ = 0... L – 1, probabilidades posteriores del modelo de h^ℓ vectores binarios ocultos (las salidas de las capas ocultas) dados v^ℓ vectores de entrada a las capas ocultas, mientras que las L capas superiores modelan la clase posterior deseada según las Ecuaciones (1), (2), y (3). Las Ecuaciones (1), (2), y (3) usan W^ℓ matrices de peso y a^ℓ vectores de sesgo, donde h_j^ℓ y $z_j^\ell(v^\ell)$ son los componentes j-ésimos de h^ℓ y $z^\ell(y^\ell)$, respectivamente.

$$P^{\ell}_{\mathbf{h}|\mathbf{v}}(h^{\ell}|v^{\ell}) = \prod_{j=1}^{N^{\ell}} \frac{e^{z_j^{\ell}(v^{\ell})\cdot h_j^{\ell}}}{e^{z_j^{\ell}(v^{\ell})\cdot 1} + e^{z_j^{\ell}(v^{\ell})\cdot 0}}, \qquad 0 \le \ell < L \tag{1}$$

$$P^{L}_{\mathbf{s}|\mathbf{v}}(s|v^{L}) = \frac{e^{z_s^L(v^L)}}{\sum_{s'} e^{z_{s'}^L(v^L)}} = \mathrm{softmax}_s(z^L(v^L)) \tag{2}$$

$$z^{\ell}(v^{\ell}) = (W^{\ell})^T v^{\ell} + a^{\ell} \quad ; \quad v^{\ell} \overset{def}{=} E^{\ell-1}\{h^{\ell-1}\} \tag{3}$$

La sumatoria total sobre todas las variables ocultas es a veces inviable. En ese caso, esta sumatoria puede ser aproximada mediante una “aproximación de campo medio” donde las v^ℓ entradas a las capas ocultas son tomadas como las expectativas de los vectores h^ℓ de salida correspondientes de la capa previa. Además, para el uso con la DNN 304, los estados posteriores $P_{s|o}(s|o)$ pueden ser convertidos a probabilidades escaladas dividiendo por su anterior.

En consecuencia, el módulo 326 de determinación de la actualización y el módulo 328 de actualización pueden entrenar la DNN 304 según el criterio D de entropía cruzada mostrado en la Ecuación (4):

$$D = \sum_{t=1}^{T_{\text{corpus}}} \log P_{\mathbf{s}|\mathbf{o}}(s(t)|o(t)) \qquad (4)$$

mediante el uso del descenso de gradiente estocástico (SGD) como se muestra en la Ecuación (5), con tasa de aprendizaje ε:

$$(W^{\ell}, a^{\ell}) \leftarrow (W^{\ell}, a^{\ell}) + \varepsilon \frac{\partial D}{\partial (W^{\ell}, a^{\ell})} \quad , \quad 0 \leq \ell \leq L \qquad (5)$$

Los gradientes $\partial D/\partial \cdot$ son mostrados en las Ecuaciones (6), (7), (8), y (9) con señales $e^{\ell}(t)$, las derivadas de componentes $\sigma_j,(z) = \sigma_j(z) \cdot (1 - \sigma_j(z))$, y $(\log \text{softmax})_j,(z) = \delta_{s(t),j} - \text{softmax}_j(z)$, y la delta de Kronecker δ.

$$\frac{\partial D}{\partial W^{\ell}} = \sum_t v^{\ell}(t)\,(\omega^{\ell}(t)\,e^{\ell}(t))^T \,;\, \frac{\partial D}{\partial a^{\ell}} = \sum_t \omega^{\ell}(t)\,e^{\ell}(t) \qquad (6)$$

$$e^{L}(t) = (\log \text{softmax})'(z^{L}(v^{L}(t))) \qquad (7)$$

$$e^{\ell-1}(t) = W^{\ell} \cdot \omega^{\ell}(t) \cdot e^{\ell}(t) \quad \text{for} \quad 0 \leq \ell < L \qquad (8)$$

$$\omega^{\ell}(t) = \begin{cases} \text{diag}(\sigma'(z^{\ell}(v^{\ell}(t))) & \text{for} \quad 0 \leq \ell < L \\ 1 & \text{else} \end{cases} \qquad (9)$$

Estas ecuaciones proporcionan una fórmula que puede optimizar la entropía cruzada matemáticamente.

En al menos un ejemplo, el módulo 326 de determinación de actualizaciones determina los gradientes, por ejemplo, valores $\partial D/\partial \cdot$, Ecuación (6), de la DNN 304 en base a un mini lote 218 seleccionado a partir de los datos 316 de entrenamiento. El módulo de determinación de la actualización es configurado para determinar los valores de modificación mediante el uso de un algoritmo de descenso de gradiente estocástico. El módulo 328 de actualización modifica el modelo de cálculo almacenado, por ejemplo, de la DNN 304, en base a los gradientes (valores de modificación) desde uno o más nodos, Ecuación (5). El módulo 324 de cuantificación y el módulo 330 de transferencia cooperan para proporcionar los gradientes determinados como los necesitan los nodos. Ejemplos de transferencia son discutidos a continuación con referencia a las Figuras 5 y 6.

El entrenamiento de la DNN 304 puede ser alcanzado mediante la canalización de cálculos de propagación hacia atrás en un modo paralelizado (esto es, mediante la ejecución de manera simultanea de múltiples cálculos) mediante el uso de múltiples nodos, por ejemplo, múltiples unidades 306 de procesamiento. En varios ejemplos, la canalización no se usa. En algunos ejemplos, uno o más de los nodos se comunican de manera concurrente con los cálculos.

La convergencia (esto es, finalización del entrenamiento) puede ser alcanzada mediante la realización del descenso de gradiente estocástico, como se describió anteriormente en la Ecuación (5), mediante el uso de lotes de tamaños discretos de tramas 218 muestreadas aleatoriamente de los datos 316 de entrenamiento, referidos en este documento como “mini lotes”. El tamaño de los mini lotes puede estar limitado por la naturaleza del cálculo paralelizado del algoritmo 220. Por ejemplo, las actualizaciones del modelo de la DNN 304, que implica el intercambio de datos entre unidades de procesamiento, son usadas para el cálculo de iteraciones del algoritmo 220. Sin embargo, las actualizaciones del modelo a través de múltiples unidades de procesamiento pueden usar una alta cantidad de ancho de banda durante la ejecución del algoritmo 220. En un ejemplo, la DNN 304 (con siete capas ocultas) puede incluir 100 millones de parámetros. En tal ejemplo, el procesamiento de un mini lote de tamaño razonable de tramas de muestras con respecto a la DNN 304 puede traducirse en la reunión y redistribución de 400 megabytes (MB) de valores de gradientes por cada uno de los nodos.

El tamaño de un mini lote individual que es usado para entrenar las DNN puede ser restringido por dos factores. La restricción superior para el tamaño del mini lote es la frecuencia de las actualizaciones del modelo. Un tamaño de mini lote más grande para los mini lotes 218 de tramas de muestras puede resultar en menos actualizaciones del modelo. Sin embargo, aumentar el tamaño del mini lote puede resultar en la pérdida de precisión de cálculo, especialmente durante las iteraciones de cálculo iniciales del algoritmo 220. Tal pérdida de precisión de cálculo

puede resultar en un tiempo de ejecución prolongado para el algoritmo 220 para alcanzar la convergencia, esto es, para completar el entrenamiento de la DNN 304. En casos extremos, el tiempo de ejecución prolongado puede incluso resultar en un fallo del algoritmo 220 para alcanzar la convergencia, esto es, fallo de completar el entrenamiento. La restricción inferior del tamaño del mini lote es la eficiencia en el uso de los nodos y las unidades 306 de procesamiento en ellos. La eficiencia en el uso de los ciclos de cálculo realizados por los nodos puede disminuir con el tamaño del mini lote pues los mini lotes 218 de tramas de muestreo son reducidos. Así, la reducción excesiva en el tamaño de los mini lotes puede también llevar a ineficiencias que prolongan el tiempo de ejecución para el algoritmo 220 para alcanzar la convergencia.

En al menos un ejemplo, un módulo 322 de selección de lotes puede particionar los datos 316 de entrenamiento en mini lotes 218 de tramas muestreadas en base al tamaño de mini lote configurado, por ejemplo, como se indicó anteriormente con referencia a la selección 224 de lotes. Un ejemplo de mini lotes es discutido a continuación con referencia a la Figura 5. El módulo 322 de selección de lotes puede configurar el tamaño del mini lote para el mini lote 218 de trama de muestra en base a las tasas de transferencias de datos entre las unidades de procesamiento y el número de operaciones por segundo que las unidades 306 de procesamiento son capaces de ejecutar. Para aumentar la eficiencia de cálculo, el tamaño del mini lote puede establecerse para que el tiempo requerido para realizar cálculos en un mini lote sea aproximadamente igual al tiempo requerido para transferir datos relacionados con ese mini lote hacia y desde un nodo. En al menos un ejemplo, el tamaño del mini lote es seleccionado tan alto como sea posible mientras proporcione al menos una precisión seleccionada, y el número de nodos es seleccionado para que el tiempo para realizar cálculos que usan el tamaño seleccionado de mini lote sea substancialmente igual al tiempo requerido para comunicar los gradientes para un mini lote del tamaño seleccionado. Esto permite transferencias de datos para solapar los cálculos. Por ejemplo, dada una matriz de 2-4 GPGPU que son capaces de 2-4 tera operaciones de punto flotante por segundo (TFLOPS), y tasas de transferencia de 6 gigabytes (GB)/s entre las GPGPU, el tamaño del mini lote puede estar en un intervalo de 256 a 1024 tramas de muestra por mini lote de muestra.

En al menos un ejemplo, el módulo 322 de selección de lotes puede configurar un tamaño de mini lote más grande cuando las tasas de trasferencias de datos para las unidades 306 de procesamiento son relativamente superiores a las velocidades de ejecución de las unidades 306 de procesamiento. A la inversa, el módulo 322 de selección de lotes puede configurar un tamaño de mini lote más pequeño cuando las velocidades de ejecución de las unidades 306 de procesamiento son relativamente superiores a las tasas de transferencias de datos entre las unidades 306 de procesamiento.

Un ejemplo de iteraciones de cálculos realizadas por el algoritmo 220 puede ejecutar los siguientes pasos: propagación hacia delante de datos de entrada, propagación hacia atrás de errores, y actualización del modelo. Estos pasos pueden ser ejecutados en el orden listado u otro orden. La propagación hacia delante de los datos de entrada puede ser descrita por las Ecuaciones (1), (2), y (3), la propagación hacia atrás del error puede ser descrita por la Ecuación (8), y la actualización del modelo puede ser descrita por la Ecuación (5).

