



(19)中華民國智慧財產局

(12)發明說明書公告本

(11)證書號數：TW I719348 B

(45)公告日：中華民國 110 (2021) 年 02 月 21 日

(21)申請案號：107136964

(22)申請日：中華民國 107 (2018) 年 10 月 19 日

(51)Int. Cl. : G06F9/30 (2018.01)

G06F9/38 (2006.01)

G06F7/58 (2006.01)

(30)優先權：2017/10/20 英國

1717306.3

(71)申請人：英商葛夫科有限公司 (英國) GRAPHCORE LIMITED (GB)

英國

(72)發明人：菲立克斯 史蒂芬 FELIX, STEPHEN (GB)；諾爾斯 賽門 克里斯汀 KNOWLES, SIMON CHRISTIAN (GB)；達 柯斯塔 高德弗瑞 DA COSTA, GODFREY (GB)

(74)代理人：陳長文

(56)參考文獻：

TW I235343

審查人員：黃鴻鈞

申請專利範圍項數：16 項 圖式數：4 共 21 頁

(54)名稱

執行單元，用於執行電腦程式之方法，以及電腦程式

(57)摘要

本發明係關於一種用於執行包括一指令序列之一電腦程式之執行單元，該序列包含一遮罩指令。該執行單元經組態以執行該遮罩指令，該遮罩指令在由該執行單元執行時遮罩來自 n 個值之一來源運算元之隨機選擇值且保留來自該來源運算元之其他原始值以產生一結果，該結果包含來自該來源運算元之原始值及其等各自原始位置中之該等遮罩值。

The present invention relates to an execution unit for executing a computer program comprising a sequence of instructions, which include a masking instruction. The execution unit is configured to execute the masking instruction which when executed by the execution unit masks randomly selected values from a source operand of n values and retains other original values from the source operand to generate a result which includes original values from the source operand and the masked values in their respective original locations.

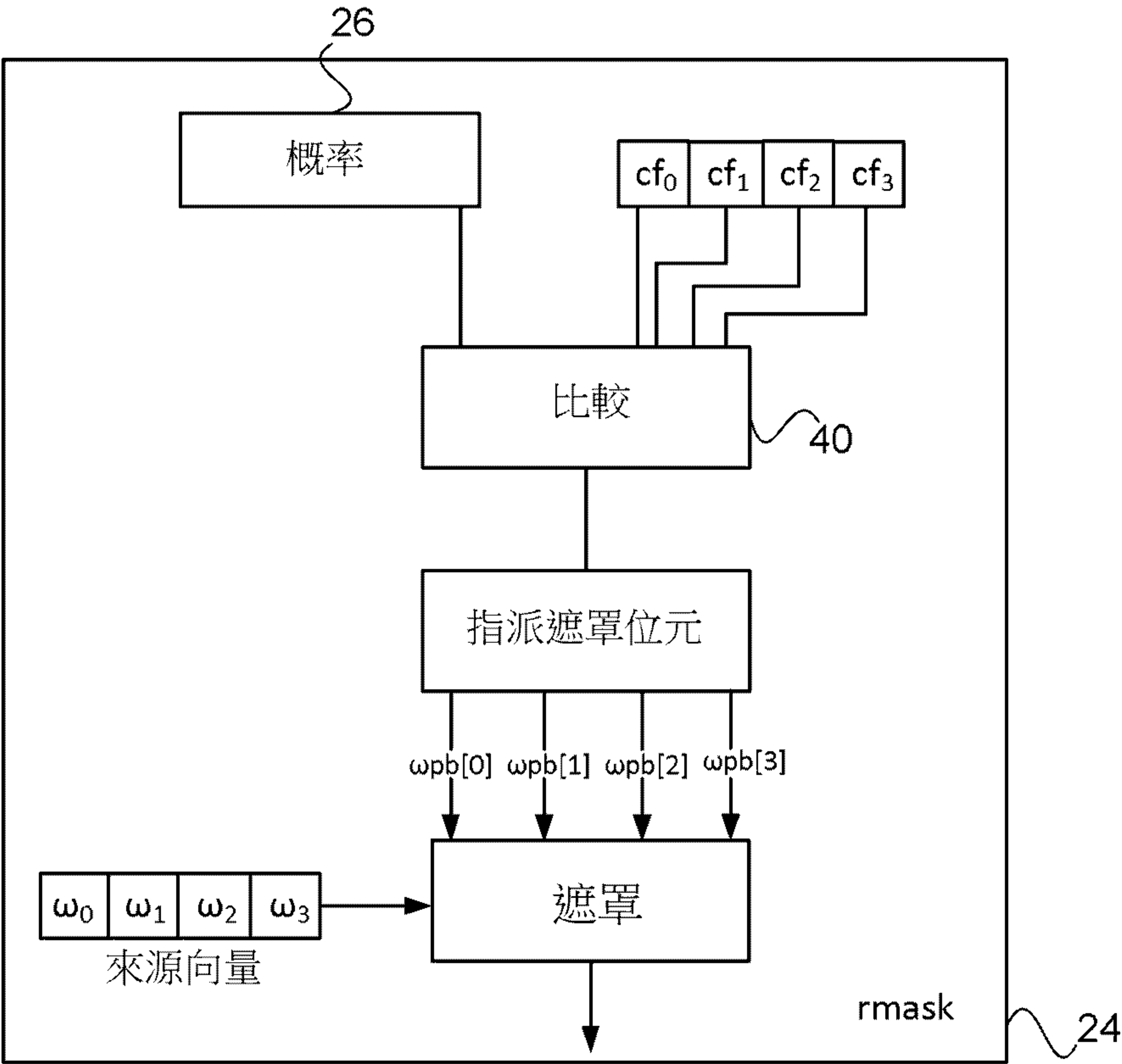
指定代表圖：

符號簡單說明：

24:rmask 模組

26:輸入緩衝器位置/概
率值

40:比較邏輯



【圖4】



I719348

【發明摘要】

【中文發明名稱】

執行單元，用於執行電腦程式之方法，以及電腦程式

【英文發明名稱】

EXECUTION UNIT, METHOD FOR EXECUTING COMPUTER PROGRAM, AND COMPUTER PROGRAM

【中文】

本發明係關於一種用於執行包括一指令序列之一電腦程式之執行單元，該序列包含一遮罩指令。該執行單元經組態以執行該遮罩指令，該遮罩指令在由該執行單元執行時遮罩來自 n 個值之一來源運算元之隨機選擇值且保留來自該來源運算元之其他原始值以產生一結果，該結果包含來自該來源運算元之原始值及其等各自原始位置中之該等遮罩值。

【英文】

The present invention relates to an execution unit for executing a computer program comprising a sequence of instructions, which include a masking instruction. The execution unit is configured to execute the masking instruction which when executed by the execution unit masks randomly selected values from a source operand of n values and retains other original values from the source operand to generate a result which includes original values from the source operand and the masked values in their respective original locations.

