



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 268 084**

51 Int. Cl.:
G11C 8/10 (2006.01)
G06F 12/02 (2006.01)
G06F 15/18 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Número de solicitud europea: **02767715 .2**
86 Fecha de presentación : **14.10.2002**
87 Número de publicación de la solicitud: **1488425**
87 Fecha de publicación de la solicitud: **22.12.2004**

54 Título: **Memoria digital direccionable de manera inexacta.**

30 Prioridad: **28.03.2002 GB 0207372**

45 Fecha de publicación de la mención BOPI:
16.03.2007

45 Fecha de la publicación del folleto de la patente:
16.03.2007

73 Titular/es: **Cogniscience Limited**
Incubator Building, Grafton Street
Manchester M13 9XX, GB

72 Inventor/es: **Furber, Stephen Byram**

74 Agente: **Torner Lasalle, Elisabet**

ES 2 268 084 T3

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

Memoria digital direccionable de manera inexacta.

La presente invención versa acerca de una memoria digital.

El empleo cada vez más generalizado de sistemas informatizados y de dispositivos electrónicos digitales ha dado como resultado que se propongan varias estructuras diferentes de memorias. La amplia gama de aplicaciones que precisan de memoria digital significa que hacen falta memorias dotadas de propiedades diferentes.

Por ejemplo, las memorias para el empleo en un ordenador convencional para programas almacenados requieren niveles elevados de integridad de datos, dado que un solo *bit* incorrecto hace que el ordenador funcione indebidamente. Una memoria convencional de ordenador requiere que se conozca con exactitud una dirección de lectura para la recuperación de la información almacenada en una dirección concreta. Esto es así porque una memoria convencional almacena una cadena de N dígitos binarios (denominada *palabra* en lo sucesivo) en una ubicación específica dentro de la memoria. Solamente se almacena una palabra en una ubicación concreta en un momento dado.

En algunas aplicaciones que implican el almacenamiento de grandes cantidades de datos es aceptable cierto grado de error en los datos recuperados de la memoria, y pueden ser pasados por alto por el sistema. Además, en algunos casos puede resultar necesario permitir la recuperación de datos empleando una dirección inexacta de lectura. La presente invención tiene que ver con una memoria adecuada para tales aplicaciones.

Una estructura de memoria conocida ha sido desarrollada por Kanerva y aparece descrita en Kanerva P., *Sparse Distributed Memory* [Memoria dispersa distribuida], MIT Press, 1988, y también en Kanerva P., "Self-propagating Search: A Unified Theory of Memory" [Búsqueda de autopropagación: Teoría unificada de la memoria], Informe N° CSLI-84-7, Stanford University, 1984. Aquí hay conectados muchos decodificadores de dirección a una memoria de datos que consta de varias líneas de palabra. Cada decodificador de dirección va conectado a una línea de palabra diferente dentro de la memoria de datos. Cada decodificador está identificado por una dirección en forma de un número binario grande.

Cuando se escriben datos a memoria, se presenta al conjunto de decodificadores una dirección de entrada, y se activan los decodificadores que tienen un identificador dentro de una distancia de Hamming predeterminada respecto de la dirección de entrada. Las líneas de palabra conectadas a los decodificadores activados son seleccionadas a continuación para recibir datos de escritura. Cuando han de recuperarse datos de la memoria, a los decodificadores se les presenta una dirección, y se activan nuevamente varias líneas de palabra. La suma de cada *bit* de cada línea de palabra activada, y la aplicación del umbral apropiado permite que se recuperen los datos escritos a la memoria. Escribir datos y recuperarlos de esta manera permite que se minimicen los efectos de los errores de dirección y de datos.

La estructura de memoria propuesta por Kanerva tiene varios inconvenientes. Uno de estos inconvenientes es que los cálculos de la distancia de Hamming requieren *hardware* diseñado específicamente si han de efectuarse de manera eficiente.

En la patente WO 90/04830 (Universities Space Research Association) se describe una estructura de memoria alternativa que procura superar este inconveniente. Este sistema adopta la arquitectura básica propuesta por Kanerva, pero determina qué decodificadores de dirección deben activarse en respuesta a una dirección concreta de entrada empleando un criterio diferente. En este sistema, se asignan valores específicos de "0" y "1" a un subconjunto de *q bits* de cada dirección de decodificador. Cuando se presenta al sistema una dirección de entrada, se comparan los valores de los *q bits* de la dirección que se presenta con los de cada decodificador de dirección. Si la dirección que se presenta tiene los mismos valores en esos *q bits* que un decodificador de dirección concreto, se activa ese decodificador. El preámbulo de la reivindicación 1 está basado en este documento.

Es un objeto de la presente invención proporcionar una estructura de memoria alternativa a las descritas más arriba.

De acuerdo con un primer aspecto de la presente invención, se plantea una configuración de memoria para su empleo en un sistema informático, constanding la memoria de una pluralidad de decodificadores de dirección, a cada uno de los cuales se le asigna un identificador dotado de un número predefinido de *bits*, teniendo cada *bit* estados seleccionables primero y segundo, y una memoria de datos dotada de una pluralidad de líneas de palabra de una longitud predefinida, siendo cada uno de dichos decodificadores de dirección susceptible de ser activado para seleccionar una de entre la pluralidad de líneas de palabra, y constanding los decodificadores de dirección de medios para recibir una dirección de entrada que tenga un número predefinido de *bits* y de medios para comparar el identificador de un decodificador de dirección con la dirección de entrada, donde la memoria conste además de medios para activar un decodificador de dirección si al menos un número mínimo predefinido de *bits* puestos en el primer estado seleccionable en la dirección de entrada se corresponden con *bits* puestos en el primer estado seleccionable en el identificador del decodificador.

Los beneficios y ventajas proporcionadas por la presente invención surgen en parte de la activación de los decodificadores de dirección basándose en los *bits* puestos en el primer estado seleccionable en una dirección de entrada, mientras que se ignoran los *bits* puestos en el segundo estado seleccionable. En cambio, los sistemas descritos más

arriba correspondientes al estado previo de la especialidad activan los decodificadores de dirección comparando los *bits* puestos en los estados seleccionables primero y segundo.

Preferiblemente, el medio para comparar el identificador de un decodificador de dirección con la dirección de entrada considera la correspondencia posicional entre *bits* puestos en el primer estado seleccionable en la dirección de entrada y *bits* puestos en el primer estado seleccionable en los identificadores de los decodificadores.

Cada identificador de los decodificadores de dirección puede tener un número igual de *bits* puestos en el primer estado seleccionable. El medio para recibir una dirección de entrada puede estar configurado para recibir direcciones que contengan un número predefinido de *bits* puestos en el primer estado seleccionable. Adecuadamente, el número predefinido de *bits* puestos en el primer estado seleccionable en una dirección de entrada es igual al número de *bits* puestos en el primer estado seleccionable en cada uno de los identificadores de decodificadores de dirección.

Preferiblemente, la memoria de datos comprende una pluralidad de memorias de *bit* único, de tal modo que cada *bit* de cada línea de palabra esté almacenado en una memoria de un solo *bit*. La memoria de datos puede consistir en una línea de entrada de datos que contenga un número igual de *bits* que cada una de la pluralidad de líneas de palabra. La configuración de memoria puede además constar de un medio para escribir datos para copiar datos de la línea de entrada de datos a líneas de palabra activadas por los decodificadores de dirección. La línea de entrada de datos puede estar configurada para recibir datos de entrada que contengan un número predefinido de *bits* puestos en el primer estado seleccionable.

La configuración de memoria puede además constar de un medio para sumar los valores almacenados en cada *bit* de las líneas de palabra activadas por un decodificador de dirección para generar un valor de nivel de activación para cada *bit*. La configuración de memoria puede además constar de un medio para generar una palabra de salida que contenga un número predefinido de *bits* puestos en el primer estado seleccionable. Los *bits* puestos en el primer estado seleccionable en la salida pueden ser un número predefinido de *bits* que tengan el nivel de activación más elevado.

La memoria puede ser implementada utilizando una pluralidad de neuronas artificiales conectadas entre sí para formar una red neuronal. La pluralidad de decodificadores de dirección puede ser representada por una pluralidad de neuronas decodificadoras de dirección, y la memoria de datos puede ser representada por una pluralidad de neuronas de datos.

La invención proporciona también una configuración de memoria de red neuronal para su empleo en un sistema informático, como se indica en la reivindicación 15. La memoria consta de una pluralidad de neuronas de decodificación de dirección, cada una de las cuales está conectada a un número predefinido de neuronas de entrada, y de una memoria de datos dotada de una pluralidad de neuronas de datos, siendo susceptible de activación cada una de dichas neuronas de decodificación de dirección para seleccionar alguna de entre la pluralidad de las neuronas de datos, y constando las neuronas de decodificación de dirección de un medio para recibir una señal que represente un disparo de una neurona de entrada a la que esté conectado, donde una neurona de decodificación de dirección conste de medios para activar neuronas de datos si se reciben señales de disparo de al menos un número mínimo predefinido de neuronas de entrada a las que esté conectada la neurona de decodificación de dirección.

En la práctica, cada neurona de decodificación de dirección puede conectarse con cada neurona de entrada, asignándose a cada una de estas conexiones un peso de "1" o "0". En tal implementación, un peso de "0" tiene el mismo efecto que el que no exista conexión entre una neurona concreta de entrada y una neurona de decodificación de dirección. De esto modo, únicamente las señales de disparo procedentes de neuronas de entrada conectadas a una neurona de decodificación de dirección, de modo que la conexión tenga un peso de "1", representarán una conexión, tal como se describe en el párrafo anterior.

De acuerdo con un segundo aspecto de la presente invención, se plantea un método para hacer funcionar una memoria para su empleo en un sistema informatizado, constando la memoria de una pluralidad de decodificadores de dirección, a cada uno de los cuales se le asigna un identificador dotado de un número predefinido de *bits*, teniendo cada *bit* estados seleccionables primero y segundo, y una memoria de datos dotada de una pluralidad de líneas de palabra de una longitud predefinida, siendo susceptible de activación cada uno de dichos decodificadores de dirección para seleccionar una de entre la pluralidad de líneas de palabra, donde se facilite una dirección de entrada dotada de un número predefinido de *bits* al decodificador de dirección, donde el identificador de un decodificador de dirección se compara con la dirección de entrada y los decodificadores de dirección se activan si al menos un número mínimo predefinido de *bits* puestos en el primer estado seleccionable en la dirección de entrada se corresponden con *bits* puestos en el primer estado seleccionable en el identificador del decodificador.

