(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété Intellectuelle
Bureau international

(43) Date de la publication internationale
02 novembre 2017 (02.11.2017)

WIPO | PCT

(10) Numéro de publication internationale
WO 2017/186830 A1

(51) Classification internationale des brevets :
G06N 3/04 (2006.01) *G06N 3/063* (2006.01)

(21) Numéro de la demande internationale :
PCT/EP2017/060017

(22) Date de dépôt international :
27 avril 2017 (27.04.2017)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
1653744 27 avril 2016 (27.04.2016) FR

(71) Déposant : **COMMISSARIAT A L'ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES** [FR/FR] ; 25 rue Leblanc -, Bâtiment "Le Ponant D", 75015 PARIS (FR).

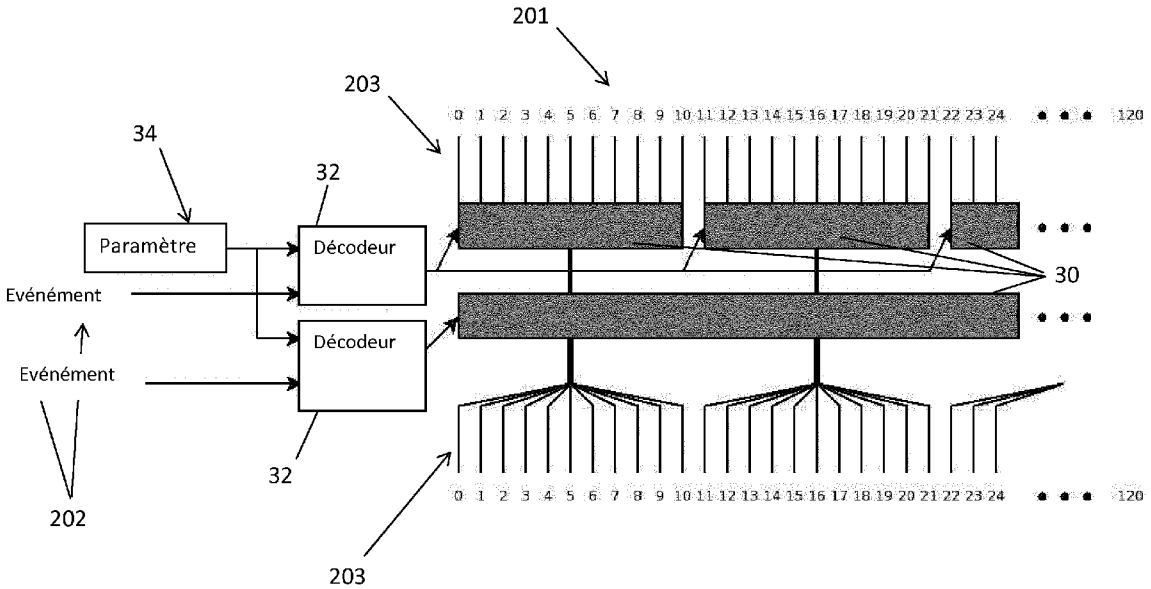
(72) Inventeurs : **BICHLER, Olivier** ; 99 avenue de Paris, 91300 MASSY (FR). **DUPRET, Antoine** ; 28, rue du Parc, 91400 ORSAY (FR). **LORRAIN, Vincent** ; 121 rue de Paris, 91120 PALAISEAU (FR).

(74) Mandataire : **HNICH-GASRI, Naïma** et al. ; Marks & Clerk France, Immeuble "Visium", 22, Avenue Aristide Briand, 94117 ARCUEIL (FR).

(81) États désignés *(sauf indication contraire, pour tout titre de protection nationale disponible)* : AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW,

(54) **Title:** DEVICE AND METHOD FOR DISTRIBUTING CONVOLUTIONAL DATA OF A CONVOLUTIONAL NEURAL NETWORK

(54) **Titre :** DISPOSITIF ET PROCEDE DE DISTRIBUTION DE DONNEES DE CONVOLUTION D'UN RESEAU DE NEURONES CONVOLUTIONNEL



34 Parameter
32 Decoder
202 Event

FIGURE 8

(57) **Abstract:** The invention proposes a device for distributing convolution coefficients of at least one convolution kernel of a convolutional neural network, carried by an input bus (201), to a set of processing units in a calculator based on a convolutional neural network architecture. The device comprises at least one permutation network (30) controlled by at least one control unit (32), the permutation network comprising a set of permutation units (30) arranged to perform circular offsets of at least a portion of the input bus. For each convolution kernel, each control unit is configured to dynamically control at least some of the permutation units of the permutation networks (30) in response to an input event applied to the convolution kernel and at least one parameter representing the maximum size of the convolution kernels.

WO 2017/186830 A1

[Suite sur la page suivante]

KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) États désignés *(sauf indication contraire, pour tout titre de protection régionale disponible)* : ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), eurasien (AM, AZ, BY, KG, KZ, RU, TJ, TM), européen (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Publiée:

— *avec rapport de recherche internationale (Art. 21(3))*

(57) Abrégé : L'invention propose un dispositif pour distribuer des coefficients de convolution d'au moins un noyau de convolution d'un réseau de neurones convolutionnel portés par un bus d'entrée (201) vers un ensemble d'unités de traitement dans un calculateur basé sur une architecture de réseau de neurones convolutionnel. Le dispositif comprend au moins un réseau de permutation (30) pilotée par au moins une unité de contrôle (32), le réseau de permutation comprenant un ensemble de permutateurs (30) agencés pour effectuer des décalages circulaires d'au moins une partie du bus d'entrée. Pour chaque noyau de convolution, chaque unité de contrôle est configurée pour piloter dynamiquement certains au moins des permutateurs des réseaux de permutation (30) en réponse à un événement d'entrée appliqué sur le noyau de convolution et d'au moins un paramètre représentant la taille maximale des noyaux de convolution.

DISPOSITIF ET PROCEDE DE DISTRIBUTION DE DONNEES DE CONVOLUTION D'UN RESEAU DE NEURONES CONVOLUTIONNEL

Domaine technique

La présente invention concerne de manière générale les réseaux de neurones convolutionnels et en particulier un dispositif et un procédé pour distribuer les coefficients d'au moins un noyau de convolution à des unités de calcul dans un calculateur à base d'architecture de réseaux de neurones convolutionnels.

Les réseaux de neurones artificiels constituent des modèles de calculs imitant le fonctionnement des réseaux de neurones biologiques. Les réseaux de neurones artificiels sont constitués principalement de neurones interconnectés entre eux par des synapses qui peuvent être implémentées par des mémoires numériques ou par des composants résistifs dont la conductance varie en fonction de la tension appliquée à leurs bornes.

Les réseaux de neurones convolutionnels correspondent à un modèle particulier de réseau de neurones artificiels. Les réseaux de neurones convolutionnels ont été décrits initialement dans l'article de K. Fukushima, « Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980. ISSN 0340-1200. doi: 10.1007/BF00344251 ». Les réseaux de neurones convolutionnels (désignés en langue anglo-saxonne par les expressions "convolutional neural networks", ou "deep (convolutional) neural networks" ou encore "ConvNets") sont des réseaux de neurones sans rétro-action (« feedforward »), inspirés par les systèmes visuels biologiques.

Les réseaux de neurones convolutionnels (CNN) sont utilisés dans différents domaines de traitement du signal (visuel, sonore, ou autre), comme par exemple dans le domaine de la classification d'image. Cependant, le fonctionnement intrinsèquement parallèle et la complexité des classificateurs classiques de type réseau de neurones convolutionnels mettent un frein à leur implémentation efficace dans les systèmes embarqués.

Des solutions ont été proposées pour implémenter les réseaux de neurones sur processeur graphique GPU afin d'améliorer significativement leurs performances, comme par exemple la solution décrite dans l'article de D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, « Flexible, high performance convolutional neural networks for image classification. Proceedings of the Twenty-Second International Joint Conference on Artificial

Intelligence - Volume Two», IJCAI' 11, pages 1237–1242, 2011. ISBN 978-1-57735-514-4. doi: 10.5591/978-1-57735-516-8/ IJCAI11-210.

Plusieurs implémentations matérielles de réseau convolutionnel impulsionnel ont notamment été proposées dans un premier temps L. Camunas-Mesa, C. Zamarreno-Ramos, A. Linares-Barranco, A. Acosta-Jimenez, T. Serrano- Gotarredona, and B. Linares-Barranco. An event-driven multi-kernel convolution processor module for event-driven vision sensors. *Solid-State Circuits, IEEE Journal of*, 47(2):504–517, 2012. ISSN 0018-9200. doi: 10.1109/JSSC.2011.2167409. Une telle implémentation de convolution utilise une mémoire numérique séparée pour stocker les coefficients des noyaux de convolution et nécessite de copier ces coefficients du noyau de la mémoire à l'unité de calcul à chaque arrivée d'impulsion.

Les implémentations matérielles de réseau convolutionnel impulsionnel existantes reposent sur l'utilisation d'un calculateur pour calculer les couches de convolution du réseau de neurones convolutionnel, le calculateur comprenant au moins une unité de traitement (tel qu'un processeur) et des mémoires numériques stockant les données.

Dans certaines structures de calculateur et pour certaines applications, les unités de traitement doivent accéder à un jeu de données en parallèle. Cependant, un tel accès aux données pose des problèmes de routage et de concurrence d'accès aux données dus à la lecture parallèle.

Il est connu de résoudre ce problème d'accès en utilisant des réseaux de permutation. Les réseaux de permutation sont des structures de distribution de données et de communication parallèle. Il existe des réseaux de permutation connus appelés réseaux MINs (acronyme pour « Multistage Interconnection Networks » signifiant Réseaux d'Interconnexion Multi-étages), tels que les réseaux « Butterfly Network » (réseau Papillon), « Omega Network » (réseau Omega), « Baseline Network » (réseau de base) et le « Cube Network » (réseau Cube). De tels réseaux sont utilisés pour connecter N entrées vers N sorties avec plusieurs étages de permutation. Leurs complexités en nombre de permutateurs est de (N/2)*log2(N), ce qui les rend très compacts. Les réseaux « Butterfly », « Baseline » et « Cube » ont un nombre d'entrées/sorties en puissance de deux tandis que le réseau Omega a un nombre d'entrées/sorties qui est un multiple de deux. Ces réseaux sont utilisés dans différentes architectures telles que les architectures de calcul parallèle, les commutateurs, etc. Les réseaux MIMs ont été implémentés dans plusieurs calculateurs parallèles.

Dans les solutions existantes, un réseau de type MIM peut être instancié avec deux types de permutateurs, soit à deux états, soit à quatre états. Les permutateurs à deux états sont contrôlés avec un bit. Dans un état dit « décroisé » des permutateurs, les entrées ne sont pas permutées. Dans un état dit « croisé », les entrées sont permutées. Les permutateurs à quatre états ont également les deux états « décroisé » et « croisé » mais comprennent deux états supplémentaires appelés « état de regroupement » et « état de dégroupement ».

Dans le cas des permutateurs à quatre états, les réseaux MIMs classiques peuvent effectuer toutes les connexions N vers N. Cependant, ce type de permutateur n'est pas compatible avec des signaux analogiques. En effet, les fonctions de regroupement et dégroupement nécessitent que la donnée transporte des informations de routage avec elle, ce qui n'est pas possible pour une donnée analogique codée en tension.

En particulier, dans un réseau Butterfly, les premiers nœuds sont des structures de permutation 4 vers 4, les nœuds de profondeurs supérieurs sont des réseaux 8 vers 8, etc.

Les réseaux MIMs sont compacts et performants. Toutefois, le nombre d'entrées n'est pas libre, ce qui implique une réduction de la flexibilité ou/et l'utilisation d'une mémoire plus grande que nécessaire. Par ailleurs, ce type de réseaux ne peut effectuer qu'un nombre de décalages circulaires limité en utilisant des permutateurs deux états. Les réseaux MIMs ne peuvent donc pas être utilisé dans des calculateurs à base d'architecture de réseau de neurones convolutionnels pour permettre l'accès aux données en parallèle

Définition générale de l'invention

L'invention vient améliorer la situation en proposant un dispositif pour distribuer des coefficients de convolution d'au moins un noyau de convolution d'un réseau de neurones convolutionnel portés par un bus d'entrée vers un ensemble d'unités de traitement dans un calculateur basé sur une architecture de réseau de neurones convolutionnel, caractérisé en ce que le dispositif comprend au moins un réseau de permutation pilotée par au moins une unité de contrôle, le réseau de permutation comprenant un ensemble de permutateurs agencés pour effectuer des décalages circulaires d'au moins une partie du bus d'entrée. Pour chaque noyau de convolution, chaque unité de contrôle est configurée pour piloter dynamiquement certains au moins des permutateurs des réseaux de permutation en réponse à un événement d'entrée appliqué sur le noyau de convolution et d'au moins un paramètre représentant la taille maximale des noyaux de convolution.

L'invention propose en outre un calculateur neuromorphique comprenant une mémoire pour stocker les coefficients de noyaux de convolution et un ensemble d'unités de traitement pour calculer la réponse d'un réseau de neurones à un événement d'entrée, le calculateur comprenant un dispositif de routage selon l'une des caractéristiques précédentes pour distribuer les coefficients aux unités de traitement.

Les modes de réalisation de l'invention permettent ainsi à un ensemble d'unités de traitement d'un calculateur à base de réseau de neurones convolutionnel d'accéder à des données stockées en mémoire en parallèle.

Brève description des dessins

D'autres caractéristiques et avantages de l'invention apparaîtront à l'aide de la description qui suit et des figures des dessins annexés dans lesquels:

- La figure 1 représente un exemple de réseau convolutionnel ;
- La figure 2 est un schéma montrant une couche de convolution constituée de plusieurs cartes de sortie ;
- La figure 3 illustre le principe de fonctionnement d'une couche de convolution dans un réseau de neurones convolutionnel ;
- La figure 4 montre un exemple de calculateur à base de réseau de neurones convolutionnel dans lequel le dispositif de routage de données peut être implémenté, selon certains modes de réalisation ;
- La figure 5 illustre la distribution des coefficients de poids en fonction d'un événement en entrée ;
- La figure 6 montre la structure d'un module de calcul du calculateur à base de réseau de neurones convolutionnel dans lequel le dispositif de routage de données peut être implémenté, selon certains modes de réalisation ;
- La figure 7 montre les entrées/sorties du dispositif de routage de données, selon certains modes de réalisation ;
- La figure 8 montre la structure du dispositif de routage de données de convolution, selon certains modes de réalisation ;
- La figure 9 illustre la structure du dispositif de routage configurable dynamiquement à partir d'une topologie d'arbre de décalage, selon certains modes de réalisation ;

- La figure 10 montre un exemple de mise en œuvre du dispositif de routage de données de convolution pour un filtre maximal de 5*5*1, selon certains modes de réalisation ;
-La figure 11 est un schéma représentant un exemple de permutateurs à deux états pouvant être utilisé dans des réseaux de permutation, selon un exemple de réalisation ;
-La figure 12 est un schéma illustrant des permutateurs à deux états répartis en strates sur les lignes de bus, selon un exemple de réalisation ;
-La figure 13 est un schéma illustrant des strates de permutateurs de différents degrés, selon un autre exemple de réalisation ;
-La figure 14 montre un exemple de réseau de permutation pour un bus de 5 lignes avec des couches de permutateur en 2 puissance n ;
- La figure 15 montre un exemple d'arbre de décalage pour un filtre maximal de 5*5*1;
-La figure 16 est un schéma du dispositif de routage pour une multi-convolution, selon un exemple de réalisation ;
- La figure 17 est un organigramme représentant le procédé de configuration selon certains modes de réalisation ;
- La figure 18 illustre le procédé de configuration pour un décalage de 5 lignes sur un bus de 5 lignes pour les 5 décalages possibles sur un réseau de permutation en couche 2 puissance n ;
- La figure 19 représente un exemple de permutateur à deux états numériques, selon certains modes de réalisation ; et
- La figure 20 représente un exemple de réseau de permutation pour un bus de 11 lignes avec des couches en 2 puissance n.

Les dessins comprennent, pour l'essentiel, des éléments de caractère certain. Ils pourront donc non seulement servir à mieux faire comprendre la description, mais aussi contribuer à la définition de l'invention, le cas échéant.

Description détaillée

Un réseau de neurones artificiel (encore appelé réseau de neurones « formel » ou désigné simplement par l'expression « réseau de neurones » ci-après) est constitué d'une ou plusieurs couches de neurones, interconnectées entre elles. Chaque couche est constituée d'un ensemble de neurones, qui sont connectés à une ou plusieurs couches précédentes. Chaque neurone d'une couche peut être connecté à un ou plusieurs neurones d'une ou plusieurs couches précédentes. La dernière couche du réseau est appelée « couche de sortie ».

Les neurones sont connectés entre eux par des synapses, ou poids synaptiques (encore appelés « coefficients de poids» ou « pondérations » ou « coefficients de convolution »), qui pondèrent l'efficacité de la connexion entre les neurones. Les poids synaptiques constituent des paramètres réglables du réseau de neurones et stockent l'information comprise dans le réseau de neurones. Les poids synaptiques peuvent être positifs ou négatifs.

Les réseaux de neurones dit « convolutionnels » (ou encore « convolutional », « deep convolutional », « convnets ») sont en outre composés de couches de type particulier qui peuvent comprendre des couches de convolution, des couches de regroupement (« pooling » en langue anglo-saxonne) et des couches complètement connectés (« fully connected »).

