

①9 RÉPUBLIQUE FRANÇAISE
INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE
COURBEVOIE

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

3 109 002

②1 N° d'enregistrement national : **20 03326**

⑤1 Int Cl⁸ : **G 06 N 3/02 (2019.12), G 06 N 3/08**

⑫

DEMANDE DE BREVET D'INVENTION

A1

②2 Date de dépôt : 02.04.20.

③0 Priorité :

④3 Date de mise à la disposition du public de la demande : 08.10.21 Bulletin 21/40.

⑤6 Liste des documents cités dans le rapport de recherche préliminaire : *Se reporter à la fin du présent fascicule*

⑥0 Références à d'autres documents nationaux apparentés :

Demande(s) d'extension :

⑦1 Demandeur(s) : COMMISSARIAT A L'ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES Etablissement public — FR et UNIVERSITE GRENOBLE ALPES Etablissement public — FR.

⑦2 Inventeur(s) : SOLINAS Miguel Angel, REYBOZ Marina, ROUSSET Stephane, MERMILLOD Martial et GALIEZ Clovis.

⑦3 Titulaire(s) : COMMISSARIAT A L'ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES Etablissement public, UNIVERSITE GRENOBLE ALPES Etablissement public.

⑦4 Mandataire(s) : CABINET BEAUMONT.

⑤4 Dispositif et procédé pour le transfert de connaissance d'un réseau neuronal artificiel.

⑤7 Dispositif et procédé pour le transfert de connaissance d'un réseau neuronal artificiel

La présente description concerne un procédé de génération de données d'apprentissage pour transférer de la connaissance d'un réseau neuronal artificiel entraîné (402) vers un autre réseau neuronal artificiel (406), le procédé comprenant: a) injecter un échantillon aléatoire (SEED) dans le réseau neuronal artificiel entraîné; b) réinjecter un pseudo-échantillon (X_m), généré sur la base d'un échantillon reproduit présent au niveau desdites une ou plusieurs sorties du réseau neuronal artificiel entraîné, dans le réseau neuronal artificiel entraîné afin de générer un nouvel échantillon reproduit; et c) répéter b) une ou plusieurs fois, dans lequel les données d'apprentissage pour l'apprentissage de l'autre réseau neuronal artificiel (406) comprennent au moins deux des pseudo-échantillons réinjectés (X_m , $X_{(m+1)}$,...) issus du même échantillon aléatoire et des valeurs de sortie correspondantes (Y_m , $Y_{(m+1)}$,...) générées par le réseau neuronal artificiel entraîné.

Figure pour l'abrégié: Fig. 4

FR 3 109 002 - A1



Description

Titre de l'invention : *Dispositif et procédé pour le transfert de connaissance d'un réseau neuronal artificiel*

Domaine technique

[0001] La présente description concerne de façon générale le domaine des réseaux neuronaux artificiels, et en particulier un dispositif et un procédé pour transférer de la connaissance entre des réseaux neuronaux artificiels.

Technique antérieure

[0002] Les réseaux neuronaux artificiels (ANN) sont des architectures qui visent à imiter, dans une certaine mesure, le comportement d'un cerveau humain. De tels réseaux sont en général constitués de circuits de neurones et d'interconnexions entre les circuits de neurones, connues sous le nom de synapses.

[0003] Comme cela est connu de la personne du métier, les architectures ANN, comme les architectures de perceptrons multicouches, comprennent une couche d'entrée de circuits de neurones, une ou plusieurs couches cachées de circuits de neurones, et une couche de sortie de circuits de neurones. Chacun des circuits de neurones dans la ou les couches cachées applique une fonction d'activation, comme la fonction sigmoïde, à des entrées reçues de la couche précédente afin de générer une valeur de sortie. Les entrées sont pondérées par des paramètres θ au niveau des entrées des neurones de la ou des couches cachées. Bien que la fonction d'activation soit en général sélectionnée par le concepteur, les paramètres θ sont trouvés pendant l'apprentissage.

[0004] Pour un problème donné, une fonction à approximer est par exemple une fonction qui génère, sur la base d'entrées X , des étiquettes de sortie vraies $y_t = F(x)$, où $F(x)$ est la fonction qui mappe X vers Y . Le réseau entraîné $y_p = f(x; \theta)$ est entraîné pour générer une valeur y_p qui est la plus proche possible de la valeur vraie y_t en minimisant une fonction de perte. Les performances d'un ANN entraîné dans la résolution de la tâche pour laquelle un apprentissage est effectué reposent sur son architecture, sur le nombre de paramètres θ , et sur la manière dont l'ANN est entraîné. En général, plus l'ANN est grand et complexe, plus ses performances sont bonnes.

[0005] Dans certains modes de réalisation, il peut être souhaitable d'effectuer l'apprentissage de plus qu'un seul ANN pour effectuer la même fonction. Une solution pourrait consister à entraîner chaque ANN en utilisant le même ensemble d'échantillons de données brutes constituant les données d'apprentissage. Toutefois, cela impliquerait de conserver les données d'apprentissage afin de permettre l'apprentissage de nouveaux ANN, ce qui est coûteux en termes de ressources ma-

térielles, et dans certains cas le jeu de données d'apprentissage d'origine peut ne plus être disponible lorsque l'on souhaite transférer la connaissance.

[0006] Une solution pour résoudre ce problème, connue sous le nom d'apprentissage par transfert, est de copier les paramètres d'un ANN entraîné sur un deuxième ANN non entraîné, évitant par cela le besoin de réaliser l'apprentissage du deuxième ANN. Une telle technique peut donner de bons résultats, mais repose sur le fait que les architectures des deux ANN sont basées sur le même modèle, en d'autres termes sur le fait que le nouveau modèle du deuxième ANN doit conserver l'architecture d'origine de l'ANN entraîné afin de transférer la fonction d'origine $f(x; \theta)$. En effet, si le deuxième ANN a une profondeur différente de celle de l'ANN entraîné, qu'elle soit moins profonde ou plus profonde, il ne sera pas capable de prendre en charge les paramètres d'origine. Puisque l'architecture d'origine doit rester fixe, cette technique n'offre pas une solution souple.

[0007] Il existe donc un besoin dans la technique d'une solution permettant de transférer de la connaissance entre des ANN ayant des profondeurs différentes.

Résumé de l'invention

[0008] Un objet des modes de réalisation de la présente invention est de pallier tout ou partie d'un ou plusieurs problèmes connus de l'art antérieur.

[0009] Selon un mode de réalisation, on prévoit un procédé de génération de données d'apprentissage pour transférer de la connaissance d'un réseau neuronal artificiel entraîné vers un autre réseau neuronal artificiel, le procédé comprenant : a) injecter un échantillon aléatoire dans le réseau neuronal artificiel entraîné, le réseau neuronal artificiel entraîné étant agencé pour mettre en œuvre au moins une fonction auto-associative pour reproduire des échantillons d'entrée au niveau d'une ou plusieurs de ses sorties ; b) réinjecter un pseudo-échantillon, généré sur la base de l'échantillon reproduit présent au niveau desdites une ou plusieurs sorties du réseau neuronal artificiel entraîné, dans le réseau neuronal artificiel entraîné afin de générer un nouvel échantillon reproduit au niveau desdites une ou plusieurs sorties ; et c) répéter b) une ou plusieurs fois pour générer une pluralité de pseudo-échantillons réinjectés ; dans lequel les données d'apprentissage pour l'apprentissage de l'autre réseau neuronal artificiel comprennent au moins deux des pseudo-échantillons réinjectés issus du même échantillon aléatoire et des valeurs de sortie correspondantes générées par le réseau neuronal artificiel entraîné.