Además, la técnica de paralelización 232 de transferencia de datos implica la paralelización de transferencia de datos con cálculos. Una primera parte de la paralelización 232 de transferencia de datos puede ocurrir después de la realización de un paso de propagación hacia atrás del error. En esta parte, los datos de salida de un cálculo en un nodo que procesa una parte de los datos del modelo o datos 316 de entrenamiento puede ser transmitida a otro nodo que procesa una parte diferente del modelo de la DNN 304 o datos 316 de entrenamiento. Tal transmisión puede ser realizada en paralelo o parcialmente en paralelo con un paso de actualización del modelo o un paso de propagación hacia delante de datos de entrada, dado que el paso de actualización del modelo y el paso de propagación hacia delante usan datos que son diferentes de los datos de salida. Igualmente, después de la realización del paso de propagación hacia delante de los datos de entrada, los datos de salida de un cálculo en un nodo pueden ser transmitidos a otro nodo. Tal transmisión puede ser realizada en paralelo o parcialmente en paralelo con el cálculo de un error de otro paso de propagación hacia atrás del error. Así, en ejemplos en los cuales el tiempo de transmisión es más corto que el tiempo de cálculo, el uso de la paralelización 232 de trasferencias de datos puede reducir o eliminar cualquier retraso de tiempo resultante del intercambio de datos entre múltiples unidades de procesamiento.

Como se indicó anteriormente, la separación 228 del modelo es la paralelización del procesamiento de partes de la DNN 304 sobre múltiples nodos que incluyen unidades de procesamiento, tales como las unidades 306 de procesamiento. Cada nodo puede calcular una franja (parte) de los gradientes. Las franjas de gradientes pueden entonces ser, por ejemplo, distribuidas a otros nodos, por ejemplo, unidades de procesamiento de las unidades 306 de procesamiento, o intercambiadas entre nodos para la finalización del cálculo de la iteración actual del modelo (por ejemplo, cálculos de entrenamiento que usan el mini lote actual de los datos de entrenamiento). En varios ejemplos la separación 228 del modelo puede ser usada junto con el paralelismo de datos, por ejemplo, mediante la ejecución de datos en paralelo sobre múltiples grupos de nodos de modelo paralelo. El paralelismo de datos es descrito a continuación, por ejemplo, con referencia a la Ecuación (16). En un ejemplo, la separación 228 del modelo puede ser usada junto con el paralelismo de capas para permitir el procesamiento de capas de manera más flexible en paralelo.

En al menos un ejemplo de un sistema para optimización matemática de modelos de cálculo, por ejemplo, para entrenamiento de redes neuronales tales como entrenamiento de DNN, cada uno de los nodos incluye una pluralidad de unidades 306 de procesamiento operables acopladas con los medios 312 legibles por un ordenador respectivos. Cada unidad 306 de procesamiento en este ejemplo es configurada para ejecutar al menos el módulo 326 de determinación de actualización. En un ejemplo, el sistema puede incluir una barra cruzada (por ejemplo, red 106, Figura 1) que conecta de manera comunicativa los nodos. Los nodos se pueden configurar para ejecutar el módulo 330 de transferencia para transmitir al menos algunos de los valores de modificación cuantificados a través de la barra cruzada de manera concurrente con la ejecución del módulo 326 de determinación de la actualización. La al menos una unidad 306 de procesamiento de cada nodo puede, en algunos ejemplos, incluir una unidad de procesamiento gráfico de propósito general (GPGPU) configurada para ejecutar el módulo 328 de actualización y el módulo 324 de cuantificación, y una unidad de procesamiento central (CPU) informática configurada para ejecutar el módulo 330 de transferencia.

Los medios 312 legibles por un ordenador según varios ejemplos tienen en ellos instrucciones ejecutables en un ordenador. Las instrucciones ejecutables en un ordenador, tras la ejecución, configuran un ordenador, por ejemplo, el dispositivo 300 informático o la unidad 306 de procesamiento en él, para realizar operaciones. Las operaciones pueden incluir operaciones discutidas a continuación con referencia a las Figuras 4-6. En al menos un ejemplo, las operaciones pueden incluir determinar primeros valores de un gradiente de un modelo de red neuronal mediante el uso de un conjunto de muestras de entrenamiento. Esto se puede hacer mediante el módulo 326 de determinación de la actualización. Las técnicas de propagación hacia atrás que incluyen los métodos de descenso de gradiente estocástico y libres de Hessian pueden ser controladas por las instrucciones en el módulo 326 de determinación de la actualización.

Las operaciones además incluyen transmitir una primera parte de los primeros valores del gradiente y recibir segundos valores correspondientes a una segunda parte diferente de los primeros valores del gradiente, por ejemplo, mediante la transferencia del módulo 330. Las operaciones pueden incluir solapar la transmisión y la recepción. Ejemplos de solapamiento se muestran en la Figura 5.

La primera y segunda partes de los valores se pueden corresponder con diferentes franjas del modelo, por ejemplo, como se discute a continuación con referencia a la Figura 5. Por ejemplo, la primera parte de los primeros valores del gradiente transmitidos por el Nodo 1, Figura 5, puede ser la franja 518, correspondiente a un Subconjunto 2 de la DNN 304. El Nodo 1 puede recibir la franja 522, que se corresponde con un Subconjunto 1 diferente de la DNN 304. La segunda parte correspondiente de los primeros valores es la franja 516, en este ejemplo.

Las operaciones pueden además incluir agregar la segunda parte, por ejemplo, la franja 516, con los segundos valores recibidos, por ejemplo, la franja 522. Esto se puede hacer mediante la agregación del módulo 332. Las operaciones pueden entonces incluir transmitir los valores agregados, por ejemplo, mediante el uso del módulo 330 de transferencia. Esto permite un entrenamiento de la DNN paralelo de datos efectivo.

En al menos un ejemplo, las operaciones además incluyen cuantificar los primeros valores determinados antes de transmitir la primera parte. Las operaciones pueden además incluir determinar valores reconstruidos (cuantificación inversa) y transformar los segundos valores recibidos mediante el uso de los valores reconstruidos antes de la agregación. Estas operaciones pueden ser realizadas por el módulo 324 de cuantificación, que puede determinar los valores reconstruidos, por ejemplo, al menos parcialmente basados en los valores del gradiente. Por ejemplo, el valor reconstruido para un valor cuantificado de q puede ser la media de todos los gradientes en un mini lote previo que se cuantificó a q (véase, por ejemplo, la Ecuación (14)). El histórico de valores reconstruidos para los respectivos q valores cuantificados pueden ser registrados y suavizados, por ejemplo, mediante el uso de una media móvil ponderada exponencialmente (EWMA) o ventana móvil. Los valores reconstruidos pueden ser recalculados cada n mini lotes, para una n seleccionada, o a intervalos aleatorios. En cualquier técnica que promedia o de otro modo mediante el uso de gradientes para determinar los valores de reconstrucción, algunos gradientes pueden ser omitidos del promedio o combinación de gradientes. Por ejemplo, los valores atípicos pueden ser omitidos del promedio, o elementos seleccionados aleatoriamente pueden ser omitidos. En varios ejemplos, los valores reconstruidos son determinados por el módulo 324 de cuantificación y son transmitidos con los datos cuantificados. Esto aun resulta en ahorro en la transmisión de datos. Por ejemplo, la dimensión X de un grupo de valores cuantificados juntos puede ser, por ejemplo, 2048, y las muestras pueden ser valores flotantes. Los datos no cuantificados son 32X bits. Cuantificar a 1 bit, y transmitir dos valores reconstruidos flotantes de 32-bits (uno para “0” bits y uno para “1” bits) con los datos, reduce el requisito de transferencia de datos a X + 2 x 32. Esto es un ahorro para cualquier entero $X \geq 3$.

En varios ejemplos de un sistema de entrenamiento de DNN, por ejemplo, que implementa instrucciones de programa informático tal como las descritas anteriormente, al menos uno de los nodos en el sistema puede transmitir los valores de modificación cuantificados directamente a al menos uno de los otros nodos. Esto es, los nodos pueden comunicarse en una topología entre pares mejor que en una topología de maestro-esclavo. Sin embargo, los ejemplos en este documento no están limitados a entre pares y puede operar con varias topologías de interconexión de nodos. Usar transferencias entre pares puede permitir de manera ventajosa que múltiples nodos transfieran datos de manera simultánea y un uso mucho más eficiente del ancho de banda disponible.

En al menos un ejemplo, nodos individuales en el sistema incluyen memorias 320 respectivas acopladas a la respectiva al menos una unidad 306 de procesamiento. En este ejemplo, una memoria 320 individual puede configurarse para almacenar el estado de cuantificación privado respectivo para el nodo correspondiente, por ejemplo, incluir valores de errores almacenados descritos anteriormente con referencia a la cuantificación 226, Figura 2. En algunos de estos ejemplos, los nodos de manera ventajosa no comparten estados relativos al error de cuantificación. Esto permite a los nodos operar de manera independiente en cuanto a la cuantificación, lo que reduce la cantidad de datos de estados a ser transferidos y aumenta la velocidad de entrenamiento.

En algunos ejemplos, un nodo incluye una CPU conectada a la interfaz 308 de comunicaciones, y una GPGPU configurada para llevar a cabo instrucciones en el módulo 326 de determinación de actualización. En algunos de esos ejemplos, el módulo 330 de transferencia puede, por ejemplo, configurarse para transferir segundos valores de modificación cuantificados desde la GPGPU a la CPU en paralelo con la transferencia de al menos algunos de los valores de modificación cuantificados a al menos otro de los nodos u otras unidades 306 de procesamiento. De este modo, el cálculo y las transferencias de datos pueden solaparse no solo entre nodos, sino también dentro de los nodos. Esto puede permitir mantener altos factores de uso en tanto los recursos de cálculo como de comunicación en el nodo, lo que mejora la velocidad de entrenamiento. En sistemas que soportan acceso de memoria directa (DMA) entre dispositivos como CPU, GPGPU, o controladores de red, las transferencias de DMA pueden ser usadas para mover datos dentro de un nodo en paralelo con los cálculos.

Procesos ilustrativos

La Figura 4 es un diagrama que muestra un proceso 400 ejemplar para entrenar una red neuronal profunda.

En el bloque 402, un motor de entrenamiento, tal como el motor 118 de entrenamiento de la Figura 1, el motor 202 de entrenamiento de la Figura 2, o el motor 302 de entrenamiento de la Figura 3, determina matrices de gradientes de un modelo de red neuronal. Como se indicó anteriormente con referencia al módulo 326 de determinación de la actualización, el motor 202 de entrenamiento puede realizar, por ejemplo, un paso de descenso de gradiente estocástico (SGD, por ejemplo, Ecuación (5)), u otro algoritmo de entrenamiento de red neuronal, o un algoritmo de entrenamiento de red neuronal combinado que incluye SGD y otras técnicas. Los términos “matriz” y “matrices” no requieren ninguna dimensión particular o tamaño de matrices. Las matrices de gradientes pueden ser tan pequeñas como un único escalar y tener cualquier número de dimensiones y cualquier extensión en esas dimensiones. En un ejemplo, las matrices de gradientes son 2048 x 2048.