Un réseau de neurones est constitué d'une ou plusieurs couches de convolution, pouvant inclure des couches de regroupement (« pooling » en langue anglo-saxonne). Les couches de convolution peuvent être suivies par un classifieur de type perceptron multicouches.

La sortie d'une couche de convolution peut être connectée à l'entrée de la couche suivante ou à la sortie de la couche suivante. Elle peut également reboucler sur l'entrée ou être connectée en sortie à d'autres types de couches qui ne sont pas des couches de convolution.

Dans une couche de convolution donnée, chaque neurone est connecté à une sous-matrice de la matrice d'entrée. Les sous-matrices ont la même taille. Elles sont décalées les unes des autres de manière régulière et peuvent se chevaucher. La matrice d'entrée peut être de dimension quelconque. Cependant, la matrice d'entrée est généralement de dimension 2D lorsque les données à traiter sont des données visuelles, les deux dimensions correspondant alors aux dimensions spatiales X et Y d'une image 3D si image en couleur.

La figure 1 représente un exemple de réseau convolutionnel simplifié, avec une couche d'entrée « env » correspondant à la matrice d'entrée, deux couches de convolution, « conv1 » et « conv2 », ainsi que deux couches complètement connectés « fc1 » et « fc2 ». Dans cet exemple, la taille des noyaux de convolution est de 5×5 pixels et ils sont décalés entre eux de 2 pixels (soit un décalage ou « stride » de 2) :

- « conv1 » a une matrice d'entrée « env » et 6 noyaux de convolution différents produisant 6 cartes de sorties ;
- « conv2 » a 12 noyaux de convolution différents et donc 12 cartes de sorties, et chaque carte de sortie prend en entrée l'ensemble des 6 cartes de sortie de la couche précédente.

Dans un réseau de neurones, les neurones sont connectés à leur sous-matrice d'entrée *I* par les synapses dont le poids est réglable. La matrice *K* des poids synaptiques appliquée aux sous-matrices d'entrée des neurones est la même pour tous les neurones d'une même carte de sortie ("feature map" en langue anglo-saxone). Une telle matrice K est encore appelée « noyau de convolution ». Le noyau de convolution est ainsi partagé pour l'ensemble des neurones d'une même carte de sortie O, et est donc appliqué sur l'ensemble de la matrice d'entrée, ce qui diminue la mémoire nécessaire pour le stockage des coefficients, ce qui optimise les performances. Par exemple pour la reconnaissance d'images, cela permet de minimiser le nombre de filtres ou de représentations intermédiaires qui codent au mieux les caractéristiques de l'image et qui sont réutilisables sur toute l'image. Les coefficients d'un noyau de convolution K (c'est-à-dire les poids synaptiques) peuvent correspondre à des filtres classiques de traitement du signal (Gaussien, Gabor, Laplace...), ou être déterminés par apprentissage, supervisé ou non-supervisé, par exemple en utilisant l'algorithme de rétro-propagation du gradient utilisé dans les réseaux de neurones de type perceptrons multi-couches. Les coefficients des noyaux de convolution peuvent être positifs ou négatifs, et sont généralement normalisés entre -1 et 1, tout comme les valeurs d'entrée et de sortie des neurones.

Les réseaux de neurones peuvent être transposés en codage impulsionnel (« spike » en langue anglo-saxonne). Dans ce cas, les signaux propagés en entrée et en sortie des couches du réseau ne sont plus des valeurs numériques, mais des impulsions électriques (assimilables à des impulsions Dirac). L'information qui était codée dans la valeur des signaux (normalisé entre -1 et 1) est alors codée temporellement avec l'ordre d'arrivée des impulsions (codage par rang, ou « rank order coding ») ou avec la fréquence des impulsions.

Dans le cas d'un codage par rang, l'instant d'arrivée de l'impulsion est inversement proportionnel à la valeur absolue du signal à coder. Le signe de l'impulsion détermine alors le signe de la valeur du signal à coder. Dans le cas d'un codage fréquentielle, la fréquence des impulsions, comprise entre f_{min} et f_{max}, est proportionnelle à la valeur absolue du signal à coder. Le signe de l'impulsion détermine le signe de la valeur du signal à coder. Par exemple, en considérant une matrice d'entrée du réseau correspondant à la composante de luminance d'une image, normalisée entre 0 et 1, un pixel (ou coefficient de la matrice) blanc, codé par une valeur 1, émettra des impulsions à une fréquence f_{max}, un pixel noir, codé par une valeur 0, émettra des impulsions à une fréquence f_{min}, tandis qu'un pixel gris, codé par une valeur x, émettra des impulsions à une fréquence $f = f_{min} + x(f_{max} - f_{min})$. Le codage peut également être pseudo-fréquentiel, par exemple poissonien : dans ce cas f_{max} et f_{min}

représentent des fréquences moyennes uniquement. La phase initiale des impulsions peut être aléatoire.

Les impulsions peuvent également provenir directement d'un capteur, tel qu'une rétine ou une cochlée artificielle, imitant le fonctionnement de leur équivalent biologique.

Un neurone est défini par une fonction algébrique non linéaire, paramétrée, à valeurs bornées, et ayant des variables réelles appelées « entrée » en fonction du modèle de neurone utilisé. Un neurone est en outre caractérisé par une fonction d'activation g(), un seuil et les poids synaptiques.

Le modèle du neurone est défini par une fonction algébrique non linéaire. Cette fonction peut prendre en argument la valeur de l'intégration (représentant la valeur interne à un neurone) mais aussi, selon les modèles, le temps ou la sortie d'un compteur interne. Telle qu'utilisée ici, l'expression « intégration » désigne l'intégrale selon le temps des trains d'impulsions (« spike » en langue anglo-saxone) pondérées en entrée du neurone (i.e. intégrale temporelle d'un train d'impulsions pondérées (peigne de Dirac par exemple)). Cette intégration peut être remise à zéro lorsque le neurone se déclenche (i.e. produit une impulsion en sortie).

Un neurone réalise dans un premier temps la somme pondérée *h* des coefficients de sa sous-matrice d'entrée par le noyau de convolution (autrement dit le produit scalaire entre la sous-matrice d'entrée I et la matrice K), en appliquant une fonction d'agrégation g = <I,W>. La sortie du neurone correspond à la valeur de la fonction d'activation g du neurone appliquée à cette somme : g(h). Classiquement, g peut prendre la forme d'une fonction sigmoïde, typiquement la fonction tangente hyperbolique.

Dans les formes de réalisations de réseau de neurones impulsionnel avec codage fréquentiel, le calcul de la somme pondérée *h* se fait par accumulation du coefficient du noyau de convolution 14 à chaque arrivée d'une impulsion sur l'entrée correspondante. La fonction d'activation du neurone *g* peut dans ce cas être remplacée par un seuil. Lorsque la valeur absolue de *h* dépasse le seuil suite à l'arrivée d'une impulsion sur la sous-matrice d'entrée, le neurone de sortie émet une impulsion du signe de *h* et remet la somme pondérée *h* à la valeur 0. Le neurone entre alors dans une période dite « réfractaire » durant laquelle il ne peut plus émettre d'impulsion durant une période fixée. Les impulsions peuvent par conséquent être positives ou négatives, selon le signe de *h* au moment du dépassement du seuil. Une impulsion négative en entrée inverse le signe du coefficient du noyau correspondant pour l'accumulation.

Une couche de convolution peut contenir un ou plusieurs noyaux de convolution qui ont chacun une matrice d'entrée (la matrice d'entrée des différents noyaux de convolution peut être la même) mais qui ont des coefficients de convolution différents correspondant à des filtres différents.

Comme illustré sur la figure 2, une couche de convolution ou de « pooling » peut être constituée d'une ou plusieurs matrices de sortie 14 (encore appelées « cartes de sortie » ou « output feature map » en langue anglo-saxonne), chaque carte de sortie pouvant être connectée à une ou plusieurs matrices d'entrée 11 (encore appelées « cartes d'entrée »).

Chaque noyau de convolution 12 dans une couche de convolution donnée produit une carte de sortie 14 différente de sorte que les neurones de sortie sont différents pour chaque noyau. Les réseaux convolutionnels peuvent également inclure des couches de "pooling" locales ou globales qui combinent les sorties de groupe de neurones d'une ou plusieurs cartes de sortie. La combinaison des sorties peut consister par exemple à prendre la valeur maximale ou moyenne des sorties du groupe de neurones, pour la sortie correspondante, sur la carte de sortie de la couche de « pooling ». Les couches de « pooling » permettent de réduire la taille des cartes de sorties d'une couche à l'autre dans le réseau, tout en améliorant ses performances en le rendant plus tolérant à de petites déformations ou translations dans les données d'entrée.

Les réseaux convolutionnels peuvent également inclure des couches complètement connectés de type perceptron.

Comme illustré sur la figure 3, une matrice de sortie (14), notée O, comprend des coefficients $O_{i,j}$ et a une taille notée (O_h, O_w). Cette matrice correspond à une matrice de neurones et les coefficients $O_{i,j}$ correspondent aux valeurs de sortie de ces neurones, calculée à partir des entrées et des poids synaptiques.

Une matrice ou carte d'entrée 11 peut correspondre à une carte de sortie d'une couche précédente, ou à une matrice d'entrée du réseau qui reçoit des stimuli ou une partie des stimuli à traiter. Un réseau de neurones peut être constitué d'une ou plusieurs matrices d'entrée 11. Il peut s'agir par exemple des composantes RGB, HSV, YUV ou de toute autre composante classique d'une image, avec une matrice par composante. Une matrice d'entrée notée I comprend des coefficients $I_{i,j}$, et a une taille notée (I_h, I_w).

Une carte de sortie O est connectée à une matrice d'entrée I par une opération de convolution, via un noyau de convolution 12 noté K (le noyau de convolution est encore appelé filtre, ou matrice de convolution), de taille (n, m) et comprenant des coefficients $K_{k,l}$. Chaque neurone de la carte de sortie 14 est connecté à une partie de la matrice d'entrée 11, cette partie étant encore appelée « sous-matrice d'entrée » ou « champ récepteur du neurone » et étant de même taille que la matrice de convolution K. La matrice de convolution K comprenant les poids synaptiques est commune pour l'ensemble des neurones de la carte de sortie 0 (les poids de la matrice K sont alors dits « poids partagés). Chaque coefficient de sortie de la matrice de sortie $O_{i,j}$ satisfait alors la formule suivante :

$$O_{i,j} = \mathrm{g}\left(\sum_{k=0}^{\min(n-1, I_h - i.s_i)} \sum_{l=0}^{\min(m-1, I_w - j.s_j)} I_{i.s_i+k, j.s_j+l}.K_{k,l}\right)$$

Dans la formule ci-dessus, $g()$ désigne la fonction d'activation du neurone, tandis que s_i et s_j désignent les paramètres de décalage (« stride » en langue anglo-saxone) selon deux dimensions, notamment dans une dimension verticale et dans une dimension horizontale respectivement. Un tel décalage « stride » correspond au décalage entre chaque application du noyau de convolution sur la matrice d'entrée. Par exemple, si le décalage est supérieur ou égal à la taille du noyau, alors il n'y a pas de chevauchement entre chaque application du noyau.

Une carte de sortie O est connectée à une matrice d'entrée I par une opération de « pooling » qui réalise un sous-échantillonnage de la matrice d'entrée, ce qui fournit une matrice sous-échantillonnée. Le sous-échantillonnage peut être de deux types :

- Un type de sous-échantillonnage dit « MAX pooling » (regroupement maximal) selon l'équation ci-dessous:

$$O_{i,j} = \mathrm{g}\left(max_{k=0}^{\min(n-1, I_h - i.s_i)} max_{l=0}^{\min(m-1, I_w - j.s_j)} I_{i.s_i+k, j.s_j+l}\right)$$

- Un type d'échantillonnage dit « AVERAGE pooling » (regroupement moyen) selon l'équation ci-dessous :

$$O_{i,j} = \mathrm{g}\left(\frac{1}{n.m} \sum_{k=0}^{\min(n-1, I_h - i.s_i)} \sum_{l=0}^{\min(m-1, I_w - j.s_j)} I_{i.s_i+k, j.s_j+l}\right)$$

Les poids synaptiques associés aux connexions dans le cas d'une couche de « pooling » sont généralement unitaires et n'apparaissent donc pas dans les formules ci-dessus.

Une couche complètement connectée comprend un ensemble de neurones, chaque neurone étant connecté à toutes les entrées de la couche. Chaque neurone O_j dispose de ses propres poids synaptiques $W_{i,j}$ avec les entrées I_i correspondantes et effectue la somme pondérée des coefficients d'entrée avec les poids qui est ensuite passée à la fonction d'activation du neurone pour obtenir la sortie du neurone.

$$O_j = \mathrm{g}\left(\sum_i I_i . W_{i,j}\right)$$

La fonction d'activation des neurones $\mathrm{g}()$ est généralement une fonction sigmoïde, comme par exemple la fonction $tanh()$. Pour les couches de « pooling », la fonction d'activation peut être par exemple la fonction Identité.

La figure 4 représente schématiquement un exemple de calculateur 100 à base d'architecture de réseau de neurones convolutionnel dans lequel peut être implémenté un dispositif de distribution de données d'au moins un noyau de convolution, selon certains modes de réalisation.

Le calculateur 100 (encore appelé « calculateur neuromorphique ») est configuré pour calculer au moins une couche de convolution d'un réseau de neurones convolutionnel. Le calculateur peut comprendre au moins un module de convolution 10 (encore appelé « bloc de convolution » ou « module de calcul de convolution ») configuré pour calculer chaque couche de convolution, chaque couche de convolution étant constituée d'une ou plusieurs cartes de sortie du réseau de neurones convolutionnel. Le calcul d'une couche de convolution consiste à calculer la valeur interne (encore appelée « valeur d'intégration ») des neurones qui ont reçu un événement d'entrée (tel qu'une impulsion), dits neurones « déclenchés » ou « activés » et le calcul de la valeur de sortie de ces neurones. Le calcul d'une couche de convolution consiste ainsi à déterminer la réponse des neurones de la couche de convolution à un événement d'entrée (tel qu'une « impulsion »). Dans les modes de réalisation où le calculateur comprend plusieurs modules de calcul de convolution 10, les modules de convolution 10 peuvent être interconnectés par un système d'interconnexion 101, par exemple en utilisant un système de réseaux sur puce (NOC), des systèmes d'interconnexions programmables (e.g. de type FPGA), des systèmes d'interconnexions à routage fixe, etc. Le système d'interconnexion 101 permet de rediriger les événements entre

les modules et/ou les entrées/sortie et assure la connexion entre les différentes couches du réseau de neurone.

Chaque module de convolution 10 effectue une opération de convolution sur une ou une partie d'une couche de convolution du réseau de neurone. Les modules de convolution 10 peuvent être utilisés pour calculer des couches de convolution différentes. En variante, chaque module de convolution 10 peut être utilisé pour réaliser plusieurs filtres dans une couche de convolution donnée. Dans certains modes de réalisation, lorsqu'un module de convolution 10 n'est pas suffisant pour calculer une couche de convolution donnée, plusieurs modules de convolution peuvent être utilisés pour calculer la couche de convolution.

Dans certaines applications de l'invention, le calculateur 100 peut être un calculateur multi-cœur à mémoires distribuées, chaque cœur pouvant être interconnecté par un système d'interconnexion, chaque module de calcul 10 formant un cœur de calcul qui peut être utilisé pour calculer une ou des opérations de convolutions. Les différentes couches de convolution du réseau de neurones convolutionnel sont distribuées sur les différents modules de calcul 10.

La figure 5 représente la distribution des coefficients de poids en fonction d'un événement en entrée, pour un réseau de neurones convolutionnel et impulsionnel.

Chaque unité de traitement peut être configurée pour calculer la valeur de neurones indépendants déclenchés par des événements d'entrée, en fonction du coefficient de poids associé à l'unité de calcul pour chaque événement d'entrée.

Dans un mode de réalisation de réseau de neurones à impulsions, l'événement d'entrée est représenté par une impulsion arrivant sur le réseau. Un réseau de neurones à impulsions peut recevoir dans le temps un train d'impulsions, chaque impulsion pouvant déclencher un ou plusieurs neurones.

L'événement d'entrée peut alors être défini par une adresse d'entrée sur la couche de convolution. La suite de la description sera faite en référence à un réseau de neurones convolutionnel à impulsions pour faciliter la compréhension de l'invention.

Telle qu'utilisée ici, « l'adresse d'entrée » d'une impulsion représente l'adresse de l'impulsion émise par la couche précédente (impulsions d'entrée) et reçue par la couche considérée. Dans le système, les impulsions peuvent être transmises par un bus série. Les impulsions

propagent donc avec elles au moins leur adresse d'émission. L'adresse d'entrée dans le repère choisi peut comprendre les coordonnées de l'événement d'entrée sur la carte d'entrée.

En réponse à l'émission d'un événement sous la forme d'une impulsion électrique encore appelée « potentiel d'action » (« spike » en langue anglo-saxonne) par un neurone d'entrée (neurone pré-synaptique), l'impulsion traverse les synapses, ce qui induit une impulsion sur un nombre fini de neurones pst-synaptiques qui sont alors sollicités. La connexion entre les neurones de sortie et les neurones d'entrée sur la carte d'entrée fait que cette impulsion n'est pas reçue par tous les neurones de cette couche. Une telle propriété implique l'existence de neurones indépendants dans une couche du réseau de neurones 10 soumise à une impulsion. Deux neurones sont dits indépendants si, pour un événement d'entrée donnée, ils ne peuvent pas se déclencher en même temps. Ainsi, pour une impulsion entrante donnée, il existe des neurones qui ne peuvent pas recevoir cette impulsion, en même temps (neurones dits indépendants). Selon certains modes de réalisation, chaque module de calcul 10 peut être configuré pour mutualiser certaines fonctions pour des groupes de neurones indépendants d'une même couche.