Los datos de entrada pueden ser representados en una entrada de datos de la memoria de datos, y los datos pueden escribirse a líneas de palabra activadas por los decodificadores de dirección activados. Preferiblemente, el número mínimo predefinido de *bits* se pone de tal modo que se activen menos de 100 decodificadores de dirección por cualquier dirección válida de entrada. Más preferiblemente, el número mínimo predefinido de *bits* se pone de tal modo que se activen menos de 50 decodificadores de dirección por cualquier dirección válida de entrada. Lo más preferible es que el número mínimo predefinido de *bits* se ponga de tal modo que se activen no más de 20 y no menos de 11 decodificadores de dirección por cualquier dirección válida de entrada.

La invención plantea además un medio portador que lleva un medio de código legible para ordenadores para hacer que un ordenador ejecute un procedimiento en conformidad con el método expuesto más arriba, y un programa de ordenador para realizar el método expuesto más arriba.

A continuación se describirá un ejemplo de realización de la presente invención, a título de ejemplo, haciendo referencia a los dibujos adjuntos, en los que:

la Figura 1 es una ilustración esquemática de una configuración de memoria conocida en el estado previo dentro de la especialidad;

la Figura 2 es una ilustración esquemática de una configuración de memoria en conformidad con la presente invención;

la Figura 3 es un trazado tridimensional que muestra la capacidad de almacenamiento de una configuración concreta de la memoria ilustrada en la figura 2;

la Figura 4 es un trazado de contorno del trazado tridimensional de la figura 3;

la Figura 5 es el trazado de contorno de la Figura 4 corregido para que tenga en cuenta decodificadores de dirección activados independientemente;

la Figura 6 es una ilustración esquemática de una red neuronal artificial que puede emplearse para implementar una memoria en conformidad con la presente invención;

la Figura 7 es una ilustración esquemática de una primera configuración para conexiones entre neuronas de entrada y neuronas de decodificación de dirección en una red neuronal artificial dotada de una forma general similar a la de la figura 6; y

la Figura 8 es una ilustración esquemática de una configuración para conexiones entre neuronas de entrada y neuronas de decodificación de dirección alternativa a la mostrada en la Figura 7.

La Figura 1 ilustra la configuración de direcciones descrita por Kanerva y esbozada más arriba. Se facilita una pluralidad de líneas de direcciones 1 en las que puede transmitirse una dirección. La dirección es un número binario, y cada línea de direcciones 1 transmite un *bit* de la dirección, de modo que hagan falta n líneas para representar una dirección de n *bits*. Las líneas de dirección constituyen la entrada a un sistema de decodificadores de dirección 2, y cada decodificador de dirección 2 tiene un identificador en la forma de un número binario de n *bits*. Los decodificadores de dirección 2 están conectados a una memoria de datos 3, que consta de un número de líneas de palabra para almacenar datos. Cada línea de palabra tiene una capacidad de almacenamiento de M *bits*. Cada *bit* de cada línea de palabra se almacena en el contador 4. Los datos que han de escribirse a la memoria de datos se transmiten en una línea de entrada 5. Los datos leídos de la memoria pasan por líneas de datos 8 a bloques sumadores 6 y son objeto de salida desde los bloques sumadores 6 por líneas de salida de datos 7.

Típicamente, el número de líneas de dirección 1 es de la magnitud de mil, de modo que a los decodificadores de dirección 2 se les presente un número binario de mil *bits*. Dado que con un número binario de mil *bits* puede representarse un número muy grande de direcciones (aproximadamente 10^{301}), no resulta práctico tener una línea aparte de palabra para almacenar datos para cada dirección posible. En consecuencia, se implementa un gran muestreo aleatorio de direcciones dentro del espacio de direcciones posible. Típicamente se implementa un millón de direcciones, y cada una de estas se asigna a un decodificador de dirección 2. Por esta razón se describe como “dispersa” la memoria propuesta por Kanerva, pues el millón (10^6) de direcciones de decodificación está distribuido de forma dispersa dentro del espacio disponible para direcciones (10^{301} direcciones).

Cuando se usa, una dirección se transmite por las líneas de dirección 1 y es recibida por los decodificadores de dirección 2. Se efectúa una comparación entre la dirección transmitida y la dirección de cada decodificador de dirección. Esta comparación se realiza empleando un cálculo de distancia de Hamming, que computa cuántos *bits* de las direcciones transmitidas son diferentes a las direcciones asociadas con cada decodificador de dirección 2. El cálculo de la distancia de Hamming se presenta en la ecuación (1):

$$H(x, y) = \sum_{i=1}^N |x_i - y_i| \quad (1)$$

donde:

x es un vector binario dotado de N elementos;

y es un vector binario dotado de N elementos; y

H es la distancia de Hamming entre x e y , siendo el número de posiciones de *bits* en los que $x_i \neq y_i$.

ES 2 268 084 T3

Se activan los decodificadores dentro de una distancia predefinida de Hamming r de la dirección transmitida x . Es posible imaginar esta región de activación para cada dirección como una esfera centrada en x y dotada de un radio r . Por ejemplo, Kanerva ha mostrado que si cada dirección tiene 1000 *bits*, y la distancia predefinida de Hamming r es 451, entonces la región esférica de activación contiene aproximadamente $\frac{1}{1000}$ direcciones asociadas con los decodificadores de dirección. Cuando se usa, un procesador computa una distancia de Hamming entre cada dirección de decodificación y una dirección de entrada presentada. Si esta distancia es menor que r para un decodificador de dirección concreto, entonces el decodificador se activa.

A continuación se describirán detalles de un proceso para escribir datos a memoria en el sistema propuesto por Kanerva. Cuando se activa un decodificador de dirección dado 2, los datos presentados en la entrada de datos 5 de la memoria se escriben a la línea de palabra que está conectada al decodificador de direcciones activado. Puesto que la activación de un decodificador de dirección se inicia siempre que se da una dirección que esté dentro de la distancia predefinida de Hamming de ese decodificador de dirección, los datos se escriben a un número de líneas diferentes de palabra. De modo similar, hay varias direcciones diferentes que activan el mismo decodificador de dirección. Esto quiere decir que los datos se escribirán a una línea de palabra concreta para varias direcciones diferentes de entrada para escritura.

Cada línea de palabra consta de M contadores, cada uno de los cuales almacena un único *bit* de datos. Los contadores de una línea dada de palabra permiten el acúmulo lineal de varias palabras de M *bits* almacenadas en esa línea de palabra. Si se presenta un “1” en un *bit* concreto de una línea de palabra, se incrementa el correspondiente contador 4. Si se presenta un “0” en un *bit* concreto, disminuye el contador 4 asociado con ese *bit*.

Debería notarse que los contadores tiene una capacidad finita, y una vez se haya alcanzado esta capacidad mediante escrituras sucesivas de datos, los intentos adicionales de escritura no surtirán efecto alguno. Por ejemplo, si el contador ha alcanzado su valor máximo mediante escrituras sucesivas de unos, escribir otro “1” a ese contador no puede tener efecto alguno. De modo similar, si el contador ha alcanzado su valor mínimo escribiendo sucesivamente ceros, escribir otro “0” a ese contador no puede tener efecto alguno. Normalmente es suficiente que se empleen contadores de 8 *bits*, cada uno dotado de un intervalo de ± 127 . En la práctica, puede resultar suficiente implementar contadores de 2 o de 4 *bits*. Sin embargo, debería notarse que los contadores ascendentes y descendentes de cualquier longitud de *bits* suponen una recarga considerable en la memoria propuesta por Kanerva.

Los decodificadores de dirección 2 están configurados de tal modo que para un millón de decodificadores de dirección 2, una dirección al azar esté dentro de la distancia predefinida de Hamming de aproximadamente 1000 de los decodificadores. Por lo tanto, para una dirección aleatoria de entrada, se escribirán datos de entrada a 1000 líneas de palabra dentro de la memoria de datos 3, incrementándose o decrementándose los contadores apropiados 4 de cada línea de palabra tal como se ha descrito más arriba.

A continuación se describirán detalles de un proceso para recuperar datos de la memoria en el sistema propuesto por Kanerva.

Se recuperan datos de la memoria facilitando una dirección de entrada en las líneas de dirección 1. Si esta dirección de entrada es idéntica a la empleada para escribir los datos que han de recuperarse, se activarán los mismos decodificadores de dirección, y se recuperarán los datos de las debidas líneas de palabra.

Si se presenta una dirección cercana a la dirección usada para escribir, se activarán para leer los datos muchos de los decodificadores de dirección 2 activados para escribir los datos. Sin embargo, debería notarse que no se activarán algunos decodificadores activados para escribir, y que entonces se activarán otros que no estaban activados con anterioridad. Esto lleva a que haya presente ruido en los datos recuperados. Suponiendo que haya un número suficiente de decodificadores en común entre los procedimientos de lectura y de escritura, siguen saliendo los datos correctos de los sumadores 6, y los efectos del ruido se mitigan. Esto constituye una característica ventajosa de la memoria descrita por Kanerva.

Los valores mantenidos en los contadores 4 de cada línea de palabra conectada a un decodificador de dirección activado son objeto de salida por las líneas de datos 8 durante un ciclo de lectura. Los valores que salen por cada línea de datos 8 son sumados por un sumador 6. Los sumadores cuya salida excede un umbral predefinido generan una salida de “1” por las líneas de salida de datos 7, y los que no exceden el umbral generan una salida de “0”.