[0010] Selon un mode de réalisation, le réseau neuronal artificiel entraîné, ou un autre réseau neuronal artificiel entraîné, est agencé pour mettre en œuvre une fonction de classification, et les valeurs de sortie correspondantes des données d'apprentissage comprennent des pseudo-étiquettes générées par la fonction de classification sur la base des

pseudo-échantillons réinjectés.

- [0011] Selon un mode de réalisation, le procédé comprend en outre une détection, sur la base des pseudo-étiquettes, du moment où une frontière entre deux espaces de pseudo-étiquettes est traversée entre des réinjections consécutives de deux des pseudo-échantillons, et lesdits au moins deux pseudo-échantillons réinjectés formant les données d'apprentissage comprennent au moins les deux pseudo-échantillons réinjectés consécutivement.
- [0012] Selon un mode de réalisation, les pseudo-étiquettes sont des sorties non normalisées de la fonction de classification.
- [0013] Selon un mode de réalisation, l'autre réseau neuronal artificiel est agencé pour mettre en œuvre au moins une fonction auto-associative pour reproduire des échantillons d'entrée au niveau d'une ou plusieurs de ses sorties, et les valeurs de sortie correspondantes des données d'apprentissage comprennent les nouveaux échantillons reproduits générés par la fonction auto-associative du réseau neuronal artificiel entraîné sur la base des pseudo-échantillons réinjectés.
- [0014] Selon un mode de réalisation, le procédé comprend en outre : d) répéter a), b) et c) au moins une fois sur la base de nouveaux échantillons aléatoires afin de générer, à chaque répétition, au moins deux autres pseudo-échantillons réinjectés formant les données d'apprentissage.
- [0015] Selon un mode de réalisation, le procédé comprend en outre la génération de l'échantillon aléatoire sur la base d'une distribution normale ou sur la base d'une distribution uniforme ajustée.
- [0016] Selon un mode de réalisation, la génération du pseudo-échantillon comprend l'injection de bruit dans l'échantillon reproduit présent au niveau desdites une ou plusieurs sorties du réseau neuronal artificiel entraîné.
- [0017] Selon un autre aspect, on prévoit un procédé de transfert de connaissance à partir d'un réseau neuronal artificiel entraîné vers un ou plusieurs autres réseaux neuronaux artificiels, le procédé comprenant : générer des données d'apprentissage en utilisant le procédé susmentionné ; et entraîner l'autre réseau neuronal artificiel sur la base des données d'apprentissage générées.
- [0018] Selon un autre aspect, on prévoit un système de génération de données d'apprentissage pour transférer de la connaissance à partir d'un réseau neuronal artificiel entraîné vers un autre réseau neuronal artificiel, le système comprenant un générateur de données agencé pour : a) injecter un échantillon aléatoire dans le réseau neuronal artificiel entraîné, le réseau neuronal artificiel entraîné étant agencé pour mettre en œuvre au moins une fonction auto-associative pour reproduire des échantillons d'entrée au niveau d'une ou plusieurs de ses sorties ; b) réinjecter un pseudo-échantillon, généré sur la base de l'échantillon reproduit présent au niveau

desdites une ou plusieurs sorties du réseau neuronal artificiel entraîné, dans le réseau neuronal artificiel entraîné afin de générer un nouvel échantillon reproduit au niveau desdites une ou plusieurs sorties ; et c) répéter b) une ou plusieurs fois pour générer une pluralité de pseudo-échantillons réinjectés ; dans lequel le générateur de données est en outre agencé pour générer les données d'apprentissage pour l'apprentissage de l'autre réseau neuronal artificiel de manière à comprendre au moins deux des pseudo-échantillons réinjectés issus du même échantillon aléatoire et des valeurs de sortie correspondantes générées par le réseau neuronal artificiel entraîné.

- [0019] Selon un mode de réalisation, le système comprend en outre l'autre réseau neuronal artificiel, et un système d'apprentissage agencé pour entraîner l'autre réseau neuronal artificiel sur la base des données d'apprentissage.
- [0020] Selon un mode de réalisation, le réseau neuronal artificiel entraîné, ou un autre réseau neuronal artificiel entraîné, est agencé pour mettre en œuvre une fonction de classification, et le générateur de données est agencé pour générer des données d'apprentissage de manière à comprendre en outre des pseudo-étiquettes générées par la fonction de classification sur la base des pseudo-échantillons réinjectés, et l'autre réseau neuronal artificiel est capable de mettre en œuvre une fonction de classification.
- [0021] Selon un mode de réalisation, l'autre réseau neuronal artificiel est agencé pour mettre en œuvre au moins une fonction auto-associative pour reproduire des échantillons d'entrée au niveau d'une ou plusieurs de ses sorties, et les données d'apprentissage comprennent en outre les nouveaux échantillons reproduits générés par la fonction auto-associative du réseau neuronal artificiel entraîné sur la base des pseudo-échantillons réinjectés.
- [0022] Selon un mode de réalisation, le système comprend en outre un générateur de germes agencé pour générer l'échantillon aléatoire sur la base d'une distribution normale ou sur la base d'une distribution uniforme ajustée.
- [0023] Selon un mode de réalisation, le générateur de données est agencé pour générer le pseudo-échantillon en injectant du bruit dans l'échantillon reproduit présent au niveau desdites une ou plusieurs sorties du réseau neuronal artificiel entraîné.

Brève description des dessins

- [0024] Ces caractéristiques et avantages, ainsi que d'autres, seront exposés en détail dans la description suivante de modes de réalisation particuliers faite à titre non limitatif en relation avec les figures jointes parmi lesquelles :
- [0025] [fig.1] la figure 1 représente une architecture ANN à perceptron multicouche selon un exemple de réalisation ;
- [0026] [fig.2] la figure 2 représente un espace bidimensionnel donnant un exemple de modèle qui classe des éléments en trois classes, et un exemple d'échantillons aléatoires

dans cet espace ;

- [0027] [fig.3] la figure 3 représente schématiquement une architecture ANN selon un exemple de réalisation de la présente description ;
- [0028] [fig.4] la figure 4 représente schématiquement un système de transfert de connaissance selon un exemple de réalisation de la présente description ;
- [0029] [fig.5] la figure 5 est un organigramme illustrant des étapes dans un procédé de transfert de connaissance selon un exemple de réalisation de la présente description ;
- [0030] [fig.6] la figure 6 représente un espace bidimensionnel donnant un exemple de modèle qui classe des éléments en trois classes, et un exemple de trajectoire de pseudo-échantillons dans cet espace ;
- [0031] [fig.7] la figure 7 est un graphique illustrant des exemples de distributions aléatoires d'échantillons aléatoires selon un exemple de réalisation de la présente description ;
- [0032] [fig.8] la figure 8 est un graphique illustrant un exemple de fonction d'activation selon un exemple de réalisation de la présente description ;
- [0033] [fig.9] la figure 9 représente schématiquement un circuit de génération d'échantillons selon un exemple de réalisation de la présente description ;
- [0034] [fig.10] la figure 10 représente schématiquement un système de transfert de connaissance selon un autre exemple de réalisation de la présente description ;
- [0035] [fig.11] la figure 11 représente schématiquement une architecture ANN selon un autre exemple de réalisation de la présente description ;
- [0036] [fig.12A] la figure 12A représente schématiquement un système de transfert de connaissance selon un autre exemple de réalisation de la présente description ; et
- [0037] [fig.12B] la figure 12B représente schématiquement un système de transfert de connaissance selon encore un autre exemple de réalisation de la présente description.