La figure 6 illustre la structure d'un module de convolution 10, selon certains modes de réalisation. Le module de convolution 10 comprend une ou plusieurs unités de traitement 20 (encore appelées « unité de traitement élémentaires » ou « unité de calcul»), tels que des processeurs élémentaires, configurées pour calculer la réponse des neurones d'une couche de convolution à un événement d'entrée (tel qu'une « impulsion »). En réponse à l'événement d'entrée, le module de convolution 10 est configuré pour distribuer les coefficients du noyau de convolution 12 à certaines au moins des unités de traitements élémentaire 20 en parallèle. Chaque module de convolution 10 comprend une mémoire de poids synaptiques 21 dans laquelle sont stockés les coefficients des noyaux de convolution (14) et un dispositif de distribution de données 200 (encore appelé « dispositif de routage » ou « dispositif d'acheminement de données ») selon certains modes de réalisation de l'invention.

La mémoire de coefficient de poids 21 est configurée pour stocker les poids de convolution et est accessible en lecture parallèle par les différentes unités de traitement 20.

Dans une couche de convolution donnée, les neurones partagent les mêmes coefficients de poids (valeur du filtre de convolution). Dans chaque module de calcul 10, la mémoire des coefficients de poids 21 est distribuée parallèlement aux unités de traitement en fonction des

événements d'entrées. Ainsi, la mémoire des coefficients de poids 21 n'est pas redondante. Une telle répartition mémoire permet d'optimiser l'espace mémoire.

Pour chaque événement d'entrée appliqué à un noyau de convolution, chaque unité de traitement 20 peut être associée à un coefficient de poids du noyau de convolution qui est distribué par le dispositif de distribution de données 200 après décalage circulaire des données du bus d'entrée 201 qui porte les coefficients de la mémoire 21.

La suite de la description sera faite en référence à un réseau de neurones convolutionnel et impulsionnel. Dans un tel réseau, l'événement d'entrée est une impulsion arrivant sur une couche de convolution. Cette impulsion est émise par la couche de convolution précédente (impulsion d'entrée) et est reçue par la couche de convolution considérée. L'impulsion peut être définie par une adresse d'entrée, c'est-à-dire l'adresse de l'impulsion émise. Les impulsions peuvent être transmises par un bus série et propagent alors avec elles au moins leur adresse.

Les unités de traitement 20 effectuent le calcul de la réponse d'une couche convolutionnelle à l'impulsion en entrée à partir des coefficients de poids distribués en parallèle par le dispositif de distribution de données 200. Les unités de traitement élémentaires 20 génèrent alors les événements de sortie en fonction d'un événement d'entrée pour chaque neurone d'entrée déclenché. Les unités de traitement 20 peuvent dépendre d'un modèle de neurones choisi (encore appelé « modèle computationnel de neurone »). Le modèle du neurone peut être défini lors de la programmation du système. Dans certains modes de réalisation, le modèle de neurone est identique pour tous les neurones d'une couche.

Un modèle de neurone de type impulsionnel génère une intégration temporelle de l'information. La réponse d'un neurone à une stimulation est le temps d'émission de la prochaine impulsion. Une impulsion est émise au moment où l'excitation du neurone dépasse un seuil. Une horloge virtuelle peut être utilisée pour temporiser les émissions des impulsions des neurones qui peuvent alors être considérées comme des événements datés. Il existe plusieurs types de modèles de neurone impulsionnel comme par exemple :

- Le modèle de Hodgkin & Huxley (HH), défini par quatre équations différentielles inter-dépendantes ;
- Le modèle Integrate & Fire (IF) décrit par une équation différentielle unique ;
- Des modèles liés au modèle IF tels que le modèle Leaky Integrate & Fire (LIF) ou le modèle Quadratic Integrate&Fire (QIF), le modèle à conductance Integrate & Fire (gIF) ;

- Le modèle Spike Response Model (SRM) basé sur le comportement du neurone (modélisation phénoménologique du neurone) et sur une fonction noyau ;
- Le modèle d'Izhikevich (Izhikevich, 2003) qui utilise deux équations différentielles.

Chaque unité de traitement 20 (encore appelée « sous-calculateur du modèle neuromorphique ») peut représenter un ou plusieurs neurones. Chaque unité de traitement de données élémentaire 20 est configurée pour pouvoir générer une valeur de sortie représentant la valeur en sortie de chaque neurone qu'elle contient qui résulte de l'intégration des données reçues par le neurone. Cette somme temporelle (représentant une intégration temporelle) est déterminée en utilisant la fonction d'activation (encore appelée fonction de transition ou fonction de transfert) qui dépend du modèle de neurone.

Chaque module de convolution 10 comprend une unité de gestion de sortie 22 pour délivrer en sortie les valeurs calculées par les unités de traitement élémentaire. Cette unité 22 peut être configurée pour sérialiser les sorties des unités de traitement 20 et/ou pour créer une fonction de regroupement (« pooling » en langue anglo-saxone). L'unité de gestion de sorties 20 peut comprendre un système de concurrence pour gérer la concurrence entre les événements de sortie en fonction de règles de concurrence prédéfinies et une unité de sérialisation configurée pour sérialiser les événements de sortie. L'unité de gestion de sortie 22 peut recevoir en entrée un bus parallèle de données et délivrer ces données en sortie, une par une, sur un bus série, selon un ordre de priorité prédéfini, dans certains modes de réalisation.

La figure 7 représente les entrées et les sorties du dispositif de routage 200, selon certains modes de réalisation. dans le doivent lire parallèlement des valeurs du filtre en fonction des coordonnées de l'impulsion en entrée du système. Le dispositif de routage 200 est configuré pour contrôler la distribution des coefficients de poids aux unités de traitement 20 pour le calcul d'une couche de convolution neuromorphique et permettre la lecture parallèle du filtre de convolution par les unités de traitements 20.

En particulier, dans un réseau de neurones convolutionnel à impulsion, les unités de traitement 20 doivent pouvoir lire en parallèle des valeurs du filtre en fonction des coordonnées de l'impulsion en entrée du calculateur 100, alors que les unités de traitement ne lisent pas les mêmes données en même temps.

Le dispositif de routage 200 permet un acheminement dynamique de ces données vers les unités de traitements 20 qui permet une lecture parallèle des données de la mémoire 2

(coefficients de filtres). Comme montré sur la figure 7, le dispositif de routage 20 reçoit les données de la mémoire 2 sous la forme d'un bus d'entrée 201 depuis la mémoire 2 qui stocke les coefficients de filtre du réseau de neurones. Le dispositif de routage 20 reçoit en outre l'adresse de l'événement d'entrée à traiter 202 (appelé ci-après « adresse d'entrée »). Le dispositif de routage 200 est configuré pour réordonner les coefficients du filtres (données représentant des noyaux de convolution) en fonction de l'adresse d'entrée, et délivre les coefficients du filtre réordonnés aux unités de traitement 20 de manière à pouvoir utiliser le coefficients de filtres en parallèle dans les unités de traitement 200.

L'adresse d'entrée 202 de la couche de convolution représente l'adresse du point d'application du noyau de convolution 12. L'adresse d'entrée correspond à l'adresse de l'impulsion émise par la couche précédente (impulsion d'entrée) et reçue par la couche considérée. Dans le système, les impulsions peuvent être transmises par un bus série. Les impulsions propagent donc avec elles au moins leur adresse d'émission.

L'adresse d'entrée peut comporter deux composantes X et Y selon le format de représentation choisi pour coder les impulsions. Par exemple, les impulsions transitant entre les couches du réseau de neurones 10 peuvent être codées selon le protocole ou format AER (« Address-Event Representation » signifiant représentation par événement d'adresse). Dans un tel format, chaque impulsion est numérique et est constituée d'un train de bits codant l'adresse (X, Y) de destination de l'impulsion suivant deux axes perpendiculaires X et Y, le repère (X, Y) correspondant au repère de la matrice d'entrée, ainsi que le signe de l'impulsion. Lorsqu'une impulsion arrive sur la matrice d'entrée, l'adresse (X, Y) codée représente l'emplacement du neurone d'entrée à activer. Ainsi, lorsqu'une impulsion arrive sur la matrice d'entrée, son adresse (X, Y) donne l'emplacement à activer $I_{i,j,}$ avec X → j et Y → i.

Le dispositif de routage 200, placé entre la mémoire 2 des coefficients des filtres et les unités de traitement 20, change ainsi de configuration à chaque évènement d'entrée. Le dispositif de routage 200 peut être implémenté pour un nombre de dimensions maximal et une taille maximale de noyau de convolution (le noyau de convolution est représenté par une matrice). Cependant, le dispositif de routage 200 peut être paramétré pour des tailles de noyau de convolution plus petites ou égales à la taille maximale, ou encore pour plusieurs noyaux de convolution différents.

L'implémentation du dispositif de routage 200 peut donc être de taille quelconque. Le dispositif de routage 200 peut reposer sur une structure régulière tout en passant à l'échelle en termes de taille de matrice maximale et de nombre de dimensions.

La figure 8 illustre la structure du dispositif de routage 200 selon un mode de réalisation.

Selon un aspect de l'invention, le dispositif de routage 200 comprend au moins un réseau de permutation 30 piloté par au moins une unité de contrôle paramétrable 32 (les unités de contrôle sont encore appelées ci-après « décodeur »). Chaque réseau de permutation 30 est configuré pour effectuer des décalages et/ou des sous-décalages circulaires du bus de données d'entrée 201 ou d'une partie du bus 201 en décalant les valeurs des coefficients portés par les fils du bus 201. Chaque fil (ou « ligne ») du bus d'entrée 201 transmet ou porte un coefficient du noyau de convolution numérique ou analogique. Chaque décodeur 32 (encore appelé « unité de contrôle ») est configuré pour piloter les permutateurs 30 en fonction de l'adresse d'entrée du noyau de convolution et d'au moins un paramètre 34 représentant la taille des noyaux de convolution à traiter.

Selon certains modes de réalisation, le dispositif de routage 200 peut s'appuyer sur une représentation en arbre (appelé ci-après « arbre de décalage ») pour effectuer l'acheminement des données d'un filtre de convolution vers les unités de traitement 20.

La représentation en arbre est construite par le dispositif de routage en fonction de la taille maximale du filtre de convolution à traiter. L'arbre peut être en outre construit à partir de d'autres paramètres qui peuvent comprendre :

- La taille de bus d'entrée 201 ; et/ou
- La division (ou « découpe ») du bus d'entrée en parties ; et/ou
- La profondeur de l'arbre.

La taille du bus en entrée 201 peut être définie comme étant égale au nombre d'éléments dans le filtre de convolution maximal. La découpe du bus d'entrée peut être définie comme étant égale au nombre de colonnes (respectivement de lignes) du filtre de convolution dans sa représentation matricielle (12) ou plus généralement à partir de ce nombre de colonnes (respectivement de lignes). La profondeur de l'arbre peut être définie comme étant égale au nombre de dimensions du filtre de convolution ou plus généralement à partir de ce nombre de dimensions. La profondeur de l'arbre définit ainsi le nombre de niveaux de l'arbre.

La suite de la description sera faite en référence à une découpe du bus d'entrée selon une découpe en colonnes de la matrice de convolution à titre d'exemple non limitatif, chaque sous-bus comprenant les composantes (coefficient de convolution) d'une colonne de la matrice de convolution.

Chaque nœud de l'arbre représente un réseau de permutation 30 configuré pour effectuer les décalages circulaires d'un vecteur de données, représentant les coefficients de convolution d'entrée d'une partie du bus d'entrée 201, ou des décalages circulaires des sous-ensembles (correspondant aux entrées de sous-bus 2010) de ce vecteur.

Selon certains modes de réalisation, chaque niveau de l'arbre correspond à une dimension donnée du repère dans lequel la matrice de convolution est représentée. Les réseaux de permutation agencés sur un niveau donné de l'arbre peuvent être avantageusement configurés pour effectuer des décalages circulaires des données reçues en entrée selon la dimension du repère choisi associée au niveau de l'arbre (le repère étant utilisé pour représenter les éléments du réseau de neurones).

La figure 9 illustre l'équivalence entre un filtre de convolution maximal et la topologie en arbre, selon certains modes de réalisation. Dans l'exemple de la figure 10, les matrices du réseau de neurones, et notamment les matrices de convolution 12, sont représentées dans un repère tri-dimensionnel (X, Y, Z). Dans une découpe en colonnes, le nombre de colonne de la matrice de convolution étant égal à $\boldsymbol{p}$, le bus d'entrée 201 peut être divisé en $\boldsymbol{p}$ sous-bus {2010-1, …, 2010-$\boldsymbol{p}$ }, chaque sous bus comprenant les composantes d'une colonne de la matrice de convolution. Il convient de noter que chaque coefficient de convolution dans la matrice de convolution peut être associé à une adresse ou position définie par des coordonnées dans le repère utilisé.

Dans l'exemple de la figure 9, le repère ayant 3 dimensions X, Y et Z, l'arbre de décalage comprend 3 niveaux, chaque niveau étant associé à une dimension du repère. Les réseaux de permutation 30 sur un niveau donné de l'arbre sont donc configurés pour effectuer des décalages circulaires des données reçues, selon la dimension associée à ce niveau de l'arbre. Par exemple, les réseaux de permutation 30-1 du niveau 1 effectuent des décalages circulaires des sous-bus 2010-i (composantes de chaque colonne i de la matrice de convolution) selon l'axe X, les réseaux de permutation 30-2 du niveau 2 effectuent des décalages circulaires des sorties des réseaux de permutation du niveau 1 selon l'axe Y, et les réseaux de permutation 30-3 du niveau 3 effectuent des décalages circulaires des sortie du niveau 2 selon l'axe Z.

La figure 10 représente un exemple de représentation en arbre du dispositif de routage 200 adapté pour traiter une matrice de convolution maximale de taille 5x5 dans un repère bi-dimensionnel (X, Y). Le bus d'entrée 201 comporte donc 25 éléments (5x5). Le nombre de colonnes de la matrice de convolution étant égal à 5, 5 sous-bus 2010 de 5 éléments (notés {20101-1,..., 2010-i, ..., 2010-5}) peuvent être créés. Le filtre ayant deux dimensions (matrices 2D définies selon deux axes X, Y), la profondeur de l'arbre est de 2 (l'arbre comprend deux niveaux).

Chacun des sous-bus 2010 peut être permuté indépendamment des autres sous-bus par un réseau de permutation 30 distinct (sous-décalages selon la dimension X), dans le premier niveau de l'arbre (5 réseaux de permutation au niveau 1 de l'arbre). Les sorties des 5 sous-bus 2010 peuvent être réunies dans un bus intermédiaire 204 comprenant 5 éléments (5 fils). Les données du bus secondaire 204 peuvent être permutées par un réseau de permutation agencé sur un deuxième niveau de l'arbre qui reçoit en entrée les données du bus secondaire 204 (décalage selon la dimension Y).

L'arbre ainsi créé comporte donc 5 nœuds sur le premier niveau et un nœud sur le deuxième niveau.

Chaque réseau de permutation 30 peut être constitué d'un ensemble d'opérateurs 30 de décalage circulaire, appelés ci-après « permutateurs ». Les permutateurs 30 peuvent être à 2 états et répartis en strates superposées (encore appelées « couches » ci-après) sur les lignes du bus. Une couche de permutation peut effectuer un décalage circulaire. Chaque couche décale ainsi les données d'un nombre choisi de lignes du bus vers la droite. Les valeurs à droite ne sont pas écrasées mais sont ramenées vers la gauche. L'ensemble des couches de permutation peut effectuer tous les décalages circulaires utiles.

La figure 11 représente un exemple de permutateur 300 à deux états qui peut être utilisé dans chaque réseau de permutation 30. Un permutateur à deux états 300 peut être contrôlé avec un bit. Dans l'état dit « décroisé » du permutateur 300 (représenté en haut), les entrées ne sont pas permutées. Dans l'état dit « croisé » (représenté en bas), les entrées sont permutées.

La figure 12 représente un exemple de réseau de permutation 30 selon deux représentations (partie gauche et partie droite) comprenant 3 permutateurs 300 à deux états répartis en strates sur les lignes de bus portant les valeurs 0, 1, 2, 3.

Les trois permutateurs 300 effectuent un décalage circulaire modifiant l'ordre des données initiales {0, 1 , 2, 3}. Le réseau de permutation 30 délivre ainsi les données {3, 0, 1 , 2} réordonnées par décalage circulaire.

Le réseau de permutation 30 peut être configuré pour réaliser des décalages circulaires de différents degrés sur les N fils du sous-bus d'entrée 2010 auquel il est associé. En effet, chaque strate de permutateurs peut agir sur des lignes plus ou moins éloignées. La distance entre deux lignes permutées dans une strate donnée définit le degré de cette strate.

La figure 13 représente des exemples de réseaux de permutation 30-1, 30-2, 30-3 et 30-4 comprenant des permutateurs configurés en strates selon différents degrés
- les strates de permutateurs du réseau de permutation 30-1 sont de degré 1 ;
- les strates de permutateurs du réseau de permutation 30-2 sont de degré 2 ;
- les strates de permutateurs du réseau de permutation 30-3 sont de degré 3 ; et
- les strates de permutateurs du réseau de permutation 30-4 sont de degré 4.

