(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

COURBEVOIE

(11) **N° de publication :** **3 090 163**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **18 73141**

(51) Int Cl[8] : ***G 06 N 3/06*** *(2019.01)*

(12) **BREVET D'INVENTION** **B1**

(54) Processeur de traitement de données, procédé et programme d'ordinateur correspondant.

(22) **Date de dépôt :** 18.12.18.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 19.06.20 Bulletin 20/25.

(45) **Date de la mise à disposition du public du brevet d'invention :** 30.04.21 Bulletin 21/17.

(56) **Liste des documents cités dans le rapport de recherche :**

Se reporter à la fin du présent fascicule

(60) **Références à d'autres documents nationaux apparentés :**

◯ **Demande(s) d'extension :**

(71) **Demandeur(s) :** *UNIVERSITE DE BOURGOGNE Etablissement public national scientifique, culturel et professionnel* —FR et *UNIVERSITÉ DE TECHNOLOGIE DE TROYES Etablissement public national scientifique, culturel et professionnel* — FR.

(72) **Inventeur(s) :** DOUSSOT Michel et PAINDAVOINE Michel.

(73) **Titulaire(s) :** UNIVERSITE DE BOURGOGNE Etablissement public national scientifique, culturel et professionnel, UNIVERSITÉ DE TECHNOLOGIE DE TROYES Etablissement public national scientifique, culturel et professionnel.

(74) **Mandataire(s) :** Cabinet Vidon brevets et stratégies.

FR 3 090 163 - B1

Description

Titre de l'invention : Processeur de traitement de données, procédé et programme d'ordinateur correspondant.

1. Domaine Technique

[0001] L'invention se rapporte à la matérialisation de réseaux de neurones. Plus particulièrement, l'invention se rapporte à la mise en œuvre physique de réseaux de neurones adaptables et configurables. Plus spécifiquement encore, l'invention se rapporte à la mise en œuvre d'un réseau de neurones génériques dont la configuration et le fonctionnement peut être adaptée en fonction des besoins.

2. Art antérieur

[0002] Dans le domaine du traitement informatisé de données, un réseau de neurones est un système numérique dont la conception est originellement inspirée du fonctionnement des neurones biologiques. Un réseau de neurones est plus généralement modélisé sous la forme d'un système comprenant une algorithmique de traitement et des données statistiques (comprenant notamment des poids). L'algorithmique de traitement permet de traiter des données d'entrée, lesquelles sont combinées avec les données statistiques pour obtenir des résultats en sortie. L'algorithmique de traitement consiste à définir les calculs qui sont réalisées sur les données d'entrée en combinaison avec les données statistiques du réseau pour fournir des résultats de sortie. Parallèlement, les réseaux de neurones informatisés sont divisés en couches. Ils présentent généralement une couche d'entrée, une ou plusieurs couches intermédiaires et une couche de sortie. Le fonctionnement général du réseau de neurones informatisé, et donc le traitement général appliqué sur les données d'entrées consiste à mettre en œuvre un processus algorithmique itératif de traitement, dans lequel les données d'entrées sont traitées par la couche d'entrée, laquelle produit des données de sortie, ces données de sortie devenant des données d'entrées de la couche suivante et ainsi de suite, autant de fois qu'il y a de couches, jusqu'à obtenir les données finales de sortie, qui sont délivrées par la couche de sortie.

[0003] Comme l'objet initial du réseau de neurones artificiels était de mimer le fonctionnement d'un réseau de neurones biologiques, l'algorithmique utilisée pour combiner les données d'entrée et les données statistiques d'une couche du réseau comprend des traitements qui tentent d'imiter le fonctionnement d'un neurone biologique. On considère ainsi, dans un réseau de neurones artificiels (simplement appelé réseau de neurones dans la suite), qu'un neurone comprend généralement d'une part une fonction de combinaison et une fonction d'activation. Cette fonction de combinaison et cette fonction d'activation sont mises en œuvre de manière informatisée par

l'utilisation d'un algorithme associé au neurone ou à un ensemble de neurones situés dans une même couche.

[0004] La fonction de combinaison sert à combiner les données d'entrées avec les données statistiques (les poids synaptiques). Les données d'entrées sont matérialisées sous la forme d'un vecteur, chaque point du vecteur représentant une valeur donnée. Les valeurs statistiques (i.e. poids synaptiques) sont également représentées par un vecteur. La fonction de combinaison est donc formalisée comme étant une fonction vecteur-à-scalaire, ainsi :

- dans les réseaux de neurones de type MLP (perceptron multicouches), un calcul d'une combinaison linéaire des entrées est effectué, c'est-à-dire que la fonction de combinaison renvoie le produit scalaire entre le vecteur des entrées et le vecteur des poids synaptiques ;
- dans les réseaux de neurones de type RBF (*« radial basis function »*), un calcul de la distance entre les entrées est effectué, c'est-à-dire que la fonction de combinaison renvoie la norme euclidienne du vecteur issu de la différence vectorielle entre le vecteur d'entrée et le vecteur correspondant aux poids synaptiques.

[0005] La fonction d'activation, pour sa part, est utilisée pour effectuer une rupture de linéarité dans le fonctionnement du neurone. Les fonctions de seuillage présentent généralement trois intervalles :

- en dessous du seuil, le neurone est non-actif (souvent dans ce cas, sa sortie vaut 0 ou -1) ;
- aux alentours du seuil, une phase de transition ;
- au-dessus du seuil, le neurone est actif (souvent dans ce cas, sa sortie vaut 1).

[0006] Parmi les fonctions d'activation classiques, on retrouve par exemple :

- La fonction sigmoïde ;
- La fonction tangente hyperbolique ;
- La fonction de Heaviside.

[0007] D'innombrables publications portent sur les réseaux de neurones. D'une manière générale, ces publications portent sur des aspects théoriques des réseaux de neurones (comme la recherche de nouvelles fonctions d'activations, ou encore sur la gestion des couches, ou encore sur la rétroaction ou encore sur l'apprentissage ou plus précisément sur la descente de gradient dans les sujets relatifs au « machine learning »). D'autres publications portent sur l'utilisation pratique de systèmes mettant en œuvre des réseaux de neurones informatisés pour répondre à telle ou telle problématique. Moins fréquemment, on trouve également des publications relatives à l'implémentation, sur un composant spécifique, de réseaux de neurones particuliers. C'est par exemple le cas de la publication *« FPGA Implementation of Convolutional Neural Networks with Fixed -*

Point Calculations » de Roman A. Solovye et Al (2018), dans laquelle il est proposé de localiser les calculs effectués au sein d'un réseau de neurones sur un composant matériel. L'implémentation matérielle proposée dans ce document est cependant limitée en termes de portée. En effet, elle se limite à la mise en œuvre d'un réseau de neurones convolutif dans lequel de nombreuses réductions sont réalisées. Elle apporte cependant une mise en œuvre de calculs en virgule fixe ou en virgule flottante. L'article « *Implementation of Fixed-point Neuron Models with Threshold, Ramp and Sigmoid Activation Functions* » de Lei Zhang (2017) porte également sur la mise en œuvre d'un réseau de neurones comprenant la mise en œuvre de calculs à virgule fixe pour un neurone particulier et trois fonctions d'activation particulières, unitairement implémentées.

[0008] Cependant, les solutions décrites dans ces articles ne permettent pas de résoudre les problématiques d'implémentation matérielles de réseaux de neurones génériques, à savoirs des réseaux de neurones implémentant des neurones généraux, qui peuvent mettre en œuvre une multiplicité de type de réseaux de neurones, incluant des réseaux de neurones mixtes comprenant plusieurs fonctions d'activation et/ou plusieurs fonctions de combinaison.

[0009] Il existe donc un besoin de fournir un dispositif qui permette de mettre en œuvre un réseau de neurones, implémentant des neurones de manière fiable et efficace, qui soit de plus reconfigurable et qui puisse prendre place sur une surface de processeur réduite.

3. Résumé de l'invention

[0010] L'invention ne pose pas au moins un des problèmes de l'art antérieur. Plus particulièrement, l'invention se rapporte à un processeur de traitement de données, ledit processeur comprenant au moins une mémoire de traitement et une unité de calcul, le dit processeur étant caractérisé en ce que l'unité de calcul comprend un ensemble d'unités de calculs configurables appelées neurones configurables, chaque neurone configurable de l'ensemble de neurones configurables comprenant un module de calcul de fonctions de combinaison et un module de calcul de fonctions d'activation, chaque module de calcul de fonctions d'activation comprenant un registre de réception d'une commande de paramétrage, de sorte que ladite commande détermine une fonction d'activation à exécuter parmi au moins deux fonctions d'activation exécutables par le module de calcul de fonctions d'activation.