Kanerva ha determinado que la estructura y la función de la memoria significa que de los mil *bits* de una dirección de entrada, solo hace falta que sean correctos seiscientos *bits* en una dirección de lectura para permitir una recuperación satisfactoria de esa misma dirección de entrada. Por ejemplo, si un *bit* está puesto a “1” en mil de las líneas de palabra durante la escritura de los datos, se esperaría que el sumador 6 conectado a los contadores que mantienen valores para ese *bit* sacaría 1000. Sin embargo, si solamente se recupera, mediante una dirección de recuperación, una mayoría de las mil líneas de palabra a las que se escribieron datos, el sumador en general sacará un valor que se interprete correctamente. Así, la memoria propuesta por Kanerva tiene propiedades útiles de corrección de errores y permite que se mitiguen los efectos del ruido descritos con anterioridad.

ES 2 268 084 T3

Un código N de M es un código en el que N *bits* de un número binario de M *bits* estén puestos a “1”. Por lo tanto, cualquier código N de M consta de N *bits* puestos a “1”, y de M-N *bits* puestos a “0”. Cada valor representado por el código difiere en el orden en el que aparecen los unos y los ceros. Los códigos N de M se cronometran solos, por cuanto cuando se han recibido los N unos se sabe que el código está completo (o sea, se ha recibido), pero si se han recibido menos de N unos, se sabe que el código está incompleto.

La configuración de memoria de la presente invención toma las propiedades dispersas distribuidas de la memoria de la figura 1, correspondiente al estado previo de la especialidad, e introduce varias aplicaciones de los códigos N de M para dotar a la memoria de un número de mejoras que se exponen de manera más detallada más abajo. Una configuración de memoria que constituye un ejemplo de realización de la presente invención se ilustra a alto nivel en la figura 2. Puede verse que la configuración general es similar a la de la figura 1, y se emplean numerales homólogos de referencia cuando resulta apropiado. Únicamente se analizarán aquí con más detalle las diferencias entre la memoria de la figura 1 y la facilitada por la invención.

En general, la memoria consta de W decodificadores de dirección 2, w de los cuales están activados para una lectura o escritura concretos. Cada uno de los W decodificadores de dirección consta de una dirección de A *bits*, y cada una de estas direcciones consta de a *bits* que tienen un valor de “1”. Por lo tanto, la dirección de cada decodificador de dirección es un código “a de A”. Cuando se introduce una dirección a los decodificadores de dirección 2, tiene i *bits* puestos al valor de “1” y A-i *bits* puestos al valor “0”. Por lo tanto, cualquier dirección de entrada es un código “i de A”. En algunos ejemplos de realización ocurre que todas las direcciones de entrada tienen un número de *bits* con valor “1” igual al de las direcciones de los decodificadores ($i = a$). Sin embargo, hay normalmente un beneficio marginal en permitir que i sea independiente de a.

Cuando los decodificadores de dirección 2 reciben una dirección de entrada, se efectúa una comparación entre el código a de A de cada decodificador y el código i de A de la dirección de entrada. Cada decodificador de dirección tiene un umbral T asociado. El umbral T determina el número de unos que deben coincidir entre la dirección de entrada y la dirección del decodificador.

A continuación se presentarán ejemplos simplificados del procedimiento de activación de los decodificadores. En estos ejemplos, se da por sentado que hay tres decodificadores de dirección (A, B y C), dotado cada uno de un código 3 de 8. Las direcciones de estos decodificadores son como siguen:

A: 01001001

B: 10001010

C: 01101000

En la descripción que sigue, cada una de las direcciones va indiciada de izquierda a derecha, empezando en un índice 0.

En un primer escenario, el umbral para cada decodificador es tal que al menos dos *bits* de una dirección de entrada deben coincidir con al menos dos *bits* de la dirección de decodificación. Por lo tanto, cuando se les presente:

Dirección1: 01001000 (un código 2 de 8)

puede verse que los decodificadores A y C se activan, ya que los *bits* 1 y 4 de la dirección de entrada son unos y los *bits* 1 y 4 de las direcciones de los decodificadores A y C también son unos. Por lo tanto, se supera el umbral de dos *bits* coincidentes. El decodificador B no se activa, pues únicamente es coincidente uno de los unos (el *bit* 4) entre la Dirección 1 y el decodificador B.

De modo similar, cuando se les presente una dirección

Dirección2: 01000011 (un código 3 de 8)

y suponiendo que el umbral T permanezca inalterado, puede verse que la Dirección2 tiene unos en los *bits* 1 y 7 y que el decodificador A tiene unos en las mismas posiciones, lo que significa que se activará el decodificador A. Los decodificadores B y C no se activan, puesto que únicamente coincide un *bit* puesto a “1” de cada decodificador con un “1” en la Dirección2.

El procedimiento de activación descrito más arriba activa w decodificadores de un total de W. Por lo tanto, el modelo de activación es un código w de W. Esto es similar al proceso descrito más arriba con referencia a la figura 1, aunque es importante notar que el umbral T es preferible que esté puesto para garantizar que se dispare un número óptimo o casi óptimo de decodificadores de dirección durante un ciclo dado de lectura o escritura. Se ha descubierto que este número óptimo típicamente es mucho menor que el número activado en la memoria propuesta por Kanerva. El inventor se ha dado cuenta de que los datos pueden ser leídos o escritos a la memoria con suficiente precisión empleando un número menor de activaciones de decodificador, y que activar mil decodificadores, como en el sistema de Kanerva, resulta innecesario. Más abajo se presenta un análisis más detallado de esta característica del sistema de memoria.

A su vez, los w decodificadores activados seleccionan w líneas de palabra de las que han de leerse datos o a las que han de escribirse. Por eso, la comunicación entre los decodificadores de dirección 2 y las líneas de palabra de la memoria de datos 3 es un código w de W (o sea, w decodificadores activados de entre W decodificadores en total). Debería apreciarse que normalmente no resulta práctico garantizar que w sea constante, dado el procedimiento de activación que implica decodificadores de dirección activados independientemente, tal como se ha descrito más arriba. La comunicación de este código w de W de los decodificadores de dirección 2 a la memoria de datos 3 se denota con la flecha 9 en la figura 2.

Los w decodificadores activados seleccionan w líneas de palabra. Se presentan datos de escritura para escribir datos a la memoria de datos 3, y se leen datos de lectura de la memoria de datos 3. La lectura o la escritura es efectuada en cada una de las w líneas activadas de palabra. Se considera que los datos de entrada 5 representan un código “d de D”, donde cada línea de palabra consta de D bits, d de los cuales están puestos a unos. Puesto que cada entrada es un código “d de D”, cada operación de escritura de datos escribirá datos que contengan d unos.

Cada bit de cada línea de palabra es almacenado en una memoria convencional binaria de bit único. Todos los bits de todas las líneas de palabra están puestos inicialmente a “0”. Cuando deba escribirse un “1” a un bit, el valor de “0” es sustituido por un “1”. Una vez que un bit se pone a “1”, nunca vuelve a ponerse a “0”, ni siquiera si un ciclo subsiguiente de escritura intenta escribir “0” en ese bit. O sea, una vez que una operación de escritura de datos ha puesto un valor, cualquier intento adicional en este valor es, simplemente, ruido, y, por lo tanto, puede ser descartado. Este procedimiento de almacenamiento está basado en la naturaleza unipolar de códigos N de M, y las razones de su fiabilidad se dan más abajo.

Cuando han de leerse datos de la memoria de datos 3, se presenta una dirección a los decodificadores de dirección 2, lo que da como resultado que se activen w decodificadores de dirección. A su vez, estos w decodificadores de dirección activan w líneas de palabra en la memoria de datos 3. Cada palabra contiene D bits, y se suma el contenido de cada bit de cada línea activada de palabra, de modo que dé D salidas. En lo sucesivo denominaremos estas sumas niveles de activación. Se sabe que todos los datos de escritura constituyen un código “d de D”, por lo que también será cierto que los datos leídos de memoria constituyan un código “d de D”.

Si un solo ciclo de escritura escribe un valor en una dirección concreta, y esta misma dirección es presentada por una operación de lectura de datos (sin un ciclo intermedio adicional de escritura), puede esperarse que el nivel de activación de cada bit puesto a “1” en los datos de escritura sea w , mientras que el nivel de activación de cada bit puesto a “0” en los datos de escritura será cero. Por lo tanto, puede verse con facilidad qué bits deberían ponerse a “1” en los datos de salida.

Sin embargo, si dos ciclos de escritura escriben datos a direcciones de entrada de escritura diferentes, puede ocurrir que algunos de los decodificadores de dirección activados durante el primer ciclo de escritura se activen también durante el segundo ciclo de escritura. O sea, algunas de las líneas de palabra dentro de la memoria de datos 3 contendrán algunos bits puestos a “1” como resultado del primer ciclo de escritura, y algunos bits puestos a “1” como resultado del segundo ciclo de escritura.

Cuando se presenta una dirección igual a la de la primera dirección de escritura a los decodificadores de dirección, el procedimiento de lectura de datos activa w líneas de palabra, y calcula los niveles de activación, tal como se ha descrito más arriba. Dada la acción del ciclo de escritura intermedio, estos niveles de activación contendrán algunos valores derivados de la segunda escritura, o sea, algunos bits que fueron puestos a “0” en el primer ciclo de escritura tendrán ahora un nivel de activación distinto de cero, aunque estos niveles de activación deberían ser menos que el de los bits puestos a “1” durante el primer ciclo de escritura. O sea, tendrán niveles de activación distintos de cero más de d bits.

Como se sabe que los datos de salida deberían ser un código “d de D”, se aplica un algoritmo modificado de tipo “el ganador se queda con todo” (*winner takes all*, WTA) para determinar los d bits dotados del nivel de activación más elevado. Lo más probable es que estos d bits sean los que fueron puestos a “1” durante la escritura de datos a esa dirección, y, por lo tanto, son puestos a “1” en los datos de salida.

Un algoritmo WTA normalmente determina un único ganador de entre una colección de datos. En este caso, el algoritmo modificado, denominado en lo sucesivo algoritmo k -WTA, determina d valores que tienen el nivel de activación más elevado, y los bits que tienen estos valores son puestos a “1” en la salida. O sea, estos d bits son los “ganadores” en el algoritmo. Puede implementarse un algoritmo k -WTA ordenando los D valores empleando cualquier algoritmo adecuado de ordenación, y seleccionando los d valores más altos de entre esos valores ordenados.