En el bloque 404, el motor 202 de entrenamiento cuantifica las matrices de gradientes mediante el uso de las matrices de errores almacenadas correspondientes. Por ejemplo, las matrices de errores pueden ser incorporadas en la cuantificación como se discute a continuación, por ejemplo, con referencia a la Ecuación (10). La cuantificación puede incluir, por ejemplo, determinar representaciones de un único bit tales como representación de un único bit aproximado para elementos respectivos de las matrices de gradientes. En algunos ejemplos, las representaciones cuantificadas pueden tener dos bits por valor, tres bits, o cualquier número b de bits por valor para b menor que la cuenta B de bits de las matrices de gradientes antes de la cuantificación. En el bloque 406, el motor 202 de entrenamiento actualiza las matrices de errores mediante el uso (por ejemplo, incorporación) de las matrices de gradientes cuantificados correspondientes.

En al menos un ejemplo, la cuantificación (bloque 404) y la actualización (bloque 406) son realizadas según las Ecuaciones (10) y (11), en las cuales $G_{ij\ell}(t)$ es un parámetro de gradiente, $G_{ij\ell}^{quant}(t)$ es la representación cuantificada del mismo, $Q(\cdot)$ es la función de cuantificación, $Q^{-1}(\cdot)$ es la función de cuantificación inversa (reconstrucción) correspondiente, $\Delta_{ij\ell}(t)$ es el error de cuantificación, N es el tamaño del mini lote, y t es el índice de muestra.

$$G_{ij\ell}^{quant}(t) = Q(G_{ij\ell}(t) + \Delta_{ij\ell}(t-N)) \qquad (10)$$

$$\Delta_{ij\ell}(t) = G_{ij\ell}(t) - Q^{-1}(G_{ij\ell}^{quant}(t)) \qquad (11)$$

Como se puede ver, el error de cuantificación $\Delta_{ij\ell}(t\text{-}N)$ para la muestra t-N en un mini lote es usado para determinar el valor cuantificado $G_{ij\ell}^{quant}(t)$ de la muestra correspondiente en el siguiente mini lote. Además, la matriz de error Δ es actualizada ($\Delta_{ij\ell}(t)$) de forma que el error será corregido tanto como sea posible para la Q dada en la siguiente cuantificación, de la muestra t + N.

En al menos un ejemplo, la función de cuantificación es una función de umbral como se muestra en la Ecuación (12):

$$Q(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \qquad (12)$$

Esta función proporciona una salida Q(x) de 1 bit para el valor x a ser cuantificado. Otras funciones de cuantificación pueden usarse, por ejemplo, para dividir un intervalo seleccionado para x (por ejemplo, [0, 1]) en un número seleccionado de pasos espaciados igualmente (por ejemplo, 2^n pasos para un valor cuantificado de n bits, o, en varios ejemplos, k pasos, k > 2). En algunos ejemplos, el umbral de cuantificación es establecido mediante el uso del gradiente, como se muestra en la Ecuación (13):

$$Q(x) = \begin{cases} 1, & x \geq R \\ 0, & x < R \end{cases} \qquad (13)$$

Donde $R = \overline{G_{i,j,k}(t)}$ para usar datos desde el mini lote actual, o $R = \overline{G_{i,j,k}(t \quad N)}$ para usar datos desde el mini lote anterior. A lo largo de esta descripción, t – N puede ser reemplazado con t – kN para k > 0 entero. Esto es, para cuantificación, reconstrucción, o cualquier propósito en este documento, se pueden usar datos más antiguos que un mini lote.

En el bloque 408, el motor 202 de entrenamiento intercambia las matrices de gradientes cuantificados con varios nodos. El bloque 408 puede incluir el motor 202 de entrenamiento que transmite algunas o todas las matrices de gradientes cuantificados a al menos otro nodo, por ejemplo, al menos otro dispositivo 210 informático o unidad 212 de procesamiento, Figura 2. El bloque 408 puede incluir una pluralidad de dispositivos 210 informáticos que transmiten las matrices de gradientes cuantificados respectivas a otros de la pluralidad de dispositivos informáticos. Los nodos pueden intercambiar las matrices de gradientes cuantificados de manera síncrona. Un ejemplo de este intercambio es discutido a continuación con referencia a la Figura 5. El intercambio puede incluir reconstruir valores de gradientes a partir de los valores cuantificados recibidos (por ejemplo, mediante el uso de las Ecuaciones (14) o (15) a continuación). En un ejemplo, el bloque 408 de intercambio comprende intercambiar solo las matrices de gradientes cuantificados. En al menos un ejemplo, el bloque 408 de intercambio comprende intercambiar las matrices de gradientes cuantificados y valores de criterio de entropía cruzada (CE) de uno o más nodos. Los valores de criterio de CE pueden usarse para hacer seguimiento del progreso del entrenamiento de la red neuronal. En un ejemplo, el bloque 408 de intercambio comprende intercambiar matrices de gradientes cuantificados, reconstruir valores de gradientes, agregar valores reconstruidos, cuantificar los valores agregados, e intercambiar los valores agregados cuantificados, por ejemplo, como se discute a continuación con referencia a las Figuras 5 y 6.

En el bloque 410, el modelo de red neuronal es actualizado mediante el uso de gradientes. Esto se puede hacer, por ejemplo, como se describe anteriormente con referencia a la Ecuación (5) anterior. Los valores de gradientes reconstruidos pueden ser usados. Los gradientes agregados pueden ser usados como se discutió anteriormente con referencia a las Figuras 5 y 6.

En el bloque 412 de decisión, el motor 202 de entrenamiento puede determinar si más nodos tienen datos que procesar. Si es así, el proceso puede volver al bloque 402. Si no, el proceso puede proceder al bloque 414. De este modo, el bloque 402 de determinación, el bloque 404 de cuantificación, el bloque 406 de actualización, el bloque 408 de intercambio, y el bloque 410 de actualización del modelo, o cualquier combinación de esos bloques, pueden realizarse por los nodos individuales para las matrices de gradientes respectivas y las matrices de errores correspondientes a las matrices de gradientes respectivas. Los nodos pueden realizar este procesamiento en paralelo para reducir el tiempo de entrenamiento.

En el bloque 414, en algunos ejemplos, un factor de paralelización puede ser ajustado como una función del tamaño del lote en base al menos en parte a las mediciones de tiempo. Como se indicó anteriormente, el tamaño N del mini lote y el número K de nodos informáticos en paralelo afectan el tiempo de entrenamiento total. Estos valores pueden ajustarse en base a las mediciones del tiempo gastado en la comunicación o cálculo para aumentar la velocidad de entrenamiento.

En el bloque 416 de decisión, el motor 202 de entrenamiento puede determinar si un criterio de terminación seleccionado ha sido satisfecho. De ser así, el proceso de entrenamiento puede determinarse que está completo. Más de un criterio puede ser usado, por ejemplo, indicar la finalización cuando un criterio es satisfecho (o un número seleccionado de criterios son satisfechos) o indicar la finalización cuando todos los criterios son satisfechos. Cuando el entrenamiento es completado, el motor 202 de entrenamiento puede proporcionar la DNN 208 al motor 206 de análisis de datos, ambas en la Figura 2. Si el criterio no ha sido satisfecho, el proceso puede volver al bloque 402. De este modo, cualquiera o todos los bloques 402 de determinación, bloque 404 de cuantificación, bloque 406 de actualización, bloque 408 de intercambio, bloque 410 de actualización del modelo, bloque 412 de decisión, y bloque 414 de ajuste pueden ser repetidos para cada una de la pluralidad de mini lotes del modelo de red neuronal. En al menos un ejemplo, el intercambio de un primer mini lote es ejecutado por el bloque 408 en paralelo con la determinación, cuantificación, o actualización de un segundo mini lote.

Un ejemplo de criterio puede ser precisión. La precisión de la trama de entrenamiento de la DNN, por ejemplo, DNN 204 o 304, puede ser evaluada durante el entrenamiento, y el criterio puede ser al menos un criterio de trama de entrenamiento seleccionada. La DNN puede ser probada periódicamente en un conjunto de pruebas de entradas y la

tasa de error determinada, y el criterio puede ser como mucho una tasa de error seleccionada. Un criterio ejemplar puede ser tiempo de entrenamiento transcurrido. Por ejemplo, después de un tiempo transcurrido seleccionado un número seleccionado de épocas de entrenamiento, el entrenamiento puede ser finalizado. Esto puede finalizar el entrenamiento si la DNN no converge para los parámetros particulares usados. Un criterio ejemplar puede ser mejorar bien en la precisión de la trama de entrenamiento o tasa de error. El criterio puede ser menos que una mejora seleccionada en cualquiera de esas calidades. Por ejemplo, si la precisión de trama de entrenamiento se acerca a una asíntota o si solo aumenta, por ejemplo, <0,1% puntos por época, el entrenamiento puede ser finalizado en la suposición de que la DNN ha convergido sustancialmente.

En el bloque 418, después de que se determina que el entrenamiento está completado, por ejemplo, porque el criterio de terminación es satisfecho, la red neuronal entrenada, por ejemplo, DNN 208 o 318 entrenada, es proporcionada. En un ejemplo, los pesos W^{ℓ} y sesgos a^{ℓ} de la Ecuación (5) en la finalización del entrenamiento son almacenados en un medio legible por un ordenador, por ejemplo, un medio de almacenamiento legible por un ordenador tal como medios 312 legibles por un ordenador. En al menos un ejemplo, el bloque 418 incluye realizar un algoritmo de red neuronal que usa los pesos y sesgos almacenados, por ejemplo, en el motor 120 de operación de la DNN o motor 206 de análisis de datos, para procesar los datos 236 de entrada para producir los datos 234 de salida.

La Figura 5 es un diagrama 500 de flujo de datos que muestra los pasos en un proceso ejemplar para intercambiar datos. Este proceso puede ser llevado a cabo de manera cooperativa por múltiples nodos que tienen bloques 408 de intercambio respectivos, Figura 4. Este proceso también es referido en este documento como un “todo reducido”. En este ejemplo, tres nodos participan, como se indica por las etiquetas “Nodo 1” hasta “Nodo 3” en la Figura 5.