Description des modes de réalisation

- [0038] De mêmes éléments ont été désignés par de mêmes références dans les différentes figures. En particulier, les éléments structurels et/ou fonctionnels communs aux différents modes de réalisation peuvent présenter les mêmes références et peuvent disposer de propriétés structurelles, dimensionnelles et matérielles identiques.
- [0039] Par souci de clarté, seuls les étapes et éléments utiles à la compréhension des modes de réalisation décrits ont été représentés et sont détaillés. En particulier, des techniques pour l'apprentissage d'un réseau neuronal artificiel, basées par exemple sur la minimisation d'une fonction objective comme une fonction de coût, sont connues de la personne du métier et ne seront pas décrites ici en détail.
- [0040] Sauf précision contraire, lorsque l'on fait référence à deux éléments connectés entre eux, cela signifie directement connectés sans éléments intermédiaires autres que des conducteurs, et lorsque l'on fait référence à deux éléments couplés entre eux, cela

signifie que ces deux éléments peuvent être connectés ou être reliés par l'intermédiaire d'un ou plusieurs autres éléments.

[0041] Dans la description qui suit, lorsque l'on fait référence à des qualificatifs de position absolue, tels que les termes "avant", "arrière", "haut", "bas", "gauche", "droite", etc., ou relative, tels que les termes "dessus", "dessous", "supérieur", "inférieur", etc., ou à des qualificatifs d'orientation, tels que les termes "horizontal", "vertical", etc., il est fait référence sauf précision contraire à l'orientation des figures.

[0042] Sauf précision contraire, les expressions "environ", "approximativement", "sensiblement", et "de l'ordre de" signifient à 10 % près, de préférence à 5 % près.

[0043] Dans la description suivante, on supposera que les termes suivants ont les définitions suivantes :

- "données d'entrée réelles" : échantillons de données recueillis et utilisés pour l'apprentissage d'un ANN non entraîné, ces données d'entrée étant appelées "réelles" parce que ce ne sont pas des données générées par un ordinateur, et par conséquent ne sont pas synthétiques ;

- "échantillon aléatoire" : échantillon synthétique généré par ordinateur sur la base de valeurs aléatoires ou pseudo-aléatoires ;

- "données d'apprentissage" ou "pseudo-données" : données qui peuvent être utilisées pour l'apprentissage d'un ou plusieurs réseaux neuronaux, ces données comprenant par exemple, dans les modes de réalisation de la présente description, des données synthétiques sous la forme de pseudo-échantillons, et dans le cas d'un classificateur, de pseudo-étiquettes ;

- "pseudo-échantillon" : échantillon synthétique généré par un ordinateur, généré sur la base d'un processus de génération de données guidé ou utilisant un prétraitement ;

- "pseudo-étiquette" : étiquette générée par un réseau neuronal entraîné en réponse à l'injection d'un pseudo-échantillon, la pseudo-étiquette correspondant à la vérité de terrain à cibler pendant l'apprentissage d'un ANN utilisant des données d'apprentissage ; et

- "auto-associative" : la fonction de reproduction d'entrées, comme dans un auto-codeur. Toutefois, le terme "auto-codeur" est souvent associé à un ANN qui doit réaliser une certaine compression, par exemple impliquant une compression de l'espace latent signifiant que lesdites une ou plusieurs couches cachées contiennent moins de neurones que le nombre de neurones se trouvant dans l'espace d'entrée. En d'autres termes, l'espace d'entrée est projeté dans un espace plus petit. Le terme "auto-associative" est utilisé ici pour désigner une fonction de reproduction similaire à celle d'un auto-codeur, mais une fonction auto-associative est plus générale en ce qu'elle peut ou non impliquer de la compression.

[0044] La figure 1 illustre une architecture ANN à perceptron multicouche 100 selon un

exemple de réalisation.

- [0045] L'architecture ANN 100 selon l'exemple de la figure 1 comprend trois couches, en particulier une couche d'entrée (INPUT LAYER), une couche cachée (HIDDEN LAYER) et une couche de sortie (OUTPUT LAYER). Dans des variantes de réalisation, il pourrait y avoir plusieurs couches cachées. Chaque couche comprend par exemple un certain nombre de neurones. Par exemple, l'architecture ANN 100 définit un modèle dans un espace bidimensionnel, et il y a ainsi deux neurones visibles dans la couche d'entrée recevant les valeurs X1 et X2 correspondantes d'une entrée X. Le modèle comporte une couche cachée avec sept neurones de sortie cachés, et correspond ainsi à une matrice de dimensions $\mathbb{R}^{2 \times 7}$. L'architecture ANN 100 de la figure 1 correspond à un réseau de classification, et le nombre de neurones dans la couche de sortie correspond ainsi au nombre de classes, l'exemple de la figure 1 comportant trois classes.
- [0046] Le mappage $y = f(x)$ appliqué par l'architecture ANN 100 est une fonction d'agrégation, comprenant une fonction associative g_n dans chaque couche, ces fonctions étant connectées en chaîne pour mapper $y = f(x) = g_1(g_2(\dots(g_n(x))\dots))$. Il y a seulement deux fonctions de la sorte dans l'exemple simple de la figure 1, correspondant à celles de la couche cachée et de la couche de sortie.
- [0047] Chaque neurone de la couche cachée reçoit le signal provenant de chaque neurone d'entrée, un paramètre correspondant θ_j^i étant appliqué à chaque neurone j de la couche cachée à partir de chaque neurone d'entrée i de la couche d'entrée. La figure 1 illustre les paramètres θ_1^1 à θ_7^1 appliqués aux sorties d'un premier des neurones d'entrée vers chacun des sept neurones cachés.
- [0048] Le but du modèle neuronal défini par l'architecture 100 est d'approximer une certaine fonction $F : X \rightarrow Y$ par l'intermédiaire de l'ensemble de paramètres θ . Le modèle correspond à un mappage $y = f(x; \theta)$, les paramètres θ étant par exemple modifiés pendant l'apprentissage sur la base d'une fonction objective, comme une fonction de coût. Dans certains modes de réalisation, la fonction de mappage est basée sur une projection non linéaire φ , appelée en général fonction d'activation, de sorte que la fonction de mappage f peut être définie comme $y_p = f(x; \theta, w) = \varphi(x; \theta)^T w$, où θ sont les paramètres de φ , et w est une valeur de vecteur. En général, on utilise une même fonction pour toutes les couches, mais il est aussi possible d'utiliser une fonction différente pour chaque couche. Dans certains cas, une fonction d'activation linéaire φ pourrait aussi être utilisée, le choix entre une fonction linéaire et non linéaire dépendant du modèle particulier et des données d'apprentissage.