【指定代表圖】

圖4

【代表圖之符號簡單說明】

24	rmask模組
26	輸入緩衝器位置/概率值
40	比較邏輯

【發明說明書】

【中文發明名稱】

執行單元，用於執行電腦程式之方法，以及電腦程式

【英文發明名稱】

EXECUTION UNIT, METHOD FOR EXECUTING COMPUTER PROGRAM, AND COMPUTER PROGRAM

【技術領域】

【0001】 本發明係關於處理類神經網路中之資料。

【先前技術】

【0002】 類神經網路用於機器學習及人工智慧領域。類神經網路包括藉由鏈路互連且彼此互動之節點組之配置。運算中之類神經網路之原理基於關於電刺激如何在人腦中傳達資訊之資訊。出於此原因，節點通常被稱為類神經元。節點亦可稱為頂點。鏈路有時稱為邊緣。網路可獲取輸入資料且特定節點對資料執行運算。將此等運算之結果傳遞至其他節點。各節點之輸出稱為其激勵(activation)或節點值。各鏈路與一權重相關聯。一權重定義類神經網路之節點之間的連接性。已知類神經網路能夠藉由其等學習之諸多不同技術，該學習係藉由改變權重值來進行。

【0003】 圖1展示一類神經網路中之一種節點配置之一極簡化版本。此類型之配置通常用於學習或訓練且包括一節點輸入層、一節點隱藏層及一節點輸出層。實際上，各層中將存在諸多節點，且現今每個區段可存在一個以上層。輸入層 N_i 之各節點能夠在其輸出處產生一激勵或節點值，該激勵或節點值係藉由對提供給該節點之資料實行一函數而產生。藉由隱藏層中之各節點之輸入處之各自權重之一向量來按比例調整來自輸入層之節點值之一向量，各權重定義該特定節點與其在隱藏層中之連接節點

之連接性。在實踐中，網路可具有數百萬個節點且多維連接，因此向量通常係一張量。施加於節點 N_h 之輸入處之權重標記為 $w_0 \dots w_2$ 。輸入層中之各節點至少最初連接至隱藏層中之各節點。隱藏層中之各節點可對提供給其等之資料執行一激勵函數且可類似地產生一輸出向量，該輸出向量供應給輸出層 o 中之節點 N_o 之各者。各節點例如藉由實行節點之輸入激勵及各自傳入鏈路之唯一權重之點積來對其傳入資料進行加權。接著，各節點對加權資料執行一激勵函數。該激勵函數可為例如一S形(sigmoid)。參見圖1A。網路藉由對輸入層處輸入之資料進行運算，將權重指派給來自各節點之激勵且對輸入至隱藏層中之各節點之資料進行運算(藉由對其進行加權且執行激勵函數)來學習。因此，隱藏層中之節點對資料進行運算且將輸出供應給輸出層中之節點。輸出層之節點亦可將權重指派給其等傳入資料。各權重之特徵在於一各自誤差值。此外，各節點可與一誤差條件相關聯。各節點處之誤差條件給定節點權重誤差是否低於一特定可接受性位準或程度之一量度。存在不同學習方法，但在各情況下，誤差存在透過網路之一正向傳播(圖1中自左至右)、一總體誤差計算及透過網路之一反向傳播(圖1中自右至左)。在下一循環中，各節點考量反向傳播誤差且產生一權重修訂集。以此方式，可訓練網路以執行其期望運算。

【0004】 一類神經網路可能出現之一個問題係「過度擬合」。具有數百萬或數十億個參數(權重)之大型網路可能容易過度擬合。過度擬合引起一網路記住已提供給其之各訓練樣本(將資料提供給輸入節點之一訓練樣本)，而非經訓練以提取相關特徵使得類神經網路適於在其已經訓練之後應用以更一般地自樣本提取特徵。已開發廣泛範圍之技術以藉由使類神經網路正規化以避免過度擬合/記憶來解決此問題。

【0005】在JE Hinton等人撰寫之標題為「Improving Neural Networks by Preventing Co-Adaption of Feature Detectors」CoRR abs/1207.0580 (2012)之論文中論述稱為「捨棄(Drop out)」之一技術。根據此技術，針對各訓練實例，正向傳播涉及隨機刪除各層中之一半激勵。接著，僅透過剩餘激勵反向傳播誤差。已表明，此顯著減少過度擬合且改良淨效能。

【0006】在由紐約電腦科學系出版之Li Wan等人撰寫之標題為「Regularisation of Neural Network using Drop connect」之論文中論述稱為「棄連(Drop connect)」之另一技術。根據此技術，在正向傳播期間，針對各訓練實例，將一權重子集隨機歸零。已類似地表明，此技術改良類神經網路之效能。

【0007】使用已知電腦技術實施類神經網路具有各種挑戰。例如使用一CPU或GPU實施特定技術(如同Drop out及Drop connect)並非易事且且會影響有效實施方案可達成之全部益處。

【發明內容】

【0008】本發明人已開發一種用於一處理器之執行單元，該處理器可基於該處理單元之一指令序列中之單個指令有效地實施Drop out或Drop connect。

【0009】根據本發明之一態樣，提供一種用於執行包括一指令序列之一電腦程式之執行單元，該序列包含一遮罩指令，其中該執行單元經組態以執行該遮罩指令，該遮罩指令在由該執行單元執行時遮罩來自n個值之一來源運算元之隨機選擇值且保留來自該來源運算元之其他原始值以產生一結果，該結果包含來自該來源運算元之原始值及其等各自原始位置中