El mini lote 502 incluye una pluralidad de tramas 218 muestreadas de manera aleatoria a partir de los datos 216 de entrenamiento, ambos en la Figura 2. En varios ejemplos, el mini lote 502 puede dividirse en sub mini lotes 504, 506, 508, por ejemplo, por la unidad de procesamiento configurada para coordinar los esfuerzos de los nodos. En algunos ejemplos, uno o más de los nodos pueden dividir el mini lote 502. En al menos un ejemplo, un nodo individual recupera solo su sub mini lote 504, 506, 508 correspondiente del mini lote 502.

En este ejemplo, cada uno de los Nodos 1, 2, y 3 recibe un sub mini lote 504, 506, 508 respectivo de datos 216 de entrenamiento y calcula valores de modificación de un modelo de cálculo basado en los sub mini lotes recibidos de los datos de entrenamiento. En un ejemplo, el modelo de cálculo es un modelo de red neuronal y las actualizaciones son gradientes adecuados para usar en un proceso de descenso de gradiente estocástico (SGD), por ejemplo, como el descrito anteriormente con referencia a la Ecuación (5). Este ejemplo será usado a lo largo de la discusión de la Figura 5 para claridad, pero las técnicas descritas y mostradas no se limitan a redes neuronales o a SGD.

El bloque 510 representa los gradientes calculados por el Nodo 1 mediante el uso del sub mini lote 504. El bloque 512 representa los gradientes calculados por el Nodo 2 mediante el uso del sub mini lote 506. El bloque 514 representa los gradientes calculados por el Nodo 3 mediante el uso del sub mini lote 508. Es deseable para los nodos individuales recibir información sobre los gradientes calculados por los otros nodos. Tal recepción permite que un nodo individual actualice su modelo de red neuronal en preparación para la siguiente iteración del proceso de SGD.

Como se discutió anteriormente con referencia al intercambio 230, para permitir el intercambio eficiente de información de gradientes con otros nodos, un nodo individual particiona sus gradientes en “franjas”. En este ejemplo, el bloque 510 incluye las franjas 516, 518, 520; el bloque 512 incluye las franjas 522, 524, 526; y el bloque 514 incluye las franjas 528, 530, 532. Las franjas respectivamente se corresponden con subconjuntos diferentes de los parámetros del modelo de cálculo.

En un ejemplo, las neuronas de cálculo en una única capa 222 de la DNN 204, ambas en la Figura 2, pueden ser asignadas a una de tres partes. Cada franja se relaciona con los pesos y sesgos de las neuronas de cálculo en una parte respectiva. Un ejemplo de la relación entre franjas y los datos de entrenamiento correspondientes y subconjuntos del modelo es expresado en la Tabla 1 a continuación. En este ejemplo, el sub mini lote 504 incluye 300 muestras de datos, numeradas 1-300. El sub mini lote 506 incluye las muestras 301-600 y el sub mini lote 508 incluye las muestras 601-900.

Tabla 1

Nodo	Muestras	Subconjunto 1	Subconjunto 2	Subconjunto 3
Nodo 1	1-300	516	518	520
Nodo 2	301-600	522	524	526
Nodo 3	601-900	528	530	532

Como se indica en la Tabla 1, para determinar los parámetros del modelo para el Subconjunto 1 de los parámetros del modelo, los gradientes de las franjas 516, 522, y 528 pueden combinarse en una franja 534 agregada, e igualmente las franjas 518, 524, 530 para el Subconjunto 2 en una franja 536 agregada y franjas 520, 526, 532 para el Subconjunto 3 en la franja 538 agregada.

En consecuencia, en una primera fase ("Fase 1"), los Nodos 1-3 intercambian datos entre ellos de forma que cada nodo, en este ejemplo, agrega las franjas correspondientes a un único subconjunto. Las transferencias indicadas por las líneas sólidas pueden ser realizada de manera concurrente, y las transferencias indicadas por las líneas de puntos pueden ser realizadas de manera concurrente. Las transferencias en este ejemplo son dadas en la Tabla 2 a continuación. Los grupos separados de los extremos de transferencias se pueden realizar de manera concurrente.

Tabla 2

Transfiere la franja	del Subconjunto	del nodo	al nodo	como parte de la franja agregada
518	2	1	2	536
526	3	2	3	538
528	1	3	1	534
520	3	1	3	538
522	1	2	1	534
530	2	3	2	536

En el ejemplo de la Tabla 2, el Nodo 1 agrega su propia franja 516 (esto es, franja 516 de gradientes calculada por el Nodo 1, indicada por la flecha punteada) con las franjas 522 y 528 recibidas para proporcionar la franja 534 agregada para el Subconjunto 1. La franja 534 agregada incluye los gradientes (o un agregado, por ejemplo, una suma, del mismo) del Subconjunto 1 del modelo con respecto a las muestras 1-900 de entrenamiento. De manera similar, el Nodo 2 agrega su propia franja 524 con las franjas 518 y 530 recibidas para proporcionar la franja 536 agregada para el Subconjunto 2. El Nodo 3 agrega su propia franja 532 con las franjas 520, 526 recibidas para proporcionar la franja 538 agregada para el Subconjunto 3.

En el ejemplo discutido anteriormente, los Nodos 1-3 intercambian datos ente ellos. En al menos un ejemplo, la operación u operaciones de agregación para proporcionar una o más de las franjas 534, 536, 538 es o son realizadas en un nodo diferente de los Nodos 1, 2, y 3. En algunos ejemplos, la operación u operaciones de agregación es o son realizadas en uno de los Nodos 1, 2, y 3 para otro u otros nodos, o para ese nodo y otro u otros nodos. Las franjas 534, 536, 538 agregadas pueden ser proporcionadas por un nodo que realiza la agregación del nodo o nodos que necesitan las franjas agregadas. El número de nodo o nodos que realizan la agregación puede ser el mismo o diferente del número de nodo o nodos que calcula gradientes.

Como se discutió anteriormente con referencia a la Ecuación (10), los valores de gradientes pueden ser cuantificados antes de ser transferidos en la Fase 1, Un nodo individual puede cuantificar los valores de gradientes que transmite, por ejemplo, franjas 518, 520 para el Nodo 1. Un nodo individual puede realizar una cuantificación inversa Q^{-1} en los valores recibidos para determinar los valores reconstruidos correspondientes antes de agregar las franjas 534, 536, 538 agregadas. La cuantificación inversa puede ser determinada, por ejemplo, mediante el uso de medios de una columna anterior de datos separados en sus cuadros de cuantificación (por ejemplo, un promedio de ~2000 valores). En al menos un ejemplo, la cuantificación inversa puede ser realizada como se muestra en la Ecuación (14):

$$Q^{-1}(x) = \overline{G_{i,j,\ell}(t-N)}\ para\ todo\ j,\ell\ tal\ que\ \mathrm{Q}\left(G_{ij\ell}(\mathrm{t}-\mathrm{N})\right) = \mathrm{x} \qquad (14)$$

En un ejemplo de un bit de cuantificación (q = 0 o 1), Q^{-1} (0) es el promedio de los valores en el mini lote anterior que se cuantificó a 0, y Q^{-1} (1) es el promedio de los valores en el mini lote anterior que se cuantificó a 1. Esto puede proporcionar una estimación del error de mínimo cuadrado. En al menos un ejemplo, los valores reconstruidos pueden ser el mínimo y máximo de un gradiente de mini lote anterior como se muestra en la Ecuación (15).

$$Q^{-1}(x) = \begin{cases} \max\limits_{j,\ell} G_{ij\ell}(t-N), & x = 1 \\ \min\limits_{j,\ell} G_{ij\ell}(t-N), & x = 0 \end{cases} \quad (15)$$

Entre las fases 1 y 2, los Nodos 1-3 (u otro nodo o nodos que realizan la agregación, o cualquier combinación de nodos de cálculo y agregación) pueden realizar procesamiento adicional en los datos de las franjas 534, 536, 538 agregadas. En un ejemplo, el procesamiento incluye suavizado del momento. En un ejemplo, el procesamiento incluye normalización de AdaGrad de los gradientes.

En una segunda fase ("Fase 2"), los Nodos 1-3 intercambian las franjas agregadas de forma que cada nodo, en este ejemplo, tiene el modelo completo, que incluye los tres subconjuntos. Las transferencias son expresadas en la Tabla 3 a continuación. Las líneas sólidas y punteadas en la Figura 5, y los extremos en la Tabla 3, son como se discutió anteriormente con referencia a la Fase 1. Las líneas punteadas en la Figura 5 durante la Fase 2 representan el reúso de las franjas 534, 536, 538 agregadas ya calculadas por los Nodos 1, 2, 3, respectivamente.

Tabla 3

Transfiere la franja agregada	del Subconjunto	del nodo	al nodo
534	1	1	2
536	2	2	3
538	3	3	1
534	1	1	3
536	2	2	1
538	3	3	2

Después de la Fase 2, el Nodo 1 tiene todos los gradientes para el modelo completo, las muestras 1-900, en el bloque 540. El Nodo 2 tiene el conjunto completo de gradientes en el bloque 542, y el Nodo 3 tiene el conjunto completo de gradientes en el bloque 544. Cada bloque 540, 542, 544 incluye las tres franjas 534, 536, 538 agregadas.

En varios ejemplos, los nodos cuantifican los valores de gradientes en las franjas 534, 536, 538 agregadas antes de transferirlas en la Fase 2. Los nodos que reciben los valores de gradientes agregados cuantificados pueden entonces reconstruir (Q^{-1}) esos valores y usar los gradientes reconstruidos resultantes para actualizar el modelo de cálculo, por ejemplo, la DNN 114.

La técnica descrita en la Figura 5 usa dos fases para intercambiar datos entre tres nodos. En general, para K nodos, K>1, esta técnica usa K-1 fases. Un nodo individual transfiere una K^{esima} parte de los gradientes dos veces en cada fase. En sistemas ejemplares que usan barras cruzadas o dispositivos similares para interconectar los nodos, en cada fase, todos los K nodos pueden transferir información de manera simultánea. En estos ejemplos, el tiempo requerido para cualquier transferencia de datos dada es solo el tiempo de transferencia para una K^{esima} parte de los gradientes. Representar el tamaño total de los datos de gradientes como M, el tiempo requerido para completar la transferencia ilustrada en estos ejemplos es del orden mostrado en la Ecuación (16):

$$\underbrace{M}_{\text{gradientes}} \times \underbrace{1/K}_{\text{tiempo por transferencia}} \times \underbrace{2}_{\text{transferencias por fase}} \times \underbrace{K-1}_{\text{número de fases}} \quad (16)$$

o 0(2M(K – 1)/K) ≈ 0(M). Por lo tanto, el tiempo requerido para realizar los intercambios de datos ilustrados en la Figura 5 es aproximadamente independiente del número de nodos K, cuando se usan transferencias simultáneas. Esto permite ventajosamente aumentar el número de nodos K que participan en el procesamiento paralelo, y

aumentar la velocidad de entrenamiento correspondientemente, sin sufrir rendimiento reducido debido a la sobrecarga de transmisión.