Es importante notar que la aplicación de la codificación de N de M a la memoria de datos permite que se emplee este tipo de proceso de lectura, ya que se sabe cuántos unos deberían aparecer en los datos de salida. En el sistema propuesto por Kanerva, donde los datos escritos a la memoria son binarios, no hay manera de saber cuántos unos deberían aparecer en los datos de salida, y, en consecuencia, Kanerva utiliza contadores como elementos de almacenamiento para acumular datos escritos a cada bit, sumándose el contenido de los contadores, y aplicándose un umbral para generar una salida.

Se notó con anterioridad que una memoria de un solo *bit* nunca se incrementa más allá del “1”. Esto se debe a que el valor en un *bit* concreto sólo puede ser en un momento dado “1” o “0”. Por lo tanto, aumentar más allá del “1” debe significar que se almacena algo de ruido en ese *bit*, lo que resulta indeseable. Este ruido es generado por dos direcciones de escritura que activen una línea de palabra común. En el caso presente, el algoritmo *k*-WTA puede identificar correctamente qué *bits* deberían ponerse a “1” sin la necesidad de contadores que puedan incrementarse más allá del “1”. La implementación de Kanerva de contadores ascendentes y descendentes, y de un umbral cero ya no es necesaria para determinar si un *bit* concreto de salida debería ponerse a “0” o “1”. Por ello, la presente invención proporciona una memoria en la que los datos se almacenan en memorias convencionales de un solo *bit* y no en contadores. Esto da como resultado un ahorro importante en el coste de implementación. Además, desarrollos recientes han dado como resultado que estén disponibles a bajo precio y eficientemente grandes cantidades de memoria adecuada (RAM).

Una medida común de la eficacia de una memoria es su capacidad de almacenamiento en función del número de *bits* de información que pueden almacenarse con utilidad para cada *bit* de la memoria. Usando esta medida, un sistema que sea un ejemplo de realización de la presente invención confiere ventajas considerables con respecto al propuesto por Kanerva. La razón principal de esta mayor capacidad de almacenamiento surge de escribir datos a un número menor de líneas de palabra. El inventor ha descubierto que, típicamente, mientras que Kanerva escribía datos a unas 1000 líneas de palabra, pueden escribirse datos a un número mucho menor de líneas de palabra a la vez que se conservan las ventajas de la memoria de Kanerva, y obteniendo mayor eficiencia de almacenamiento. Por ejemplo, para las características de memoria descritas más abajo, se prefiere que se escriban datos a menos de 100 líneas de palabra, se prefiere más que se escriban datos a menos de 50 líneas de palabra, pero lo más ventajoso a nivel operativo es que se escriban datos a entre 11 y 20 líneas de palabra. A continuación se presentará un análisis de la capacidad de almacenaje, junto con las ecuaciones para determinar el número óptimo de líneas de palabra, *w*, a las que deberían escribirse datos.

La memoria de datos 3 es una forma de memoria de matriz de correlación. Las matrices de correlación se han estudiando extensamente en, por ejemplo, Kohonen, T. “Correlation Matrix Memories” [Memorias de matriz de correlación], IEEE Transactions Computers [Actas IEEE sobre ordenadores] C-21(4), abril de 1972, páginas 353-359, y Turner, M. y Austin, J. “Matching Performance of Binary Correlation Matrix Memories” [Rendimiento de adaptación de las memorias binarias de matriz de correlación], Neural Networks [Redes neuronales] 10(9), 1997, páginas 1637-1648. Se presenta aquí un bosquejo de este análisis como base para la derivación de *w*.

Se da por sentado que las entradas de direcciones y de datos están distribuidas de manera uniforme por sus espacios respectivos. La memoria de datos 3 contiene inicialmente ceros en cada *bit*. Se emplea la medida de ocupación *h*, donde *h* representa la probabilidad de que un *bit* arbitrario de la memoria de datos se ponga a “1” en algún punto del proceso de escritura. Después de *Z* operaciones de escritura, la ocupación esperada viene dada por la ecuación (2):

$$h = 1 - \left[1 - \left(\frac{w}{W} \cdot \frac{d}{D} \right) \right]^Z \quad (2)$$

El término $\left(\frac{w}{W} \cdot \frac{d}{D} \right)$ de la ecuación (2) da la probabilidad de poner a “1” un *bit* concreto dentro de la memoria de datos en una sola operación de escritura. Restar este término de 1 da un valor para la probabilidad de no poner ese *bit* a “1” en una operación concreta. Esta resta se eleva luego a la potencia de *Z* para que represente el efecto de *Z* operaciones de escritura. Así el término $\left[1 - \left(\frac{w}{W} \cdot \frac{d}{D} \right) \right]^Z$ proporciona una probabilidad de no poner un *bit* arbitrario a “1” después de *Z* operaciones de escritura. Por lo tanto, para derivar *h*, la probabilidad de poner un *bit* arbitrario a “1” después de *Z* operaciones, este término se resta de 1, lo que da la ecuación (2).

Puede verse de la ecuación (2) que *h* es inicialmente 0, puesto que *Z* es 0, y la parte derecha de la ecuación se simplifica a 1 - 1. *h* se incrementa de forma monótona hacia 1 según se van almacenando más datos en la memoria.

La ecuación (2) puede volver a escribirse usando logaritmos, de modo que da:

$$\ln(1 - h) = Z \ln \left[1 - \left(\frac{w}{W} \cdot \frac{d}{D} \right) \right] \quad (3)$$

El término $\left(\frac{w}{W} \cdot \frac{d}{D} \right)$ es pequeño, dado que típicamente *d* es mucho menor que *D*, y que *w* es típicamente mucho menor que *W*. Por lo tanto, el logaritmo natural de la parte derecha de la ecuación puede aproximarse al primer término

ES 2 268 084 T3

de la serie de Taylor. Esto da una expresión para el número de elementos de datos almacenados, como se presenta en la ecuación (4).

$$Z = -\ln(1 - h) \cdot \frac{W}{w} \cdot \frac{D}{d} \quad (4)$$

La información total almacenada en la memoria después de Z operaciones de escritura puede identificarse considerando el número de símbolos que pueden representarse mediante un código “ d de D ”. Este contenido de información viene dado en la ecuación (5).

$$I(Z) = Z \cdot \log_2[C_D^d] \text{ bits} \quad (5)$$

Podemos computar la eficiencia de almacenamiento comparando la capacidad de información dada en la ecuación (5) con el número de ubicaciones binarias de almacenamiento en la memoria de datos para dar la expresión de la ecuación (6).

$$\eta(Z) = \frac{I(Z)}{D \cdot W} = \frac{Z}{D \cdot W} \cdot \log_2[C_D^d] \quad (6)$$

donde $\eta(Z)$ es la eficiencia de almacenamiento de la memoria.

Para un valor grande de D , el término combinatorio de la ecuación (6) puede aproximarse con:

$$\log_2[C_D^d] = -D \cdot \left[\frac{d}{D} \cdot \log_2 \frac{d}{D} + \left(1 - \frac{d}{D} \right) \cdot \log_2 \left(1 - \frac{d}{D} \right) \right] \quad (7)$$

Para una memoria dispersa donde d es mucho menor que D , reemplazar las ecuaciones (7) y (4) en la ecuación (6) da, tras simplificar:

$$\eta(Z) = -\ln(1 - h) \cdot \frac{1}{w} \cdot \left[\log_2 \frac{D}{d} + 1 \right] \quad (8)$$

donde $\eta(Z)$ es la eficiencia de almacenamiento de la memoria.

Esto muestra que la memoria más eficiente, para una ocupación h dada, y bajo cero condiciones de error es aquella en la que los datos se escriben como código uno de D (conocida también como código de “uno caliente”), y en la que el menor número de decodificadores de dirección se dispare para una dirección de entrada dada. O sea, $d = 1$ y $w = 1$. Sin embargo, poner d y w de acuerdo con estos criterios afectará adversamente a la robustez de la memoria bajo una capacidad elevada o errores de entrada, como se describe más abajo.

Para considerar la robustez de la memoria de datos, se considera la recuperación de una sola palabra de datos siguiendo la escritura de Z palabras de datos, donde la dirección de recuperación se presenta sin error. Los w decodificadores de dirección activados por la dirección de lectura serán los mismos w decodificadores de dirección activados durante el proceso de escritura de datos. Si el dato escrito a un *bit* concreto fue un “1”, entonces cada una de las w líneas de palabra debería almacenar un “1” en ese *bit* para dar un nivel de activación en conformidad con la ecuación (9).

$$E(X_1) = w \quad (9)$$

donde $E(X)$ es el nivel de activación.

Sin embargo, si el valor del dato escrito a un *bit* concreto fue un “0”, el nivel de activación esperado será:

$$E(X_0) = w \cdot h \quad (10)$$

donde h se define igual que antes.

Puesto que h es inferior a 1, se puede esperar que la memoria recupere los datos escritos, puesto que los *bits* puestos a “1” tendrán un nivel de activación más elevado que los puestos a “0”, como puede verse en las ecuaciones (9) y (10). Sin embargo, el nivel de activación para un *bit* puesto a “0” está sujeto a una variación estadística, y con

ES 2 268 084 T3

alguna probabilidad (pequeña), todos los pesos habrán sido puestos a “1” por otras escrituras de datos. En este caso, el nivel de activación de una salida que debería ser “0” será la misma que la de una salida que debería ser “1”. Esto da una posibilidad de una condición de error de “falso 1”. En ausencia de errores de escritura, puede verse que no hay posibilidad alguna de errores de “falso 0”.

Un error de “falso 1” sucede únicamente si cada *bit* que contribuye a una salida de “0” está puesto incorrectamente a “1”. Así, la probabilidad de que un *bit* “0” se lea incorrectamente viene dada por la ecuación (11).

$$p(x_0 < w) = [1 - h^w] \quad (11)$$

O sea, h^w da un valor para la probabilidad de que los w *bits* contribuyan a que el nivel de activación se haya puesto incorrectamente a “1”. Por eso, restar este valor de 1 da la probabilidad de que se lea correctamente un valor “0”, que es la probabilidad de que w *bits* no se hayan puesto incorrectamente a “1”.