[0011] Ainsi, l'invention permet de paramétrer, à l'exécution, un ensemble de neurones reconfigurables, afin que ceux-ci exécutent une fonction prédéterminée selon le mot de commande fourni aux neurones lors de l'exécution. Le mot de commande, reçu dans un espace mémoire, pouvant être dédié, du neurone reconfigurable, peut être différent pour chaque couche d'un réseau de neurones particulier, et ainsi faire partie des pa-

ramètres du réseau de neurones à exécuter (implémenter) sur le processeur en question.

[0012] Selon un mode de réalisation particulier, caractérisé en ce que les au moins deux fonctions d'activation exécutables par le module de calcul de fonctions d'activation appartiennent au groupe comprenant :

- la fonction sigmoïde ;
- la fonction tangente hyperbolique ;
- la fonction gaussienne ;
- la fonction RELU *(« Rectified linear Unit »).*

[0013] Ainsi, un neurone reconfigurable est en mesure de mettre en œuvre les principales fonctions d'activation utilisées pour l'industrie.

[0014] Selon un mode de réalisation particulier, le module de calcul de fonctions d'activation est configuré pour effectuer une approximation desdites au moins deux fonctions d'activation.

[0015] Ainsi, la capacité de calcul du processeur neuronal embarquant un ensemble de neurones reconfigurables peut être réduite entraînant une réduction de la taille, de la consommation et donc de l'énergie nécessaire à la mise en œuvre de la technique proposée par rapport aux techniques existantes.

[0016] Selon une caractéristique particulière, le module de calcul de fonctions d'activation comprend un sous module de calcul d'une opération de base correspondant à une approximation du calcul de la sigmoïde de la valeur absolue de λx :

[0017] [Math 1] $$f(x) = \frac{1}{1 + e^{|\lambda x|}}.$$

[0018] Ainsi, à l'aide d'une opération de base, il est possible d'approcher, par une série de calculs simples, le résultat d'une fonction d'activation particulière, définie par un mot de commande.

[0019] Selon un mode de réalisation particulier, l'approximation desdites au moins deux fonctions d'activation est effectuée en fonction d'un paramètre d'approximation λ.

[0020] Le paramètre d'approximation λ peut ainsi être utilisé, conjointement au mot de commande, pour définir le comportement de l'unité de calcul de l'opération de base pour calculer une approximation circonstanciée de la fonction d'activation du mot de commande. En d'autres termes, le mot de commande route le calcul (réalise un routage du calcul) à effectuer dans l'unité de calcul de la fonction d'activation tandis que le paramètre d'approximation λ conditionne (paramètre) ce calcul.

[0021] Selon une caractéristique particulière, l'approximation desdites au moins deux fonctions d'activation est effectuée en configurant le module de calcul de fonctions d'activation pour que les calculs soient effectués en virgule fixe ou virgule flottante.

[0022] Lorsqu'effectué en virgule fixe, ceci permet avantageusement de réduire encore les ressources nécessaires à la mise en œuvre de la technique proposée, et donc de réduire

encore la consommation en énergie. Une telle implémentation est avantageuse pour des dispositifs à faible capacité/faible consommation tels que les objets connectés.

[0023] Selon une caractéristique particulière, le nombre de bits associés aux calculs en virgule fixe ou virgule flottante est paramétré pour chaque couche du réseau. Ainsi, un paramètre complémentaire peut être enregistré dans les ensembles de paramètres de couches du réseau de neurones.

[0024] Selon un mode de réalisation particulier, le processeur de traitement de données comprend une mémoire de stockage de configuration du réseau au sein de laquelle des paramètres (PS, cmd, λ) d'exécution de réseau de neurones sont enregistrés.

[0025] Selon une autre implémentation, l'invention se rapporte également à un procédé de traitement de données, ledit procédé étant mis en œuvre par un processeur de traitement de données comprenant au moins une mémoire de traitement et une unité de calcul, l'unité de calcul comprend un ensemble d'unités de calculs configurables appelées neurones configurables, chaque neurone configurable de l'ensemble de neurones configurables comprenant un module de calcul de fonctions de combinaison et un module de calcul de fonctions d'activation, le procédé comprenant :

- une étape d'initialisation comprenant le chargement dans la mémoire de traitement d'un un ensemble de données applicatives et le chargement d'un ensemble de données, correspondant à l'ensemble des poids synaptiques et des configurations des couches dans la mémoire de stockage de configuration du réseau ;
- l'exécution du réseau de neurone, selon une mise en œuvre itérative, comprenant pour chaque couche, l'application d'une commande de paramétrage, de sorte que ladite commande détermine une fonction d'activation à exécuter parmi au moins deux fonctions d'activation exécutables par le module de calcul de fonctions d'activation, l'exécution délivrant des données traitées ;
- la transmission des données traitées à une application appelante.

[0026] Les avantages procurés par un tel procédé sont similaires à ceux précédemment énoncés. Le procédé peut cependant être mis en œuvre sur tout type de processeur.

[0027] Selon un mode de réalisation particulier, l'exécution du réseau de neurone comprend au moins une itération des étapes suivantes, pour une couche courante du réseau de neurones :

- transmission d'au moins un mot de commande, définissant la fonction de combinaison et/ou la fonction d'activation mise en œuvre pour la couche courante ;
- chargement des poids synaptiques de la couche courante ;
- chargement des données d'entrée en provenance de la mémoire de stockage

temporaire ;

- calcul de la fonction de combinaison, pour chaque neurone et chaque vecteur d'entrée, en fonction dudit au moins un mot de commande, délivrant, pour chaque neurone utilisé, un scalaire intermédiaire ;
- calcul de la fonction d'activation en fonction du scalaire intermédiaire, et dudit au moins un deuxième mot de commande, délivrant, pour chaque neurone utilisé, un résultat d'activation ;
- enregistrement du résultat d'activation dans la mémoire de stockage temporaire.

[0028] Ainsi, l'invention permet, au sein d'un processeur dédié (ou bien au sein d'un procédé de traitement spécifique) de réaliser des optimisations des calculs des fonctions non linéaires en effectuant des factorisations de calculs et des approximations qui permettent de diminuer la charge de calcul des opérations, notamment au niveau de la fonction d'activation.

[0029] Il est entendu, dans le cadre de la description de la présente technique selon l'invention, qu'une étape de transmission d'une information et/ou d'un message d'un premier dispositif à un deuxième dispositif, correspond au moins partiellement, pour ce deuxième dispositif à une étape de réception de l'information et/ou du message transmis, que cette réception et cette transmission soit directe ou qu'elle s'effectue par l'intermédiaire d'autres dispositifs de transport, de passerelle ou d'intermédiation, incluant les dispositifs décrits dans la présente selon l'invention.

[0030] Selon une implémentation générale, les différentes étapes des procédés selon l'invention sont mises en œuvre par un ou plusieurs logiciels ou programmes d'ordinateur, comprenant des instructions logicielles destinées à être exécutées par un processeur de données d'un dispositif d'exécution selon l'invention et étant conçu pour commander l'exécution des différentes étapes des procédés, mis en œuvre au niveau du terminal de communication, du dispositif électronique d'exécution et/ou du serveur distant, dans le cadre d'une répartition des traitements à effectuer et déterminés par un code source scripté.

[0031] En conséquence, l'invention vise aussi des programmes, susceptibles d'être exécutés par un ordinateur ou par un processeur de données, ces programmes comportant des instructions pour commander l'exécution des étapes des procédés tel que mentionnés ci-dessus.

[0032] Un programme peut utiliser n'importe quel langage de programmation, et être sous la forme de code source, code objet, ou de code intermédiaire entre code source et code objet, tel que dans une forme partiellement compilée, ou dans n'importe quelle autre forme souhaitable.

[0033] L'invention vise aussi un support d'informations lisible par un processeur de

données, et comportant des instructions d'un programme tel que mentionné ci-dessus.