Les décalages circulaires à appliquer peuvent être définis par des numéros de décalage. Un numéro de décalage peut définir le décalage circulaire à appliquer. Ainsi un numéro de décalage « 1 » peut déclencher l'application d'un décalage circulaire vers la droite, un numéro de décalage « 2 » peut déclencher l'application de deux décalages circulaire vers la droite, etc. Selon un mode de réalisation préféré, un bus de N lignes est au maximum décalé N-1 fois.

Les décalages circulaires effectués par les strates (ou « couches ») de permutateurs 300 peuvent s'additionner. Ainsi, par exemple, il est possible d'utiliser une couche de permutateurs de degré 2 avec une couche de degré 1 pour faire le même décalage que la couche de degré 3.

Dans un mode de réalisation, les degrés des couches de permutateurs 300 utilisées dans chaque réseau de permutation 30 du dispositif de routage 20 peuvent correspondre à la décomposition en somme des décalages circulaires du bus 201 et de tous les sous-bus 2010. Ainsi, le réseau de permutation 30 peut effectuer tous les décalages circulaires du bus d'entrée et/ou de ses sous-bus. En complément, les couches de permutateurs 300 dans chaque réseau de permutation 30 peuvent être de préférence agencées par ordre décroissant de degrés.

Dans un mode de réalisation particulier, le degré des couches de permutateurs 300 dans chaque réseau de permutation 30 peuvent être fixées à 2 puissance $\boldsymbol{n}$ ($\boldsymbol{2^n}$) avec $\boldsymbol{n} \in [\boldsymbol{0}, \lfloor \boldsymbol{log2(N)} \rfloor]$. Dans le cas d'un bus à 5 lignes ($\boldsymbol{N = 5}$), il est donc possible d'utiliser 3 couches de permutateurs 300 pour chaque réseau de permutation 30, chaque couche étant respectivement de degré 4, 2 et 1. La figure 13 représente un tel exemple de réseau de permutation 30 comportant 3 couches de permutateurs en 2 puissance $\boldsymbol{n}$ pour un bus de 5 lignes. Dans l'exemple de la figure 13, le réseau de permutation comporte une première couche de degré 4 (comportant 1 permutateur 300), une deuxième couche de degré 2 (comportant 3 permutateurs 300) et une troisième couche de degré 1 (comportant 4 permutateurs 300).

Avec de tels couches dont le degré est fixé en 2 puissance $\boldsymbol{n}$, il est possible de réaliser des décalages circulaires de 1 à 7=4+2+1 en passant par tous les valeurs intermédiaires. Une telle décomposition en couches de permutateurs de degré $\boldsymbol{2^n}$ a une complexité en nombre de permutateurs 30 égale à :

$$\sum_{n=0}^{\lfloor log2(N) \rfloor} (N - 2^n) = N.(\lfloor log2(N) \rfloor + 1) - 2^{\lfloor log2(N) \rfloor + 1} + 1 \approx N.log2(N)$$

La complexité ainsi obtenue est faible et proche des réseaux MIM classiques.

La figure 15 représente un exemple de dispositif de routage 200 dont les éléments sont agencées selon une représentation d'arbre, pour matrice de convolution maximale de taille 5x5x1.

Selon certains modes de réalisation, les réseaux de permutation 30 se trouvant à la même profondeur dans l'arbre (même niveau de l'arbre) partagent un même décodeur 32. A chaque niveau de l'arbre est ainsi associé au moins un décodeur 32 configuré pour piloter les réseaux de permutation 30 agencés sur ce niveau. Dans ces modes de réalisation, le nombre total de décodeurs 32 est au moins égal à la profondeur de l'arbre. Par exemple, si l'arbre a une profondeur de 2, 2 décodeurs (1 pour chaque profondeur) contrôlés indépendamment sont utilisés. Un tel agencement des décodeurs permet de réduire la logique de contrôle.

Chaque décodeur 32 associé à un niveau donné de l'arbre est configuré pour activer dynamiquement les permutateurs 300 des réseaux de permutation 30 agencés sur ce niveau

de l'arbre de manière à modifier le nombre de décalages appliqués en fonction de l'événement et d'une configuration d'entrée définie notamment par l'adresse de l'évènement d'entrée. Ainsi, dans l'exemple de la figure 15, le décodeur 32-1 est configuré pour contrôler dynamiquement l'activation des permutateurs 300 des réseaux de permutation 30-1 du premier niveau en fonction de l'événement d'entrée, tandis que le décodeur 32-2 est configuré pour contrôler dynamiquement l'activation des permutateurs 300 des réseaux de permutation 30-2 du deuxième niveau.

Les décodeurs 32 peuvent inhiber certains permutateurs 300 du réseau de permutation 30 pour créer des sous-espaces de décalage indépendants. A partir de ces sous-espaces, des sous arbres et des sous-ensembles du vecteur d'entrée du réseau de permutation peuvent être crées. Un permutateur 300 peut être inhibé pour toutes les couches de tous les étages. Dans un mode de réalisation, les permutateurs 300 d'une couche donnée sont activés en même temps.

La configuration d'entrée peut être utilisée par le dispositif de routage 200 pour déterminer le nombre de lignes du bus d'entrée 201 sur lesquelles un décalage circulaire est à appliquer. La configuration d'entrée peut être commune aux différents décodeurs 32. La configuration d'entrée peut être utilisée après la construction de l'arbre pour fixer la taille de la convolution traitée.

Le dispositif de routage 200 peut traiter une ou plusieurs convolutions en même temps en fonction du nombre de décalages à appliquer par rapport à la taille du bus d'entrée, comme illustré par la figure 16. La figure 16 montre la représentation en arbre de décalage utilisée pour configurer dynamiquement les réseaux de permutation 30 en fonction d'un événement d'entrée pour chaque noyau de convolution C0, C1, C2, C3. Chaque convolution utilise 3 réseaux de permutation Px (30) sur le premier niveau de l'arbre pour effectuer des sous-décalages circulaire selon la dimension X et un permutateur Py (30) sur le deuxième niveau de l'arbre pour effectuer des décalages selon la dimension Y. Sur chaque niveau et pour chaque noyau de convolution, au moins un décodeur 32 (non représenté sur la figure 16) peut piloter les réseaux de permutation en utilisant un mot de commande pour chaque réseau de permutation, le mot de commande comprenant autant de bits que de permutateurs 300 dans le réseau de permutation 30. La valeur de chaque bit dans le mot de commande contrôle l'état d'activation du permutateur 300 associé.

Les décalages circulaires ainsi réalisés sur les différents sous-bus d'entrées 2010 permettent ainsi de réordonner les coefficients portés par les fils de ces sous-bus. Le dispositif de

routage 200 distribue ensuite les coefficients ainsi réordonnés aux unités de traitement 20. Par exemple, les coefficients {021} de la convolution C0 de la ligne L0 de la matrice de convolution (selon une découpe en lignes de la matrice de convolution) et les coefficients {3,5,4} de la convolution C1 de la ligne L2 la matrice de convolution sont distribués après réordonnancement ({012} pour C0,L0 et {345} pour C1,L2 à une même unité de traitement 20. Chaque unité de traitement ne reçoit qu'une seul valeur.

Les différents filtres peuvent être découpés (ou divisés) en lignes. Ces lignes peuvent être ensuite concaténées sur un sous-bus. Le sous-bus suivant contient les lignes suivantes. Une fois que toutes les lignes de ces filtres sont placées, le filtre suivant est traité. Dans une mode de réalisation préféré, une ligne ne peut être coupée pour la placer sur 2 sous bus. Par exemple, en considérant 11 sous-bus de 11 éléments (soit 121 éléments), sur un sous-bus donné, il est possible de placer 3 lignes de 3 filtres 3*3 différents. Les 3 filtres utilisent 3 sous-bus soit 33 éléments (fils). Il est donc possible de constituer 9 filtres 3*3.

Dans certains modes de réalisation, si le nombre de décalages à appliquer est inférieur à la taille du bus d'entrée 201 ayant un nombre de lignes total égal à N, le dispositif de routage 200 peut effectuer les permutations de lignes comme suit :

- si le nombre de lignes à permuter est plus grand que la moitié du nombre de ligne total (N/2), le dispositif de routage 200 applique la permutation une seule fois sur le bus d'entrée 201 (par exemple une permutation de 4 éléments d'entrée d'un bus 201 de 5 éléments est effectuée) ;
- sinon, si le nombre de lignes à permuter est inférieur à N/2, la permutation peut être appliquée autant de fois que possible. Par exemple, une permutation de 2 éléments d'entrée d'un bus 201 de 5 éléments peut être appliquée 2 fois sur le bus pour traiter deux convolutions 2x2 en même temps. Selon une caractéristiques, les valeurs non utiles peuvent ne pas être permutées et garder leur position initiale.

Les décodeurs 32 peuvent mettre en œuvre un procédé d'activation de permutateurs pour déterminer les permutateurs 300 à activer dans les réseaux de permutation 30 pour une configuration d'entrée et pour une permutation circulaire données (la permutation pouvant être définie par un nombre de décalage), et en tenant compte de la topologie des réseaux de permutation. Le procédé d'activation (encore appelé « procédé de configuration ») peut être appliqué pour tous les décalages possibles et toutes les configurations d'entrée possibles.

La figure 17 est un organigramme représentant le procédé d'activation des permutateurs d'un réseau de permutation 30 d'un niveau donné de l'arbre.

Le réseau de permutation 30 est constitué de couches de permutateurs 300, chaque couche comprenant un ou plusieurs permutateurs 300, ayant chacun un degré.

A l'étape 500, un ensemble de couches de permutateurs 300 du réseau de permutation est sélectionné en fonction de la somme des degrés associés aux permutateurs 300 de cet ensemble de couches. L'ensemble de couche comprend au moins une couche. Dans un mode de réalisation, pour la permutation « 0 », aucun permutateur n'est activé.

Dans un mode de réalisation, l'ensemble de couches sélectionné est tel que la somme des degrés associés aux permutateurs 300 de cet ensemble de couches est égale au décalage à appliquer (défini par un nombre de décalage).

A l'étape 502, dans l'ensemble de couches sélectionné, certains des permutateurs sont activés. Les permutateurs qui sont activés sont ceux qui permettent de déplacer les données du bus 201 ou du sous-bus d'entrée 2010 depuis la gauche vers des emplacements cibles à droite. L'activation des permutateurs résulte en un décalage circulaire des données d'entrée du réseau de permutation 30.

A l'étape 504, chaque donnée ainsi permutée peut être testée. Les valeurs peuvent être testées de gauche à droite.

Ainsi, à l'étape 505, il est déterminé si une condition relative à la donnée testée après permutation (initialement la donnée la plus à gauche) est satisfaite. Cette condition peut consister à déterminer si la donnée, qui est ramenée à la position considérée par chaque permutation appliquée à l'étape 502 et qui correspond à un permutateur activé (initialement position la plus à gauche), est positionnée sur la ligne souhaitée du bus de sortie du réseau de permutation. Si la condition n'est pas satisfaite, à l'étape 506, un autre permutateur 300 du sous-ensemble sélectionné à l'étape 500 qui n'a pas été activé à l'étape 502 peut être activé pour remplacer le permutateur considéré sans affecter les permutations effectuées au préalable. Si la donnée testée est bien placée (condition 505 satisfaite), les étapes 504 et 507 sont répétées itérativement pour vérifier si les données suivantes (507) sont bien placées.

Si le nombre de lignes permutées du bus d'entrée 201 est inférieur à la moitié du nombre de lignes (201) dans le bus 201, les configurations trouvées peuvent être répétées autant de fois que possible sur le bus d'entrée.

La figure 18 illustre les étapes successives du procédé d'activation de permutateurs pour un exemple de réseau de permutation en couches 2^n réalisation, pour tous les décalages de 5 lignes sur un bus d'entrée de 5 lignes sur un réseau de permutation en couche 2 puissance n. Les permutateurs en rouge sont activés à l'étape 500 tandis que les permutateurs en vert sont activés à l'étape 504.

Lorsque toutes les configurations de permutateurs 300 ont été déterminées, ces configurations peuvent être formalisées sous la forme d'un circuit logique ou d'une table de correspondance pilotée par un bus de contrôle et par un bus de configuration. La configuration d'entrée du routage pour un évènement d'entrée donné fournit la taille du filtre traité, le contrôle, et/ou le nombre de décalages circulaires à appliquer.

La figure 19 représente un exemple d'implémentation de permutateur à 2 états numériques pouvant être utilisé dans les réseaux de permutation 30 du dispositif de routage 200.

Le permutateur 300 à 2 états de la figure 19 est réalisé en utilisant 2 multiplexeurs 301 et 302 prenant les même entrées et partageant le même bit de contrôle 303. Les entrées des multiplexeurs sont cependant inversées (dans l'exemple, le multiplexeur 301 reçoit les entrées A et B tandis que le multiplexeur 302 reçoit les entrées B et A). La taille des bus d'entrées des multiplexeurs 301/302 peut être quelconque : le permutateur 300 peut ainsi permuter des bus de taille quelconque. Dans l'exemple de la figure 19, lorsque le bit de contrôle 303 est à 0, la donnée A est routé vers C et la donnée B est routée vers D. Inversement, lorsque le bit de contrôle est à 1, la donnée A est routée vers D et la donnée B est routée vers C.

La figure 20 représente un exemple de réseau de permutation 30 d'un bus d'entrée 201 à 11 lignes avec des couches en $\boldsymbol{2^n}$ pouvant utiliser un tel permutateur à deux états 300. Le réseau de permutation du bus d'entrée comprend $[\boldsymbol{log2(11)}] = \boldsymbol{3}$ couches en $\boldsymbol{2^n}$, soit $\boldsymbol{n} \in [\boldsymbol{0,3}]$. Le réseau de permutation 30 contient donc des couches de degré 8, 4, 2 et 1 et 29 permutateurs 300. Une telle structure de réseau de permutation 30 permet de faire toutes les permutations circulaire d'un bus de 11 lignes et de tous ses sous bus. Ainsi la liste des décalages circulaires pouvant être effectués par un tel réseau de permutation comprend:

- Tous les décalages d'un bus de 11 lignes ;
- Tous les décalages d'un bus de 10 lignes ;
- Tous les décalages d'un bus de 9 lignes ;
- Tous les décalages d'un bus de 8 lignes ;

- Tous les décalages d'un bus de 7 lignes ;
- Tous les décalages d'un bus de 6 lignes ;
- Tous les décalages de 2 sous-bus de 5 lignes ;
- Tous les décalages de 2 sous-bus de 4 lignes ;
- Tous les décalages de 3 sous-bus de 3 lignes ;
- Tous les décalages de 5 sous-bus de 2 lignes.

Un tel réseau de permutation 30 peut être contrôlé par un mot de contrôle (de 29 bits dans cet exemple) fourni par le décodeur 32 associé au réseau de permutation 30. Chaque bit du mot de contrôle est associé à l'un des permutateurs 300 du réseau de permutation 30. La valeur de chaque bit du mot de contrôle commande l'état d'activation du permutateur 300 associé (par exemple, la valeur 1 commande l'activation du permutateur associé tandis que la valeur 0 laisse le permutateur à l'état inactivé).

Les permutateurs 300 étant agencés en couches de degré $\boldsymbol{deg}$, le nombre de permutateurs dans une couche donnée est :
$\boldsymbol{N - 2^n = N - deg_n}$, où $\boldsymbol{N}$ représente le nombre d'éléments dans le bus.

Le nombre de couches étant égal à $[\boldsymbol{log2(N)}]$, le nombre total de permutateurs est donc $\sum_{\boldsymbol{n=0}}^{[\boldsymbol{log2(N)}]}(\boldsymbol{N - 2^n}), \boldsymbol{2^n = deg_n}$

Dans un exemple d'implémentation du dispositif de routage 200 utilisant des réseaux de permutation selon les configurations des figures 19 et 20 (29 permutateurs par réseau de permutation), pour une matrice de convolution de taille 11x11x1, le bus d'entrée comprend 121 lignes (11x11) divisées en 11 paquets de 11 lignes formant les 11 sous-bus d'entrée 2010. Le filtre ayant deux dimensions (matrices 2D définies selon deux axes X et Y), la profondeur de l'arbre est de 2 (l'arbre comprend deux niveaux). L'arbre contient 12 nœuds représentant chacun un réseau de permutation. Le dispositif de routage 200 utilise donc 12 réseaux de permutation (30) et 29x12 = 348 permutateurs (300). L'arbre de décalage peut être paramétré pour effectuer toutes les matrices de taille 11x11 à 1x1. Le mot de paramétrage 203 peut être représenté par 4 bits et le mot de contrôle 4+4 = 8 bits.

Les modes de réalisation de l'invention permettent ainsi de trouver le décalage cible à partir de l'adresse de l'événement d'entrée pour réordonner les données d'un filtre de convolution à transférer aux unités de traitement 20 dans un calculateur 100 à base d'architecture de réseaux de neurones convolutionnel.

Le dispositif de routage 200 permet une activation dynamique des permutateurs des réseaux de permutation en utilisant des décodeurs 32 paramétrables en fonction de l'événement d'entrée, de la taille maximal du noyau de convolution et/ou du décalage à appliquer.

Il permet de réordonner les coefficients de poids portés par le vecteur d'entrée (bus d'entrée) en fonction des événements d'entrée (par exemple impulsions) et des paramètres du réseau en appliquant des permutations circulaires contrôlées.