之該等遮罩值。

【0010】該執行單元可包括一硬體偽隨機數產生器(HPRNG)，該HPRNG經組態以產生一隨機化位元串，隨機位元序列係自該隨機化位元串導出以隨機選擇該等待遮罩值。

【0011】在一項實施例中，自該HPRNG輸出之各隨機化位元串包括m個位元，且該執行單元經組態以將該m個位元分成p個序列且比較各序列與一概率值以產生一加權概率指示符以選擇性地遮罩該等值。

【0012】該遮罩指令可包括識別來源向量之一來源欄位、用於保存該結果之一目的地暫存器之一指示及定義該概率值之一概率欄位。

【0013】該執行單元可包括用於保存該來源運算元之一輸入緩衝器及用於保存該結果之一輸出緩衝器。

【0014】該來源運算元之值可表示一類神經網路中之鏈路之權重(例如，以實施Drop connect)。

【0015】該來源運算元中之值可表示定義一類神經網路中之節點之輸出值之激勵(例如，以實施Dropout)。

【0016】該指令序列可包括用於實施該結果與一進一步值集之一點積之指令、及/或用於在已執行該遮罩指令之後將該結果寫入至一記憶體位置之一指令。

【0017】該來源運算元可包括任何適當長度之任何方便數目個值，例如四個16位元值、兩個32位元值或八個8位元值(作為非限制性實例)。一般而言，n個值可為一或多個值。

【0018】提供一種對應方法及電腦程式。

【0019】一個態樣提供一種執行包括一指令序列之一電腦程式之方

法，該序列包含一遮罩指令，該方法包括：回應於該遮罩指令之執行，自 n 個值之一來源運算元隨機選擇值且保留來自該來源運算元之其他原始值以產生一結果，該結果包含來自該來源運算元之原始值及其等各自原始位置中之遮罩值。

【0020】 可提供包括 m 個位元之一隨機化位元串輸出，且該方法可包括將該 m 個位元分成 p 個序列及比較各序列與一概率值以產生一加權概率指示符以選擇性地遮罩該等值。

【0021】 該遮罩指令可包括識別來源向量之一來源欄位、用於保存該結果之一目的地暫存器之一指示及定義該概率值之一概率欄位。

【0022】 該指令序列可包括用於實施該結果與一進一步值集之一點積之指令。

【0023】 該指令序列可包括用於在已執行該遮罩指令之後將該結果寫入至一記憶體位置之一指令。

【0024】 另一態樣提供一種包括儲存於不可傳輸媒體上之一電腦可讀指令序列之電腦程式，該序列包含一遮罩指令，該遮罩指令在被執行時自 n 個值之一來源運算元隨機選擇待遮罩值且保留來自該來源運算元之其他原始值以產生一結果，該結果包含來自該來源運算元之原始值及其等各自原始位置中之該等遮罩值。

【0025】 當在本文中使用时，術語「隨機」及「隨機地」被認為意謂隨機或偽隨機。

【圖式簡單說明】

【0026】 圖1係類一類神經網路之一高度簡化示意圖；

【0027】 圖1A係一類神經元之一高度簡化示意圖；

【0028】 圖2係根據本發明之一實施例之一處理單元之一示意圖；

【0029】 圖3展示一遮罩指令之格式；及

【0030】 圖4係用於實施遮罩指令之一執行單元之一方塊圖。

【實施方式】

【0031】 圖2展示一執行單元之一示意方塊圖，該執行單元經配置以執行用於遮罩一向量中之隨機選擇值之單個指令。該指令在本文中稱為 **rmask** 指令。執行單元2在一處理單元中形成一管線4之部分。該處理單元包括一指令提取單元6，指令提取單元6自一指令記憶體10提取指令。該處理單元亦包括一記憶體存取階8，記憶體存取階8負責存取一資料記憶體12以自該記憶體載入資料或將資料儲存至該記憶體中。提供一組暫存器14以保存由管線4在任何情況下執行之指令之來源運算元及目的地運算元。將容易理解，管線4可包含用於執行各種不同指令(例如，用於執行數學運算)之諸多不同類型之執行單元。可用於本發明之一種類型之處理單元係使用桶形執行緒時槽之一處理單元，其中一管理程式執行緒(**supervisor thread**)可將不同工作執行緒分配給不同時槽以便執行。本文中所描述之 **rmask** 指令可與任何合適處理單元架構一起使用。圖3展示 **rmask** 指令格式。該指令具有：一欄位30，其定義待自其擷取一來源運算元之一源記憶體位置；及一欄位32，其用於將輸出向量放入其中之一目的地記憶體位置。該記憶體位置可為暫存器14中之一隱式或顯式定義暫存器，或該資料記憶體中之一位址。**rmask** 指令進一步定義一概率值 **Src1** 34，將利用概率值 **Src1** 34 保持來源運算元內之各個別值，如稍後更詳細描述。該指令具有運算碼36，運算碼36將該指令識別為 **rmask** 指令。**rmask** 指令具有兩種不同格式。在該指令之一種格式中，該來源運算元可

表示包括四個16位元值之一向量(半精度向量)，且在另一格式中，該來源運算元可定義包括兩個32位元值之一向量(單精度向量)。

【0032】 在各情況下，**rmask**指令具有遮罩(置為0)來源向量中之隨機選擇值之效應。此一函數具有諸多不同應用，尤其包含上文已描述、在類神經網路技術中稱為「**Drop connect**」及「**Drop out**」之函數。如已解釋，在一類神經網路中，可提供表示權重之一向量，此等權重表示該網路中之類神經元之間的鏈路。另一向量可表示激勵，此等激勵係輸出至藉由鏈路連接之個別類神經元之值。

【0033】 類神經網路中之一非常常見函數係在此兩個向量之間實行一點積。例如，該點積可係預激勵值，其係使用一激勵函數自其計算一類神經元輸出之值。在計算此點積之前將**rmask**指令應用於權重向量可用來實施**Drop-connect**函數。將**rmask**指令應用於權重向量將向量內之個別權重隨機歸零。

【0034】 在計算該點積之前將**rmask**指令應用於激勵向量可用來實施**Dropout**函數。將**rmask**指令應用於激勵向量將向量內之個別激勵值隨機歸零。