La Figura 6 es un diagrama de flujo que muestra los pasos en un proceso 600 ejemplar para intercambiar datos, por ejemplo, como se discutió anteriormente con referencia al bloque 408, Figura 4. La referencia se hace en la siguiente discusión de las Figuras 5 y 6. El procesamiento ejemplar recibe como entradas matrices de gradientes cuantificados desde el bloque 404, Figura 4.

En el bloque 602, las matrices de gradientes están particionadas. Las matrices de gradientes pueden ser particionadas según el número K de nodos o un número diferente. Como se indicó anteriormente, el nodo o nodos que realizan la agregación pueden ser el mismo o diferente del nodo o nodos que realizan el cálculo. En al menos un ejemplo, las matrices de gradientes pueden ser particionadas según el número de nodo o nodos que realizan la agregación. Las matrices de gradientes pueden ser cuantificadas, como se discutió anteriormente. Las particiones individuales resultantes de este particionamiento son referidas en este documento como “franjas”. Un ejemplo de esta partición es la división de bloques 510, 512, 514 en franjas, como se discutió anteriormente.

En el bloque 604, las franjas individuales (particiones) son proporcionadas a cada uno de los nodos respectivos. Esto es, las individuales de las particiones de las matrices de gradientes cuantificados son proporcionadas a cada uno de los nodos respectivos. En el ejemplo de la Figura 5, la franja 516 está ya residente en el Nodo 1. El bloque 602 que proporciona por lo tanto puede incluir la selección, mediante una unidad 212 de procesamiento asociada con el Nodo 1, de los datos de la franja 516 para mayor procesamiento. Las franjas 522 y 528 no son residentes en el Nodo 1, de forma que el bloque 602 que proporciona puede incluir que el Nodo 2 transfiera la franja 522 al Nodo 1 (flecha punteada) y el Nodo 3 que transfiera la franja 528 al Nodo 1 (flecha sólida). De manera similar, las franjas 518 y 530 son transmitidas al Nodo 2 y la franja 524 es seleccionada por el Nodo 2 como parte del bloque 602 que provisiona, y las franjas 520, 526 son transmitidas al Nodo 3 y la franja 532 es seleccionada por el Nodo 3. Las transferencias ejemplares mostradas en la Figura 5 están sumariadas en la Tabla 2 anterior.

En algunos ejemplos, en el bloque 606, las matrices de gradientes son reconstruidas a partir de los datos de las particiones cuantificadas. La reconstrucción puede ser realizada como se describe en este documento. En un ejemplo, el bloque 604 incluye la transmisión de una tabla de (q, Q^{-1} (q)) valores junto con los valores q cuantificados, y la reconstrucción incluye buscar valores q cuantificados en la tabla.

En el bloque 608, las particiones recibidas son agregadas en las correspondientes de los nodos. Esto se corresponde con la producción de franjas 534, 536, 538 agregadas en la Figura 5. La agregación puede incluir, por ejemplo, sumar los gradientes reconstruidos.

En algunos ejemplos, en el bloque 610, los valores de modificación, por ejemplo, los valores de gradientes, en las particiones agregadas pueden ser además procesados, por ejemplo, como se discutió anteriormente con referencia a la Figura 5. En un ejemplo, el procesamiento incluye el suavizado del momento. En un ejemplo, el procesamiento incluye normalización de AdaGrad de los gradientes.

En algunos ejemplos, en el bloque 612, los valores de modificación, por ejemplo, los valores de gradientes, en las particiones agregadas pueden ser cuantificados. La cuantificación puede usar la misma función de cuantificación usada en el bloque 404, o una función de cuantificación diferente. Los valores reconstruidos pueden determinarse como se describe en este documento con respecto a la cuantificación de las matrices de gradientes (bloque 404).

En el bloque 614, los datos agregados, por ejemplo, las particiones agregadas cuantificadas, son transmitidas desde nodos individuales, que producen las particiones, a los otros nodos. Por ejemplo, la franja 534 agregada es transmitida desde el Nodo 1 a los Nodos 2 y 3, e igualmente para las otras transferencias descritas en la Tabla 3 anterior.

En algunos ejemplos, en el bloque 616, las particiones agregadas son reconstruidas, por ejemplo, como se describió anteriormente con referencia al bloque 606. El bloque 616 puede ser seguido por el bloque 410, Figura 4.

El proceso ejemplar de la Figura 6 puede también ser usado con otros modelos de cálculo en lugar de un modelo de red neuronal, y con otros valores de modificación en lugar de los gradientes.

Resultados ilustrativos

Varios experimentos fueron realizados para probar un sistema para entrenamiento de una DNN según varios ejemplos en este documento. Una CD-DNN-HMM (“modelo”) fue entrenada en el conjunto de entrenamiento SWBD-I (309 h de audio). El modelo tenía siete capas ocultas de dimensión 2048 y una dimensión de salida de 9304, para un total de M = 46M parámetros del modelo. El conjunto de prueba usado fue Hub-5’00 (1831 declaraciones). Las pruebas fueron realizadas en un servidor equipado con 8 tarjetas GPU K20Xm TESLA NVIDIA. Las pruebas fueron también realizadas en una granja de servidores de 24 servidores duales K20Xm conectados a través de INFINIBAND.

El entrenamiento de la DNN usó cuantificación de 1 bit como se describió anteriormente, con 0 (cero) como el umbral de cuantificación. En una prueba, las primeras 24 h de datos fueron procesadas sin paralelismo ni cuantificación. Los datos restantes fueron procesados mediante el uso de cuantificación de 1 bit con realimentación del error como se discutió anteriormente (por ejemplo, Ecuaciones (10) y (11) juntas), con K = 4. La tasa de error de palabra y precisión de trama de entrenamiento no fueron alteradas de manera significativa por la adición de cuantificación con realimentación del error. En otra prueba, pesos de aprendizaje adaptativo de AdaGrad fueron aplicados a los gradientes cuantificados. Esta configuración mejoró la precisión de trama en 1,7% sobre el AdaGrad aplicado a los gradientes no cuantificados. Esta configuración, con K = 4, proporcionó un tiempo de entrenamiento de 8,1 h. Esto se compara con un tiempo de entrenamiento de 35 h. para una prueba correspondiente no paralelizada. En consecuencia, la operación de paralelización y uso de gradientes cuantificados puede proporcionar una mejora sustancial en la velocidad de entrenamiento, esto es, una reducción sustancial en el tiempo de entrenamiento. Como se indicó anteriormente, este aumento en la velocidad no sacrifica la calidad de los resultados de la DNN 114 entrenada. En otra prueba, la DNN con AdaGrad fue probada con y sin un ajuste del tamaño del lote (por ejemplo, selección 224 de lotes, Figura 2). Usar el ajuste del tamaño del lote redujo el tiempo de entrenamiento desde 41 h a 35 h. Las pruebas fueron también realizadas mediante la comparación del paralelismo de datos y modelo. En varios ejemplos, solo se usó el paralelismo. Por ejemplo, una prueba de un sistema con paralelismo de 4 x 2 (datos x modelo) tuvo una velocidad de entrenamiento de 40,9 kfps, comparado con una velocidad más alta de 50,6 kfps para un sistema con un paralelismo de 8 x 1. En una prueba, un modelo de escala de producción de 160M parámetros completó un paso a través de 3.300 horas de datos de entrenamiento bajo 24 horas de tiempo transcurrido. Estos ejemplos demuestran que la cuantificación, por ejemplo, cuantificación de 1 bit, acelera la transferencia de datos y hace el SGD de datos paralelos factible sin pérdida sustancial de precisión.

Conclusión

Aunque las técnicas han sido descritas en un lenguaje específico de características estructurales y actos metodológicos, se debe entender que las reivindicaciones anexas no limitan necesariamente las características o actos descritos. Más bien, las características y actos son descritos como implementaciones ejemplares de tales técnicas.

Las operaciones de los procesos ejemplares son ilustradas en bloques individuales y resumidas con referencia a esos bloques. Los procesos son ilustrados como flujos lógicos de bloques, cada bloque de los cuales puede representar una o más operaciones que pueden ser implementadas en hardware, software, o una combinación de ellos. En el contexto del software, las operaciones representan instrucciones ejecutables por ordenador almacenadas en uno o más medios legibles por un ordenador que, cuando son ejecutadas por uno o más procesadores, permiten que el uno o más procesadores realicen las operaciones enumeradas. De manera general, las instrucciones ejecutables por un ordenador incluyen rutinas, programas, objetos, módulos, componentes, estructuras de datos, y similares que realizan funciones particulares o implementan tipos de datos abstractos particulares. El orden en el cual las operaciones son descritas no pretende ser interpretado como una limitación, y cualquier número de las operaciones descritas pueden ser ejecutadas en cualquier orden, combinadas en cualquier orden, subdivididas en múltiples sub-operaciones, o ejecutadas en paralelo para implementar los procesos descritos. Los procesos descritos pueden ser realizados mediante recursos asociados con uno o más dispositivo o dispositivos 210 de cálculo o unidad o unidades 212 de procesamiento, tales como aceleradores u otras unidades 212 de procesamiento descritas anteriormente. Tales dispositivos pueden incluir, por ejemplo una o más CPU o GPU interna o externa, o una o más piezas de hardware lógico tal como FPGA o DSP.

Todos los métodos y procesos descritos anteriormente pueden ser realizados en, y totalmente automatizados a través de, módulos de código de software ejecutados por uno o más ordenadores o procesadores de propósito general. Los módulos de código pueden almacenarse en cualquier tipo de medio legible por un ordenador u otro dispositivo de almacenamiento informático. Algunos o todos los métodos pueden ser realizados en hardware informático especializado.

El lenguaje condicional tal como, entre otros, “puede” o “podría”, a menos que se indique específicamente lo contrario, se entienden dentro del contexto para presentar que ciertos ejemplos incluyen, mientras otros ejemplos no incluyen, ciertas características, elementos o pasos. Así, tal lenguaje condicional no pretende generalmente implicar que ciertas características, elementos o pasos son requeridos en forma alguna para uno o más ejemplos o que uno o más ejemplos necesariamente incluyen lógica para decidir, con o sin la entrada del usuario o solicitud, si ciertas características, elementos o pasos están incluidos o han de ser realizados en cualquier ejemplo particular. El lenguaje conjuntivo tal como la frase “al menos uno de X, y o Z”, a menos que se indique específicamente lo contrario, ha de entenderse como la presentación de que un elemento, término, etc. puede ser tanto X, como Y o Z o una combinación de ellos.