- [0049] La valeur de vecteur W est par exemple évaluée par la fonction non linéaire φ en tant qu'exemple d'agrégation. Par exemple, la valeur de vecteur W est constituée de poids W , et chaque neurone k de la couche de sortie reçoit les sorties de chaque neurone j de la couche cachée pondérées par un poids correspondant parmi les poids W_j^k . La valeur de vecteur peut par exemple être vue comme étant une autre couche cachée avec une fonction d'activation non linéaire φ et ses paramètres W . La figure 1 représente les poids W_1^1 à W_1^3 appliqués entre la sortie d'un neurone supérieur de la couche cachée et chacun des trois neurones de la couche de sortie.
- [0050] La projection non linéaire φ est par exemple sélectionnée manuellement, par exemple une fonction sigmoïde. Les paramètres θ de la fonction d'activation sont toutefois appris par apprentissage, par exemple sur la base de la règle de descente de gradient. D'autres caractéristiques de l'architecture ANN, comme la profondeur du modèle, le choix de l'optimiseur pour la descente de gradient et la fonction de coût, sont aussi par exemple sélectionnées manuellement.
- [0051] Il y a deux procédures qui peuvent être appliquées à un ANN comme l'ANN 100 de la figure 1, l'une étant une procédure d'apprentissage ou de propagation vers l'arrière afin d'apprendre les paramètres θ , et l'autre étant une procédure d'inférence ou de propagation vers l'avant, pendant laquelle des valeurs d'entrée X s'écoulent à travers la fonction, et sont multipliées par les calculs intermédiaires définissant la fonction de mappage f , afin de générer une sortie Y .
- [0052] La figure 2 représente un espace bidimensionnel donnant un exemple de modèle qui classe des éléments en trois classes. Dans l'exemple de la figure 2, un réseau neuronal artificiel, comme l'ANN 100 de la figure 1, est entraîné pour mapper des échantillons d'entrée définis sous forme de points représentés par des paires de valeurs d'entrée X_1 et X_2 en l'une de trois classes C , D et E .
- [0053] À titre d'exemple, $X \in \mathbb{R}^2$, où X_1 est une caractéristique de poids, X_2 est une caractéristique de hauteur correspondante, et la fonction $y_p = f(X; \theta)$ mappe les échantillons de hauteur et de poids en une classification de chat (C), chien (D) ou éléphant (E). En d'autres termes, l'ANN est entraîné pour définir une frontière non linéaire entre des chats, des chiens et des éléphants sur la base d'une caractéristique de poids et d'une caractéristique de hauteur d'un animal, chaque échantillon décrit par ces caractéristiques tombant dans l'une des trois classes.
- [0054] L'espace défini par la valeur X_1 sur l'axe y et la valeur X_2 sur l'axe x est divisé en trois régions de 202, 204 et 206 correspondant respectivement aux classes C , D et E . Dans la région de 202, chaque échantillon a une probabilité supérieure de tomber dans la classe C que dans l'une ou l'autre des autres classes D et E , et pareillement pour les

régions 204 et 206. Une frontière 208 entre les classes C et D et une frontière 210 entre les classes D et E représentent l'incertitude du modèle, c'est-à-dire que, le long de ces frontières, les échantillons ont des probabilités égales d'appartenir à chacune des deux classes séparées par la frontière. Des contours en figure 2 représentent les distributions d'échantillons dans la région associée à chaque classe, les zones centrales marquées C, D et E correspondant à la densité d'échantillons la plus haute. Un contour extérieur dans chaque région 202, 204, 206 indique la limite des échantillons, la région extérieure au contour extérieur dans chaque région de 202, 204, 206 correspondant par exemple à des échantillons situés en dehors de l'ensemble.

[0055] Comme cela a été expliqué dans la section d'art antérieure susmentionnée, dans certains modes de réalisation il peut être souhaitable d'effectuer l'apprentissage de plusieurs ANN destinés à réaliser une même fonction. Une solution pourrait consister à effectuer l'apprentissage de chaque ANN en utilisant le même ensemble d'échantillons de données brutes constituant les données d'apprentissage. Toutefois, cela impliquerait de conserver les données d'apprentissage afin de permettre l'apprentissage de nouveaux ANN, ce qui est coûteux en ressources matérielles, et dans certains cas le jeu de données d'apprentissage d'origine pourrait ne plus être disponible au moment où l'on souhaite transférer la connaissance.

[0056] Ce problème pourrait être évité en utilisant un apprentissage par transfert, mais comme cela a été expliqué précédemment, le transfert de paramètres appris à partir d'un ANN entraîné vers un deuxième ANN est seulement applicable si les architectures des deux ANN sont basées sur le même modèle, en d'autres termes si le nouveau modèle du deuxième ANN conserve l'architecture d'origine de l'ANN entraîné afin de transférer la fonction d'origine $f(x; \theta)$. En effet, si le deuxième ANN a une profondeur différente de celle de l'ANN entraîné, qu'elle soit moins profonde ou plus profonde, il ne pourra pas prendre en charge les paramètres d'origine. Puisque l'architecture d'origine doit rester fixe, cette technique n'offre pas de solution souple.

[0057] Une autre solution qui présente l'avantage de ne pas nécessiter la mémorisation des données d'apprentissage brutes consiste à utiliser un ANN entraîné pour générer des données d'apprentissage artificielles qui caractérisent la fonction f du modèle d'origine, et qui permettent ainsi à de nouveaux modèles ayant des profondeurs différentes de celle du modèle d'origine d'apprendre une approximation de la fonction f . Une telle technique est appelée ici transfert de connaissance.

[0058] La figure 2 représente une approche simpliste pour générer ces données d'apprentissage, qui implique la génération de valeurs d'entrée aléatoires, correspondant à des échantillons aléatoires dans l'espace d'échantillons. Des exemples de telles échantillons aléatoires sont représentés par de petits cercles 212 en figure 2, dont seulement certains comportent des références pour faciliter l'illustration. En appliquant

ces échantillons aléatoires à l'ANN classificateur entraîné, et en mémorisant les classifications résultantes, des données d'apprentissage peuvent être générées. En effet, chaque ensemble d'un échantillon aléatoire et d'une étiquette correspondante forme une paire d'apprentissage, et ces ensembles d'apprentissage caractérisent et représentent approximativement la fonction de classification $X \rightarrow Y$, ou $y = F(X)$, de l'ANN entraîné. Les paires d'apprentissage peuvent par conséquent être utilisées pour l'apprentissage d'un nouvel ANN. Par exemple, de telles données d'apprentissage peuvent être utilisées pour capturer, dans une certaine mesure, les frontières de décision 208, 210 du modèle d'origine. Toutefois, une limitation d'un tel procédé est que, à moins que le jeu d'apprentissage soit très grand, des régions intéressantes de l'espace d'entrée peuvent être omises du jeu d'apprentissage. Cela est particulièrement le cas lorsque l'espace d'entrée a des dimensions relativement grandes. Cela signifie que plus le nombre de dimensions à échantillonner est grand, plus la probabilité est faible qu'une région soit préservée. Il y a donc un problème technique pour générer des données d'apprentissage pour l'apprentissage d'ANN non entraînés qui permettent de capturer effectivement le modèle d'origine.

[0059] La figure 3 représente schématiquement une architecture ANN 300 selon un exemple de réalisation de la présente description. L'ANN 300 de la figure 3 est similaire à l'ANN 100 de la figure 1, mais il comprend en plus une partie auto-associative capable de reproduire les données d'entrée en utilisant des neurones de la couche de sortie. Ainsi, ce modèle réalise une incorporation à partir de $\mathbb{R}^n \rightarrow \mathbb{R}^n \times \{1, 2, \dots, c\}$, où n sont les caractéristique, et c les classes. Comme dans l'exemple de la figure 1, dans l'ANN 300 de la figure 3, chaque échantillon d'entrée comporte deux valeurs, correspondant à un espace d'entrée bidimensionnel, et il y a ainsi aussi deux neurones de sortie additionnels correspondants (FEATURES) pour générer un pseudo-échantillon de sortie (X') reproduisant l'échantillon d'entrée. Par exemple, comme dans l'exemple des figures 1 et 2, les valeurs d'entrée de chaque échantillon représentent un poids (W) et une hauteur (H), et l'ANN 300 classe ces échantillons comme étant soit des chats (C), des chiens (D) ou des éléphants (E), correspondant à l'étiquette (LABELS) formant la valeur de sortie Y .