Una palabra de datos es leída correctamente si cada uno de sus $D - d$ *bits* puestos a 0 se lee correctamente. La probabilidad de leer correctamente una palabra viene dada en la ecuación (12):

$$p(\text{palabra de datos correcta}) = [1 - h^w]^{D-d} \quad (12)$$

O sea, la expresión de la ecuación (11) elevada a la potencia de $D - d$, dado que han de recuperarse un total de $D - d$ *bits*.

Si se han escrito Z palabras de datos a la memoria, la probabilidad de que cada palabra de datos pueda leerse correctamente viene dada por la ecuación (13):

$$P_c = [1 - h^w]^{Z \cdot (D-d)} \quad (13)$$

Según va aumentando h , la probabilidad de recuperación sin error disminuye hacia cero.

Reescribir la ecuación (13) usando logaritmos da:

$$\ln(P_c) = Z \cdot (D - d) \cdot \ln(1 - h^w) \quad (14)$$

Puesto que h^w es muy pequeño, donde w es considerablemente mayor que 1, el primer término de la serie de Taylor da una aproximación cercana para el logaritmo de la parte derecha de la ecuación (14), de modo que da:

$$\ln(P_c) = -Z \cdot (D - d) \cdot h^w \quad (15)$$

Para configurar de manera óptima la memoria de datos, se considera que el número de decodificadores de dirección W y la longitud de palabra D son fijos.

También se considera que la codificación de datos (“ d de D ”) es fija. Para maximizar la probabilidad de recuperación correcta de datos, puede verse en la ecuación (13) que h^w o que, equivalentemente, $(w \ln h)$ deberían minimizarse. Por lo tanto, buscamos una solución donde:

$$\frac{d}{dw}(w \ln h) = \ln h + \frac{w}{h} \cdot \frac{dh}{dw} = 0 \quad (16)$$

Recordando la ecuación (4):

$$Z = -\ln(1 - h) \cdot \frac{W}{w} \cdot \frac{D}{d} \quad (4)$$

que puede reescribirse como:

$$w \cdot \frac{Z \cdot d}{W \cdot D} = -\ln(1 - h) \quad (17)$$

ES 2 268 084 T3

Diferenciando la ecuación (17) con respecto a w y sustituyendo un valor de Z obtenido de la ecuación (17) da:

$$\frac{dh}{dw} = -\frac{(1-h) \cdot \ln(1-h)}{w} \quad (18)$$

Sustituyendo el valor de $\frac{dh}{dw}$ de la ecuación (18) en la ecuación (16) da la siguiente fórmula para h :

$$\ln h + \frac{w}{h} \cdot \left[-\frac{(1-h) \cdot \ln(1-h)}{w} \right] = 0 \quad (19)$$

Simplificando, da:

$$\ln h + \frac{1}{h} \cdot [-(1-h) \cdot \ln(1-h)] = 0 \quad (20)$$

Multiplicando ambos lados de la ecuación (20) por h :

$$h \ln h - (1-h) \cdot \ln(1-h) = 0 \quad (21)$$

La ecuación (21) tiene la solución $h = 0,5$.

Se ha mostrado previamente que el máximo de contenido de información de una memoria de matriz de correlación se da cuando $h = 0,5$ (véase Nadal, J. P., y Toulouse, G., "Information Storage in Sparsely Coded Memory Nets" [Almacenamiento de información en redes de memoria codificada de forma dispersa], *Network 1* (1990), páginas 61 a 74). La optimización de h presentada más arriba no estaba interesada en cuánta información puede almacenar la memoria, sino en cuánta puede registrarse fiablemente. El contenido máximo recuperable coincide, por lo tanto, con el contenido máximo.

Sustituyendo un valor de Z de la ecuación (4) en la ecuación (15) da:

$$h^{-w} \cdot w = \frac{\ln(1-h)}{\ln(P_c)} \cdot W \cdot \frac{D}{d} \cdot (D-d) \quad (22)$$

Usando el valor óptimo de h deducido más arriba, o sea, $h = 0,5$, y dado que W , d y D son conocidos, y usando una probabilidad aceptable de recuperación sin error P_c , puede computarse un valor de w .

Sin embargo, en algunas circunstancias, una recuperación totalmente libre de errores (como se ha dado por sentada hasta el momento) resulta un requerimiento demasiado restrictivo, dada la naturaleza de la memoria. Una definición alternativa para la capacidad de información de la memoria es establecer el número máximo de elementos de datos que pueden recuperarse sin error, permitiendo que algunos otros elementos puedan recuperarse con errores.

Recordando la ecuación (12):

$$p(\text{palabra de datos correcta}) = [1 - h^w]^{D-d} \quad (12)$$

es posible derivar una expresión para el número de elementos de datos que pueden ser recuperados con exactitud, mientras que otros se recuperarán con errores E_c . Esta expresión es dada en la ecuación (23):

$$E_c = Z \cdot [1 - h^w]^{D-d} \quad (23)$$

donde Z es el número de palabras contenidas en la memoria.

ES 2 268 084 T3

Para un valor dado de Z , w varía para maximizar E_c , y su valor máximo se da cuando $h = 0,5$.

Si se escriben datos a memoria hasta que cada nueva entrada cause uno o más errores adicionales de lectura, se ha alcanzado la capacidad máxima útil de la memoria. El punto se alcanza cuando:

$$\frac{dE_c}{dZ} = 0 \quad (24)$$

Reescribiendo la ecuación (23) como:

$$\ln E_c = (D - d) \ln Z \cdot [1 - h^w] \quad (25)$$

Diferenciando la ecuación (25) da:

$$\frac{1}{E_c} \cdot \frac{dE_c}{dZ} = \frac{1}{Z} - \frac{(D - d)}{(1 - h^w)} \cdot \ln h \cdot h^w \cdot \frac{dw}{dZ} = 0 \quad (26)$$

Reescribiendo la ecuación (4):

$$wZ = -\ln(1 - h) \cdot \frac{WD}{d} \quad (27)$$

Puede verse en la ecuación (27) que para una h constante, wZ es constante, dado que W , D y d son fijos. Por lo tanto:

$$\frac{dw}{dZ} = -\frac{w}{Z} \quad (28)$$

Sustituyendo la ecuación (28) en la ecuación (26) da:

$$\frac{(h^{-w} - 1)}{w} = -(D - d) \cdot \ln h \quad (29)$$

La ecuación (29) da un valor alternativo para el w óptimo, cuando el objetivo es recuperar correctamente tantas palabras como resulte posible, permitiendo la recuperación de algunas palabras con errores. El valor de w que puede obtenerse de la ecuación (29) es menor que el obtenido de la ecuación (22). Esto refleja el hecho de que activar un número mayor de decodificadores de dirección (o sea, aumentando w), da como resultado una memoria que tiene menos errores pero menor capacidad. A la inversa, un valor menor de w da como resultado una memoria que produce más errores pero que tiene una capacidad mayor, ya que cada operación de escritura de datos pone valores en menos líneas de palabra de la memoria de datos. En ambos casos, el máximo de información extraíble se produce cuando $h = 0,5$.

El inventor, habiendo deducido las expresiones presentadas más arriba, ha determinado que una memoria que sea un ejemplo de realización de la presente invención puede ser muy mejorada si w se pone de modo que esté entre los valores determinados por las ecuaciones (22) y (29).

A continuación se describirá el análisis descrito con anterioridad haciendo referencia a las figuras 3 y 4. La figura 3 muestra un trazado tridimensional de la ecuación (23). Un eje etiquetado como Z representa el número de elementos escritos a la memoria; un eje etiquetado como eje w representa el número de decodificadores de dirección activados por una operación de lectura o escritura; y un eje etiquetado como E_c representa la cantidad dada por la ecuación (23). La figura 4 es un trazado de contorno del trazado de la figura 3, donde los ejes w y Z representan las mismas cantidades que los ejes respectivos de la figura 3. El sistema ejemplar ilustrado en las gráficas es uno en el que hay 4096 decodificadores de dirección (o sea, $W = 4096$), y los datos se almacenan usando un código 11 de 256 (o sea, $d = 11$ y $D = 256$). Otras configuraciones muestran resultados similares.

Refiriéndonos primero a la figura 3, puede verse que el número de palabras E_c que pueden recuperarse correctamente aumenta según se van escribiendo más datos a memoria (o sea, según aumenta Z), pero desciende bruscamente cuando Z se hace demasiado grande (una formación 10 en forma de acantilado). O sea, tras un número de escrituras de datos, la memoria se satura y el número de palabras que puede recuperarse correctamente baja señaladamente.

A continuación se presentará un análisis adicional haciendo referencia a la figura 4. Las líneas continuas de contorno muestran que el número esperado de palabras correctas (E_c) varía con el número de entradas almacenadas (Z)

y con el número de decodificadores de dirección activados (w). El valor pico de E_c es 5332, con una tasa de error de aproximadamente el 11%. Esta capacidad pico es mayor que el número de líneas de palabra ($W = 4096$). La línea central de contorno 11 se dibuja por $E_c = 5000$, y se dibujan contornos subsiguientes en incrementos de 500. La cara sur del trazado es casi plana y E_c está muy cerca de Z , salvo en la cara oeste del trazado, donde w es pequeño. El rendimiento descende rápidamente si w es demasiado pequeño, o bien si Z es demasiado grande, como se ve por las pendientes acusadas de las caras oeste y norte del trazado, respectivamente.

Sobrepuesto en los contornos continuos descritos más arriba, hay otros dos juegos de datos. Un primer juego de datos contiene tres hipérbolas 12 dibujadas en líneas hechas con trazos largos. Cada hipérbola representa los valores de w y Z que generan un valor dado para h . La primera de estas hipérbolas A representa puntos en los que $h = 0,2$; una segunda hipérbola B representa $h = 0,5$; y una tercera hipérbola C representa $h = 0,8$. Puede verse que la línea que denota ocupación óptima ($h = 0,5$) pasa por el pico del trazado.