【0035】 向量可依單精度或半精度兩者來表示浮點數。一種類型之向量可包括四個16位元值(半精度向量)，而另一類型之向量可包括兩個32位元值(單精度向量)。在圖2中，展示對四個16位元值或兩個32位元值進行運算之一執行單元。圖2中之執行單元2具有一輸入緩衝器20，輸入緩衝器20可保存四個16位元值或兩個32位元值。輸入緩衝器20被展示為保存四個16位元值，各值表示一權重 $w_0...w_3$ 。該單元具有一偽隨機數產生器22，偽隨機數產生器22係以硬體實施以將隨機數提供給一**rmask**模組

24。提供一輸入緩衝器位置26以保存一概率，將使用該概率保持向量內之各個別值。此概率值可提供於該指令中(如圖3中所展示)，或可由一較早指令設定且可利用一rmask指令自一暫存器或記憶體位址存取。例如，該概率值可為資訊中之一17位元立即值。該rmask模組使用由PRNG 22產生之隨機數及概率值26來隨機遮罩(置為0)輸入向量之值。輸出向量放入一輸出緩衝器28。在圖2中，第一及第三值被展示為已遮罩為0。應注意，該概率係將利用其保持(即，未遮罩)各個別值之概率。例如，若將該概率設定為.5，則各個別值具有將被遮罩之一概率.5。因此，所有四個值被遮罩為0之概率係 $(.5)^4$ ，即，1/16。隨機數產生器基於該概率，藉由提供應遮罩哪些值之一指示來將所需隨機性引入遮罩。

【0036】 參考圖4，該rmask模組如下般運算。偽隨機數產生器22產生一64位元輸出。針對各輸出，產生四個16位元欄位 $cf_0...cf_3$ 且根據來源運算元中之概率值進行測試。應注意，針對單精度模式，產生兩個32位元欄位且根據來源運算元中之概率值進行類似測試。

【0037】 當發起遮罩指令時，提供PRNG之一64位元輸出。針對各輸出，形成四個16位元欄位。Res0[15:0]指代PRNG輸出之最低有效16個位元，且此語法適用於其他三個欄位。將此等欄位之各者指派給一cf欄位(cf代表比較欄位)且比較欄位 $cf_0..cf_3$ 之各者係16位元長。

```
assign cf0[15:0] = res0[15:0];
assign cf1[15:0] = res0[31:16];
assign cf2[15:0] = res0[47:32];
assign cf3[15:0] = res0[63:48];
```

【0038】 接著，藉由使用比較邏輯40比較各比較欄位 $cf_0...3$ 與來自來源運算元src1[15:0]之概率值來導出四個加權隨機位wpb[0]...wpb[3]。

```

assign wpb[0] = (src1[16]==1 | cf0 < src1[15:0])? 1'b1 : 1'b0;
assign wpb[1] = (src1[16]==1 | cf1 < src1[15:0])? 1'b1 : 1'b0;
assign wpb[2] = (src1[16]==1 | cf2 < src1[15:0])? 1'b1 : 1'b0;
assign wpb[3] = (src1[16]==1 | cf3 < src1[15:0])? 1'b1 : 1'b0;

```

【0039】 上述語法意謂著若 $cf0 < src1$ ，則位元wpb[0]被指派「1」。

在此，cf0係一16位元不帶正負號隨機值，其可具有範圍{0...65535}內之值；src1係一32位元寬運算元，但僅最低有效17個位元用於rmask；src1係{0至65536}範圍內之允許值，使得「解除遮罩」之概率係 $src1/65536$ 。

【0040】 應注意，針對所允許src1範圍{0...65536}，src1之第17位元(src1[16])僅設定為值65536。由於cf0之最大值為65535且 $65535 < 65536$ ，因此當 $src1[16]==1$ 時wpb[0]自動為「1」。

【0041】 隨後分別使用四位wpb[0..3]來分別解除遮罩src0[63:0]中之4個16位元值，因此：

```

assign aDst[15:0] = (wpb[0]==1)? aSrc0[15:0] : 16'b0;
assign aDst[31:16] = (wpb[1]==1)? aSrc0[31:16] : 16'b0;
assign aDst[47:32] = (wpb[2]==1)? aSrc0[47:32] : 16'b0;
assign aDst[63:48] = (wpb[3]==1)? aSrc0[63:48] : 16'b0;

```

【0042】 其中16'b0表示具有值「0」之一16位元寬序列，1'b0意謂一位元值「0」且1'b1表示一位元值「1」。

【0043】 例如，若wpb[1]==1，則 $aDst[31:16]=aSrc0[31:16]$ ；否則 $aDst[31:16]=0$ 。

【0044】 在一項實施例中，概率值aSrc1用來選擇利用其隨機「解除遮罩」(即：未遮罩)值之概率。

【0045】 若運算元aSrc1=65536，則始終解除遮罩所有值

【0046】 若運算元aSrc1=49152，則各值利用一概率0.25獨立遮罩且利用一概率0.75解除遮罩

【0047】 若運算元aSrc1=32768，則各值利用一概率0.50獨立遮罩

且利用一概率0.50解除遮罩

【0048】 若運算元aSrc1=16384，則各值利用一概率0.75獨立遮罩
且利用一概率0.25解除遮罩

【0049】 若運算元aSrc1=0，則始終遮罩所有值

【0050】 「解除遮罩」一輸入意謂著在輸出處再生由輸入表示之該
值。輸出格式及精度不一定相同於輸入格式。

【0051】 「遮罩」輸入意謂著在輸出處產生表示零值而非由輸入表
示之值之一正負號。

【0052】 應注意，一解除遮罩概率==1-遮罩概率

【0053】 在本文中所描述之rmask之實施方案中，概率值
aSrc1==65536*解除遮罩概率。對應遮罩概率可自遮罩概率導出：
65536*(1-遮罩概率)。

【0054】 因此，將明白，定義一遮罩概率或一解除遮罩概率(兩個概
率之和為「1」)。

【0055】 rmask指令具有數個不同應用。例如，當實施Drop out
時，可在將類神經輸出寫入至記憶體之前使用Drop out。接著，類神經網
路之下一階中之所有類神經元將自先前層拾取遮罩激勵。