[0060] La partie auto-associative de l'ANN 300 se comporte de manière similaire à un auto-codeur. Les auto-codeurs sont un type d'ANN connu de la personne du métier dans lequel, plutôt que d'être entraînés pour réaliser une classification, ils sont entraînés pour reproduire leurs entrées au niveau de leurs sorties. Comme indiqué précédemment, le terme "auto-associative" est utilisé ici pour désigner une fonctionnalité similaire à celle d'un auto-codeur, excepté que l'espace latent n'est pas nécessairement comprimé. En outre, comme pour l'apprentissage d'un auto-codeur, l'apprentissage de la partie auto-associative pour l'ANN peut être réalisé avec certaines contraintes afin

d'éviter que l'ANN converge rapidement vers la fonction identité, comme cela est bien connu de la personne du métier.

- [0061] L'ANN 300 est par exemple mis en œuvre par du matériel dédié, comme un ASIC (circuit intégré spécifique d'application), ou par une émulation logicielle exécutée sur un dispositif informatique, ou par une combinaison de matériel dédié et de logiciel.
- [0062] Dans l'exemple de la figure 3, le réseau est commun pour la partie auto-associative et pour la partie de classification, excepté dans la couche de sortie. En outre, chacun des neurones de sortie W et H de la partie auto-associative reçoit des sorties de chacun des neurones de la couche cachée. Toutefois, dans des variantes de réalisation, il pourrait y avoir une plus faible quantité de chevauchement, ou pas de chevauchement du tout, entre la partie auto-associative et la partie de classification de l'ANN 300. En effet, comme cela est décrit plus en détail ci-après, dans certains modes de réalisation, les fonctions auto-associative et hétéro-associative pourraient être mises en œuvre par des réseaux neuronaux séparés. Dans certains modes de réalisation, en plus des neurones communs dans la couche d'entrée, il y a au moins un autre neurone commun dans les couches cachées entre la partie auto-associative et la partie de classification de l'ANN 300. Un neurone commun implique que ce neurone fournit sa sortie directement, ou indirectement, c'est-à-dire par l'intermédiaire d'un ou plusieurs neurones d'autres couches, à au moins l'un des neurones de sortie de la partie auto-associative et au moins l'un des neurones de sortie de la partie de classification.
- [0063] Comme cela est illustré en figure 3, une réinjection est réalisée à partir des sorties auto-associatives vers les entrées de l'ANN. Une telle réinjection est réalisée afin de générer des données d'apprentissage, et comme on va le décrire plus en détail ci-après, la réinjection est par exemple réalisée par un générateur de données qui est couplé à l'ANN. Ainsi, la partie auto-associative du modèle d'ANN est utilisée comme fonction récursive, en ce que ses sorties sont utilisées en tant que ses entrées. Cela entraîne une trajectoire des sorties, dans laquelle, après chaque réinjection, les échantillons générés se rapprochent des échantillons bruts réels dans des régions d'intérêt de l'espace d'entrée. Avantageusement, selon les modes de réalisation décrits ici, pour chaque germe injecté dans l'ANN, au moins deux points se trouvant sur cette trajectoire sont par exemple utilisés pour former des données d'apprentissage pour l'apprentissage d'un autre ANN.
- [0064] On va maintenant décrire la génération de données d'apprentissage pour un transfert de connaissance basé sur l'ANN 300 plus en détail en faisant référence aux figures 4 à 8.
- [0065] La figure 4 représente schématiquement un système 400 de transfert de connaissance selon un exemple de réalisation de la présente description.
- [0066] Le système 400 comprend un ou plusieurs réseaux neuronaux artificiels 402, chacun

correspondant par exemple à un ANN similaire à celui de la figure 3, et comprenant, en particulier, au moins une partie auto-associative. Dans l'exemple de la figure 4, les fonctions appliquées par les ANN sont référencées f1 à fn.

- [0067] Dans un exemple, il y a un seul ANN entraîné 402, et l'on souhaite générer des données d'apprentissage afin de transférer la connaissance acquise de l'unique ANN 402 vers au moins un autre ANN ayant un modèle différent de celui de l'ANN entraîné 402.
- [0068] Dans un autre exemple, il y a une pluralité d'ANN 402 entraînés, et l'on souhaite transférer la connaissance de la pluralité d'ANN entraînés 402 vers au moins un autre ANN, chaque autre ANN étant entraîné pour mettre en œuvre la totalité des fonctions de la pluralité d'ANN entraînés 402. En d'autres termes, la connaissance peut être fédérée à partir de multiples ANN, comme de multiples ANN classificateurs, vers un seul ANN, comme un seul ANN classificateur.
- [0069] Dans encore un autre exemple, il y a une pluralité d'ANN entraînés 402, et l'on souhaite transférer la connaissance de la pluralité d'ANN 402 entraînés vers une pluralité d'autres ANN.
- [0070] Le système 400 comprend aussi un générateur de données (DATA GENERATOR) 404 agencé pour faire usage de fonctions auto-associatives d'un ou plusieurs des ANN entraînés 402 afin de générer des pseudo-données (PSEUDO DATA) pour l'apprentissage d'un ou plusieurs autres ANN 406.
- [0071] Le générateur de données 404 reçoit par exemple une valeur de germe (SEED) générée par un générateur de germes (SEED GEN) 408. Le générateur de germes 408 est par exemple mis en œuvre par un générateur pseudo-aléatoire ou similaire, et génère des valeurs basées sur une distribution aléatoire donnée, comme on va le décrire plus en détail ci-après.
- [0072] Le générateur de données 404 génère par exemple des valeurs d'entrée (INPUTS) fournies auxdits un ou plusieurs ANN 402, reçoit des valeurs de sortie (OUTPUTS) provenant desdits un ou plusieurs ANN 402, et génère des données d'apprentissage (PSEUDO DATA) comprenant les pseudo-échantillons et des pseudo-étiquettes résultantes, comme on va le décrire plus en détail ci-après. Les pseudo-données sont par exemple utilisées au vol pour l'apprentissage desdits un ou plusieurs autres ANN 406, ou elles sont mémorisées dans un ou plusieurs fichiers, qui sont par exemple mémorisés par une mémoire, comme un dispositif mémoire non transitoire. Par exemple, les pseudo-données sont mémorisées dans un seul fichier, ou, dans le cas où il y a une pluralité d'autres ANN 406 différents à entraîner, les pseudo-données sont par exemple mémorisées dans une pluralité de fichiers associés aux fonctions f1 à fn mises en œuvre par les ANN.
- [0073] Dans certains modes de réalisation, les fonctionnalités du générateur de données 404

sont mises en œuvre par un dispositif de traitement (P) 410, qui par exemple exécute des instructions de logiciel mémorisées dans une mémoire (M) 412. En variante, le générateur de données 404 pourrait être mis en œuvre par du matériel dédié, comme par un ASIC.