[0034] Le support d'informations peut être n'importe quelle entité ou dispositif capable de stocker le programme. Par exemple, le support peut comporter un moyen de stockage, tel qu'une ROM, par exemple un CD ROM ou une ROM de circuit microélectronique, ou encore un moyen d'enregistrement magnétique, par exemple un support mobile (carte mémoire) ou un disque dur ou un SSD.

[0035] D'autre part, le support d'informations peut être un support transmissible tel qu'un signal électrique ou optique, qui peut être acheminé via un câble électrique ou optique, par radio ou par d'autres moyens. Le programme selon l'invention peut être en particulier téléchargé sur un réseau de type Internet.

[0036] Alternativement, le support d'informations peut être un circuit intégré dans lequel le programme est incorporé, le circuit étant adapté pour exécuter ou pour être utilisé dans l'exécution du procédé en question.

[0037] Selon un mode de réalisation, l'invention est mise en œuvre au moyen de composants logiciels et/ou matériels. Dans cette optique, le terme "module" peut correspondre dans ce document aussi bien à un composant logiciel, qu'à un composant matériel ou à un ensemble de composants matériels et logiciels.

[0038] Un composant logiciel correspond à un ou plusieurs programmes d'ordinateur, un ou plusieurs sous-programmes d'un programme, ou de manière plus générale à tout élément d'un programme ou d'un logiciel apte à mettre en œuvre une fonction ou un ensemble de fonctions, selon ce qui est décrit ci-dessous pour le module concerné. Un tel composant logiciel est exécuté par un processeur de données d'une entité physique (terminal, serveur, passerelle, set-top-box, routeur, *etc.*) et est susceptible d'accéder aux ressources matérielles de cette entité physique (mémoires, supports d'enregistrement, bus de communication, cartes électroniques d'entrées/sorties, interfaces utilisateur, *etc.*).

[0039] De la même manière, un composant matériel correspond à tout élément d'un ensemble matériel (ou hardware) apte à mettre en œuvre une fonction ou un ensemble de fonctions, selon ce qui est décrit ci-dessous pour le module concerné. Il peut s'agir d'un composant matériel programmable ou avec processeur intégré pour l'exécution de logiciel, par exemple un circuit intégré, une carte à puce, une carte à mémoire, une carte électronique pour l'exécution d'un micrologiciel (firmware), etc.

[0040] Chaque composante du système précédemment décrit met bien entendu en œuvre ses propres modules logiciels.

[0041] Les différents modes de réalisation mentionnés ci-dessus sont combinables entre eux pour la mise en œuvre de l'invention.

4. Présentation des dessins

[0042] D'autres caractéristiques et avantages de l'invention apparaîtront plus clairement à la

lecture de la description suivante d'un mode de réalisation préférentiel, donné à titre de simple exemple illustratif et non limitatif, et des dessins annexés, parmi lesquels :

- [fig 1] décrit un processeur dans lequel l'invention est mise en œuvre ;
- [fig 2] illustre le découpage de la fonction d'activation d'un neurone configurable selon l'invention ;
- [fig 3] décrit l'enchaînement des blocs dans un mode de réalisation particuliers, pour le calcul d'une valeur approchante de la fonction d'activation ;
- [fig 4] décrit un mode de réalisation d'un procédé de traitement de données au sein d'un réseau de neurones selon l'invention.

5. Description détaillée

[0043] 5.1. Exposé du principe technique

[0044] 5.1.1. Généralités

[0045] Confrontés à la problématique de mise en œuvre d'un réseau de neurones adaptable et configurable, les inventeurs se sont penchés sur la matérialisation des calculs à mettre en œuvre dans différentes configurations. Comme explicité précédemment, il s'avère que les réseaux de neurones se différencient entre eux principalement par les calculs effectués. Plus particulièrement, les couches qui composent un réseau de neurones mettent en œuvre des neurones unitaires qui réalisent à la fois des fonctions de combinaison et des fonctions d'activation qui peuvent être différentes d'un réseau à l'autre. Or, sur un dispositif électronique donné, tel qu'un smartphone, une tablette ou un ordinateur personnel, de nombreux réseaux de neurones différents peuvent être mis en œuvre, chacun de ces réseaux de neurones étant utilisé par des applications ou des processus différents. Dès lors, dans un souci d'implémentation matérielle efficace de tels réseaux de neurones, il n'est pas envisageable de disposer d'un composant matériel dédié par type de réseau de neurones à mettre en œuvre. C'est pour cette raison que majoritairement, les réseaux de neurones actuels sont mis en œuvre de manière purement logicielle et non matériellement (c'est-à-dire en utilisant directement des instructions de processeurs). Partant de ce constat, comme exposé précédemment, les inventeurs ont développé et mis au point un neurone spécifique qui peut être matériellement reconfigurable. Grâce à un mot de commande, un tel neurone peut prendre la forme qui convient dans un réseau de neurones en cours d'exécution. Plus particulièrement, dans au moins un mode de réalisation, l'invention se matérialise sous la forme d'un processeur générique. Les calculs effectués par ce processeur générique peuvent, en fonction de modes de réalisation, être réalisés en virgule fixe ou en virgule flottante. Lorsqu'ils sont réalisés en virgules fixe, les calculs peuvent avantageusement être mis en œuvre sur des plateformes disposant de peu de ressources de calcul et de traitement, tels que de petits dispositifs de type objets connectés. Le processeur fonctionne avec un apprentissage hors-ligne. Il comprend une mémoire comprenant

notamment : les poids synaptiques des différentes couches ; le choix de la fonction d'activation de chaque couche ; ainsi que des paramètres de configuration et d'exécution des neurones de chaque couche. Le nombre de neurones et de couches cachées dépend de la mise en œuvre opérationnelle et de considérations économiques et pratiques. Plus particulièrement, la mémoire du processeur est dimensionnée en fonction de la capacité maximale que l'on souhaite offrir au réseau de neurones. Une structure de mémorisation des résultats d'une couche, également présente au sein du processeur, permet de réutiliser les mêmes neurones pour plusieurs couches cachées consécutives. Cette structure de mémorisation est, dans un objectif de simplification, appelée mémoire de stockage temporaire. Ainsi, le nombre de neurones reconfigurable du composant (processeur) est lui aussi sélectionné en fonction du nombre maximal de neurones que l'on souhaite autoriser pour une couche donnée du réseau de neurones.

[0046] [fig.1] La figure 1 illustre succinctement le principe général de l'invention. Un processeur, comprend une pluralité de neurones configurable (seize neurones sont représentés sur la figure). Chaque neurone est composé de deux unités distinctes : une unité de calcul de la fonction de combinaison et une unité de calcul de la fonction d'activation (AFU). Chacune de ces deux unités est configurable par un mot de commande (cmd). Les neurones sont adressés par des bus de connexion (CBUS) et des routages de connexion (CROUT). Les données d'entrée sont représentées sous la forme d'un vecteur ($\overrightarrow{X_i}$) qui contient un certain nombre de valeurs d'entrées (huit valeurs dans l'exemple). Les valeurs sont routées dans le réseau pour produire huit scalaires résultats (z0,..., z7). Les poids synaptiques, les commandes et le paramètre d'ajustement λ sont décrits par la suite. Ainsi, l'invention se rapporte à un processeur de traitement de données, ledit processeur comprenant au moins une mémoire de traitement (MEM) et une unité de calcul (CU), le dit processeur étant caractérisé en ce que l'unité de calcul (CU) comprend un ensemble d'unités de calculs configurables appelées neurones configurables, chaque neurone configurable (NC) de l'ensemble de neurones configurables (ENC) comprenant un module de calcul de fonctions de combinaison (MCFC) et un module de calcul de fonctions d'activation (MCFA), chaque module de calcul de fonctions d'activation (AFU) comprenant un registre de réception d'une commande de paramétrage, de sorte que ladite commande détermine une fonction d'activation à exécuter parmi au moins deux fonctions d'activation exécutables par le module de calcul de fonctions d'activation (AFU). Le processeur comprend également une mémoire de stockage de configuration du réseau (MEMR) au sein de laquelle des paramètres (PS, cmd, λ) d'exécution de réseau de neurones sont enregistrés. Cette mémoire peut être la même que la mémoire de traitement (MEM).

[0047] On expose par la suite différentes caractéristiques du processeur objet de l'invention et plus particulièrement la structure et les fonctions d'un neurone reconfigurable.