【0056】 當實施Drop connect時，可剛好在自記憶體讀取激勵之
後、剛好在計算點積之前使用rmask指令。該指令不知道來源向量表示什
麼且因此存在除Drop out及Drop connect之外之其他應用。該指令亦不知
道來源向量內之16位元或32位元純量值之格式。在輸出中，各純量值保
持不變或所有位元遮罩為零。因此，該遮罩指令可用來遮罩四個16位元帶
正負號或不帶正負號整數之一向量或各種16位元浮點數格式。純量值格式

之唯一要求係由所有位元表示為零之數之值為「0」。

【0057】 在一些數格式中，可能存在表示零值之一個以上可能符號。

【0058】 例如，在IEEE半精度浮點數格式中，以下兩個符號可用來表示零：

【0059】 16'b0000000000000000 (正零)

【0060】 16'b1000000000000000 (負零)

【0061】 本文中所使用之術語「隨機」可意謂「真隨機」或「偽隨機」。rmask指令可使用一偽隨機或一真隨機位元序列產生器。

【0062】 由一「偽隨機數產生器」或「PRNG」產生偽隨機數。PRNG可以軟體或硬體實施。由一「真隨機數產生器」或「TRNG」產生真隨機數。一TRNG之一實例係一「轉變效應環形振盪器」。PRNG優於TRNG之處在於確定性(在相同起始條件下運行相同程式兩次始終具有相同結果)。

【0063】 TRNG優於PRNG之處在於輸出係真隨機(而一PRNG之輸出滿足任意選擇數學性質之一有限集；一PRNG之狀態及輸出始終可自當前狀態預測且因此并非真隨機)。

【0064】 雖然已描述特定實施例，但一旦給定本發明，所揭示技術之其他應用及變體對於熟習此項技術者而言將變得顯而易見。本發明之範疇不受限於所描述實施例，而僅受限於隨附申請專利範圍。

【符號說明】

【0065】

2 執行單元

4	管線
6	指令提取單元
8	記憶體存取階
10	指令記憶體
12	資料記憶體
14	暫存器
20	輸入緩衝器
22	偽隨機數產生器
24	rmask模組
26	輸入緩衝器位置/概率值
28	輸出緩衝器
30	欄位
32	欄位
34	概率值Src1
36	運算碼
40	比較邏輯

【發明申請專利範圍】

【第1項】

一種用於執行包括多個指令之一序列之一電腦程式之執行單元，該序列包含一遮罩指令，其中該執行單元經組態以執行該遮罩指令，該遮罩指令在由該執行單元執行時遮罩來自 n 個值之一來源運算元之隨機選擇多個值且保留來自該來源運算元之其他原始值以產生一結果，該結果包含來自該來源運算元之原始值及其等各自原始位置中之該等經遮罩值，其中該執行單元包括一硬體偽隨機數產生器(HPRNG)，其經組態以產生一隨機化位元串，自該隨機化位元串導出隨機位元序列而用於隨機選擇待遮罩之該等值。

【第2項】

如請求項1之執行單元，其中自該HPRNG輸出之各隨機化位元串包括 m 個位元，該執行單元經組態以將該 m 個位元分成 p 個序列且將各序列與一概率值比較以產生一加權概率指示符而用於選擇性地遮罩該等值。

【第3項】

如請求項2之執行單元，其中該遮罩指令包括識別該來源運算元之一來源欄位、用於保存該結果之一目的地暫存器之一指示及定義該概率值之一概率欄位。

【第4項】

如請求項1至3中任一項之執行單元，其包括用於保存該來源運算元之一輸入緩衝器及用於保存該結果之一輸出緩衝器。

【第5項】

如請求項1至3中任一項之執行單元，其中該來源運算元之該等值表

示一類神經網路中之鏈路之權重。

【第6項】

如請求項1至3中任一項之執行單元，其中該來源運算元中之該等值表示定義一類神經網路中之節點之輸出值之激勵(activations)。

【第7項】

如請求項1至3中任一項之執行單元，其中該等指令之該序列包括用於實施該結果與一進一步值集(a further set of values)之一點積之指令。

【第8項】

如請求項1至3中任一項之執行單元，其中該等指令之該序列包括在已執行該遮罩指令之後用於將該結果寫入至一記憶體位置之一指令。

【第9項】

如請求項1至3中任一項之執行單元，其中該來源運算元包括四個16位元值。

【第10項】

如請求項1至3中任一項之執行單元，其中該來源運算元包括兩個32位元值。

【第11項】

一種執行一電腦程式之方法，該方法實施於一執行單元中，該電腦程式包括多個指令之一序列，該序列包含一遮罩指令，該方法包括：

回應於該遮罩指令之執行，自n個值之一來源運算元隨機選擇多個值且保留來自該來源運算元之其他原始值以產生一結果，該結果包含來自該來源運算元之原始值及其等各自原始位置中之該等經遮罩值；及

藉由該執行單元之一硬體偽隨機數產生器產生一隨機位元串，自該

隨機化位元串導出隨機位元序列而用於隨機選擇待遮罩之該等值。

【第12項】

如請求項11之方法，其包括提供包括 m 個位元之一隨機化位元串輸出，將該 m 個位元分成 p 個序列及將各序列與一概率值比較以產生一加權概率指示符而用於選擇性地遮罩該等值。