- [0074] Lesdits un ou plusieurs autres ANN 406 à entraîner peuvent correspondre à une ou plusieurs architectures classiques qui sont agencées pour réaliser seulement une classification, par exemple du type décrit en relation avec la figure 1 précédente. En variante, un ou plusieurs des autres ANN 406 à entraîner pourraient comporter des parties auto-associatives ou auto-codeuses en plus de la fonction de classification, ces ANN étant par exemple du type représenté en figure 3. Il serait aussi possible qu'un ou plusieurs des autres ANN soient entraînés pour avoir seulement une fonctionnalité auto-associative, comme on va le décrire plus en détail ci-après.
- [0075] La figure 5 est un organigramme illustrant des étapes dans un procédé de transfert de connaissance selon un exemple de réalisation de la présente description. Ce procédé est par exemple mis en œuvre par le système 400 de la figure 4.
- [0076] Dans une étape 501, une variable s est initialisée, par exemple à 1, et une première valeur de germe est générée par le générateur de germes 408.
- [0077] Dans une étape 502, la première valeur de germe est par exemple appliquée par le générateur de données 404 en tant qu'entrée desdits un ou plusieurs ANN 402. Ainsi, chacun desdits un ou plusieurs ANN 402 propage le germe X_0 à travers ses couches et génère, au niveau de sa couche de sortie, des étiquettes Y_0 correspondant à la classification du germe, et des caractéristiques X_0' correspondant au germe modifié sur la base de la partie auto-associative entraînée de l'ANN.
- [0078] Dans un but de classification, il est en général souhaité que les pseudo-étiquettes générées d'un ANN soient normalisées, par exemple en utilisant un codage du type 1 parmi n , pour indiquer la classe déterminée. Toutefois, en réalité, l'ANN va générer des sorties non normalisées qui représentent la probabilité relative que l'échantillon d'entrée tombe dans chaque classe, en d'autres termes la probabilité relative pour affecter une probabilité de toutes les classes, au lieu d'une classe discrète. Avantageusement, les données d'apprentissage comprennent des pseudo-étiquettes sous la forme des données de sortie non normalisées, fournissant ainsi de plus grandes informations pour l'apprentissage des autres ANN, et en particulier comprenant les informations qui sont fournies pour toutes les classes, et pas seulement la classe qui est sélectionnée. Par exemple, on peut utiliser des procédés de logits ou de distillation pour l'apprentissage d'un modèle utilisant des pseudo-étiquettes, comme cela est connu de la personne du métier. Cela utilise par exemple l'entropie croisée binaire. La distillation est par exemple décrite plus en détail dans la publication de Geoffrey Hinton et al. intitulée "Distilling the Knowledge in a Neural Network"

(arXiv.1503.02531v1, 9 March 2015), et dans la demande de brevet américain publiée sous la référence US2015/0356461. Dans le cas d'échantillons synthétiques qui peuvent ne pas appartenir nettement à une classe particulière, on utilise par exemple un procédé de logit/distillation comme cela est connu de la personne du métier, ce procédé étant par exemple utilisé pour affecter des probabilités de toutes les classes au lieu d'une classe discrète. Les probabilités relatives indiquent comment un modèle tend à se généraliser et aide à transférer la capacité de généralisation d'un modèle entraîné vers un nouveau modèle.

- [0079] Dans une étape 503, on détermine si la variable s a atteint une valeur S , qui est par exemple une condition d'arrêt pour le nombre de réinjections en fonction de chaque germe. Dans un exemple, la valeur S est égale à 6, mais plus généralement elle pourrait être comprise entre 3 et 20, et par exemple entre 4 et 10, en fonction de la taille de l'espace d'entrée, et en fonction de la qualité de l'auto-association entraînée. En effet, lorsque l'auto-association est bien entraînée, en d'autres termes de telle sorte qu'il y a une erreur relativement faible entre les entrées dans les reproductions du réseau, on peut par exemple utiliser relativement peu de réinjections, par exemple moins de 10, pour assurer un bon échantillonnage de l'espace d'entrée. Sinon, on peut utiliser un nombre de réinjections relativement élevé, par exemple compris entre 10 et 20, pour trouver les régions d'intérêt.
- [0080] Dans des variantes de réalisation, plutôt que la condition d'arrêt dans l'étape 503 soit un nombre fixe de réinjections, elle pourrait à la place être basée sur la variation entre les reproductions, comme par exemple basée sur une mesure de la distance euclidienne, ou un autre type quelconque de distance, entre les deux dernières projections. Par exemple, si la distance euclidienne est descendue en dessous d'un seuil donné, la condition d'arrêt est satisfaite. En effet, plus les reproductions se rapprochent les unes des autres, plus les pseudo-échantillons se rapprochent de la distribution d'échantillons vraie sous-jacente.
- [0081] Initialement la variable s est mise à 1, et ainsi n'est pas égale à S . Par conséquent, l'étape suivante est une étape 504, dans laquelle le pseudo-échantillon à la sortie de chacun desdits un ou plusieurs ANN 402 est réinjecté dans l'ANN correspondant. Ensuite, dans une étape 505, le pseudo-échantillon réinjecté dans chacun desdits un ou plusieurs ANN 402 dans l'étape 504, et la pseudo-étiquette de sortie correspondante provenant de chacun desdits un ou plusieurs ANN 402, sont par exemple mémorisés pour former des données d'apprentissage, comme on va le décrire maintenant plus en détail en faisant référence à la figure 6.
- [0082] La figure 6 représente un espace bidimensionnel donnant un exemple d'un modèle qui classe des éléments en trois classes, et un exemple de pseudo-échantillons dans cet espace qui suivent une trajectoire de pseudo-échantillons à partir d'un germe aléatoire

jusqu'à un pseudo-échantillon final.

- [0083] L'exemple de la figure 6 est basé sur les mêmes classes C, D et E, et les mêmes frontières de classe 208, 210, que l'exemple de la figure 2. Un exemple du germe est représenté par une étoile 602 en figure 6, et une trajectoire de pseudo-échantillons 604, 606, 608, 610, 612 et 614 générés en commençant avec ce germe est aussi représentée. Chacun de ces pseudo-échantillons résulte par exemple d'une réinjection du pseudo-échantillon précédent. Après un certain nombre de réinjections, égal à six réinjections dans l'exemple de la figure 6, la réinjection est par exemple stoppée avec un pseudo-échantillon final représenté par une étoile 614 en figure 6. Comme cela est représenté par l'étape 505, des valeurs d'entrée et de sortie correspondant à chaque point sur la trajectoire sont par exemple mémorisées pour former les données d'apprentissage. En variante, seul un sous-ensemble des points est utilisé pour former les données d'apprentissage. Par exemple, on utilise au moins deux points sur la trajectoire.
- [0084] Étant donné que la partie auto-associative desdits un ou plusieurs ANN 402 a été entraînée pour reproduire des échantillons réels au niveau de sa sortie, ces ANN ont été entraînés sur la base de la distribution de ces échantillons réels, comme cela est représenté par les contours dans les figures 2 et 6. Ainsi, lorsqu'un échantillon aléatoire est fourni à ces ANN, ils vont générer des sorties sollicitées vers la distribution des échantillons réels. Cela explique le saut dans les pseudo-échantillons générés à la suite de chaque réinjection. Les présents inventeurs ont montré que cela est une propriété de tout modèle auto-associatif. En effet, en théorie, un modèle auto-associatif ne reproduit pas avec précision lorsqu'il est face à des échantillons aléatoires ou des pseudo-échantillons puisque il n'a pas été entraîné pour reproduire du bruit aléatoire par sa distribution d'apprentissage. La capacité à reproduire toute entrée implique que le modèle auto-associatif a appris la fonction identité. Il y a de nombreuses façons connues pour empêcher qu'un tel modèle apprenne la fonction identité, mais dans tous les cas, en général, un modèle ne va pas apprendre naturellement la fonction identité.
- [0085] En faisant de nouveau référence à la figure 5, dans une étape 506, la variable s est incrémentée, puis le procédé revient à l'étape 503. Cette boucle se répète jusqu'à ce que, dans l'étape 503, la variable s soit égale à la limite S . Ensuite, l'opération suivante est une étape 507.
- [0086] Dans l'étape 507, on détermine si un autre critère d'arrêt a été satisfait. Par exemple, cet autre critère d'arrêt pourrait être basé sur le fait qu'un nombre global de pseudo-échantillons a été généré, le procédé se terminant par exemple lorsque le nombre de pseudo-échantillons est considéré comme suffisamment élevé pour permettre l'apprentissage d'un ou plusieurs autres réseaux ANN. Cela peut dépendre par exemple la précision du modèle entraîné.
- [0087] Si, dans l'étape 507, le critère d'arrêt n'a pas été satisfait, le procédé revient à l'étape