Cualquier descripción de rutinas, elementos o bloques en los diagramas de flujo descritos en este documento o representados en las figuras anexas deberían entenderse como una representación potencial de módulos, segmentos, o partes de código que incluyen una o más instrucciones ejecutables para implementar funciones lógicas o elementos específicos en la rutina. Se debería enfatizar que muchas variaciones y modificaciones pueden hacerse a los ejemplos descritos anteriormente, los elementos de los cuales deben entenderse como que están entre otros. El alcance de la invención es definido por las siguientes reivindicaciones, ejemplos aceptables.

REIVINDICACIONES

1. Un método (400) implementado por ordenador para entrenar una red neuronal que comprende:

determinar (402) matrices de gradientes de un modelo de cálculo de un problema de optimización, el modelo de cálculo que incluye una red neuronal;

cuantificar (404) las matrices de gradientes mediante el uso de matrices de errores de cuantificación almacenados;

actualizar (406) las matrices de errores de cuantificación mediante el uso de las matrices de gradientes cuantificados correspondientes; intercambiar las matrices de gradientes cuantificados con un número de unidades de procesamiento; y modificar el modelo de cálculo según las matrices de gradientes cuantificados; y

además comprende repetir los pasos de la determinación (402), cuantificación (404), actualización (406), e intercambio para cada uno de una pluralidad de mini lotes (218) de datos entrenados del modelo de cálculo, el intercambio para un primero de los mini lotes que es realizado en paralelo con la determinación (402), cuantificación (404), o actualización (406) para un segundo de los mini lotes; y

donde la cuantificación (404) de las matrices de gradientes para un mini lote actual comprende determinar los valores de cuantificación mediante la adición de un error de cuantificación para un mini lote anterior a un valor de gradiente para el mini lote actual.

2. El método (400) como recita la reivindicación 1, donde la determinación (402), cuantificación (404), y actualización (406) son realizadas por unidades de procesamiento individuales para las respectivas matrices de gradientes y las matrices de errores de cuantificación correspondientes a las matrices de gradientes correspondientes.

3. El método (400) como recita la reivindicación 2, el intercambio comprende intercambiar (408) las matrices de gradientes cuantificados.

4. El método (400) como recitan tanto la reivindicación 2 como la reivindicación 3, donde las unidades de procesamiento intercambian las matrices de gradientes cuantificados de manera síncrona.

5. El método (400, 600) como en cualquiera de las reivindicaciones 2-4, el intercambio (408) comprende:

particionar (602) las matrices de gradientes cuantificados;

proporcionar (604) particiones individuales de las matrices de gradientes cuantificados a las unidades de procesamiento respectivas;

agregar (608) las particiones recibidas en las unidades de procesamiento; y

transmitir (614) los datos agregados desde las unidades de procesamiento individuales a las otras unidades de procesamiento.

6. El método (400) como recitan cualquiera de las reivindicaciones anteriores, además incluye ajustar (414) un factor de paralelización como una función del tamaño del lote en base al menos en parte a las mediciones de tiempo.

7. El método (400) como recitan cualquiera de las reivindicaciones anteriores, la cuantificación (404) comprende determinar una representación de bit único aproximada para cada elemento de las matrices de gradientes.

8. Un medio (312) legible por un ordenador que tiene en él instrucciones ejecutables por un ordenador, las instrucciones ejecutables por un ordenador tras su ejecución configuran un ordenador para llevar a cabo el método (400) de cualquiera de las reivindicaciones precedentes.

9. Un sistema para entrenar una red neuronal que comprende:

uno o más medios (312) legibles por un ordenador que tienen en ellos una pluralidad de módulos y un modelo de cálculo de un problema de optimización, el modelo de cálculo que incluye una red neuronal; y

una pluralidad de unidades de procesamiento, cada una que incluye al menos una unidad (306) de procesamiento, cada unidad de procesamiento acoplada operativamente a al menos uno de los medios (312) legibles por un ordenador, las unidades de procesamiento adaptadas para intercomunicarse y para ejecutar módulos de la pluralidad de módulos que comprenden:

un módulo (326) de determinación-actualización configurado para determinar valores de gradientes del modelo de cálculo;

un módulo (324) de cuantificación configurado para los valores de gradientes determinados mediante el uso de valores de errores de cuantificación almacenados y para actualizar los valores de errores de cuantificación mediante el uso de los valores de los gradientes determinados y los valores de los gradientes cuantificados;

un módulo (330) de transferencia configurado para transmitir al menos algunos de los valores de gradientes cuantificados a al menos una más de las unidades (306) de procesamiento; y

un módulo (328) de actualización configurado para modificar el modelo de cálculo almacenado según los valores de gradientes cuantificados recibidos;

donde los módulos están configurados para operar para cada uno de una pluralidad de mini lotes (218) de datos de entrenamiento del modelo de cálculo, la transmisión para un primero de los mini lotes que es realizada en paralelo con operaciones de los módulos para un segundo de los mini lotes;

donde los módulos están configurados tal que la cuantificación (404) de los valores de gradientes para un mini lote actual comprende determinar los valores de cuantificación mediante la adición de un error de cuantificación para un mini lote anterior a un valor de gradiente para el mini lote actual.

10. El sistema como recita la reivindicación 9, donde al menos una de las unidades de procesamiento transmite los valores de gradientes cuantificados directamente a al menos otra de las unidades de procesamiento.

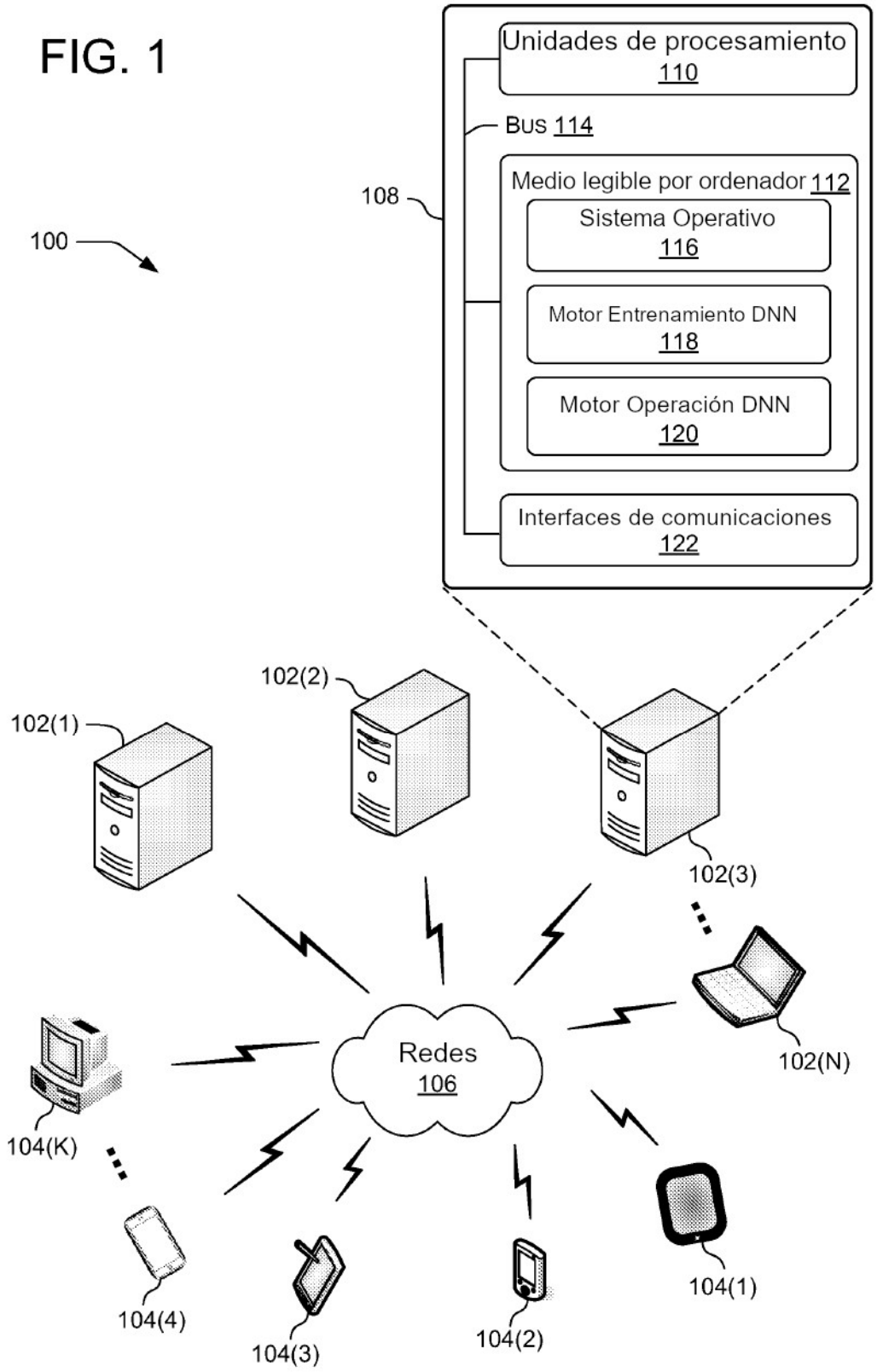
11. El sistema como recitan tanto la reivindicación 9 como 10, donde cada unidad de procesamiento incluye una memoria (320) acoplada con al menos una unidad (306) de procesamiento respectiva y configurada para almacenar estados de cuantificación privados respectivos que incluyen los valores de errores almacenados.

12. El sistema como recitan cualquiera de las reivindicaciones 9-11, donde el módulo (326) de determinación de la actualización es configurado para determinar los valores de gradientes mediante el uso de un algoritmo de descenso de gradiente estocástico.

13. El sistema como recitan cualquiera de las reivindicaciones 9-12, donde cada una de las unidades (306) de procesamiento es acoplada operativamente al medio (312) legible por un ordenador y configurada para ejecutar al menos el módulo (326) de determinación de la actualización.

14. El sistema como recitan cualquiera de las reivindicaciones 9-13, además incluye una barra cruzada que conecta comunicativamente las unidades de procesamiento, donde las unidades de procesamiento están configuradas para ejecutar el módulo (330) de transferencia para transmitir al menos algunos de los valores de gradientes cuantificados a través de la barra cruzada en paralelo con la ejecución del módulo (326) de determinación de la actualización.