L'invention offre une solution de routage dynamique de données de noyaux de convolution dans les calculateurs neuromorphiques et notamment dans les calculateurs neuromorphiques massivement parallèles.

L'invention permet notamment un routage parallèle et paramétrable de données pour des calculs neuromorphiques de convolutions, une distribution et un ordonnancement des coefficients de poids parallèle pour une convolution. Elle offre un routage flexible et notamment la création de sous routages indépendants. Le dispositif de routage 200 s'adapte dynamiquement à la taille et au nombre de dimensions des convolutions. Il en résulte une complexité réduite de N^2 par rapport aux solutions existantes (complexité N^3).

Le dispositif de routage 200 selon les modes de réalisation de l'invention peut changer dynamiquement la configuration des permutateurs 300 des réseaux de permutation, à chaque évènement d'entrée sur un noyau de convolution. Le dispositif de routage 200 est implémentable non seulement pour une dimension maximale et une taille maximale de filtre de convolution, mais aussi pour toutes les tailles de matrices de convolution (12) plus petites ou égales à la taille maximale, voire pour plusieurs matrices de convolution en mêmes temps. Le dispositif de routage 200 permet donc un passage à l'échelle en termes de taille de matrice maximale et de nombre de dimensions du noyau de convolution.

Le dispositif de routage 200 requière une surface d'implémentation plus petite que dans les réalisations existantes à base de multiplexeur.

L'arbre de décalage permet de représenter l'interconnexion des réseaux de permutation 30 et de traiter les filtres convolutions en fonction du noyau maximal à traiter, et du nombre de dimensions du filtre de convolution (définissant la profondeur de l'arbre).

La topologie des réseaux de permutation 30 et des couches de permutateurs 30 à l'intérieur des réseaux de permutation permettent d'effectuer des permutations circulaires du bus d'entrée et/ou des sous bus d'entrée en contrôlant dynamiquement l'activation des permutateurs 30. Le contrôle dynamique des permutateurs 300 et plus généralement la configuration dynamique des réseaux de permutations 30 en fonction de l'événement d'entrée sont tels que les réseaux de permutation peuvent effectuer tous les décalages circulaires.

Le dispositif de routage 200 selon les modes de réalisation de l'invention peut s'appliquer à un bus d'entrée de taille quelconque de façon complètement parallèle, en utilisant que des permutateurs 300 pouvant être numériques ou analogique.

Le dispositif de routage 200 peut être implémenté pour des signaux numériques ou analogiques. Il offre en outre une grande flexibilité d'implémentation (taille du bus d'entrée, profondeur de l'arbre, etc.). Les décalages circulaires peuvent être réalisés au niveau du bus d'entrée 201 mais aussi au niveau de ses sous-bus.

Bien que l'invention présente un avantage particulier dans une application à un calculateur de convolution neuromorphique impulsionnel pour transférer les valeurs d'un filtre de convolution vers des unités de traitement 20 représentant des unités de calcul neuromorphiques 20, l'invention n'est pas limitée à une telle application. En variante, l'invention s'applique à tout dispositif dans lequel les coefficients d'au moins un noyau de convolution doivent être distribuer en parallèle vers des unités de traitements.

L'homme du métier comprendra que le procédé d'activation des permutateurs 300 selon les modes de réalisation peut être mis en œuvre de diverses manières par matériel (« hardware »), logiciel, ou une combinaison de matériel et de logiciels, notamment sous la forme de code de programme pouvant être distribué sous la forme d'un produit de programme, sous diverses formes. En particulier, le code de programme peut être distribué à l'aide de supports lisibles par ordinateur, qui peuvent inclure des supports de stockage lisibles par ordinateur et des supports de communication. Les procédés décrits dans la présente description peuvent être notamment implémentés sous la forme d'instructions de programme d'ordinateur exécutables par un ou plusieurs processeurs dans un dispositif informatique d'ordinateur. Ces instructions de programme d'ordinateur peuvent également être stockées dans un support lisible par ordinateur.

L'invention n'est pas limitée aux modes de réalisation décrits ci-avant à titre d'exemple non limitatif. Elle englobe toutes les variantes de réalisation qui pourront être envisagées par

l'homme du métier. En particulier, l'invention n'est pas limitée à un réseau de neurones convolutionnel de type à impulsions. Par ailleurs, l'invention n'est pas limitée à un nombre de dimension égal à 2 ou 3 comme illustré dans les exemples ci-avant.

5

Revendications

1.Dispositif pour distribuer des coefficients de convolution d'au moins un noyau de convolution d'un réseau de neurones convolutionnel portés par un bus d'entrée (201) vers un ensemble d'unités de traitement dans un calculateur basé sur une architecture de réseau de neurones convolutionnel, caractérisé en ce que le dispositif comprend au moins un réseau de permutation (30) pilotée par au moins une unité de contrôle (32), le réseau de permutation comprenant un ensemble de permutateurs (30) agencés pour effectuer des décalages circulaires d'au moins une partie du bus d'entrée, et en ce que, pour chaque noyau de convolution, chaque unité de contrôle est configurée pour piloter dynamiquement certains au moins des permutateurs (300) des réseaux de permutation (30) en réponse à un événement d'entrée appliqué sur le noyau de convolution et d'au moins un paramètre représentant la taille maximale des noyaux de convolution.

2. Dispositif selon la revendication 1, caractérisé en ce que chaque noyau de convolution est représenté par une matrice de convolution de taille donnée dans un repère de dimensions choisies, et en ce que les réseaux de permutation sont interconnectés entre eux sous la forme d'une structure en arbre, dit arbre de décalage, ledit arbre comprenant un ensemble de nœuds, chaque nœud représentant un réseau de permutation, et un ensemble de niveaux, le nombre de niveaux dans l'arbre étant déterminé à partir du nombre de dimensions dudit repère, la taille du bus d'entrée (201) étant définie à partir de la taille maximale de noyau de convolution.

3. Dispositif selon la revendication 2, caractérisé en ce que chaque niveau de l'arbre correspond à une dimension donnée dudit repère, et en ce que les réseaux de permutation (30) agencés sur un niveau donné de l'arbre sont configurés pour effectuer des décalages circulaires des données du bus d'entrée selon la dimension associée audit niveau.

4. Dispositif selon l'une des revendications 2 et 3, caractérisé en ce que la matrice de convolution représentant chaque noyau de convolution (12) comprend un ensemble de colonnes, et en ce que le dispositif de routage (200) est configuré pour diviser le bus d'entrée en un ensemble de sous-bus d'entrée en fonction du nombre de colonnes dans la matrice de convolution, les coefficients de convolution de chaque sous-bus d'entrée (2010) étant associé à un desdits réseaux de permutation (30) sur un premier niveaux de l'arbre, chaque réseau de permutations sur ledit premier niveau étant configuré pour réaliser des décalages circulaires des coefficients des sous-bus d'entrée en fonction d'un événement d'entrée sur le noyau de convolution, la sortie de chaque réseau de permutation du premier

niveau de l'arbre étant délivrée en entrée d'un bus de données intermédiaire sous la forme d'un vecteur de données d'entrée (204), les vecteurs de données d'entrée du bus de données secondaires étant permutées par un réseau de permutation agencé sur le deuxième niveau de l'arbre.

5. Dispositif selon l'une des revendications 2 à 4, caractérisé en ce qu'il comprend au moins une unité de contrôle (32) pour piloter les réseaux de permutations d'un niveau donné de l'arbre.

6. Dispositif selon l'une des revendications précédentes, caractérisé en ce que chaque réseau de permutation comprend un ensemble de permutateurs (300), agencés en couches superposées, pour effectuer un décalage circulaire des lignes du bus (201) en entrée du réseau de permutation (30), chaque couche de permutateurs (300) étant activable par l'unité de contrôle pilotant le réseau de permutation, chaque couche de permutateurs étant en outre configurée pour effectuer des permutations entre deux lignes d'entrée.

7. Dispositif selon la revendication 6, caractérisé en ce que les permutateurs sont des permutateurs à deux états.

8. Dispositif selon la revendication 7, caractérisé en ce que chaque couche de permutateurs (300) a un degré représentant la distance entre deux lignes du bus d'entrée permutées, et en ce que le degré d'une couche de permutateurs (300) d'un réseau de permutation (30) est égal à 2 élevé à une valeur de puissance choisie (2^n) .

9. Dispositif selon la revendication 8, caractérisé en ce que la valeur de puissance choisie pour une couche de permutateurs donnée est égale à la partie entière de $log2(N)$, où N désigne le nombre de lignes du bus d'entrée (201).

10. Dispositif selon l'une des revendications 5 à 6, caractérisé en ce que le réseau de permutation est configuré pour réaliser plusieurs décalages circulaires de taille inférieure au nombre de lignes du bus d'entrée (201).

11. Dispositif selon l'une des revendications 5 à 6, caractérisé en ce que le réseau de permutation est configuré pour réaliser un décalage circulaire de taille égale au nombre de lignes du bus d'entrée (201).

12. Dispositif selon l'une des revendications précédentes, caractérisé en ce que le réseau de neurones convolutionnel (100) comprend une matrice d'entrée, la matrice d'entrée comprenant un ensemble de neurones d'entrée, la matrice d'entrée étant connectée à une matrice de sortie par des noyaux de convolution, et en ce que l'événement d'entrée est l'adresse du neurone d'entrée à activer sur la matrice d'entrée du réseau de neurones convolutionnel, le dispositif de routage (200) étant apte à déterminer le nombre de lignes du bus d'entrée (201) sur lesquelles un décalage circulaire est à appliquer en fonction de l'adresse d'entrée.

13. Dispositif selon l'une des revendications précédentes, caractérisé en ce que le réseau de neurones convolutionnel est à impulsions et en ce que l'événement d'entrée est défini par l'adresse de l'impulsion arrivant sur la matrice d'entrée.

14. Calculateur neuromorphique comprenant une mémoire pour stocker les coefficients de noyaux de convolution et un ensemble d'unités de traitement pour calculer la réponse d'un réseau de neurones à un événement d'entrée, caractérisé en ce qu'il comprend un dispositif de routage (200) selon l'une des revendications précédentes pour distribuer les coefficients aux unités de traitement.

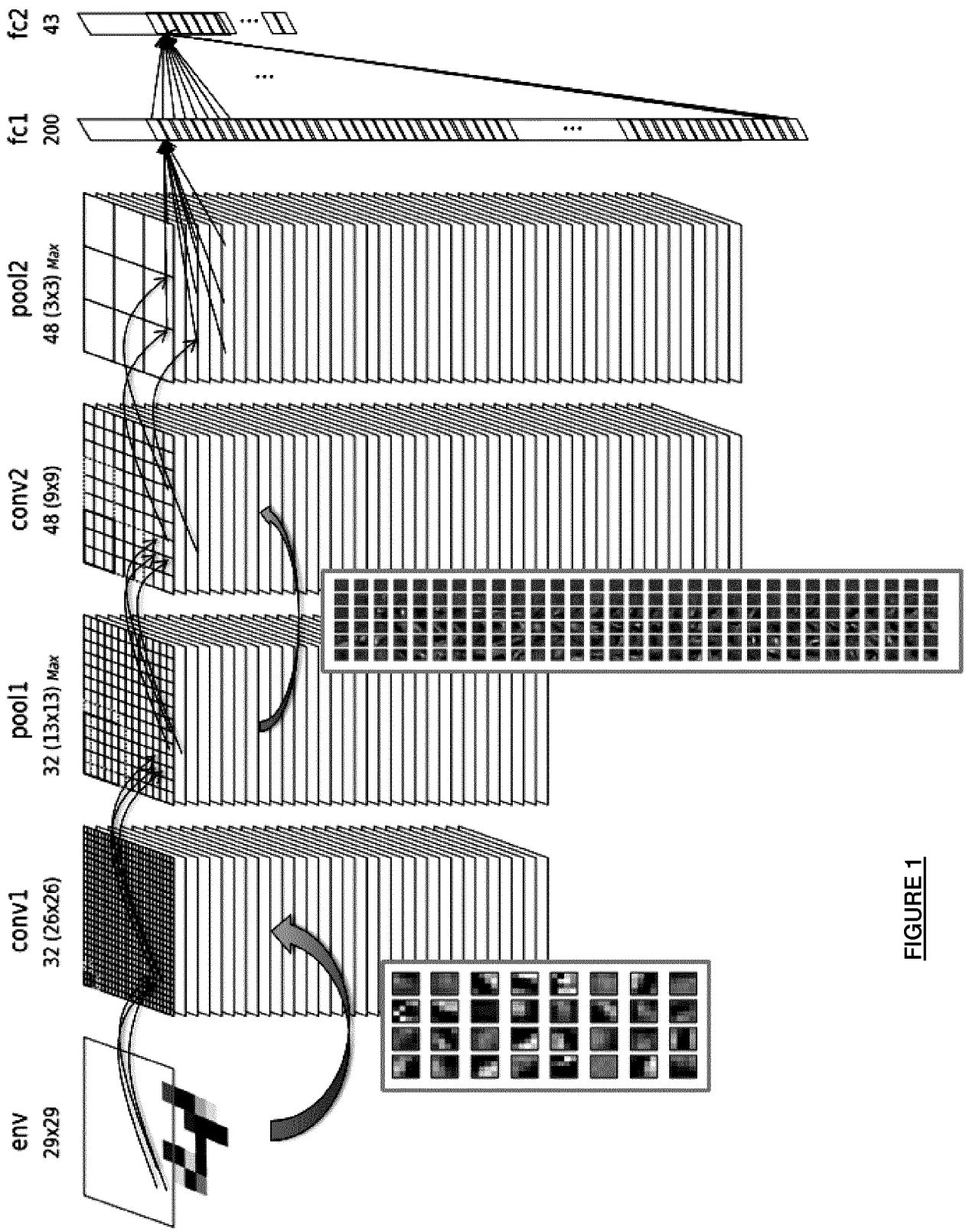
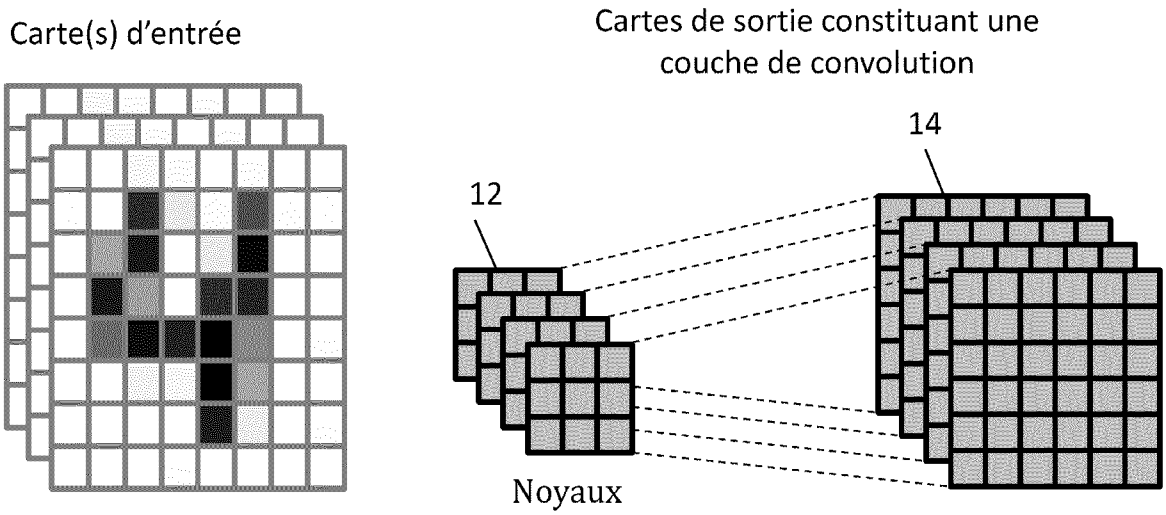

fc2
43
fc1
200
pool2
48 (3x3) Max
conv2
48 (9x9)
pool1
32 (13x13) Max
conv1
32 (26x26)
env
29x29