501, de sorte qu'un nouveau germe est généré, et un nouvel ensemble de pseudo-échantillons est généré pour ce nouveau germe.

- [0088] Lorsque, dans l'étape 507, le critère d'arrêt a été satisfait, dans une étape 508, lesdits un ou plusieurs autres ANN 406 sont par exemple entraînés sur la base des données d'apprentissage générées. En effet, les pseudo-données recueillies contiennent le modèle de la fonction interne f , et sont par exemple mémorisées sous forme d'un seul fichier qui caractérise le modèle entraîné. Un ou plusieurs autres ANN sont ensuite capables d'apprendre le modèle en utilisant les données d'apprentissage du pseudo-ensemble de données en utilisant des outils d'apprentissage profond qui sont bien connus de la personne du métier.
- [0089] En variante, plutôt que de générer un fichier contenant toutes les données d'apprentissage générées, l'apprentissage desdits un ou plusieurs autres ANN 406 pourrait être réalisé progressivement pendant la génération de données d'apprentissage. En d'autres termes, l'apprentissage est réalisé au moins partiellement en parallèle avec la génération de pseudo-échantillons, ce qui par exemple éviterait d'avoir besoin de mémoriser tous les pseudo-échantillons jusqu'à la fin de la génération des données d'apprentissage.
- [0090] On notera que, dans l'exemple de la figure 5, le premier pseudo-échantillon à être mémorisé est par exemple celui résultant de la première réinjection. Ainsi, le germe lui-même n'est pas utilisé comme valeur d'entrée d'un pseudo-échantillon. En effet, des échantillons aléatoires bruts ne sont pas considérés comme caractérisant efficacement la fonction f qui doit être transférée.
- [0091] En outre, comme cela a été indiqué précédemment, il est aussi possible de sélectionner seulement certains des points de la trajectoire des pseudo-échantillons pour former une partie des données d'apprentissage. Par exemple, dans certains modes de réalisation, on sélectionne des points qui se trouvent près d'une frontière de classe. Par exemple, en faisant référence à la figure 6, dans le cas de la trajectoire allant de 602 à 614, on choisit par exemple au moins les points 608 et 610 pour faire partie des données d'apprentissage, puisque ces points sont particulièrement pertinents pour la définition de la frontière 208. L'étape 505 de la figure 5 peut par conséquent impliquer de détecter si la pseudo-étiquette générée par l'échantillon réinjecté dans l'étape 504 est différente de la pseudo-étiquette générée par l'échantillon réinjecté immédiatement précédent, et si oui, ces deux pseudo-échantillons consécutifs sont par exemple sélectionnés pour faire partie des données d'apprentissage.
- [0092] La figure 7 est un graphique illustrant des exemples de distributions aléatoires d'échantillons aléatoires générés par le générateur de germes 408 de la figure 4 selon un exemple de réalisation de la présente description.
- [0093] Une courbe 702 représente un exemple dans lequel la distribution est une distribution

gaussienne de la forme $\mathbf{X} \sim \mathbf{N}(\mu = 0, \sigma^2 = I)$, bien que de manière plus générale on puisse utiliser toute distribution normale.

- [0094] Une courbe 704 représente un autre exemple dans lequel la distribution est une distribution uniforme ajustée de la forme $\mathbf{X} \sim U(-3, 3)$, bien que de manière plus générale on puisse utiliser une distribution uniforme ajustée de la forme $\mathbf{X} \sim U(-A, A)$, pour $A \geq 1$.
- [0095] Quelle que soit la distribution aléatoire choisie, la même distribution est par exemple utilisée pour générer indépendamment la totalité des germes qui seront utilisés comme point de départ pour les trajectoires de pseudo-échantillons. Autant de valeurs aléatoires que de neurones se trouvant dans la couche d'entrée sont par exemple échantillonnées dans la distribution sélectionnée afin de générer chaque vecteur d'entrée. Ce vecteur d'entrée a ainsi la même longueur que celle de la couche d'entrée du modèle, et appartient à l'espace d'entrée des échantillons vrais.
- [0096] La figure 8 est un graphique illustrant un exemple de fonction d'activation $\varphi(x)$ de l'ANN selon un exemple de réalisation de la présente description. Comme cela est illustré, dans certains modes de réalisation la fonction fournit des sorties non nulles seulement en réponse à des entrées non nulles, ce qui implique que des valeurs négatives générées de manière aléatoire vont être filtrées par le réseau. En effet, le modèle auto-associatif va approximer tout point de la distribution apprise sans tenir compte du point de départ ou de sa fonction d'activation.
- [0097] Dans certains modes de réalisation, plutôt que de réinjecter les valeurs de sortie auto-associatives de l'ANN en tant qu'échantillon d'entrée suivant de l'ANN, les valeurs de sortie sont d'abord modifiées, comme on va le décrire maintenant plus en détail en faisant référence à la figure 9.
- [0098] La figure 9 représente schématiquement un circuit de génération d'échantillons 900 selon un exemple de réalisation de la présente description. Ce circuit 900 est par exemple partiellement mis en œuvre par le générateur de données 404 de la figure 4, et partiellement par l'ANN 300 formant l'un des ANN 402 de la figure 4.
- [0099] Le générateur de données 404 fournit des échantillons d'entrée X_m à l'ANN 300. La partie de classification de l'ANN 300 génère ainsi des pseudo-étiquettes Y_m correspondantes, et la partie auto-associative génère ainsi des pseudo-échantillons X_m' correspondants. Les pseudo-échantillons X_m' sont fournis à un module d'injection de bruit (NOISE INJECTION) 902, qui par exemple ajoute un certain degré de bruit aléatoire aux pseudo-échantillons afin de générer le pseudo-échantillon suivant $X_{(m+1)}$ à fournir à l'ANN 300. Par exemple, dans certains modes de réalisation, le bruit aléatoire est sélectionné dans une distribution gaussienne, comme par exemple la gaussienne $\mathbf{N}(0, I)$ et est par exemple pondéré par un coefficient Z . Par exemple, le coefficient Z est choisi de telle sorte que, après l'injection, la portion de bruit représente

entre 1 % et 30 % de l'amplitude du pseudo-échantillon, et dans certains cas entre 5 % et 20 % de l'amplitude du pseudo-échantillon.