Tres curvas adicionales 13 muestran que los valores de w y Z afectan al valor de P_c . Las tres curvas, leyendo de izquierda a derecha representan valores de P_c de 0,1, 0,5 y 0,9, respectivamente. Puede verse en la figura 4 que el pico de cada una de las curvas 13 está en la línea $h = 0,5$. Debería notarse que las tres curvas 13 están cerca una de otra, indicando que la capacidad es relativamente insensible a la probabilidad P_c .

Dada la presentación precedente, puede verse que el valor óptimo de w para la máxima recuperación correcta de datos, recuperándose otros datos con error, se da en el pico del trazado. O sea, $w = 11$. El valor óptimo para la recuperación sin error se lee desde el punto en el que la hipérbola central B cruza la curva central de las curvas 13. (Ese es el punto en el que la línea $h = 0,5$ cruza la línea $P_c = 0,5$.) En este caso, $w = 20$, que está por encima del doble del valor para la recuperación con errores. Para determinar el valor de w para la recuperación sin errores es necesario ubicar el máximo de la curva P_c . Sin embargo, como esta curva es muy plana, detectar el máximo es muy difícil. Como se sabe que este máximo se da en un punto en el que $h = 0,5$, es posible tomar el punto de intersección de la línea $P_c = 0,5$ y $h = 0,5$ para determinar el valor de w para el máximo de la recuperación de datos libre de error. Por lo tanto, de acuerdo con los desarrollos hechos por la invención, cualquier memoria dotada de los parámetros expuestos más arriba funciona óptimamente con un valor de w entre 11 y 20.

La exposición anterior ha propuesto una forma novedosa para determinar un valor óptimo de w , y se han presentado las ventajas de tal determinación. Sin embargo, debería notarse que, dado que cada decodificador de dirección se dispara independientemente, es difícil garantizar que estén activadas exactamente w direcciones por parte de cada dirección de entrada, o sea, habrá cierta variación en w . El análisis presentado más arriba se modifica ahora a la luz de esta observación.

Se ha explicado que se aplica un umbral T a un decodificador de dirección para determinar el número de *bits* que necesitan corresponderse entre una dirección de entrada y un identificador de decodificación si ha de activarse un decodificador. Los decodificadores de dirección activados son aquellos que tienen al menos T *bits* de su entrada correspondiéndose con la dirección de entrada.

La probabilidad de que un primer código N_1 de M tenga T unos en común con un segundo código N_2 de M es dada por la ecuación (30):

$$p(t) = \frac{C_{N_1}^T \cdot C_{M-N_1}^{N_2-T}}{C_M^{N_2}} \quad (30)$$

Puesto que están activados los decodificadores de dirección que tienen en común T *bits* con la dirección de entrada, el valor medio de w , denotado por $\hat{w}$, puede deducirse usando la ecuación (31):

$$\hat{w} = \frac{W}{C_A^0} \cdot \sum_{k=T}^n C_i^k \cdot C_{A-i}^{a-k} \quad (31)$$

Si la memoria está funcionando con entrada libre de errores, siempre estarán activados los mismos decodificadores de dirección por parte de una dirección de entrada dada. Sin embargo, habrá cierta variación estadística en el número de decodificadores que son activados por diferentes direcciones de entrada. Para una dirección de entrada dada, la probabilidad de que cualquier decodificador de dirección se active es dada por:

$$p_a = \frac{\hat{w}}{W} \quad (32)$$

ES 2 268 084 T3

Los W decodificadores de dirección dan una distribución binómica para w , con una media en conformidad con la ecuación (31), y una varianza en conformidad con la ecuación (33):

$$\sigma_a^2 = W \cdot pa \cdot (1 - pa) \quad (33)$$

La probabilidad de que w tome un valor particular w' es:

$$P_{w'} = (1 - pa)^{W-w'} \cdot pa^{w'} \cdot C_W^{w'} \quad (34)$$

La ecuación (13) de más arriba da una expresión de la probabilidad de que todos los valores de datos almacenados puedan recuperarse sin error. Dada la ecuación (34), la ecuación (13) puede modificarse, de modo que dé:

$$P_c = \prod_{w'} [1 - h^{w'}]^{Z \cdot P_{w'} \cdot (D-d)} \quad (35)$$

Es decir, la ecuación (13) es modificada para tomar en cuenta la varianza en el valor de w .

De modo similar, la ecuación (23), que da un valor para el número esperado de elementos de datos recuperables correctamente (ignorando los recuperados con errores), se convierte en:

$$E_c = \sum_w P_w \cdot Z \cdot [1 - h^w]^{(D-d)} \quad (36)$$

Dados estos cambios a los valores de E_c y P_c para tener en cuenta la distribución binómica de w , se modifica el trazado de contorno de la figura 4 al mostrado en la figura 5. Las líneas de contorno, hipérbolas y curvas denotan la misma información que las de la figura 4. El eje vertical (Z) de la figura 5 representa la misma cantidad que el eje homólogo de la figura 4. El horizontal w se ajusta para que refleje el valor medio de w .

En este caso, el valor pico de E_c es más bajo en $E_c = 4445$, donde $Z = 5440$ y $w = 15$. Este valor pico ya no ocurre en $h = 0,5$, sino que ahora se da en $h = 0,575$. El rendimiento libre de errores es considerablemente peor con valores más bajos de w , pero la forma general del trazado no se ve afectada. Sin embargo, puede verse que, dada la distribución binómica de w , el valor de T debe ajustarse para que dé a w una media más elevada que las propuestas anteriormente. El valor requerido de T puede determinarse reordenando la ecuación (31) y utilizando valores adecuados para a , A , i y w para determinar T .

El sistema de la presente invención puede implementarse empleando tecnología de redes neuronales artificiales, y a continuación se describe tal implementación. La figura 6 muestra la estructura básica de una red neuronal artificial adecuada. La red neuronal consta de una capa de neuronas de entrada 14 conectadas a una capa de neuronas de decodificación de dirección 15. La memoria de datos consta de D neuronas 16. Cada neurona de datos 16 va conectada a cada neurona de decodificación de dirección 15. Cada neurona de datos 16 tiene una entrada (*input*, IP) de escritura de datos y una salida (*output*, OP) de lectura de datos.

El sistema resulta particularmente adecuado para una implementación basada en neuronas de integración y disparo de picos transitorios con pérdida, aunque también son aplicables otros modelos neuronales. Una neurona de integración y disparo tiene n entradas, y toma una suma ponderada de estas entradas. Esta suma se denomina nivel de activación de la neurona. Una neurona se dispara solamente si el nivel de activación excede un umbral predeterminado. La implementación descrita más abajo se basa en neuronas de picos transitorios, que dan salida a un solo pulso - o sea, una neurona o bien se dispara y genera un pulso, o no se dispara y no genera un pulso. Así, la salida de una neurona puede considerarse que es una salida digital 0/1. La propiedad de las neuronas de tener pérdida significa que una entrada es mantenida por una neurona durante un periodo de tiempo predefinido. Después de este tiempo, el nivel de activación desciende hacia cero. Así, si las entradas se disparan en oleadas solamente se tendrán en cuenta las entradas de una oleada dada en cualquier cálculo del nivel de activación, ya que otras entradas habrán vuelto a cero por causa de la pérdida. Esta propiedad de la presencia de pérdidas es importante, ya que garantiza que una neurona no se dispare incorrectamente debido a los efectos de las entradas previas. La propiedad de tener pérdida garantiza que las neuronas dentro del sistema se asienten en estados inactivos entre "oleadas" de actividad.

Se introduce una dirección con la forma de un código i de A en las neuronas de decodificación de dirección 15 desde las neuronas de entrada 14. Esta entrada es una serie de i picos transitorios en i de las A neuronas de entrada al sistema que llegan aproximadamente a la vez. Estas neuronas de picos transitorios contribuyen al nivel de activación de algunas de las neuronas de decodificación de dirección, como se describe más abajo.

El decodificador de dirección consta de W neuronas de decodificación de dirección 15. Las conexiones entre las neuronas de entrada 14 y las neuronas asociadas con los decodificadores de dirección son fijas. De modo similar, el umbral en el que se disparan las neuronas de decodificación de dirección también es fijo. Por ejemplo, en la ilustración simplificada de la figura 6, puede ser que deba dispararse al menos una de las dos neuronas de entrada 14 conectadas a

una neurona de decodificación concreta de dirección 15 si esa neurona de decodificación de dirección ha de dispararse. Es decir, cada neurona de decodificación de dirección tiene un umbral de 1.

Puede formarse una matriz que represente las conexiones entre las neuronas de entrada 14 y las neuronas de decodificación de dirección 15, y esta matriz de conexiones es completa - es decir, cada neurona de entrada se conecta con cada neurona de decodificación de dirección. Cada neurona de decodificación de dirección tiene A pesos: a pesos están puestos a "1" en la matriz de conexiones y $A - a$ pesos están puestos a "0". En la figura 7 se muestra parte de una matriz de conexiones. Aquí se muestran conexiones para una neurona de decodificación de dirección. Cada una de cinco líneas de entrada 17 está conectada con una senda línea de entrada 18 de una única neurona de decodificación de dirección (no mostrada). Se asigna a cada una de las conexiones entre una línea de entrada 17 y una línea de entrada de decodificación de dirección 18 o bien un peso de "1" o bien un peso de "0".

Leyendo las líneas de entrada 17 de izquierda a derecha y las entradas de las neuronas de decodificación de dirección de arriba abajo puede verse que se asigna a la conexión entre la primera línea de entrada y la primera entrada de las neuronas de decodificación de dirección un "0", mientras que a la conexión entre la segunda línea de entrada y la segunda entrada de las neuronas se le asigna un "1".

Hay cinco líneas de entrada 17, y cinco entradas de decodificación de dirección 18. A las conexiones entre tres de estas líneas se les asigna un peso de "1". Así, a cada neurona de decodificación de dirección se le asigna en realidad un código 3 de 5 como identificador. Para cada una de las conexiones dotadas de un peso de "1", una neurona de entrada que se dispare por la línea de entrada asociada contribuye al nivel de activación de la neurona de decodificación de dirección. A la inversa, para una conexión dotada de un peso de "0", una neurona de entrada que se dispare por la línea de entrada asociada no contribuye al nivel de activación de la neurona de decodificación de dirección.