[0048] 5.1.2. Neurone configurable

[0049] Un neurone configurable du réseau de neurones configurables objet de l'invention comprend deux modules (unités) de calcul qui sont paramétrables : un en charge du calcul de la fonction de combinaison et un en charge du calcul de la fonction d'activation. Cependant, selon l'invention, afin de rendre l'implémentation du réseau efficace et performante, les inventeurs ont en quelque sorte simplifié et factorisé (mutualisé) les calculs, de sorte qu'un maximum de calculs en commun puisse être effectué par ces modules. Plus particulièrement, le module de calcul de la fonction d'activation (également appelé AFU) optimise les calculs communs à l'ensemble des fonctions d'activation, en simplifiant et en approximant ces calculs. Une mise en œuvre illustrative est détaillée par la suite. De façon imagée, le module de calcul de la fonction d'activation effectue des calculs de sorte à reproduire un résultat proche de celui de la fonction d'activation choisie, en mettant en commun les parties de calcul qui servent à reproduire une approximation de la fonction d'activation.

[0050] Le neurone artificiel, dans ce mode de réalisation, est décomposé en deux éléments (modules) paramétrables. Le premier élément (module) paramétrable calcule soit le produit scalaire (la plupart des réseaux) ou la distance euclidienne. Le deuxième élément (module) appelé UFA *(pour Unité de Fonction d'Activation, AFU en anglais)* implémente les fonctions d'activations. Le premier module implémente une approximation du calcul de la racine carrée pour le calcul de la distance euclidienne. Avantageusement, cette approximation est réalisée en virgule fixe, dans le cas de processeurs comprenant de faibles capacités. L'UFA permet d'utiliser la sigmoïde, la tangente hyperbolique, la gaussienne, la RELU. Comme explicité précédemment, le choix des calculs qui sont effectués par le neurone est réalisé par l'utilisation d'un mot de commande nommé *cmd* comme cela est le cas d'une instruction d'un microprocesseur. Ainsi, ce circuit de neurone artificiel est paramétré par la réception d'un mot ou de plusieurs mots de commandes, en fonction de mode de réalisation. Un mot de commande est dans le cas présent un signal, comprenant un bit ou une suite de bits (par exemple un octet, permettant de disposer de 256 commandes possibles ou de deux fois 128 commandes) qui est transmis au circuit pour le paramétrer. Dans un mode de réalisation général, l'implémentation proposée d'un neurone permet de réaliser les réseaux « communs » tout comme les réseaux de neurones de dernière génération comme les *ConvNet* (réseau de neurones convolutifs). Cette architecture de calcul peut s'implémenter, de manière pratique, sous forme de librairie logicielle pour des processeurs standards ou bien sous forme d'implémentation matérielle pour des FPGA ou des ASICs.

[0051] Ainsi, un neurone configurable est composé d'un module de calcul de distance et/ou de produit scalaire qui dépend du type de neurone utilisé, et d'un module UFA.

[0052] Un neurone générique configurable, comme tout neurone, comprend des données d'entrées en virgule fixe ou flottante dont :

– X constitue le vecteur de données d'entrée ;

– W constitue le vecteur des poids synaptiques du neurone ;

[0053] et une donnée de sortie en virgule fixe ou flottante :

– z le résultat scalaire en sortie du neurone.

[0054] Selon l'invention, en sus on dispose d'un paramètre, λ, qui représente le paramètre de la sigmoïde, de la tangente hyperbolique, de la gaussienne ou bien de la RELU. Ce paramètre est identique pour tous les neurones d'une couche. Ce paramètre λ est fourni au neurone avec le mot de commande, paramétrant la mise en œuvre du neurone. Ce paramètre peut être qualifié de paramètre d'approximation en ce sens qu'il est utilisé pour effectuer un calcul approchant de la valeur de la fonction à partir de l'une des méthodes d'approximation présentée ci-dessous.

[0055] Plus particulièrement, dans un mode de réalisation général, les quatre fonctions principales reproduites (et factorisées) par l'UFA sont :

– la sigmoïde :

[0056] [Math 2] $sig(x) = \frac{1}{1+e^{-\lambda x}}$;

– la tangente hyperbolique :

[0057] $tanh(\beta x)$

– la fonction gaussienne ;

[0058] $f(x) = exp\left(\frac{-x^2}{2\sigma^2}\right)$

– la fonction RELU *(« Rectified linear Unit »)* ;

[0059] max(0,*x*) ou bien $\begin{cases} x & x \geqslant 0 \\ ax & x < 0 \end{cases}$

[0060] Selon l'invention, les trois premières fonctions sont calculées de manière approchée. Cela signifie que le neurone configurable n'implémente pas un calcul précis de ces fonctions, mais implémente à la place une approximation du calcul de ces fonctions, ce qui permet de réduire la charge, le temps, et les ressources nécessaires à l'obtention du résultat.

[0061] On expose par la suite les quatre méthodes d'approximations de ces fonctions mathématiques utilisées ainsi que l'architecture d'un tel neurone configurable.

[0062] *Première méthode :*

[0063] La relation

[0064] [Math 5] $f(x) = \frac{1}{1+e^{-x}}$,

[0065] utilisée pour le calcul de la sigmoïde, est approchée par la formule suivante (Allipi) :

[0066] [Math 6] $f(x) = \frac{x - |x| + 2}{2^{|(x)|+2}}$ pour $x \leq 0$

[0067] [Math 7] $f(x) = 1 - \frac{-x + |x| + 2}{2^{|(x)|+2}}$ pour $x > 0$

[0068] avec (x) qui est la partie entière de x

[0069] *Deuxième méthode :*

[0070] La fonction $tanh(x)$ est estimée de la façon suivante :

[0071] $tanh(x) = 2 \times Sig(2x) - 1$

[0072] avec

[0073] $Sig(x) = \frac{1}{1 + exp(-x)}$

[0074] Ou plus généralement :

[0075] $tanh(\beta x) = 2 \times Sig(2\beta x) - 1$

[0076] avec

[0077] $Sig(\lambda x) = \frac{1}{1 + exp(-\lambda x)}$

[0078] Avec

[Math.11]

$\lambda = 2\beta$

[0079] *Troisième méthode :*

[0080] Pour approcher la gaussienne :

[0081] $f(x) = exp\left(\frac{-x^2}{2\sigma^2}\right)$

[0082] On met e œuvre la méthode suivante :

[0083] $sig'(x) = \lambda\, sig(x)(1 - sig(x))$

[0084] Avec

[0085] $\lambda \approx \frac{1,7}{\sigma}$

[0086] *Quatrième méthode :*

[0087] Il n'est pas nécessaire de passer par une approximation pour obtenir une valeur de la fonction la fonction RELU *(« Rectified linear Unit »)* ;

[0088] max(0,*x*) ou bien $\begin{cases} x & x \geqslant 0 \\ ax & x < 0 \end{cases}$ avec $\lambda = a$

[0089] Les quatre méthodes qui précèdent constituent des approximations de calculs des fonctions d'origine (sigmoïdes, tangente hyperbolique et gaussienne). Les inventeurs ont cependant démontré (voir annexe) que les approximations réalisées à l'aide de la technique de l'invention fournissent des résultats similaires à ceux issus d'une expression exacte de la fonction.

[0090] [fig.2] A la vue de ce qui précède, la figure 2 expose l'architecture générale du circuit de la fonction d'activation. Cette architecture fonctionnelle tient compte des approximations précédentes (méthodes 1 à 4) et des factorisations dans les fonctions de calcul.

[0091] Les avantages de la présente technique sont les suivants

- une implémentation matérielle d'un réseau de neurones génériques avec une cellule neuronale paramétrable qui permet d'implémenter tout réseau de neurones dont les convnet.
- pour certains modes de réalisation, une approximation originale du calcul en virgule fixe ou virgule flottante, de la sigmoïde, de la tangente hyperbolique, de la gaussienne.
- une implémentation de l'AFU sous la forme de librairie logicielle pour des processeurs standards ou bien pour des FPGA.
- une intégration de l'AFU sous la forme d'une architecture matérielle pour tous les processeurs standards ou bien pour les FPGAs ou pour les ASICs.
- en fonction de modes de réalisation, une division entre 3 et 5 de la complexité des calculs par rapport aux librairies standards.

[0092] 5.2. Description d'un mode de réalisation d'un neurone configurable

[0093] Dans ce mode de réalisation, on ne discute que de la mise en œuvre opérationnelle de l'AFU.