FIG. 1


Unidades de procesamiento
110
Bus 114
Medio legible por ordenador 112
Sistema Operativo
116
Motor Entrenamiento DNN
118
Motor Operación DNN
120
Interfaces de comunicaciones
122
108
100
102(1)
102(2)
102(3)
102(N)
Redes
106
104(K)
104(4)
104(3)
104(2)
104(1)

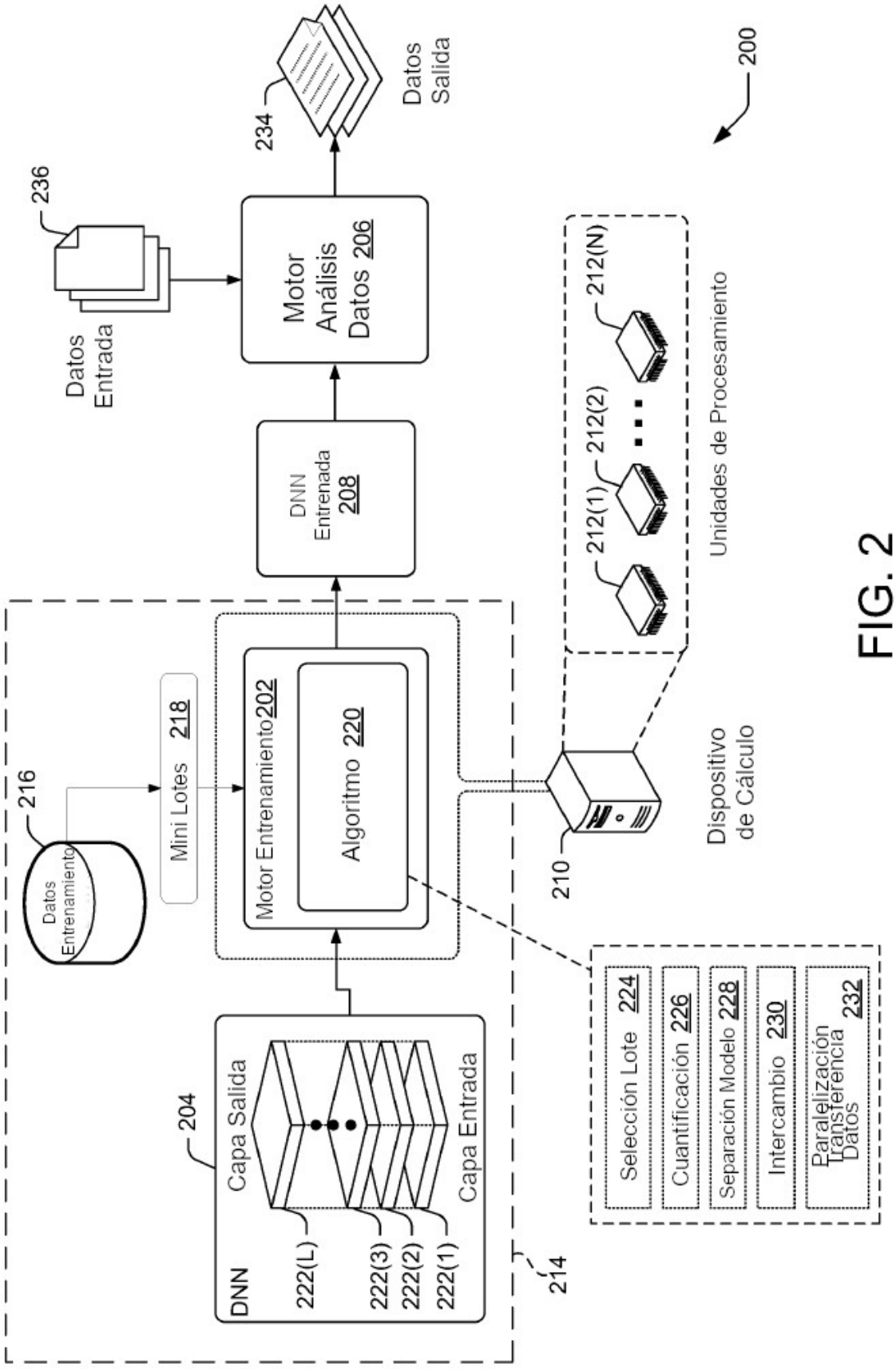

200
Datos Salida
234
Datos Entrada
236
Motor Análisis Datos 206
DNN Entrenada 208
Unidades de Procesamiento
212(N)
212(2)
212(1)
Dispositivo de Cálculo
210
214
216
Datos Entrenamiento
Mini Lotes 218
Motor Entrenamiento 202
Algoritmo 220
DNN
204
Capa Salida
Capa Entrada
222(L)
222(3)
222(2)
222(1)
Selección Lote 224
Cuantificación 226
Separación Modelo 228
Intercambio 230
Paralelización Transferencia Datos 232

FIG. 2

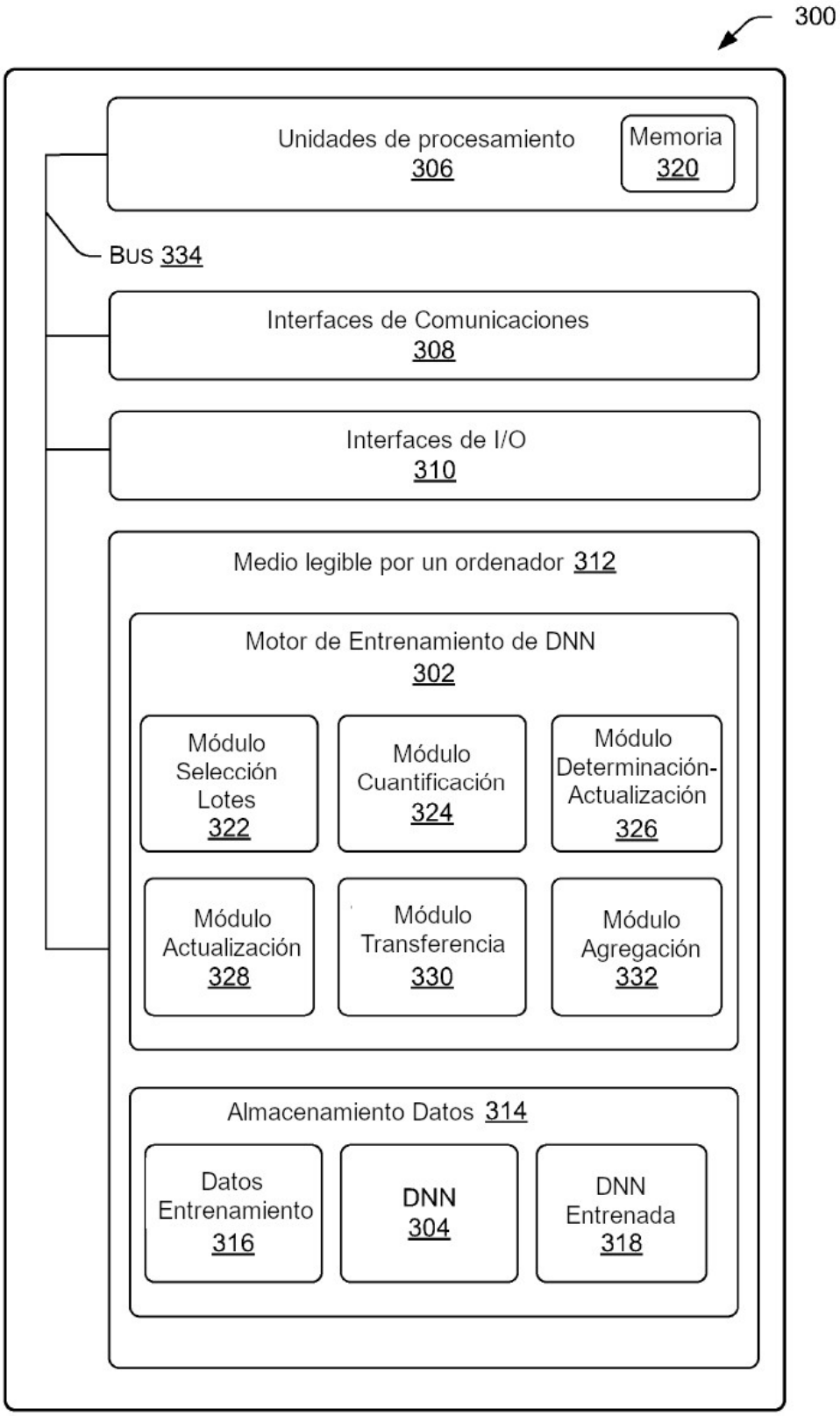
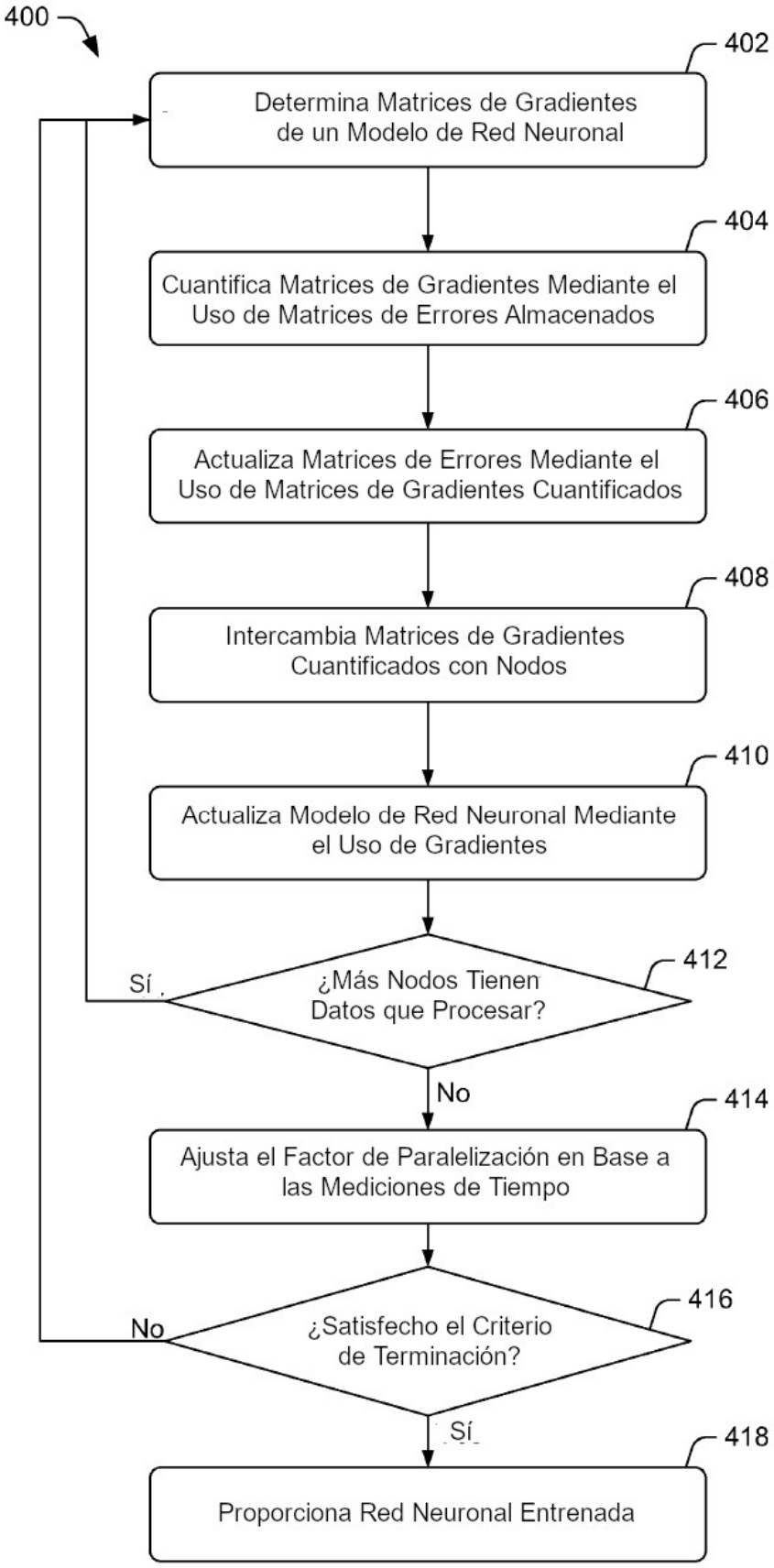