【第13項】

如請求項12之方法，其中該遮罩指令包括識別該來源運算元之一來源欄位、用於保存該結果之一目的地暫存器之一指示及定義該概率值之一概率欄位。

【第14項】

如請求項11至13中任一項之方法，其中該等指令之該序列包括用於實施該結果與一進一步值集之一點積之指令。

【第15項】

如請求項11至13中任一項之方法，其中該等指令之該序列包括在已執行該遮罩指令之後用於將該結果寫入至一記憶體位置之一指令。

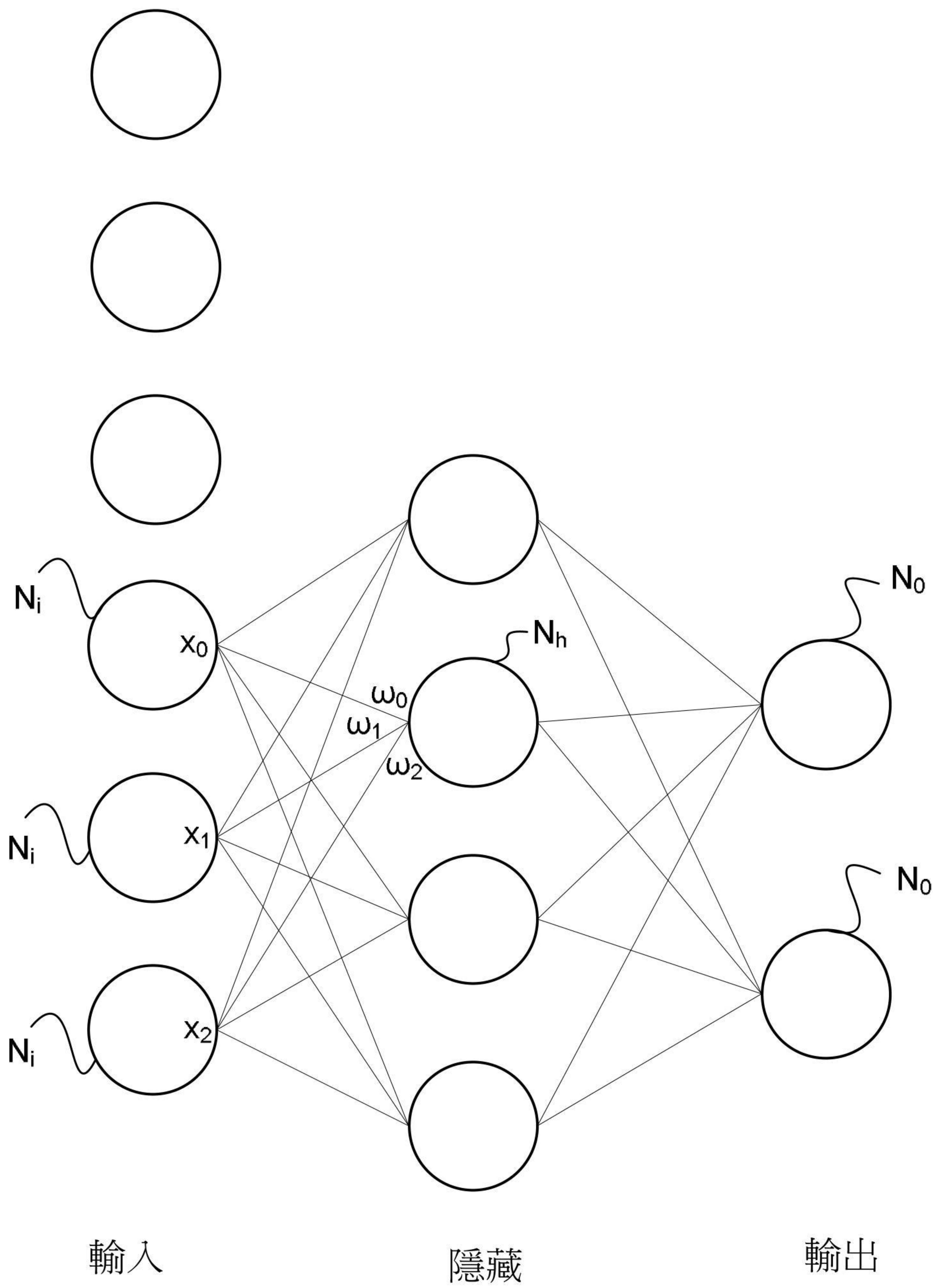
【第16項】

一種包括儲存於不可傳輸媒體上之多個電腦可讀指令之一序列之電腦程式，該序列包含一遮罩指令，該遮罩指令在藉由一執行單元被執行時：

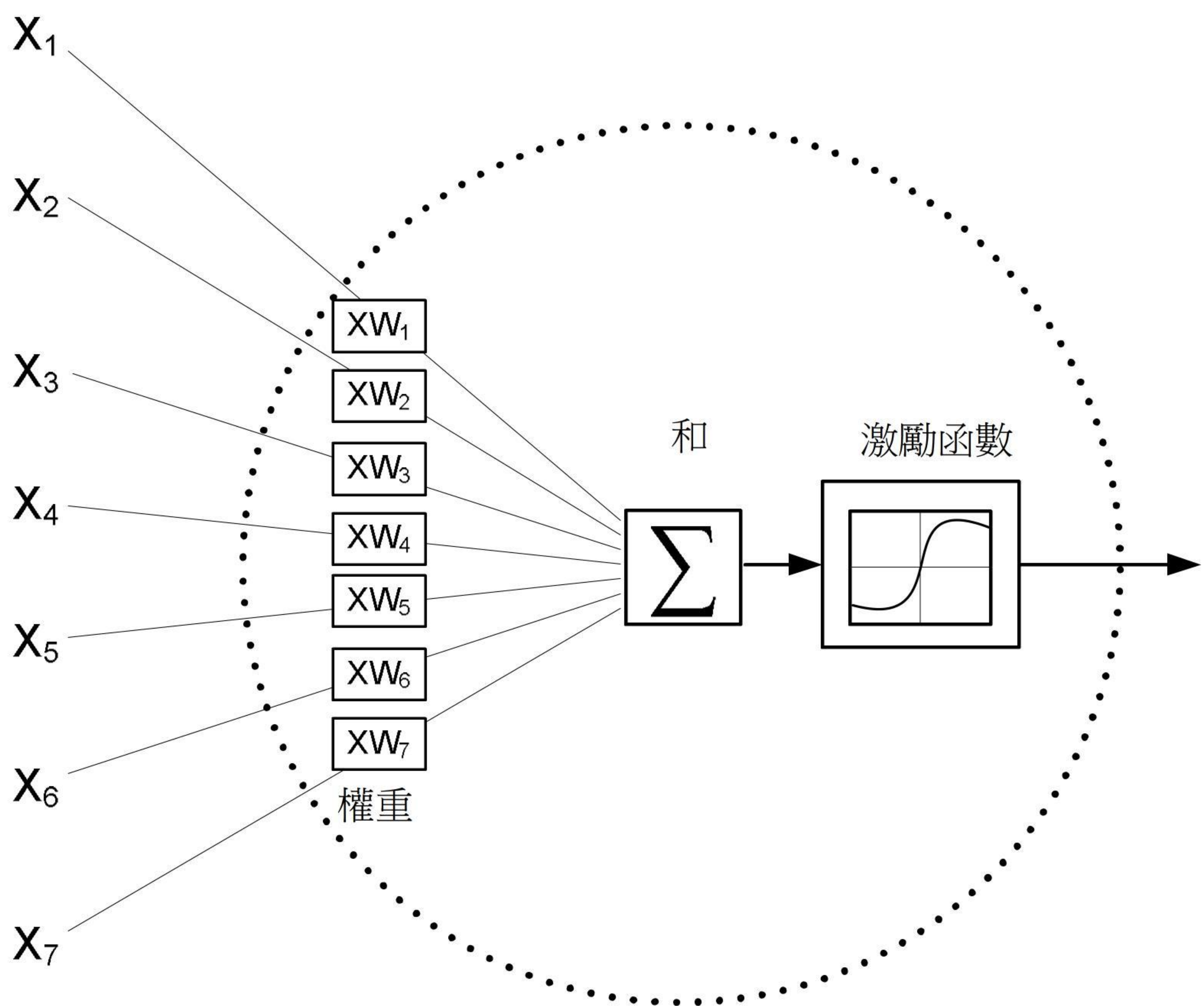
自 n 個值之一來源運算元隨機選擇多個值用以遮罩且保留來自該來源運算元之其他原始值以產生一結果，該結果包含來自該來源運算元之原始值及其等各自原始位置中之該等經遮罩值；及