[0094] L'AFU effectue le calcul quel que soit le mode de représentation des valeurs traitées virgule fixe ou virgule flottante. L'avantage et l'originalité de cette mise en œuvre réside dans la mutualisation (factorisation) des blocs de calcul (blocs n°2 à 4) pour obtenir les différentes fonctions non linéaires, ce calcul est défini comme *« l'opération de base »* dans la suite, il correspond à une approximation du calcul de la sigmoïde de la valeur absolue de λx:

[0095] [Math 15] $$f(x) = \frac{1}{1 + e^{|\lambda x|}}.$$

[0096] Ainsi *« l'opération de base »* n'est plus une opération mathématique standard comme l'addition et la multiplication que l'on trouve dans tous les processeurs classiques, mais la fonction sigmoïde de la valeur absolue de λx. Cette *« opération de base »*, dans ce mode de réalisation, est commune à toutes les autres fonctions non linéaires. Dans ce mode de réalisation, on utilise une approximation de cette fonction. On se sert donc ici d'une approximation d'une fonction de haut niveau pour effectuer les calculs de fonctions de haut niveau sans utiliser des méthodes classiques de calculs de ces fonctions. Le résultat pour une valeur positive de *x* de la sigmoïde est déduit de cette opération de base en utilisant la symétrie de la fonction sigmoïde. La fonction tangente hyperbolique est obtenue en utilisant la relation de correspondance standard qui la lie à

la fonction sigmoïde. La fonction gaussienne est obtenue en passant par la dérivée de la sigmoïde qui est une courbe approchée de la gaussienne, la dérivée de la sigmoïde est obtenue par un produit enter la fonction sigmoïde et sa symétrique. La fonction RELU qui est une fonction linéaire pour *x* positif n'utilise pas *l'opération de base* du calcul des fonctions non linéaires. La fonction leaky RELU qui utilise une fonction linéaire de proportionnalité pour x négatif n'utilise pas non plus *l'opération de base* du calcul des fonctions non linéaires.

[0097] Enfin, le choix de la fonction se fait à l'aide d'un mot de commande (cmd) comme le ferait une instruction de microprocesseur, le signe de la valeur d'entrée détermine la méthode de calcul à utiliser pour la fonction choisie. L'ensemble des paramètres des différentes fonctions utilisent le même paramètre λ qui est un réel positif quel que soit le format de représentation. [fig.3] La figure 3 illustre ce mode de réalisation plus en détail. Plus particulièrement en relation avec cette figure 3 :

– Le bloc n°1 multiplie la donnée d'entrée x par le paramètre λ dont la signification dépend de la fonction d'activation utilisée : directement λ lors de l'utilisation de la sigmoïde, $\beta = \frac{\lambda}{2}$ lors de l'utilisation de la fonction tangente hyperbolique et $\sigma \approx \frac{1.7}{\lambda}$ pour la gaussienne, le coefficient de proportionnalité « a » pour une valeur de x négative lors de l'utilisation de la fonction leakyRELU; ce calcul fournit donc la valeur x_c pour les blocs n°2 et n°5. Ce bloc effectue une opération de multiplication quel que soit le format de représentation des réels. Toute méthode de multiplication qui permet d'effectuer le calcul et de fournir le résultat, quel que soit le format de représentation de ces valeurs, identifie ce bloc. Dans le cas de la gaussienne, la division peut être incluse ou non dans l'AFU.

– Les blocs n°2 à 4 effectuent le calcul de *« l'opération de base »* des fonctions non linéaires à l'exception des fonctions RELU et leakyRELU qui sont des fonctions linéaires avec des coefficients de proportionnalité différents suivant que x est négatif ou positif. Cette opération de base utilise une approximation par segments de droites de la fonction sigmoïde pour une valeur négative de la valeur absolue de x. Ces blocs peuvent groupés par deux ou trois suivant l'optimisation souhaitée. Chaque segment de droite est défini sur un intervalle se situant entre la partie entière de x est la partie entière plus un de x :

– le bloc n°2, nommé séparateur, extrait la partie entière, en prend la valeur absolue, cela peut également se traduire par la valeur absolue de la partie entière par défaut de x : $\lfloor |x| \rfloor$. Il fournit également la valeur absolue de la partie fractionnaire de x : $|\{x\}|$. La partie tronquée fournie par ce bloc

donne le début du segment et la partie fractionnaire représente la droite définie sur ce segment. La séparation de la partie entière et de la partie fractionnaire peut s'obtenir de toutes les façons possibles et quel que soit le format de représentation de x.

- le bloc n°3 calcule le numérateur y_n de la fraction finale à partir de la partie fractionnaire $|\{x\}|$ fournie par le bloc n°2. Ce bloc fournit l'équation de la droite de la forme $2 - |\{x\}|$ indépendamment du segment déterminé avec la partie tronquée.
- le bloc n°4 calcule la valeur commune à toutes les fonctions y_1 à partir du numérateur y_n fourni par le bloc n°3 et de la partie entière fournie par le bloc n°2. Ce bloc calcule le dénominateur commun aux éléments de l'équation de la droite qui permet de fournir une droite différente pour chaque segment avec une erreur minimum entre la courbe réelle et la valeur approchée obtenue avec la droite. Le fait d'utiliser une puissance de 2, simplifie le calcul de l'opération de base. Ce bloc utilise donc une addition et une soustraction qui reste une addition en termes de complexité algorithmique suivie d'une division par une puissance de 2.
- Le bloc n°5 calcule le résultat de la fonction non linéaire qui dépend de la valeur du mot de commande *cmd,* de la valeur du signe de x et bien sûr du résultat y_1 du bloc n°4.
  - Pour une première valeur de *cmd*, il fournit la sigmoïde de paramètre λ qui est égale au résultat de l'opération de base pour x négatif ($z = y_1$ pour $x < 0$) et égal à 1 moins le résultat de l'opération de base pour x positif ($z = 1 - y_1$ pour $x \geq 0$) ; ce calcul utilise la symétrie de la fonction sigmoïde entre les valeurs positives et négatives de x. Ce calcule utilise uniquement une soustraction. Dans ce cas on obtient donc une sigmoïde avec dans le cas le plus défavorable une opération de soustraction supplémentaire.
  - Pour une deuxième valeur, il fournit la tangente hyperbolique de paramètre β qui correspond à deux fois l'opération de base moins un avec une valeur négative de x $z = 2y_1 - 1 (x < 0)$ et un moins deux fois l'opération de base pour une valeur positive de x ($z = 1 - 2y_1$ pour $x \geq 0$). La division de la valeur de x par deux est intégrée par le coefficient 1/2 dans le paramètre $\lambda = 2\beta$ ou bien effectuée à ce niveau avec $\lambda = \beta$.
  - Pour une troisième valeur, il fournit la gaussienne $z = 4y_1 (1 - y_1)$ quel que soit le signe de x. En effet l'approche

de la gaussienne est réalisée en utilisant la dérivée de la sigmoïde. Avec cette méthode on obtient une courbe proche de la fonction gaussienne. De plus la dérivée de la sigmoïde se calcule simplement en multipliant le résultat de l'opération de base par son symétrique. Dans ce cas le paramètre λ définit l'écart type de la gaussienne en divisant 1.7 par λ . Cette opération de division peut être incluse ou non dans l'AFU. Enfin ce calcul utilise une multiplication à deux opérandes et par une puissance de deux.

- Pour une quatrième valeur il fournit la fonction RELU qui donne la valeur de x pour x positif $z = x$ pour $x \geq 0$ et 0 pour x négatif $z = 0$ pour $x < 0$. Dans ce cas on utilise directement la valeur de x sans utiliser l'opération de base.
- Pour une dernière valeur une variante de la fonction relu (leakyRELU) qui donne la valeur de x pour x positif $z = x$ pour $x \geq 0$ et une valeur proportionnelle à x pour x négatif $z = x_c$ pour $x < 0$. Le coefficient de proportionnalité est fourni par le paramètre λ .

[0098] Ainsi, le bloc n°5 est un bloc qui contient les différents calculs finaux des fonctions non linéaires décrits précédemment, ainsi qu'un bloc de commutation qui effectue le choix de l'opération en fonction de la valeur du signal de commande et de la valeur du signe de x.

[0099] 5.3. Description d'un mode de réalisation d'un composant dédié apte à mettre en œuvre une pluralité de réseaux de neurones différents, procédé de traitement de données.