La estructura mostrada en la figura 7 es repetida de tal modo que cada línea de entrada 17 esté conectada a W entradas de neuronas de decodificación de dirección. A la vista de la figura 7, únicamente se muestran las entradas de la neurona de decodificación de una sola dirección, de modo que cada línea de entrada 17 esté conectada solamente a una única entrada 18.

Dado que un peso de "1" significa que una neurona de entrada que se dispare afectará a una entrada de neurona de decodificación de dirección 18 (o sea, esa línea de entrada representa un "1" en el identificador del decodificador de dirección), y que un "0" es efectivamente la representación de ausencia de conexión (o sea, una neurona que se dispare por esta línea de entrada no tendrá efecto alguno en la neurona de decodificación de dirección), cada neurona de decodificación de dirección sólo precisa estar conectada a las i neuronas de entrada que tengan unos. Esto se representa en la figura 8, donde solamente las tres líneas de entrada 17 que tienen pesos de "1" están conectadas con las entradas 18 de una neurona de decodificación de dirección 15.

De este modo, las conexiones entre las neuronas de entrada y las de decodificación de dirección pueden verse bien como una matriz de conexiones completa que conste de valores "0" y "1" (figura 7), o bien como una tabla de encaminamiento que contiene conexiones específicas (figura 8). Estas dos vistas son equivalentes. Debería notarse que, como en la vista de la figura 7, en la figura 8 únicamente se muestran las conexiones para una sola neurona de decodificación de dirección 15, y harán falta conexiones similares para cada una de las W neuronas de decodificación de dirección.

En general, las conexiones (o "unos") para cada neurona de decodificación de dirección se seleccionan de forma aleatoria.

Cuando llegan los picos transitorios (que representan una dirección de entrada), se disparan un número de neuronas de decodificación de dirección. Las neuronas que se disparan serán aquellas que hayan recibido valores de "1" en un número suficiente de líneas de entrada con peso de "1", de modo que se eleve el nivel de activación de la neurona de decodificación de dirección por encima del umbral T . Este disparo de las neuronas de decodificación de dirección 15 es un código w de W , como se ha descrito más arriba.

La memoria de datos 3 consta de D neuronas 16, cada una de las cuales tiene W neuronas de decodificación de dirección 15 como entrada. Cada una de las D neuronas de memoria de datos tiene una única entrada IP de datos de escritura. Las conexiones entre las W neuronas de decodificación de dirección y las D neuronas de memoria de datos tienen inicialmente pesos de "0". Las conexiones desde las entradas de datos de escritura (no mostradas) tienen valores que son inicialmente irrelevantes. Sin embargo, cuando se dispara una neurona de datos de escritura, debe forzarse a una senda neurona de memoria de datos para que entre en un modo hebbiano de aprendizaje que persiste durante la oleada incidente asociada de neuronas de decodificación de dirección que se disparan. Es decir, cuando han de escribirse datos, cada una de las neuronas de la memoria de datos debe ser activada para que pueda almacenar los datos presentados por las neuronas de datos de escritura.

A continuación se describe el procedimiento para escribir datos a la memoria de datos. En primer lugar, se disparan d de las D neuronas de datos de escritura (no mostradas), y las correspondientes d neuronas de memoria de datos 16 son puestas en un modo hebbiano de aprendizaje. Las neuronas de decodificación de dirección 15 se disparan, como se ha descrito más arriba, lo que da por resultado que se comunique un código w de W a las neuronas de memoria de datos 16. Las w neuronas que reciben picos transitorios procedentes de las neuronas de decodificación de dirección,

y que han sido puestas en un modo hebbiano de aprendizaje por las neuronas de entrada de datos de escritura, tienen pesos puestos a “1”.

Una vez que ha pasado la oleada de picos transitorios incidentes de neuronas de decodificación de dirección, la propiedad de todas las neuronas de tener pérdidas hace que los niveles de activación vuelvan a “0”. En este punto, las neuronas de la memoria de datos dejan de estar en un modo hebbiano de aprendizaje. Después de una demora adecuada, la memoria está en un estado apropiado para que se inicie un ciclo adicional de lectura o escritura. Por lo tanto, puede verse que la propiedad de las neuronas de tener pérdidas garantiza que la activación de un ciclo de escritura previo no afecte la operación del ciclo de escritura actual.

Cuando han de leerse datos de la memoria de datos, las neuronas de datos de escritura están inactivas. Una operación de lectura es iniciada mediante el disparo de w de W neuronas de decodificación de dirección, como se ha descrito con anterioridad. Las w neuronas disparadas de decodificación de dirección hacen que se eleven los niveles de activación en algunas neuronas de memoria de datos, dando por sentado que al menos algunas neuronas han sido modificadas por un ciclo previo de escritura.

Una lógica generalizada de salida del estilo de “el ganador se queda con todo” (k -WTA) selecciona d neuronas de memoria de datos con el nivel de activación más alto y éstas emiten picos transitorios por sus salidas respectivas OP para formar una representación de la salida de datos. Como se ha descrito con anterioridad, se requiere una demora adecuada antes de que se inicie un ciclo adicional de lectura o escritura. Pueden emplearse varios métodos conocidos para implementar el algoritmo k -WTA. En uno de tales métodos, las neuronas de salida se disparan en orden decreciente de nivel de activación, disparándose en primer lugar la neurona con el nivel de activación más elevado. Así, después de que hayan disparado d neuronas, se sabe que las d neuronas con los niveles de activación más elevados han contribuido a la salida. Puede inhibirse el disparo adicional mediante una neurona contadora que haya observado los d ciclos de disparo y que emita una señal inhibidora para evitar disparos adicionales.

La aplicación de codificación N de M a la lógica de decodificación de dirección, como se ha descrito más arriba, supone un número de beneficios. En concreto, dado que dichos códigos están interesados únicamente en la posición de los *bits* a “1” dentro de un valor de datos, resultan particularmente adecuados para aplicaciones de redes neuronales, donde un “1” queda representado por una neurona que se dispara. En un sistema binario resulta comparativamente difícil representar y detectar una neurona que no se dispara, haciendo difícil de llevar a cabo una implementación neuronal de un sistema binario. Por eso, la aplicación de codificación N de M proporciona un sistema efectivo controlado por eventos, en el que los eventos (una neurona se dispara) tienen efectos y los eventos inexistentes (una neurona no se dispara) no tienen efecto.

La selección concreta de los valores M y N para un código M de N dependerá de la aplicación, hasta cierto punto. Por ejemplo, en aplicaciones bioinformáticas, donde han de representarse datos de ADN, suele escogerse un código 1 de 4. Se hace esta elección porque el ADN consta de un número de bases, seleccionándose cada base de entre un conjunto de cuatro bases posibles, comúnmente denominadas G, A, T y C. Así, cada base está representada por un “1” en una de las cuatro entradas proporcionadas por el código. O sea, por ejemplo:

G = 0001 A = 0010

T = 0100 C = 1000

Es decir, un código 1 de 4 permite que se representen cuatro valores diferentes (un número binario de dos *bits*). Por lo tanto, una secuencia de ADN es representada como una serie de códigos 1 de 4, un código para cada base de la secuencia de ADN. Usando esta secuencia, una secuencia de ADN de cuatro bases estará representada por un código 4 de 16. Debería notarse que éste es un código restringido 4 de 16, con una restricción tal que uno de los *bits* “1” debe darse dentro de los primeros cuatro *bits* del código, el segundo en los siguientes cuatro *bits* y así sucesivamente.

Un código binario general puede transformarse en un código N de M, usando, por ejemplo, codificación de carril dual. Aquí, cada *bit* del código binario es asignado a dos “carriles”. Se transmite un pulso al primer carril de un *bit* si ese *bit* está puesto a “1”, mientras que un pulso se transmite al segundo carril de ese *bit* si el *bit* está puesto a “0”. Relacionando esta implementación de carril dual con la red neuronal ilustrada en la figura 6, cada neurona de entrada 14 acepta la entrada procedente de dos neuronas de entrada de carril dual. Es decir, hay diez neuronas de entrada de carril dual en el sistema. Cuando se recibe la entrada, ésta estará en la forma de un código 5 de 10, con una restricción tal como que un “1” deba ocurrir en una del primer par de neuronas, una en el segundo y así sucesivamente. Acto seguido, se disparan las neuronas de entrada que reciben entradas procedentes de las neuronas del carril de “1”, como se ha descrito anteriormente.

Debería notarse que añadir tecnología de carril dual de esta manera proporciona un sistema en el que las entradas de carril dual funcionan usando, como se ha descrito, un código restringido N de M, mientras que las neuronas de entrada funcionan en binario, sin certidumbre alguna en cuanto a cuántas neuronas se dispararán. Esto facilita una conversión efectiva entre un sistema N de M y un sistema binario.

ES 2 268 084 T3

Les resultará evidente a las personas versadas en la especialidad que la configuración de memoria de la presente invención puede ser implementada en términos de los decodificadores tradicionales de dirección, y de las líneas tradicionales de palabra de almacenamiento de datos o como un sistema neuronal, tal como se ha descrito más arriba. También resultará evidente que el sistema puede ser implementado bien fabricando componentes necesarios de *hardware*, o bien configurando componentes reconfigurables, tales como los sistemas de puertas programables por campos (*Field Programmable Gate Arrays*, FPGAs). Alternativamente, el sistema puede realizarse escribiendo un programa de ordenador adecuado que corra en un sistema informático convencional. Tal sistema usará el *hardware* del ordenador convencional de tal manera que simule una memoria que sea un ejemplo de realización de la invención. El programa informático puede escribirse en cualquiera del gran número de lenguajes de programación informática ampliamente utilizados, aunque se prefiera en particular una implementación orientada a objetos.