藉由該執行單元之一硬體偽隨機數產生器產生一隨機位元串，自該隨機化位元串導出隨機位元序列而用於隨機選擇待遮罩之該等值。

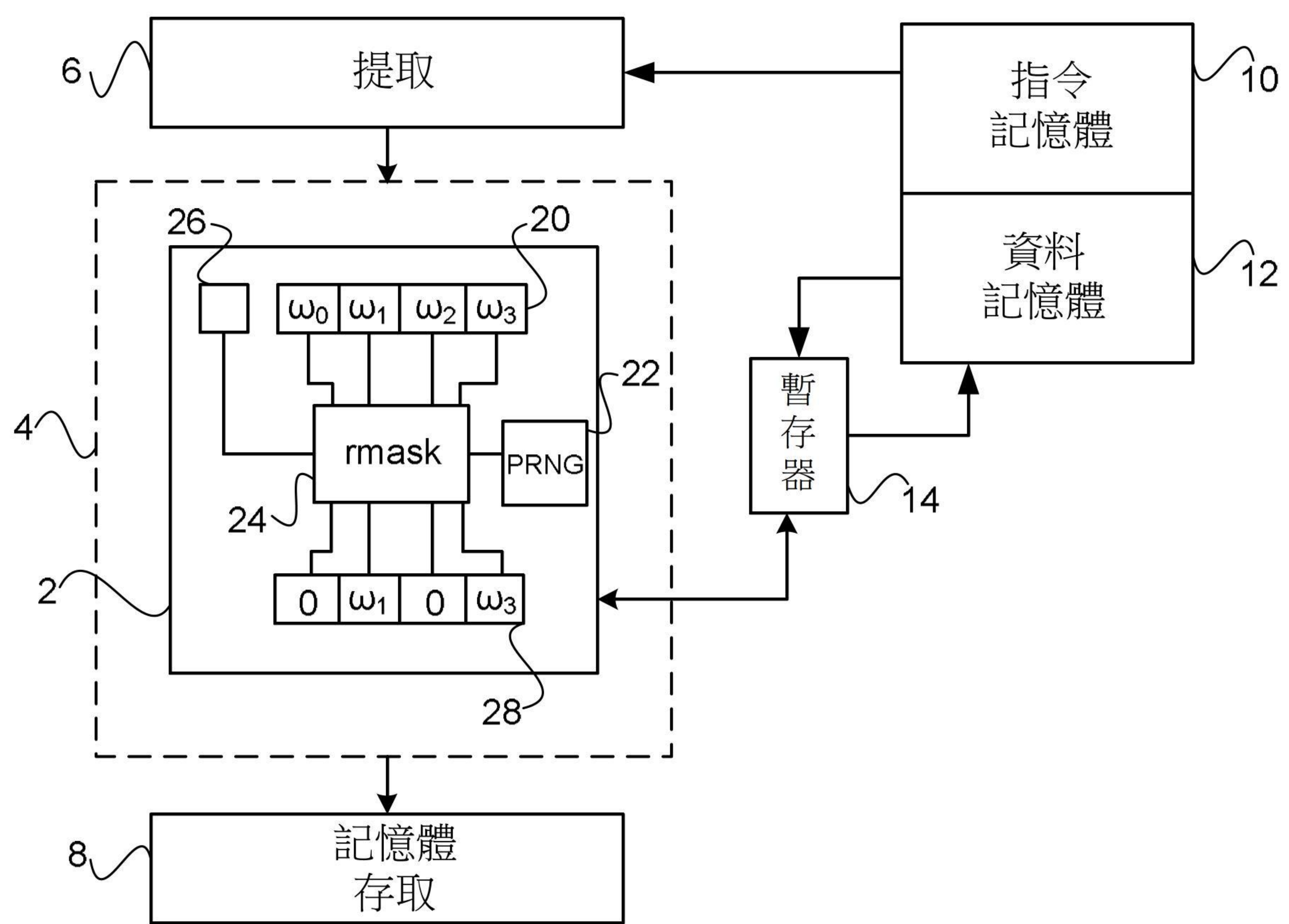
【發明圖式】