300
Unidades de procesamiento
306
Memoria
320
Bus 334
Interfaces de Comunicaciones
308
Interfaces de I/O
310
Medio legible por un ordenador 312
Motor de Entrenamiento de DNN
302
Módulo Selección Lotes
322
Módulo Cuantificación
324
Módulo Determinación-Actualización
326
Módulo Actualización
328
Módulo Transferencia
330
Módulo Agregación
332
Almacenamiento Datos 314
Datos Entrenamiento
316
DNN
304
DNN Entrenada
318

FIG. 3

FIG. 4


400
402
Determina Matrices de Gradientes de un Modelo de Red Neuronal
404
Cuantifica Matrices de Gradientes Mediante el Uso de Matrices de Errores Almacenados
406
Actualiza Matrices de Errores Mediante el Uso de Matrices de Gradientes Cuantificados
408
Intercambia Matrices de Gradientes Cuantificados con Nodos
410
Actualiza Modelo de Red Neuronal Mediante el Uso de Gradientes
412
¿Más Nodos Tienen Datos que Procesar?
Sí
No
414
Ajusta el Factor de Paralelización en Base a las Mediciones de Tiempo
416
¿Satisfecho el Criterio de Terminación?
No
Sí
418
Proporciona Red Neuronal Entrenada

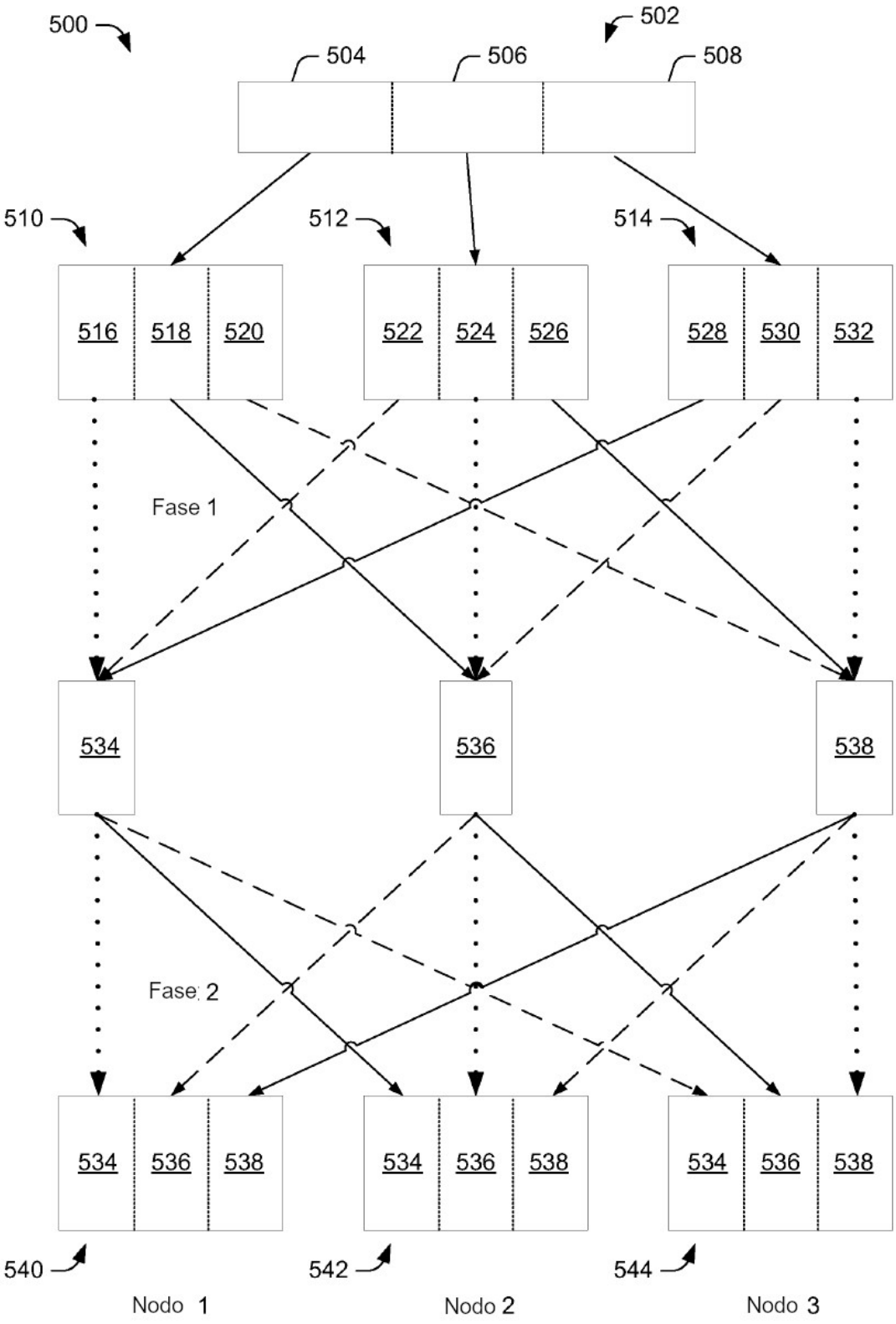

500
502
504
506
508
510
512
514
516
518
520
522
524
526
528
530
532
Fase 1
534
536
538
Fase 2
534
536
538
534
536
538
534
536
538
540
542
544
Nodo 1
Nodo 2
Nodo 3

FIG. 5

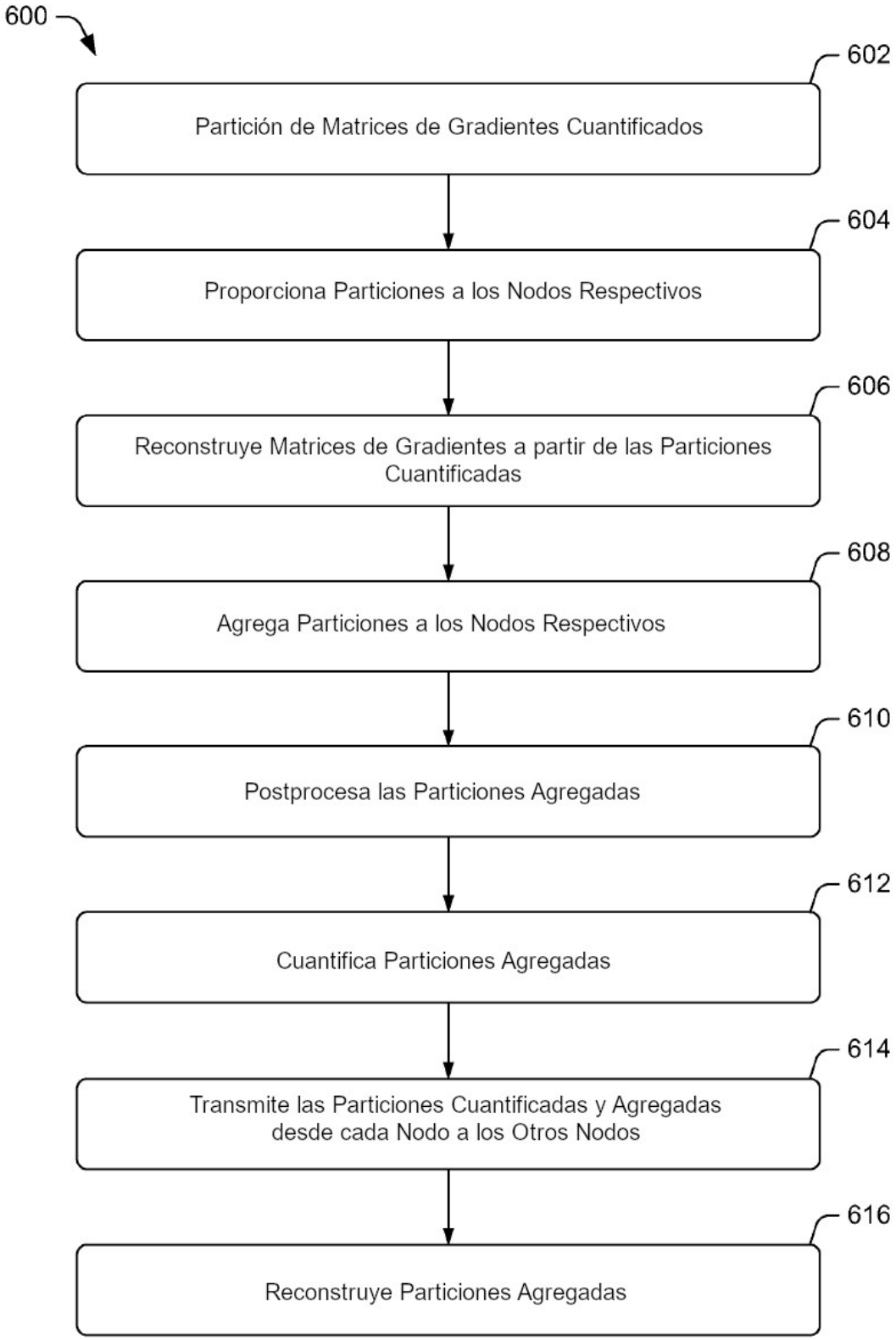

600
602
Partición de Matrices de Gradientes Cuantificados
604
Proporciona Particiones a los Nodos Respectivos
606
Reconstruye Matrices de Gradientes a partir de las Particiones Cuantificadas
608
Agrega Particiones a los Nodos Respectivos
610
Postprocesa las Particiones Agregadas
612
Cuantifica Particiones Agregadas
614
Transmite las Particiones Cuantificadas y Agregadas desde cada Nodo a los Otros Nodos
616
Reconstruye Particiones Agregadas

FIG. 6