[0100] Dans ce mode de réalisation illustratif, le composant comprenant un ensemble de 16384 neurones reconfigurables est positionné sur le processeur. Chacun de ces neurones reconfigurables reçoit ses données directement depuis la mémoire de stockage temporaire, qui comprend au moins 16384 entrées (ou au moins 32768, selon les modes de réalisation), chaque valeur d'entrée correspondant à un octet. La taille de la mémoire de stockage temporaire est donc de 16ko (ou 32ko) (kilooctets). En fonction de la mise en œuvre opérationnelle, la taille de la mémoire de stockage temporaire peut être augmentée afin de faciliter les processus de réécriture des données de résultats. Le composant comprend également une mémoire de stockage de la configuration du réseau de neurones. Dans cet exemple on suppose que la mémoire de stockage de configuration est dimensionnée pour permettre la mise en œuvre de 20 couches, chacune de ces couches comprenant potentiellement un nombre de poids synaptiques correspondant au nombre total d'entrée possible soit 16384 poids synaptiques différents pour chacune des couches, chacun d'une taille d'un octet. Pour

chaque couche, selon l'invention, on dispose également d'au moins deux mots de commandes, chacun d'une longueur d'un octet, soit au total 16386 octets par couche, et donc pour les 20 couches, un total minimal de 320 ko. Cette mémoire comprend également un ensemble de registres dédiés au stockage des données représentatives de la configuration du réseau : nombre de couches, nombre de neurones par couche, ordonnancement des résultats d'une couche, etc. L'ensemble du composant nécessite donc dans cette configuration, une taille de mémoire inférieure à 1 Mo.

[0101] 5.4. Autres caractéristiques et avantages

[0102] [fig.4] Le fonctionnement du réseau de neurones reconfigurables est présenté en relation avec la figure 4.

[0103] A l'initialisation (étape 0), un ensemble de données (EDAT), correspondant par exemple à un ensemble de données applicatives provenant d'une application matérielle ou logicielle donnée est chargée dans la mémoire de stockage temporaire (MEM). Un ensemble de données, correspondant à l'ensemble des poids synaptiques et des configurations des couches (CONFDAT) est chargé dans la mémoire de stockage de configuration du réseau (MEMR).

[0104] Le réseau de neurones est ensuite exécuté (étape 1) par le processeur de l'invention, selon une mise en œuvre itérative (tant que la couche courante est inférieure au nombre de couches du réseau, *i.e. nblyer*), des étapes suivantes exécutées pour une couche donnée du réseau de neurones, de la première couche à la dernière couche, et comprenant pour une couche courante :

- transmission (10) du premier mot de commande à l'ensemble des neurones mis en œuvre, définissant la fonction de combinaison mises en œuvre (combinaison linéaire ou norme euclidienne) pour la couche courante ;
- transmission (20) du deuxième mot de commande à l'ensemble des neurones mis en œuvre, définissant la fonction d'activation mises en œuvre pour la couche courante ;
- chargement (30) des poids synaptiques de la couche ;
- chargement (40) des données d'entrée dans la mémoire de stockage temporaire ;
- calcul (50) de la fonction de combinaison, pour chaque neurone et chaque vecteur d'entrée, en fonction du mot de commande, délivrant, pour chaque neurone utilisé, un scalaire intermédiaire ;
- calcul (60) de la fonction d'activation en fonction du scalaire intermédiaire, et du deuxième mot de commande, délivrant, pour chaque neurone utilisé, un résultat d'activation ;
- enregistrement (70) du résultat d'activation dans la mémoire de stockage temporaire.

[0105] On note que les étapes de transmission des mots de commande et de calcul des résultats des fonction de combinaison et d'activation ne constituent pas nécessairement des étapes physiquement séparées. Par ailleurs, comme explicité précédemment, un seul et même mot de commande peut être utilisé en lieu et place de deux mots de commande, et ce afin de spécifier à la fois la fonction de combinaison et la fonction d'activation utilisée.

[0106] Les résultats finaux (SDAT) sont alors retournés (étape 2) à l'application ou au composant appelant.

Revendications

[Revendication 1] Processeur de traitement de données, ledit processeur comprenant au moins une mémoire de traitement (MEM) et une unité de calcul (CU), le dit processeur étant caractérisé en ce que l'unité de calcul (CU) comprend un ensemble d'unités de calculs configurables appelées neurones configurables, chaque neurone configurable (NC) de l'ensemble de neurones configurables (ENC) comprenant un module de calcul de fonctions de combinaison (MCFC) et un module de calcul de fonctions d'activation (MCFA), chaque module de calcul de fonctions d'activation (AFU) comprenant un registre de réception d'une commande de paramétrage, de sorte que ladite commande détermine une fonction d'activation à exécuter parmi au moins deux fonctions d'activation exécutables par le module de calcul de fonctions d'activation (AFU).

[Revendication 2] Processeur de traitement de données selon la revendication 1, caractérisé en ce que les au moins deux fonctions d'activation exécutables par le module de calcul de fonctions d'activation (AFU) appartiennent au groupe comprenant :
la fonction sigmoïde ;
la fonction tangente hyperbolique ;
la fonction gaussienne ;
la fonction RELU *(« Rectified linear Unit »).*

[Revendication 3] Processeur de traitement de données selon la revendication 1, caractérisé en ce que le module de calcul de fonctions d'activation (AFU) est configuré pour effectuer une approximation desdites au moins deux fonctions d'activation.

[Revendication 4] Processeur de traitement de données selon la revendication 3, caractérisé en ce que le module de calcul de fonctions d'activation (AFU) comprend un sous module de calcul d'une opération de base correspondant à une approximation du calcul de la sigmoïde de la valeur absolue de λx :

[Math 16] $$f(x) = \frac{1}{1 + e^{|\lambda x|}}.$$

[Revendication 5] Processeur de traitement de données selon la revendication 3, caractérisé en ce que l'approximation desdites au moins deux fonctions d'activation est effectuée en fonction d'un paramètre d'approximation λ.

[Revendication 6] Processeur de traitement de données selon la revendication 3, caractérisé

en ce que l'approximation desdites au moins deux fonctions d'activation est effectuée en configurant le module de calcul de fonctions d'activation (AFU) pour que les calculs soient effectués en virgule fixe ou virgule flottante.

[Revendication 7] Processeur de traitement de données selon la revendication 5, caractérisé en ce que le nombre de bits associés aux calculs en virgule fixe ou virgule flottante est paramétré pour chaque couche du réseau.

[Revendication 8] Processeur de traitement de données selon la revendication 1, caractérisé en ce qu'il comprend une mémoire de stockage de configuration du réseau au sein de laquelle des paramètres (PS, cmd, λ) d'exécution de réseau de neurones sont enregistrés.

[Revendication 9] Procédé de traitement de données, ledit procédé étant mis en œuvre par un processeur de traitement de données comprenant au moins une mémoire de traitement (MEM) et une unité de calcul (CU), l'unité de calcul (CU) comprend un ensemble d'unités de calculs configurables appelées neurones configurables, chaque neurone configurable (NC) de l'ensemble de neurones configurables (ENC) comprenant un module de calcul de fonctions de combinaison (MCFC) et un module de calcul de fonctions d'activation (AFU), le procédé comprenant :
- une étape d'initialisation (0) comprenant le chargement dans la mémoire de traitement (MEM) d'un un ensemble de données applicatives (EDAT) et le chargement d'un ensemble de données, correspondant à l'ensemble des poids synaptiques et des configurations des couches (CONFDAT) dans la mémoire de stockage de configuration du réseau (MEMR) ;
- l'exécution (1) du réseau de neurone, selon une mise en œuvre itérative, comprenant pour chaque couche, l'application d'une commande de paramétrage, de sorte que ladite commande détermine une fonction d'activation à exécuter parmi au moins deux fonctions d'activation exécutables par le module de calcul de fonctions d'activation (AFU), l'exécution délivrant des données traitées ;
la transmission des données traitées (SDAT) à une application appelante.