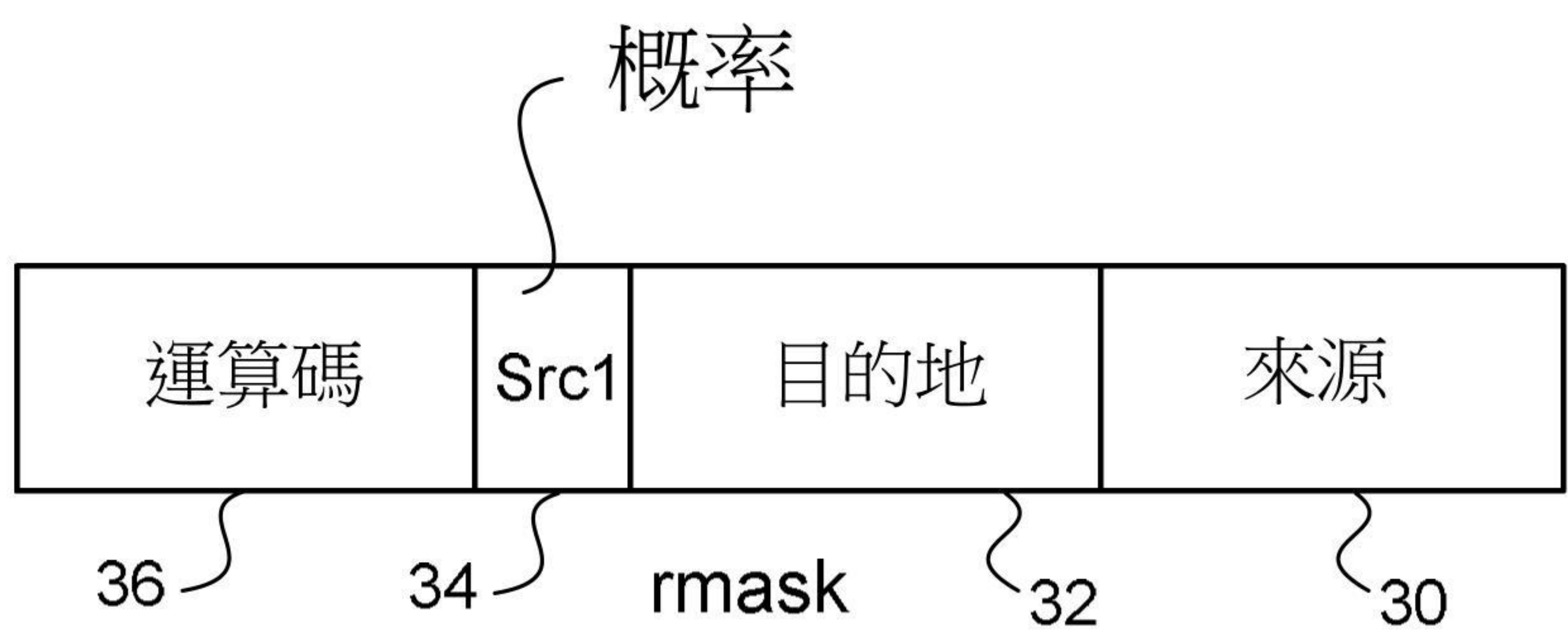
【圖1】



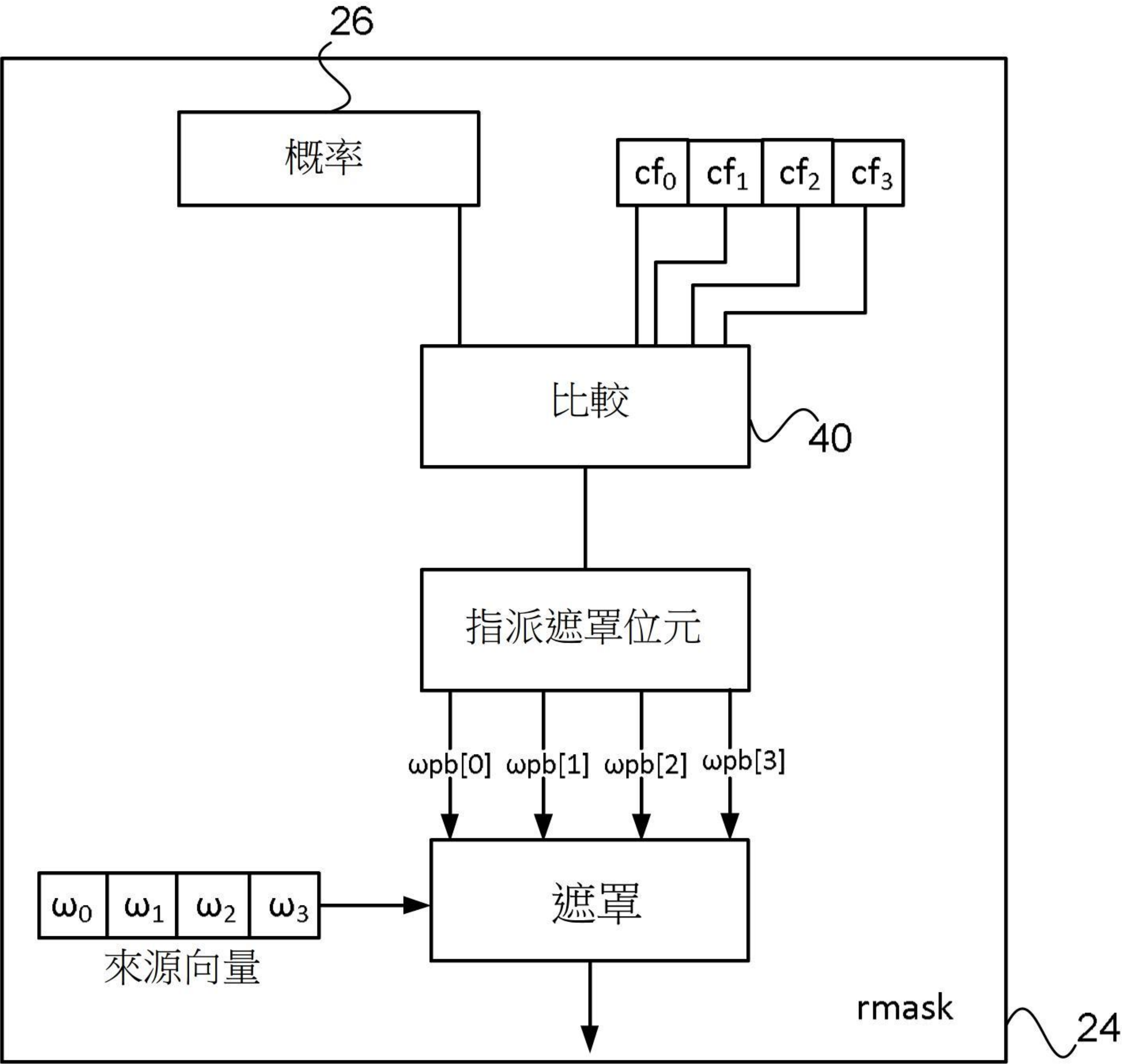
【圖1A】



【圖2】



【圖3】



【圖4】