15

20

25

30

35

40

45

50

55

60

65

REIVINDICACIONES

1. Una configuración de memoria para su empleo en un sistema informático, constando la memoria de una pluralidad de decodificadores de dirección (2), a cada uno de los cuales se le asigna un identificador que tiene un número predefinido de *bits*, teniendo cada *bit* estados seleccionables primero y segundo, y una memoria de datos (3) que tiene una pluralidad de líneas de palabra de longitud predefinida, siendo susceptible de activación cada uno de los referidos decodificadores de dirección (2) para seleccionar una de entre una pluralidad de líneas, y constando los decodificadores de dirección (2) de medios para recibir una dirección de entrada que tiene un número predefinido de *bits* y de medios para comparar el identificador de un decodificador de dirección (2) con la dirección de entrada, **caracterizada** porque la memoria consta además de medios para activar un decodificador de dirección (2) si al menos un número mínimo predefinido de *bits* puestos en el primer estado seleccionable en la dirección de entrada se corresponde con *bits* puestos en el primer estado seleccionable en el identificador del decodificador.
2. Una configuración de memoria, en conformidad con la reivindicación 1, en la que el medio para comparar el identificador de un decodificador de dirección (2) con la dirección de entrada considera la correspondencia posicional entre *bits* puestos en el primer estado seleccionable en la dirección de entrada y *bits* puestos en el primer estado seleccionable en los identificadores del decodificador (2).
3. Una configuración de memoria, en conformidad con la reivindicación 1, o con la 2, en la que cada identificador del decodificador de dirección (2) tiene un número igual de *bits* puestos en el primer estado seleccionable.
4. Una configuración de memoria, en conformidad con cualquier reivindicación precedente, en la que el medio para recibir una dirección de entrada está configurado para recibir direcciones que contienen un número predefinido de *bits* puestos en el primer estado seleccionable.
5. Una configuración de memoria, en conformidad con las reivindicaciones 3 y 4, en la que un número predefinido de *bits* puestos en el primer estado seleccionable en una dirección de entrada es igual al número de *bits* puestos en el primer estado seleccionable en cada uno de los identificadores de decodificadores de dirección (2).
6. Una configuración de memoria, en conformidad con cualquier reivindicación precedente, en la que la memoria de datos (3) consta de una pluralidad de memorias de *bit* único, de modo que cada *bit* de cada línea de palabra está almacenado en una memoria de *bit* único.
7. Una configuración de memoria, en conformidad con cualquier reivindicación precedente, en la que la memoria de datos (3) consta de una línea de entrada de datos (5) que contiene un número de *bits* igual al de cada una de la pluralidad de las líneas de palabra.
8. Una configuración de memoria, en conformidad con la reivindicación 7, que consta además de medios para escribir datos para copiar datos de la línea de entrada de datos (5) a líneas de palabra activadas por los decodificadores de dirección (2).
9. Una configuración de memoria, en conformidad con la reivindicación 7, o con la 8, en la que la línea de entrada (5) está configurada para recibir datos de entrada que contienen un número predefinido de *bits* puestos en el primer estado seleccionable.
10. Una configuración de memoria, en conformidad con cualquier reivindicación precedente, que consta además de medios para sumar valores almacenados en cada *bit* de las líneas de palabra activadas por un decodificador de dirección (2) para generar un valor de nivel de activación para cada *bit*.
11. Una configuración de memoria, en conformidad con las reivindicaciones 9 y 10, que consta además de medios para generar una palabra de salida que contiene un número predefinido de *bits* puestos en el primer estado seleccionable.
12. Una configuración de memoria, en conformidad con la reivindicación 11, en la que los *bits* puestos en el primer estado seleccionable en la salida son el número predefinido de *bits* dotados del nivel de activación más elevado.
13. Una configuración de memoria, en conformidad con cualesquiera de las reivindicaciones 1 a la 12, en la que la memoria se implemente usando una pluralidad de neuronas artificiales (16) conectadas entre sí para formar una red neuronal.
14. Una configuración de memoria, en conformidad con la reivindicación 13, en la que la pluralidad de decodificadores de dirección (2) está representada por una pluralidad de neuronas de decodificación de dirección (15), y la memoria de datos (3) está representada por una pluralidad de neuronas de datos.
15. Una configuración de memoria de red neuronal para su empleo en un sistema informático, constando la memoria de una pluralidad de neuronas de decodificación de dirección (15), cada una de las cuales está conectada a un número predefinido de neuronas de entrada (14), y de una memoria de datos dotada de una pluralidad de neuronas de

datos, siendo susceptible de activación cada una de dichas neuronas de decodificación de dirección (15) para seleccionar algunas de la pluralidad de las neuronas de datos, y constando las neuronas de decodificación de dirección (15) de un medio para recibir una señal que representa un disparo de una neurona de entrada (14) a la que va conectado, en la que una neurona de decodificación (15) consta de medios para activar neuronas de datos si se reciben señales de disparo desde al menos un número mínimo predefinido de neuronas de entrada (14) a las que va conectada la neurona de decodificación de dirección (15).

16. Un método para hacer funcionar una memoria para su empleo en un sistema informático, constando la memoria de una pluralidad de decodificadores de dirección (2), a cada uno de los cuales se le asigna un identificador que tiene un número predefinido de *bits*, teniendo cada *bit* estados seleccionables primero y segundo, y una memoria de datos (3) que tiene una pluralidad de líneas de palabra de longitud predefinida, siendo susceptible de activación cada uno de dichos decodificadores de dirección (2) para seleccionar una de entre la pluralidad de líneas de palabra, donde una dirección de entrada que tiene un número predefinido de *bits* es introducida al decodificador de dirección (2), el identificador de un decodificador de dirección (2) es comparado con la dirección de entrada y los decodificadores de dirección (2) se activan si al menos un número mínimo predefinido de *bits* puestos en el primer estado seleccionable en la dirección de entrada se corresponden con los *bits* puestos en el primer estado seleccionable en el identificador del decodificador.

17. Un método, en conformidad con la reivindicación 16, en el que los datos de entrada se presentan en una entrada de datos de la memoria de datos y los datos se escriben a líneas de palabra activadas por los decodificadores de dirección activados (2).

18. Un método, en conformidad con la reivindicación 16, o con la 17, en el que el número mínimo predefinido de *bits* está puesto de tal modo que se activen menos de 100 decodificadores de dirección (2) por cualquier dirección válida de entrada.

19. Un método, en conformidad con la reivindicación 18, en el que el número mínimo predefinido de *bits* está puesto de tal modo que se activen menos de 50 decodificadores de dirección (2) por cualquier dirección válida de entrada.

20. Un método, en conformidad con la reivindicación 19, en el que el número mínimo predefinido de *bits* está puesto de tal modo que se activen menos de 20 y más de 11 decodificadores de dirección (2) por cualquier dirección válida de entrada.

21. Un medio portador que lleva un medio de código legible para ordenadores para hacer que un ordenador ejecute un procedimiento en conformidad con el método de cualesquiera de las reivindicaciones 16 a la 20.

22. Un programa informático para efectuar el método de cualesquiera de las reivindicaciones 16 a la 20.

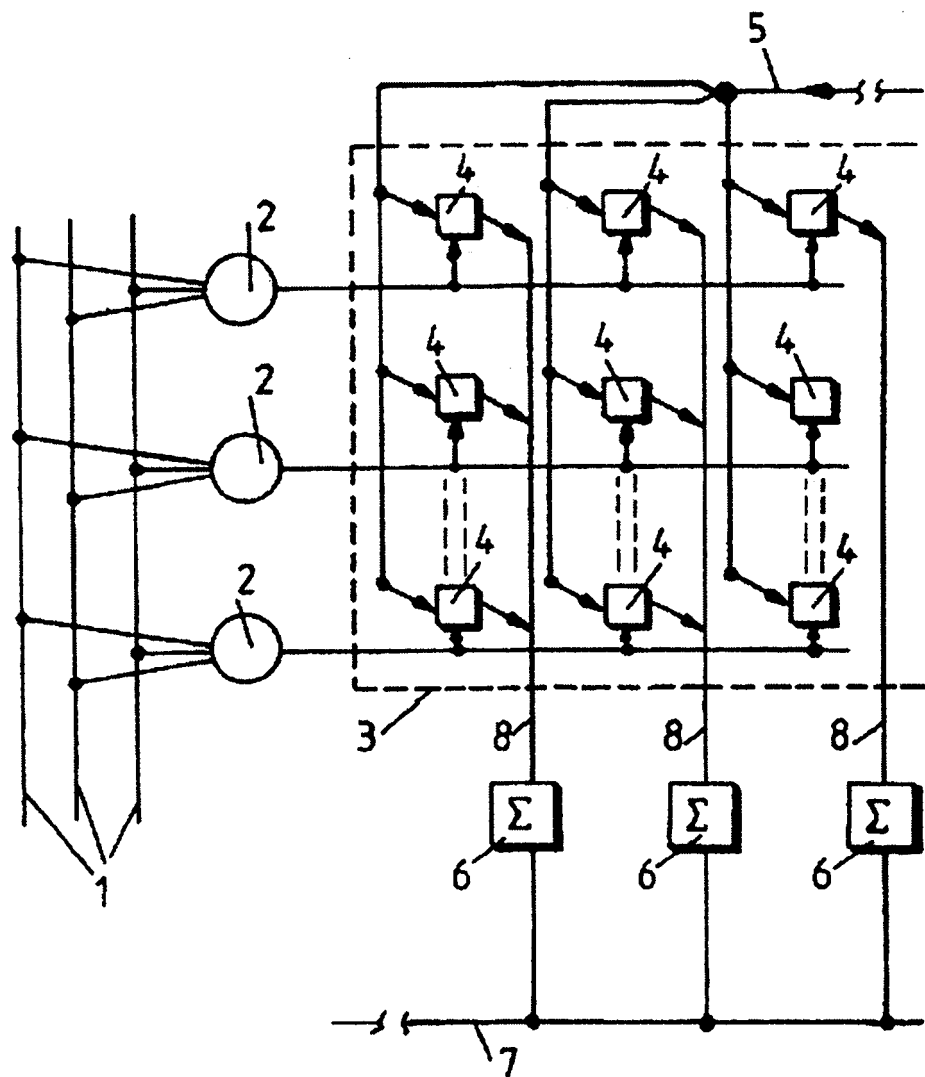


FIG. 1