[Revendication 10] Procédé selon la revendication 9, caractérisé en ce que l'exécution (1) du réseau de neurone comprend au moins une itération des étapes suivantes, pour une couche courante du réseau de neurones :
transmission (10, 20) d'au moins un mot de commande, définissant la fonction de combinaison et/ou la fonction d'activation mise en œuvre

pour la couche courante ;
chargement (30) des poids synaptiques de la couche courante ;
chargement (40) des données d'entrée en provenance de la mémoire de stockage temporaire ;
calcul (50) de la fonction de combinaison, pour chaque neurone et chaque vecteur d'entrée, en fonction dudit au moins un mot de commande, délivrant, pour chaque neurone utilisé, un scalaire intermédiaire ;
calcul (60) de la fonction d'activation en fonction du scalaire intermédiaire, et dudit au moins un deuxième mot de commande, délivrant, pour chaque neurone utilisé, un résultat d'activation ;
enregistrement (70) du résultat d'activation dans la mémoire de stockage temporaire.

[Revendication 11] Produit programme d'ordinateur téléchargeable depuis un réseau de communication et/ou stocké sur un support lisible par ordinateur et/ou exécutable par un microprocesseur, caractérisé en ce qu'il comprend des instructions de code de programme pour l'exécution d'un procédé selon la revendication 9, lorsqu'il est exécuté sur un ordinateur.

[Fig. 1]

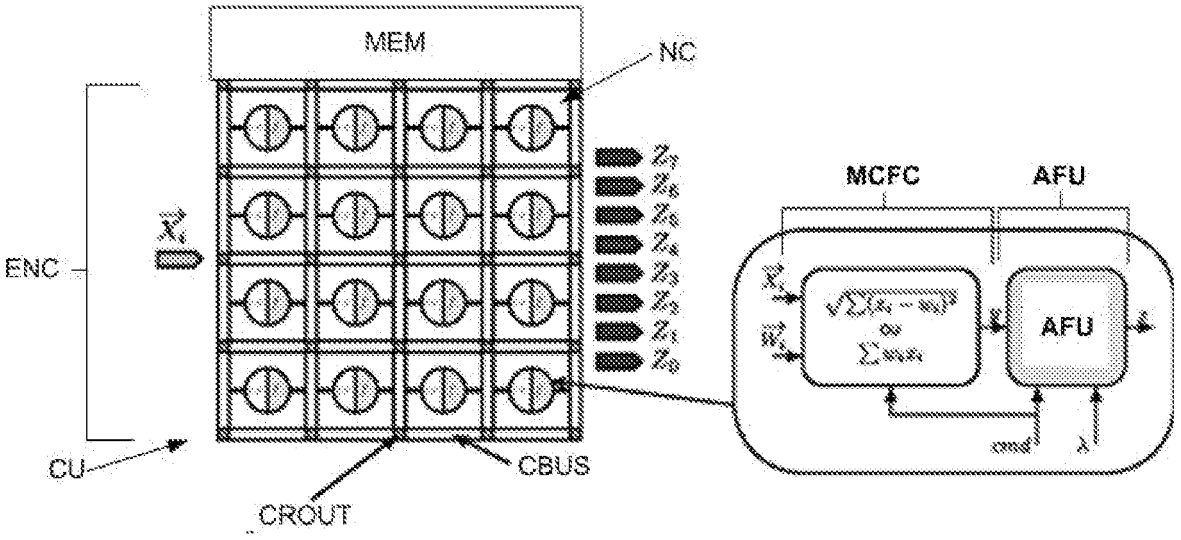

Figure 1

[Fig. 2]

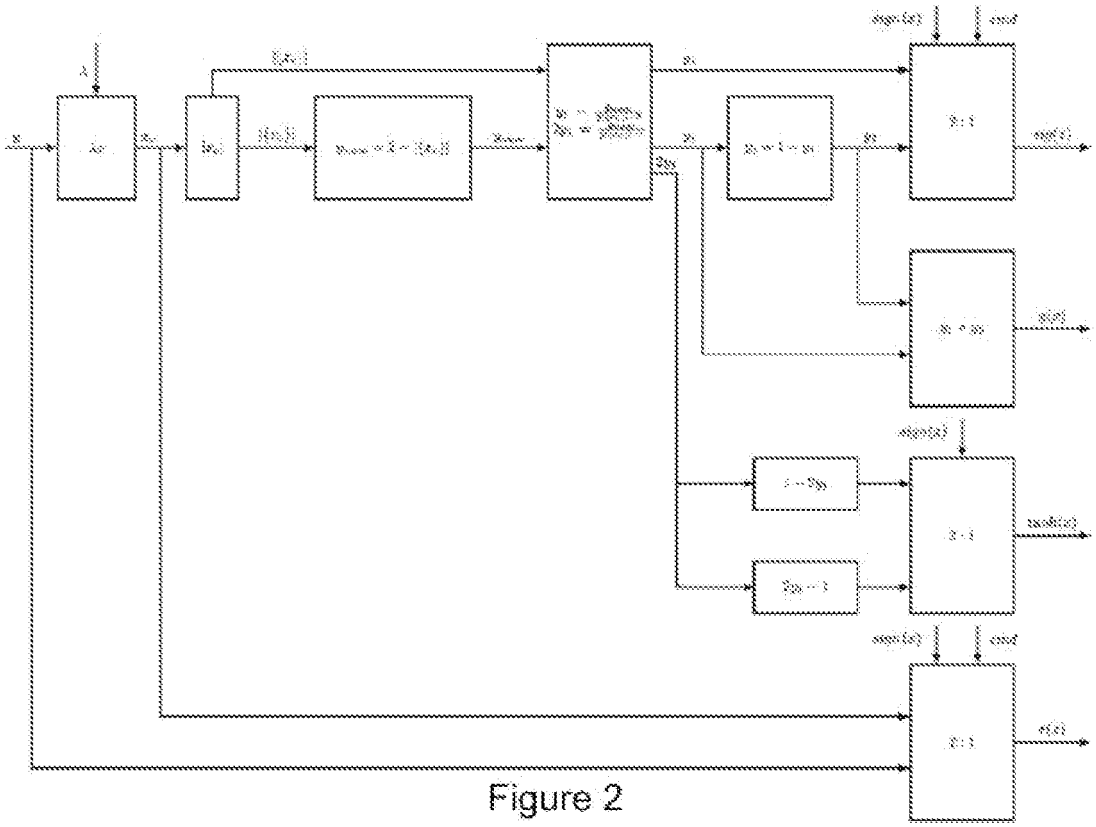

Figure 2

[Fig. 3]

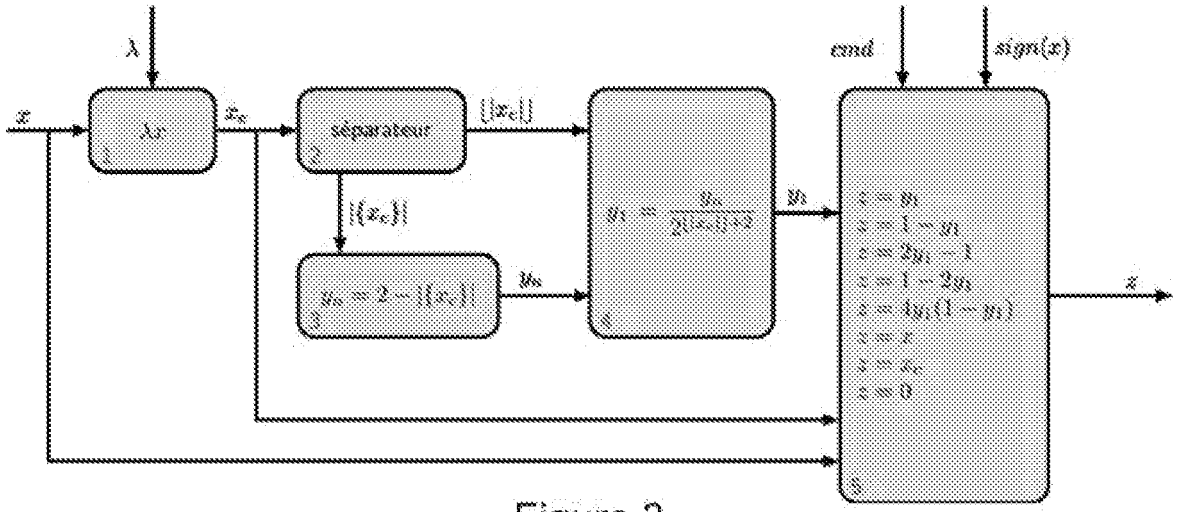

Figure 3

[Fig. 4]