FIGURE 1



	1	2	3	4	5	6
1	X				X	X
2	X	X				X
3		X	X			X
4			X	X		X
5				X	X	X

Cartes de sortie

Connexion = un noyau de convolution

Cartes d'entrée

FIGURE 2

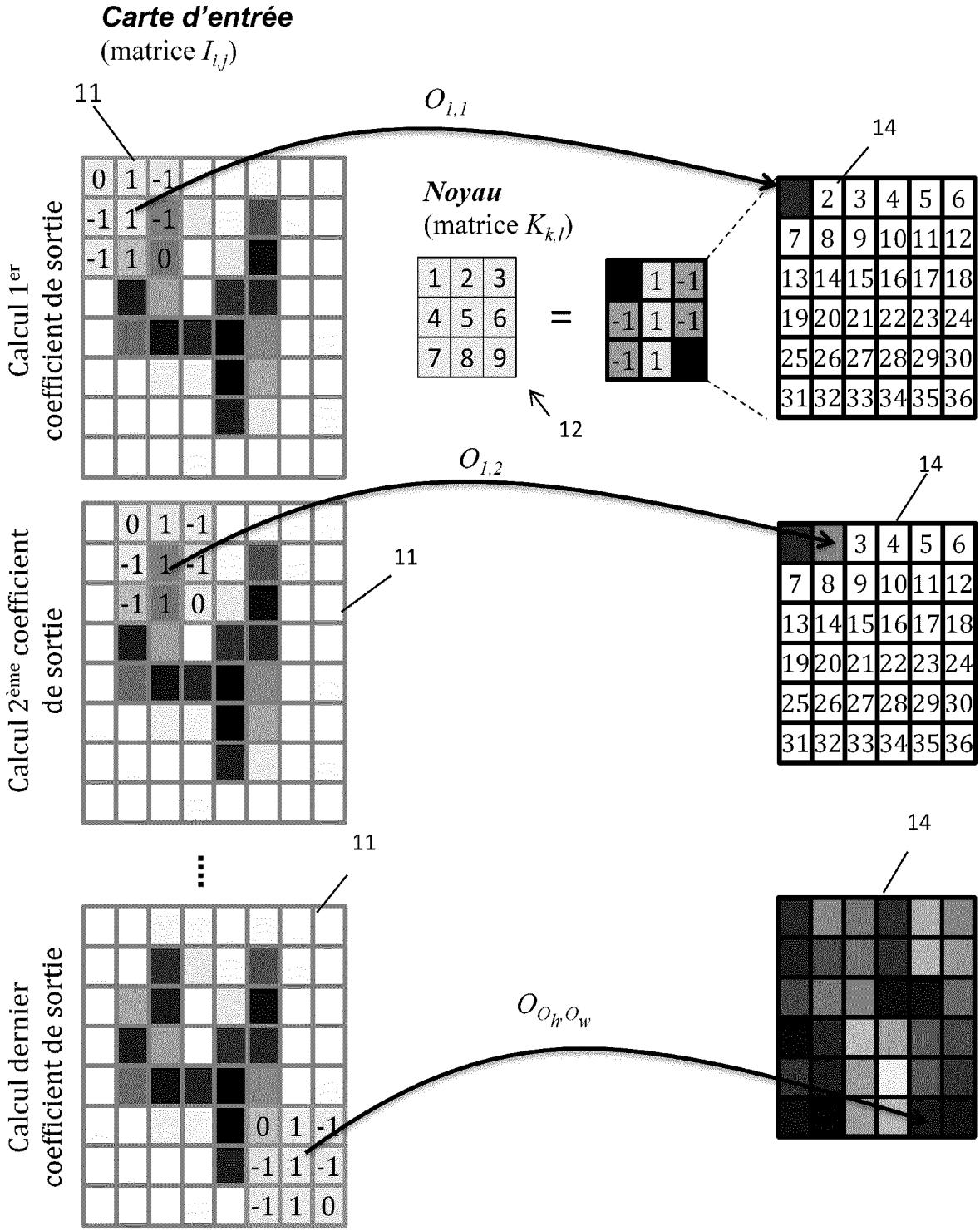

Carte d'entrée
(matrice $I_{i,j}$)
11
$O_{1,1}$
14
Noyau
(matrice $K_{k,l}$)
12
Calcul 1er coefficient de sortie
$O_{1,2}$
Calcul 2ème coefficient de sortie
Calcul dernier coefficient de sortie
$O_{O_h O_w}$

FIGURE 3

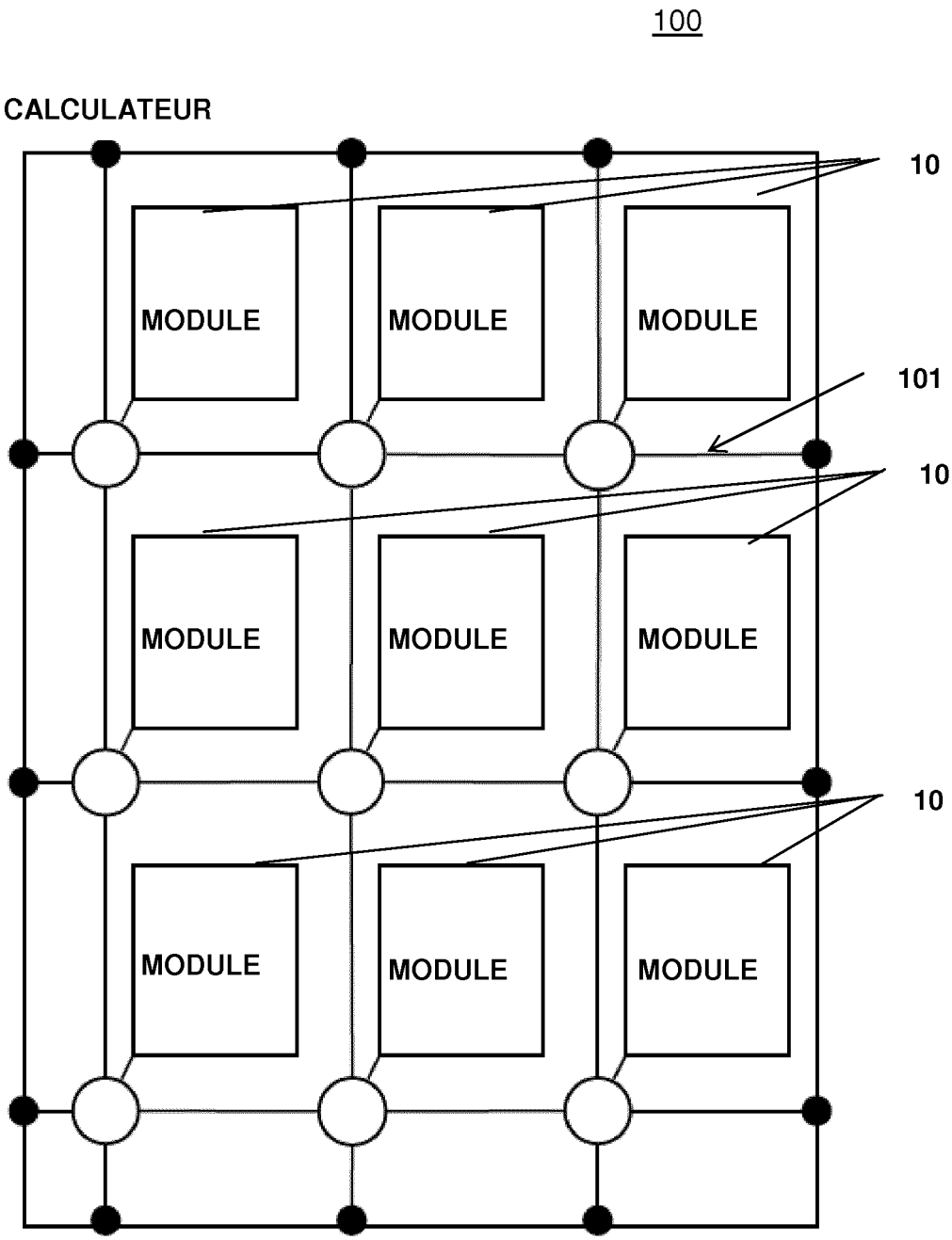

100
CALCULATEUR
10
MODULE
MODULE
MODULE
101
10
MODULE
MODULE
MODULE
10
MODULE
MODULE
MODULE

FIGURE 4

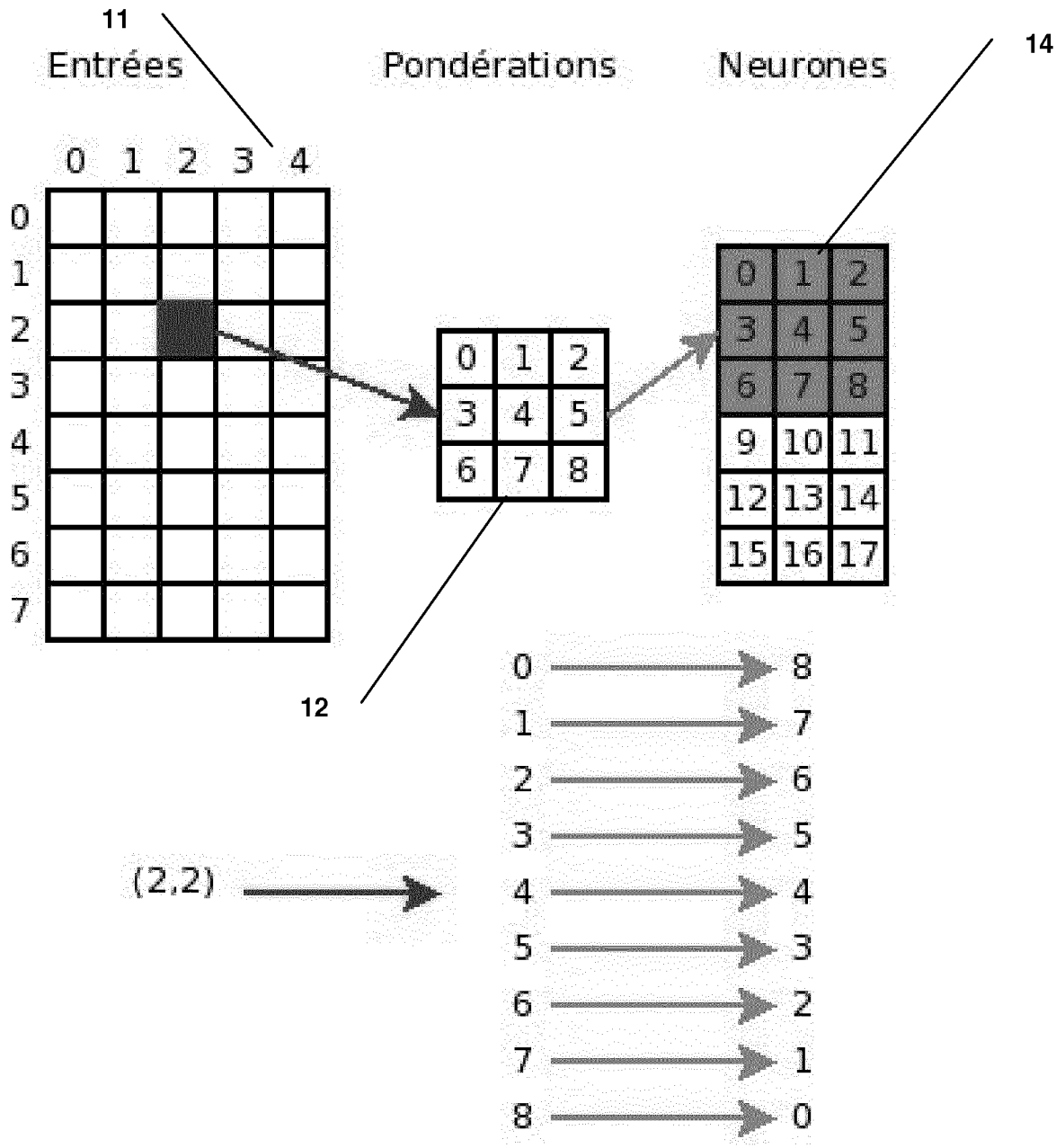

11
Entrées
Pondérations
Neurones
14
0 1 2 3 4
0
1
2
3
4
5
6
7
0 1 2
3 4 5
6 7 8
0 1 2
3 4 5
6 7 8
9 10 11
12 13 14
15 16 17
12
(2,2)
0 → 8
1 → 7
2 → 6
3 → 5
4 → 4
5 → 3
6 → 2
7 → 1
8 → 0

FIGURE 5

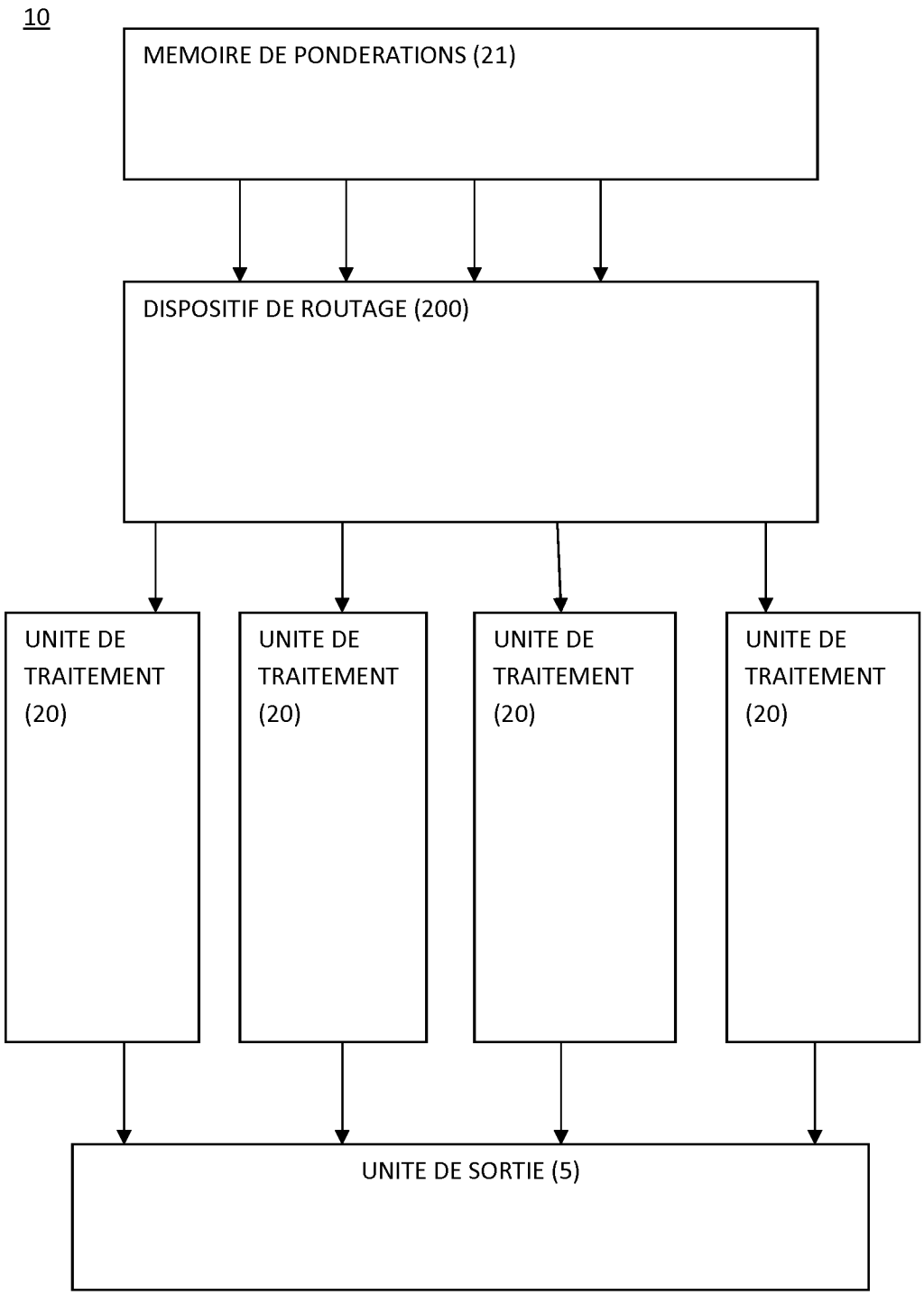

10
MEMOIRE DE PONDERATIONS (21)
DISPOSITIF DE ROUTAGE (200)
UNITE DE TRAITEMENT (20)
UNITE DE TRAITEMENT (20)
UNITE DE TRAITEMENT (20)
UNITE DE TRAITEMENT (20)
UNITE DE SORTIE (5)

FIGURE 6

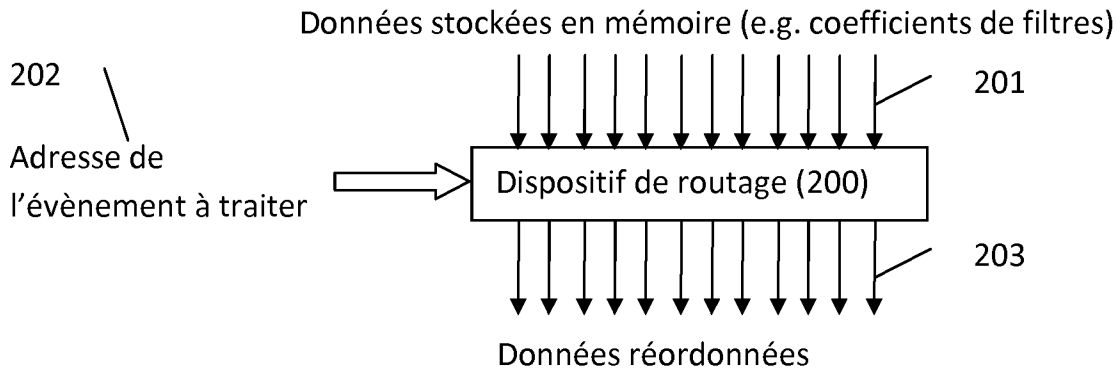

Données stockées en mémoire (e.g. coefficients de filtres)
202
201
Adresse de
l'évènement à traiter
Dispositif de routage (200)
203
Données réordonnées