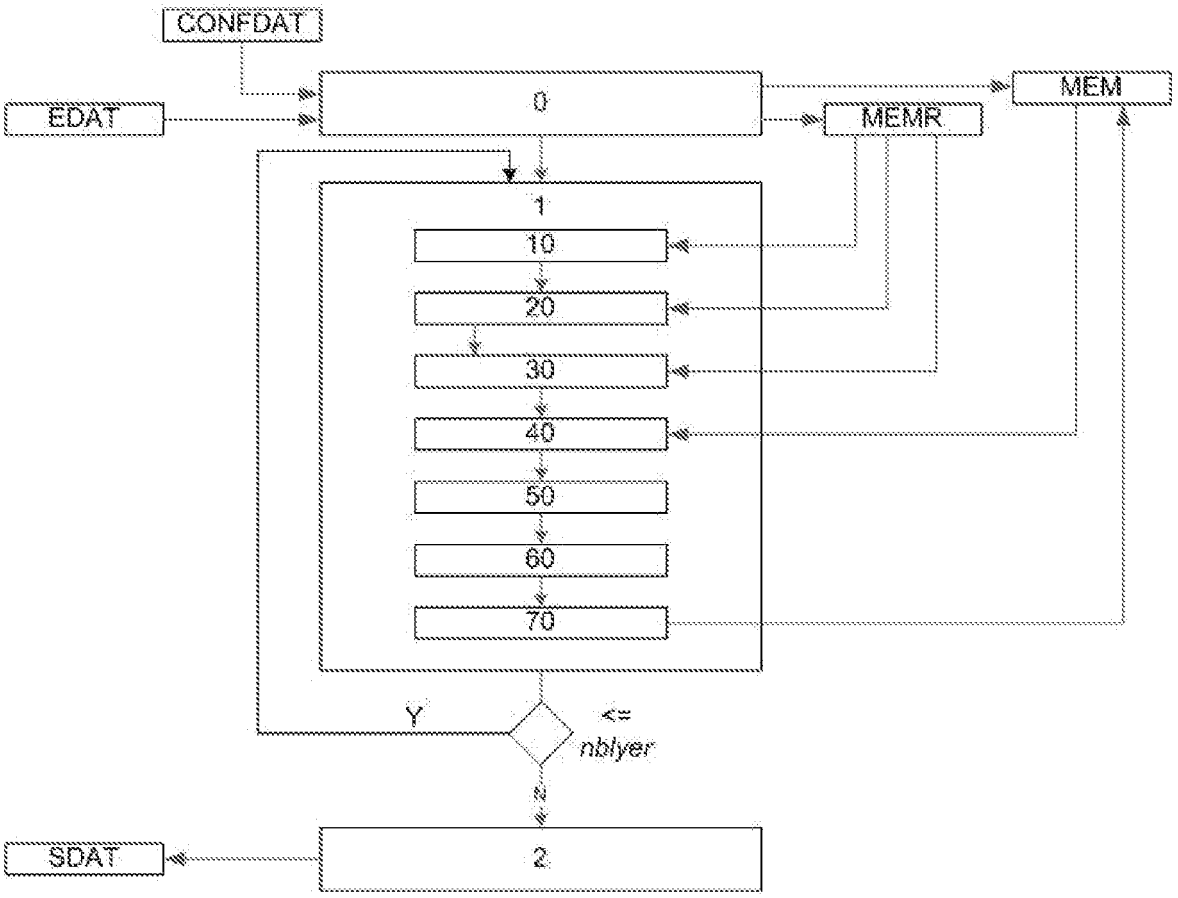

Figure 4

RAPPORT DE RECHERCHE

articles L.612-14, L.612-53 à 69 du code de la propriété intellectuelle

OBJET DU RAPPORT DE RECHERCHE

L'I.N.P.I. annexe à chaque brevet un "RAPPORT DE RECHERCHE" citant les éléments de l'état de la technique qui peuvent être pris en considération pour apprécier la brevetabilité de l'invention, au sens des articles L. 611-11 (nouveauté) et L. 611-14 (activité inventive) du code de la propriété intellectuelle. Ce rapport porte sur les revendications du brevet qui définissent l'objet de l'invention et délimitent l'étendue de la protection.
Après délivrance, l'I.N.P.I. peut, à la requête de toute personne intéressée, formuler un "AVIS DOCUMENTAIRE" sur la base des documents cités dans ce rapport de recherche et de tout autre document que le requérant souhaite voir prendre en considération.

CONDITIONS D'ETABLISSEMENT DU PRESENT RAPPORT DE RECHERCHE

[x] Le demandeur a présenté des observations en réponse au rapport de recherche préliminaire.

[x] Le demandeur a maintenu les revendications.

☐ Le demandeur a modifié les revendications.

☐ Le demandeur a modifié la description pour en éliminer les éléments qui n'étaient plus en concordance avec les nouvelles revendications.

☐ Les tiers ont présenté des observations après publication du rapport de recherche préliminaire.

☐ Un rapport de recherche préliminaire complémentaire a été établi.

DOCUMENTS CITES DANS LE PRESENT RAPPORT DE RECHERCHE

La répartition des documents entre les rubriques 1, 2 et 3 tient compte, le cas échéant, des revendications déposées en dernier lieu et/ou des observations présentées.

[x] Les documents énumérés à la rubrique 1 ci-après sont susceptibles d'être pris en considération pour apprécier la brevetabilité de l'invention.

[x] Les documents énumérés à la rubrique 2 ci-après illustrent l'arrière-plan technologique général.

☐ Les documents énumérés à la rubrique 3 ci-après ont été cités en cours de procédure, mais leur pertinence dépend de la validité des priorités revendiquées.

☐ Aucun document n'a été cité en cours de procédure.

1. ELEMENTS DE L'ETAT DE LA TECHNIQUE SUSCEPTIBLES D'ETRE PRIS EN CONSIDERATION POUR APPRECIER LA BREVETABILITE DE L'INVENTION

GOMAR SHAGHAYEGH ET AL: "Precise digital implementations of hyperbolic tanh and sigmoid function",
2016 50TH ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS AND COMPUTERS, IEEE,
6 novembre 2016 (2016-11-06), pages 1586-1589, XP033072819,
DOI: 10.1109/ACSSC.2016.7869646
[extrait le 2017-03-02]

ALIPPI C ET AL: "SIMPLE APPROXIMATION OF SIGMOIDAL FUNCTIONS: REALISTIC DESIGN OF DIGITAL NEURAL NETWORKS CAPABLE OF LEARNING",
SIGNAL IMAGE AND VIDEO PROCESSING. SINGAPORE, JUNE 11 -14, 1991; [PROCEEDINGS OF THE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS], IEEE, NEW YORK, NY,
vol. 3 OF 05, 11 juillet 1991 (1991-07-11)
, pages 1505-1508, XP000370893,
DOI: 10.1109/ISCAS.1991.176661
ISBN: 978-0-7803-0050-7

2. ELEMENTS DE L'ETAT DE LA TECHNIQUE ILLUSTRANT L'ARRIERE-PLAN TECHNOLOGIQUE GENERAL

SHANNON R. BOWLING ET AL: "A logistic approximation to the cumulative normal distribution",
JOURNAL OF INDUSTRIAL ENGINEERING AND MANAGEMENT,
vol. 2, no. 1, 1 janvier 2009 (2009-01-01)
, pages 114-127, XP055633582,
DOI: 10.3926/jiem.2009.v2n1.p114-127

Valeriu Beiu ET AL: "Close Approximations of Sigmoid Functions by Sum of Steps for VLSI Implementation of Neural Networks x",
,
1 janvier 1994 (1994-01-01), XP055633576,
Extrait de l'Internet:
URL:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.5332&rep=rep1&type=pdf

WO 2018/046415 A1 (BOSCH GMBH ROBERT [DE])
15 mars 2018 (2018-03-15)

ARMATO A ET AL: "Low-error digital hardware implementation of artificial neuron activation functions and their derivative",
MICROPROCESSORS AND MICROSYSTEMS,

vol. 35, no. 6, 31 août 2011 (2011-08-31),
pages 557-567, XP028255706,
ISSN: 0141-9331, DOI:
10.1016/J.MICPRO.2011.05.007
[extrait le 2011-05-30]

3. ELEMENTS DE L'ETAT DE LA TECHNIQUE DONT LA PERTINENCE DEPEND DE LA VALIDITE DES PRIORITES

NEANT