FIGURE 7

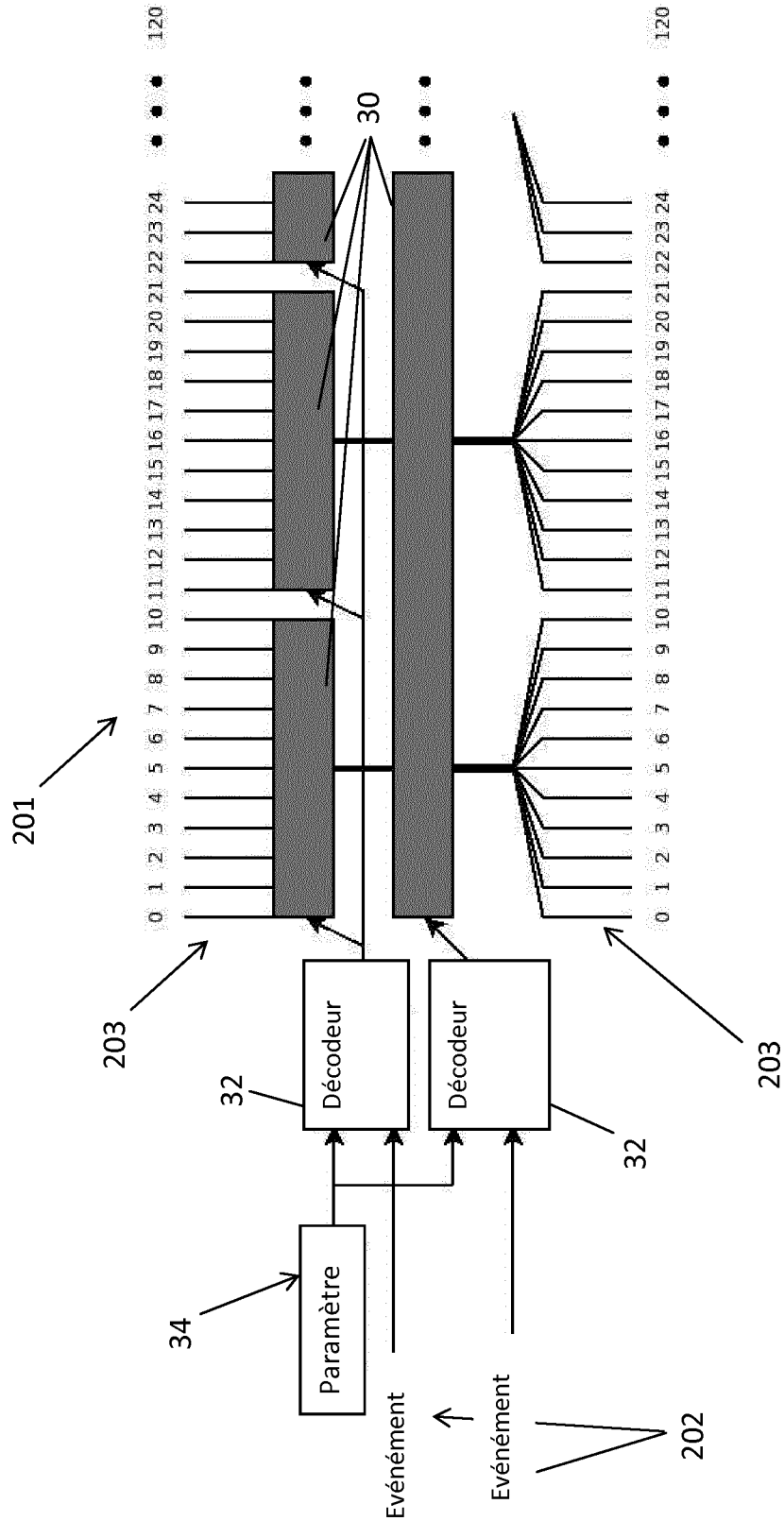

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 120
30
201
203
32
Décodeur
Décodeur
32
203
34
Paramètre
Evénement
Evénement
202

FIGURE 8

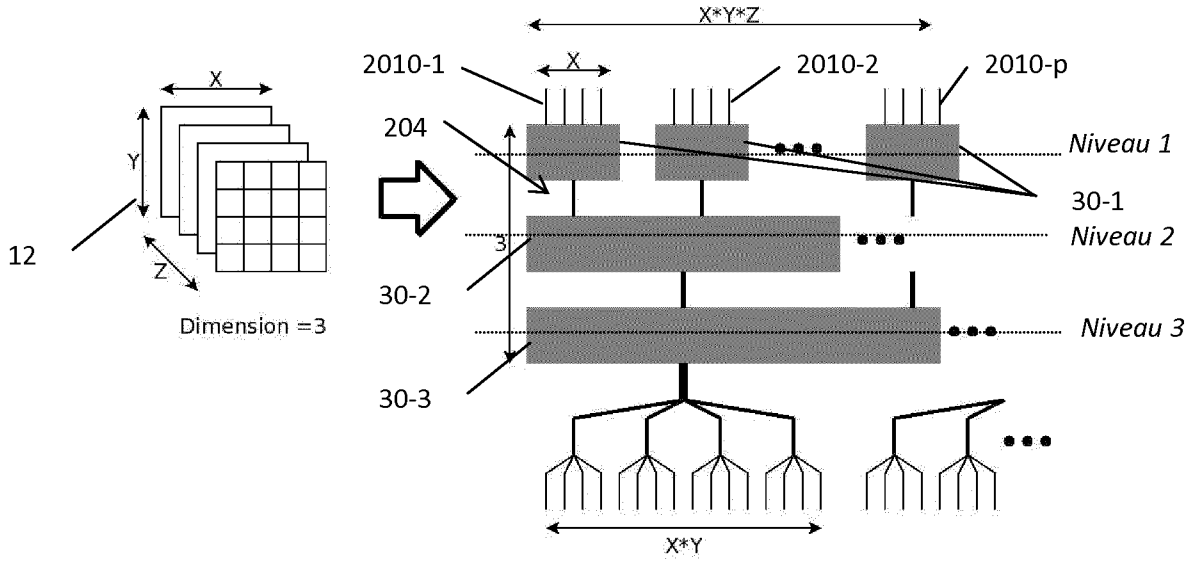

X
Y
Z
12
Dimension =3
X*Y*Z
X
2010-1
2010-2
2010-p
204
Niveau 1
30-1
Niveau 2
3
30-2
Niveau 3
30-3
X*Y

FIGURE 9

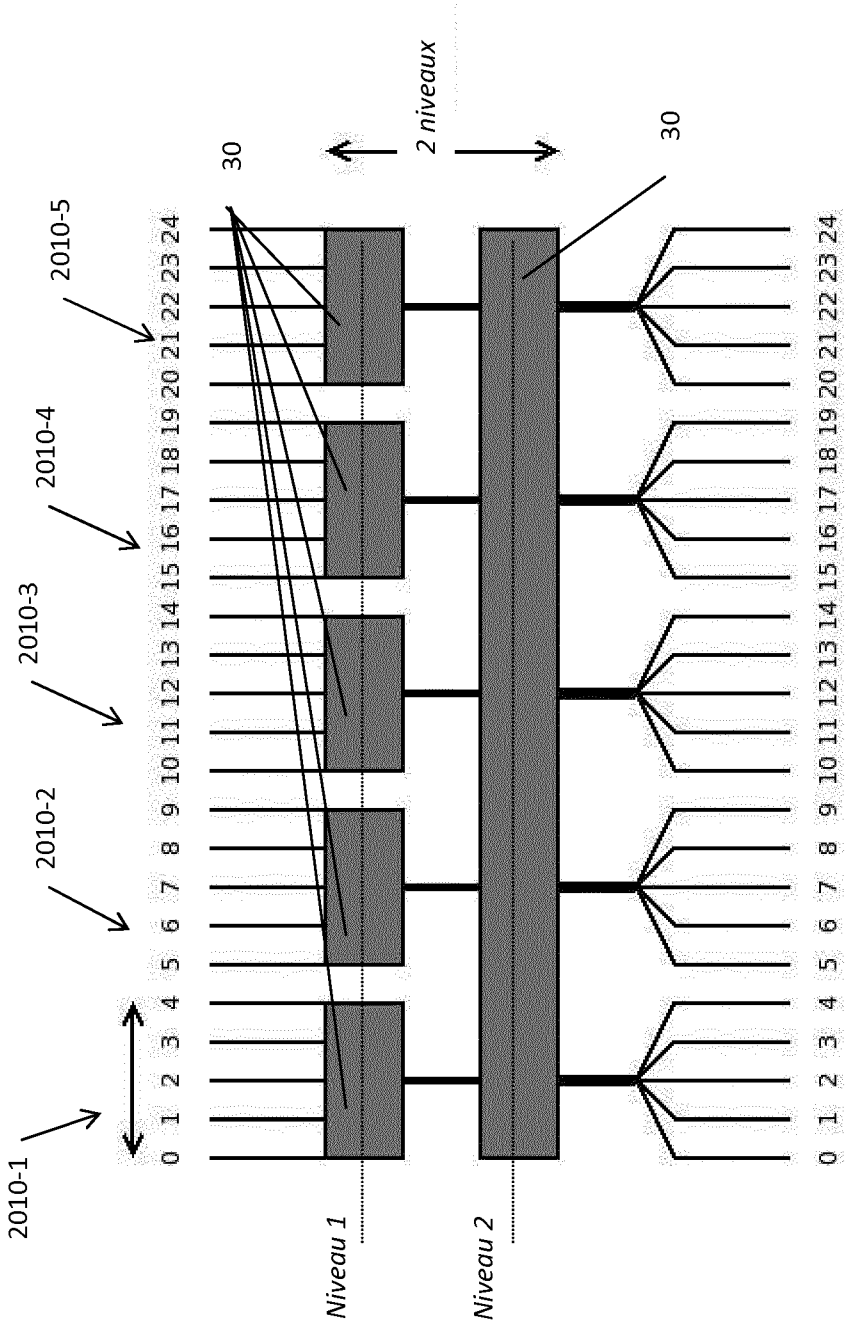

2 niveaux
30
30
2010-1
2010-2
2010-3
2010-4
2010-5
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Niveau 1
Niveau 2

FIGURE 10

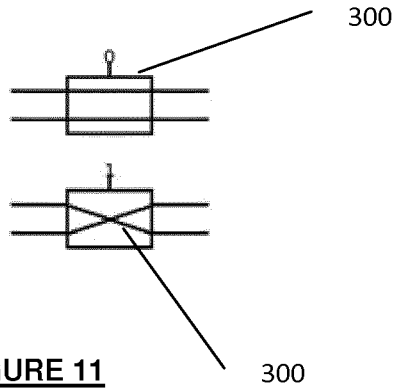


FIGURE 11

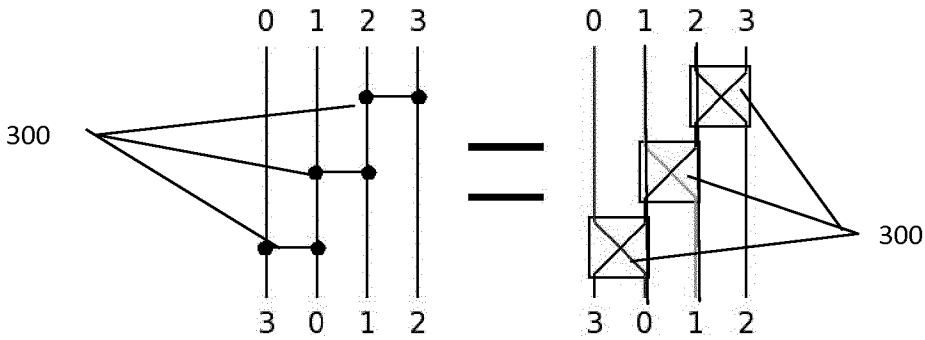


FIGURE 12

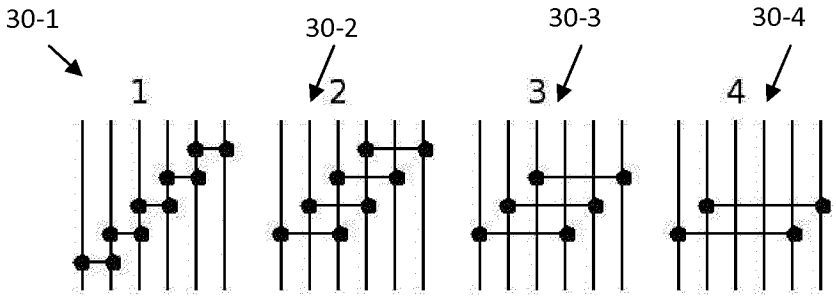


FIGURE 13

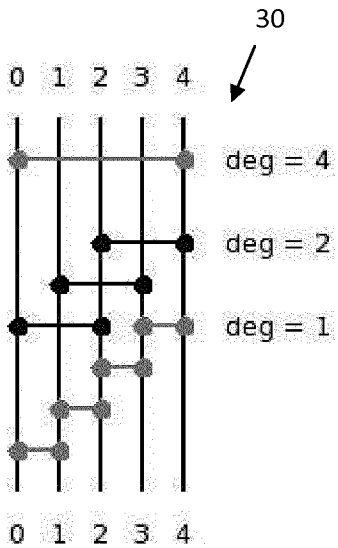


FIGURE 14

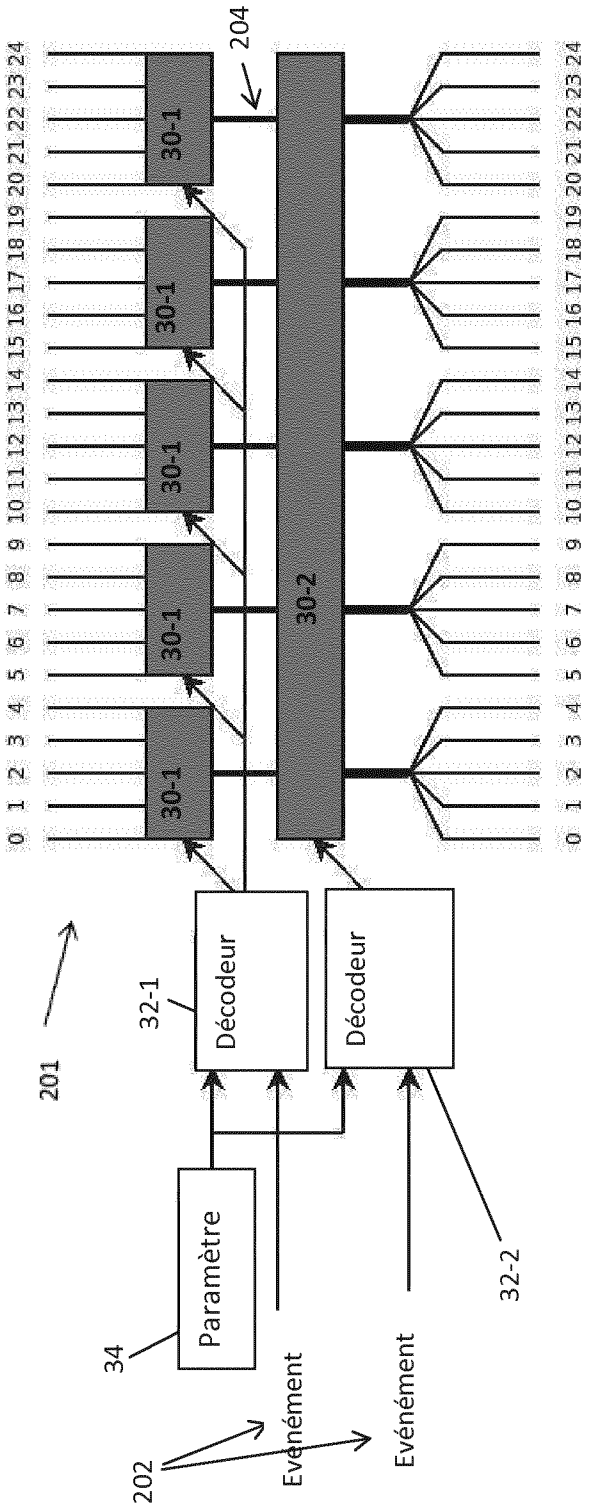

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
30-1
30-2
204
201
32-1
Décodeur
Décodeur
32-2
34
Paramètre
202
Evénement
Evénement

FIGURE 15

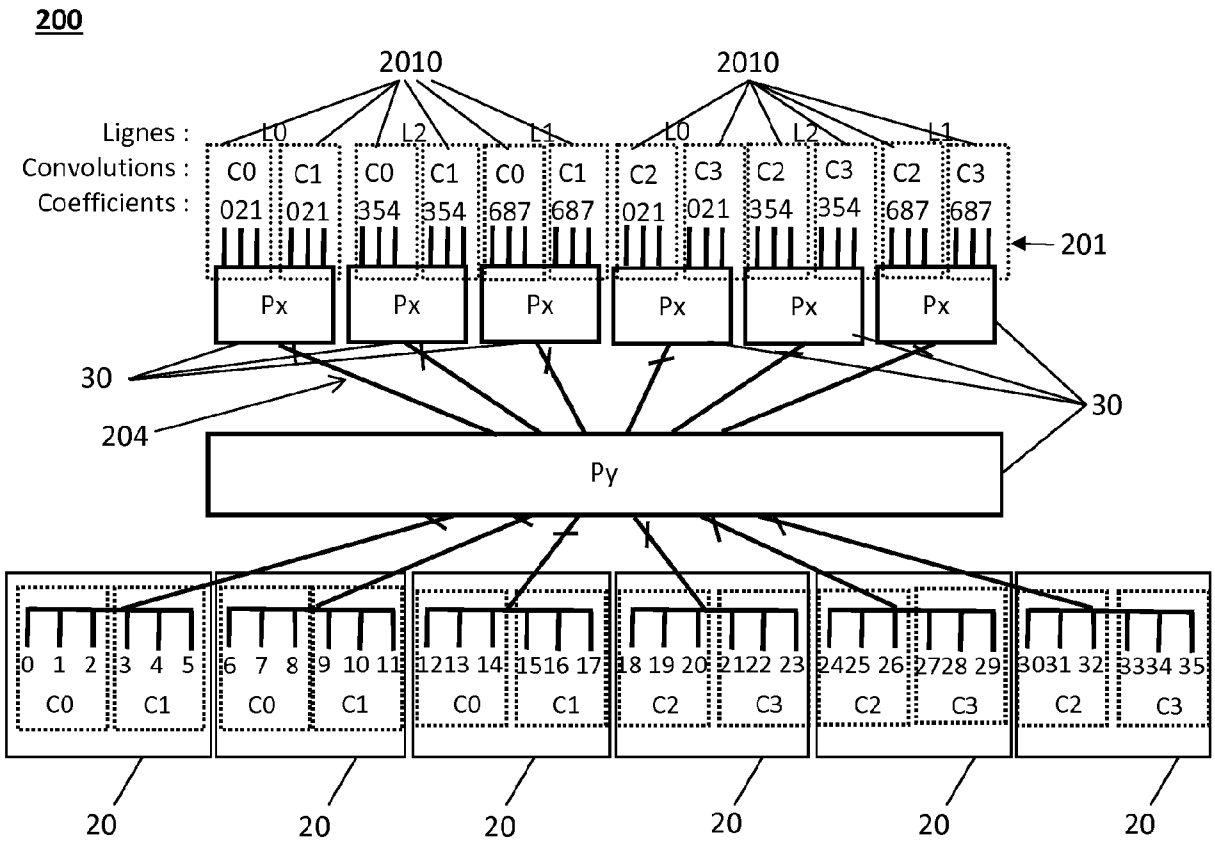

200
2010
2010
Lignes :
Convolutions :
Coefficients :
L0
L2
L1
C0
C1
021
354
687
C2
C3
201
Px
30
204
Py
0 1 2 3 4 5
6 7 8 9 10 11
12 13 14 15 16 17
18 19 20 21 22 23
24 25 26 27 28 29
30 31 32 33 34 35
20

FIGURE 16

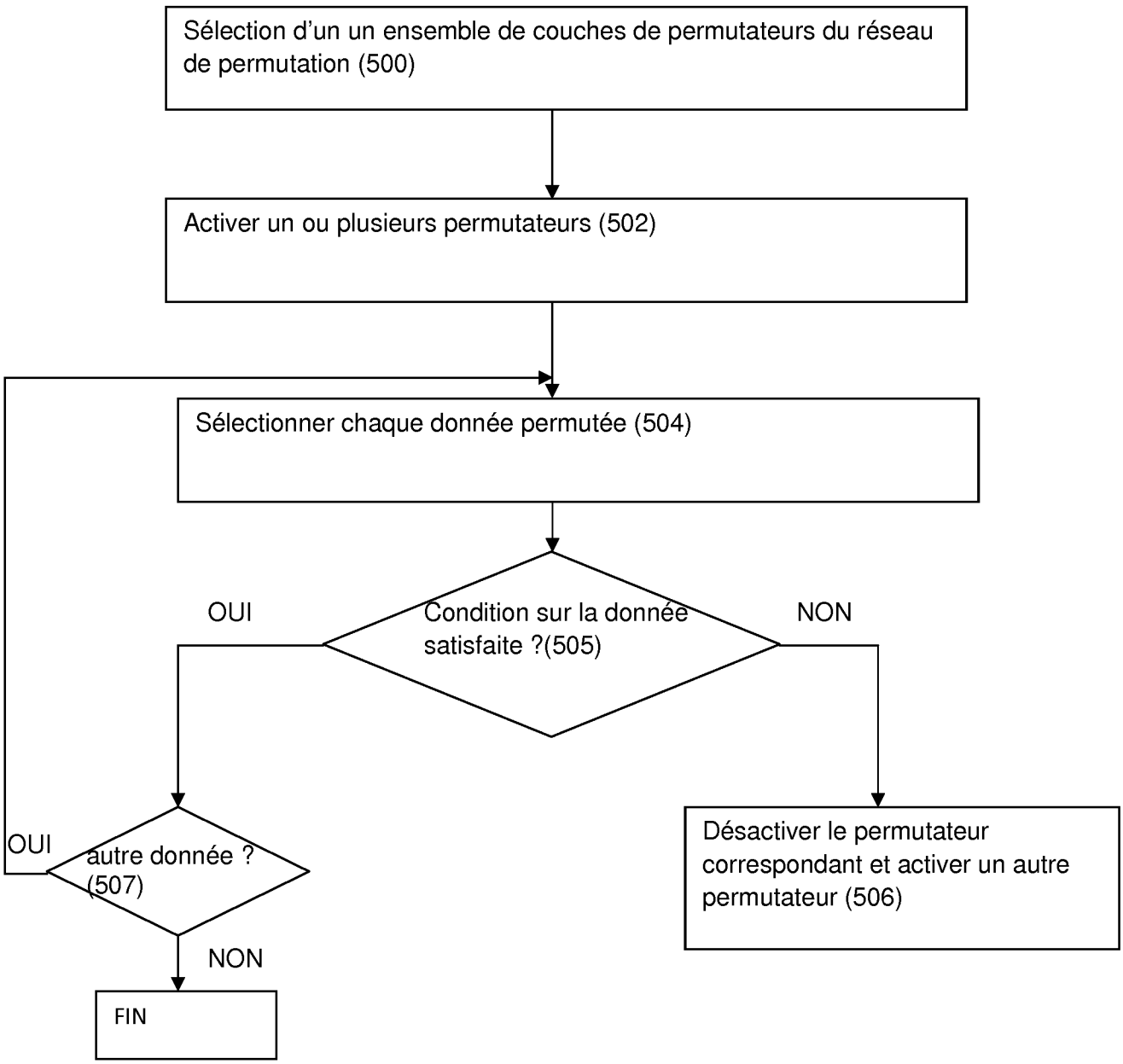

Sélection d'un un ensemble de couches de permutateurs du réseau de permutation (500)
Activer un ou plusieurs permutateurs (502)
Sélectionner chaque donnée permutée (504)
Condition sur la donnée satisfaite ?(505)
OUI
NON
Désactiver le permutateur correspondant et activer un autre permutateur (506)
autre donnée ? (507)
OUI
NON
FIN

FIGURE 17

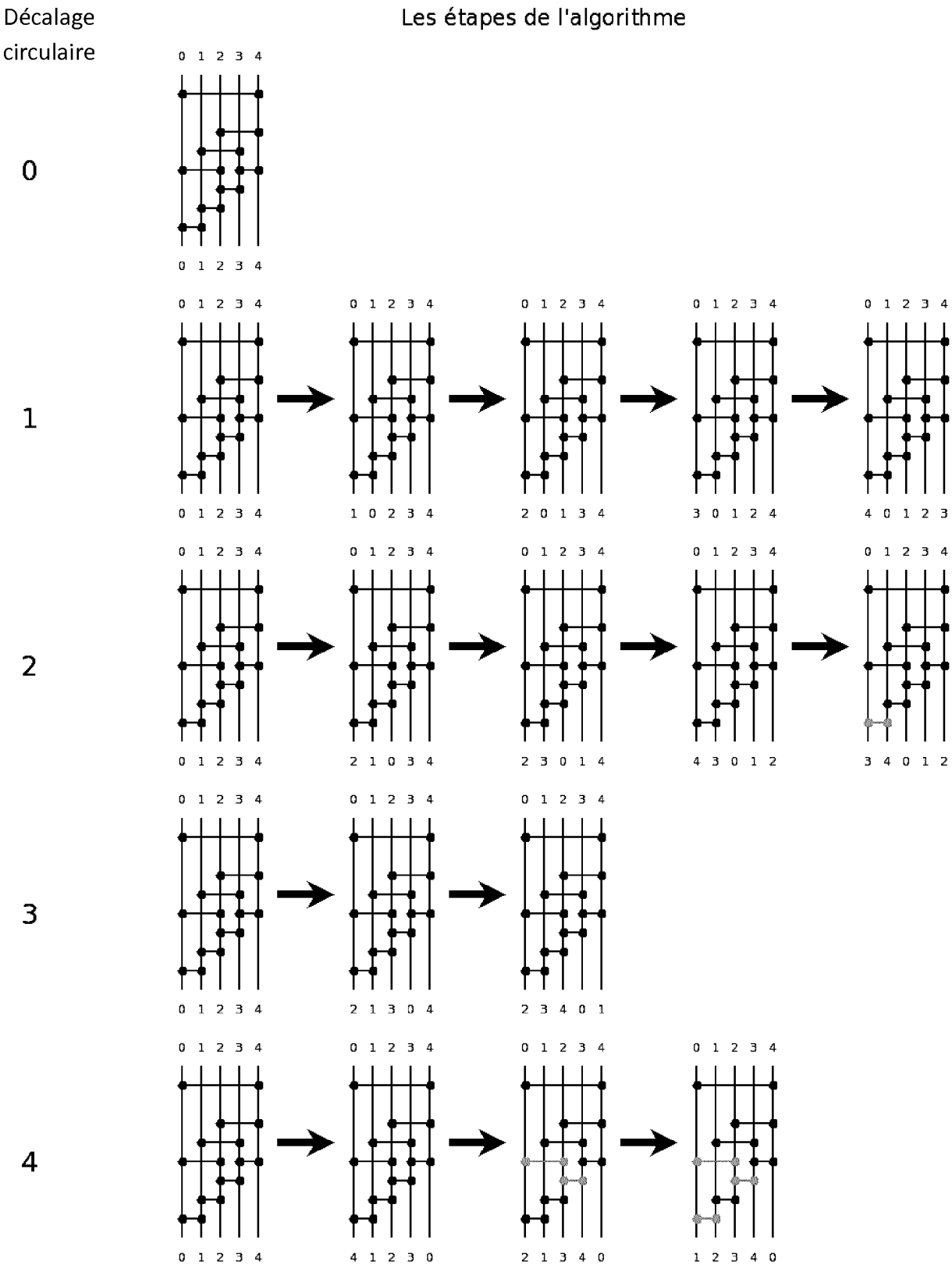

Décalage circulaire
Les étapes de l'algorithme
0
1
2
3
4
0 1 2 3 4
1 0 2 3 4
2 0 1 3 4
3 0 1 2 4
4 0 1 2 3
2 1 0 3 4
2 3 0 1 4
4 3 0 1 2
3 4 0 1 2
2 1 3 0 4
2 3 4 0 1
4 1 2 3 0
2 1 3 4 0
1 2 3 4 0

FIGURE 18

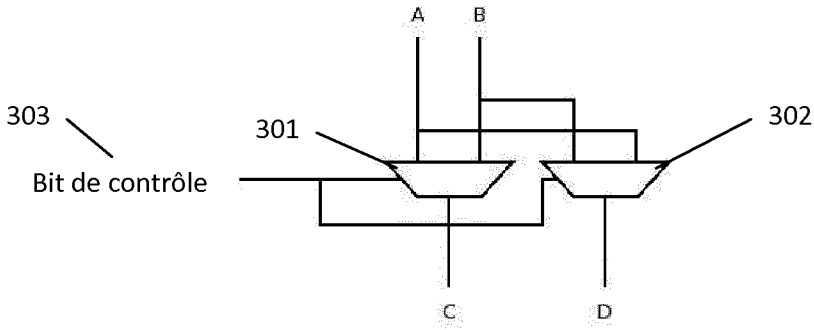


FIGURE 19

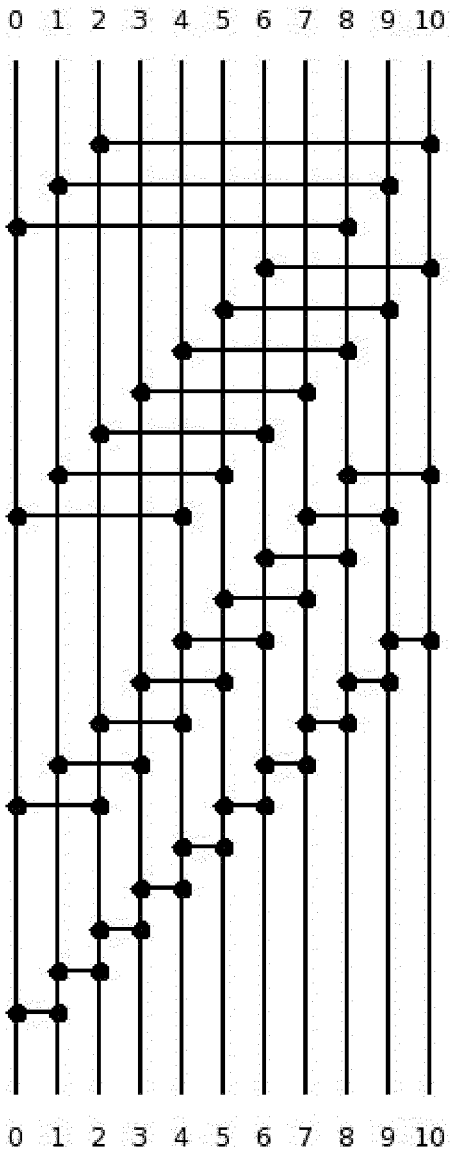


FIGURE 20

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2017/060017

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06N3/04 G06N3/063
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06N G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	FR 3 025 344 A1 (COMMISSARIAT ENERGIE ATOMIQUE [FR]) 4 March 2016 (2016-03-04) page 7, line 5 - page 7, line 27 page 20 - page 24 page 27 - page 32	1-14
A	TSE-YUN FENG: "A Survey of Interconnection Networks", COMPUTER, IEEE COMPUTER SOCIETY, vol. 14, no. 12, 1 December 1981 (1981-12-01), pages 12-27, XP001610985, ISSN: 0018-9162 the whole document	1-14

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 July 2017	20/07/2017

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Cilia, Elisa

1

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ABHAY B RATHOD ET AL: "Parallel Routing Algorithms in Benes and Clos Networks: A Survey", INTERNATIONAL JOURNAL OF ADVANCE FOUNDATION AND RESEARCH IN COMPUTER (IJAFRC), vol. 2, no. 1, 1 January 2015 (2015-01-01) , pages 21-31, XP055342092, ISSN: 2348-4853 the whole document	1-14
A	WO 2015/090885 A1 (COMMISSARIAT ENERGIE ATOMIQUE [FR]) 25 June 2015 (2015-06-25) the whole document	1-14
A	US 7 237 055 B1 (RUPP CHARLE R [US]) 26 June 2007 (2007-06-26) paragraph [0031] - paragraph [0032] paragraph [0034] - paragraph [0034]	1-14

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2017/060017

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
FR 3025344 A1	04-03-2016	EP 3186752 A1	05-07-2017
		FR 3025344 A1	04-03-2016
		WO 2016030230 A1	03-03-2016
WO 2015090885 A1	25-06-2015	EP 3084588 A1	26-10-2016
		FR 3015068 A1	19-06-2015
		US 2016292566 A1	06-10-2016
		WO 2015090885 A1	25-06-2015
US 7237055 B1	26-06-2007	US 7237055 B1	26-06-2007
		US 2007250656 A1	25-10-2007

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale n°
PCT/EP2017/060017

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
INV. G06N3/04 G06N3/063
ADD.

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
G06N G06F

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si cela est réalisable, termes de recherche utilisés)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	FR 3 025 344 A1 (COMMISSARIAT ENERGIE ATOMIQUE [FR]) 4 mars 2016 (2016-03-04) page 7, ligne 5 - page 7, ligne 27 page 20 - page 24 page 27 - page 32 -----	1-14
A	TSE-YUN FENG: "A Survey of Interconnection Networks", COMPUTER, IEEE COMPUTER SOCIETY, vol. 14, no. 12, 1 décembre 1981 (1981-12-01), pages 12-27, XP001610985, ISSN: 0018-9162 le document en entier -----	1-14
	-/--	

[X] Voir la suite du cadre C pour la fin de la liste des documents

[X] Les documents de familles de brevets sont indiqués en annexe

* Catégories spéciales de documents cités:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent

"E" document antérieur, mais publié à la date de dépôt international ou après cette date

"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)

"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens

"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention

"X" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément

"Y" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier

"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée	Date d'expédition du présent rapport de recherche internationale
11 juillet 2017	20/07/2017

Nom et adresse postale de l'administration chargée de la recherche internationale	Fonctionnaire autorisé
Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Cilia, Elisa

1

C(suite). DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	ABHAY B RATHOD ET AL: "Parallel Routing Algorithms in Benes and Clos Networks: A Survey", INTERNATIONAL JOURNAL OF ADVANCE FOUNDATION AND RESEARCH IN COMPUTER (IJAFRC), vol. 2, no. 1, 1 janvier 2015 (2015-01-01) , pages 21-31, XP055342092, ISSN: 2348-4853 le document en entier	1-14
A	WO 2015/090885 A1 (COMMISSARIAT ENERGIE ATOMIQUE [FR]) 25 juin 2015 (2015-06-25) le document en entier	1-14
A	US 7 237 055 B1 (RUPP CHARLE R [US]) 26 juin 2007 (2007-06-26) alinéa [0031] - alinéa [0032] alinéa [0034] - alinéa [0034]	1-14

1

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Demande internationale n°
PCT/EP2017/060017

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
FR 3025344 A1	04-03-2016	EP 3186752 A1	05-07-2017
		FR 3025344 A1	04-03-2016
		WO 2016030230 A1	03-03-2016
WO 2015090885 A1	25-06-2015	EP 3084588 A1	26-10-2016
		FR 3015068 A1	19-06-2015
		US 2016292566 A1	06-10-2016
		WO 2015090885 A1	25-06-2015
US 7237055 B1	26-06-2007	US 7237055 B1	26-06-2007
		US 2007250656 A1	25-10-2007