[0100] Par exemple, un multiplexeur 904 reçoit au niveau de l'une de ses entrées un échantillon aléatoire initial X_0 , et au niveau de l'autre de ses entrées les pseudo-échantillons $X_{(m+1)}$. Le multiplexeur sélectionne par exemple l'échantillon initial sur une première itération correspondant à l'étape 502 de la figure 5, et sélectionne l'échantillon $X_{(m+1)}$ sur des itérations suivantes, correspondant aux étapes 504 de la figure 5, jusqu'à ce que le nombre S de réinjections ait été atteint.

[0101] Bien qu'en figure 4 chacun desdits un ou plusieurs ANN 402 comprenne une fonction auto-associative intégrée ainsi que la fonction de classification, dans des variantes de réalisation, ces fonctions pourraient être mises en œuvre par des ANN séparés, comme on va le décrire maintenant plus en détail en faisant référence à la figure 10.

[0102] La figure 10 représente schématiquement un système 1000 de transfert de connaissance selon un autre exemple de réalisation de la présente description. Les éléments de la figure 10 qui sont communs avec des éléments de la figure 4 portent les mêmes références numériques, et ne seront pas décrits de nouveau en détail.

[0103] Dans le mode de réalisation de la figure 10, les fonctions du générateur de données 404 de la figure 4 sont réparties entre un ANN comportant une fonction auto-associative (AUTO-ASSOCIATIVE FUNCTION) 1002, qui peut correspondre à un auto-codeur et comprend par exemple un circuit de réinjection (REINJECTION) 1004, et un classificateur (CLASSIFIER) 1006.

[0104] Le fonctionnement du système 1000 de la figure 10 est par exemple le même que celui décrit en relation avec l'organigramme de la figure 5.

[0105] L'ANN 1002 est par exemple agencé pour reproduire au niveau de sa sortie un échantillon aléatoire qui est fourni par le générateur de germes (SEED GEN) 408. Le circuit de réinjection 1004 est ensuite par exemple agencé pour réinjecter les entrées reproduites présentes au niveau des sorties de l'ANN 1002 vers les entrées de l'ANN 1002, par exemple après une injection de bruit comme cela a été décrit en relation avec la figure 9. En outre, chaque entrée reproduite générée au niveau de la sortie de l'ANN 1002 forme un pseudo-échantillon, qui est fourni au classificateur 1006, et à une mémoire mémorisant les pseudo-données sous la forme d'un fichier.

[0106] Le classificateur 1006 est agencé pour réaliser une inférence sur les pseudo-échantillons et pour générer des pseudo-étiquettes correspondantes (PSEUDO LABELS), dont chacune est par exemple mémorisée en tant que partie des pseudo-données en association avec le pseudo-échantillon correspondant.

[0107] Comme cela a été décrit en relation avec la figure 4, les données d'apprentissage générées sont par exemple utilisées pour l'apprentissage d'un ou plusieurs autres ANN

406.

- [0108] Bien que dans les modes de réalisation décrits précédemment une fonction de classification se trouve dans l'ANN, dans des variantes de réalisation, l'ANN pourrait comporter seulement la fonction auto-associative, sans réaliser de classification, comme on va le décrire maintenant plus en détail en faisant référence à la figure 11.
- [0109] La figure 11 représente schématiquement une architecture ANN 1100 selon un autre exemple de réalisation de la présente description. L'architecture 1100 est similaire à l'architecture ANN 300, et de mêmes éléments portent les mêmes références et ne seront pas décrits de nouveau en détail. L'ANN 1100 comprend une couche de neurones d'entrée (INPUT LAYER), une couche de neurones de sortie (OUTPUT LAYER) et une seule couche cachée de neurones (HIDDEN LAYER), bien que dans des variantes de réalisation il puisse y avoir plusieurs couches cachées. Toutefois, l'ANN 1100 a par exemple seulement une fonction associative, et ainsi ne contient aucune fonction de classification. Dans l'exemple de la figure 11, l'ANN 1100 comporte trois neurones d'entrée correspondant à des canaux d'entrée A, B et C, et ainsi la couche de sortie génère trois canaux de sortie correspondant à A', B' et C', qui sont par exemple réinjectés directement dans la couche d'entrée à chaque itération, ou du bruit aléatoire pourrait être ajouté, comme dans l'exemple de la figure 9.
- [0110] La figure 12A représente schématiquement un système de transfert de connaissance selon un autre exemple de réalisation de la présente description. Dans l'exemple de la figure 12A, un ANN entraîné 1200 est du type de l'ANN 300 de la figure 3, comprenant à la fois une partie auto-associative et une partie hétéro-associative. L'ANN 1200 reçoit, au niveau d'une couche d'entrée 1202, un germe (SEED), et génère au niveau de sa couche de sortie des pseudo-étiquettes 1204 à partir de sa partie hétéro-associative, et des pseudo-échantillons 1206 à partir de sa partie auto-associative. Les pseudo-échantillons sont réinjectés par l'intermédiaire d'un chemin de rétroaction 1208, qui peut impliquer une injection de bruit, comme décrit précédemment.
- [0111] Des données d'apprentissage générées en utilisant l'ANN 1200 sont par exemple utilisées pour l'apprentissage d'un autre ANN 1210, et/ou d'un autre ANN 1220.
- [0112] L'ANN 1210 est aussi du type de l'ANN 300 de la figure 3, comprenant à la fois une partie auto-associative et une partie hétéro-associative, et comporte une couche d'entrée 1212 et une couche de sortie générant des pseudo-étiquettes 1214 à partir de sa partie hétéro-associative, et des pseudo-échantillons 1216 à partir de sa partie auto-associative. Un système d'apprentissage 1216, qui est par exemple mis en œuvre par du matériel et/ou par du logiciel, est par exemple agencé pour effectuer l'apprentissage du réseau 1210 en utilisant les données d'apprentissage, en fournissant des pseudo-échantillons à la couche d'entrée 1212, en recevant les données de sortie résultantes

1214 et 1216, et en ajustant en conséquence les paramètres θ du réseau 1210. Dans ce cas, les données d'apprentissage comprennent par exemple les pseudo-valeurs de données X_m , $X_{(m+1)}$, $X_{(m+2)}$, etc., qui ont été injectées dans le réseau 1200, les pseudo-étiquettes correspondantes Y_m , $Y_{(m+1)}$, $Y_{(m+2)}$, etc., résultant de l'injection de chaque pseudo-valeur de données respective X_m , $X_{(m+1)}$, $X_{(m+2)}$, etc., et les pseudo-échantillons reproduits X_m' , $X_{(m+1)}'$, $X_{(m+2)}'$, etc., résultant de l'injection de chaque pseudo-valeur de donnée respective X_m , $X_{(m+1)}$, $X_{(m+2)}$, etc. En effet, le système d'apprentissage 1216 est par exemple agencé pour entraîner non seulement la partie hétéro-associative du réseau 1210 sur la base des paires pseudo-échantillon/pseudo-étiquette, mais aussi pour entraîner la partie auto-associative du réseau 1210 sur la base des paires pseudo-échantillon/pseudo-échantillon reproduit. En effet, ce dernier apprentissage implique l'apprentissage de la partie auto-associative du réseau 1210 pour générer les mêmes différences que le réseau 1200 entre les pseudo-échantillons injectés et les pseudo-échantillons reproduits au niveau de sa sortie.

- [0113] L'ANN 1220 est un ANN classificateur, comme l'exemple de la figure 1, comprenant une couche d'entrée 1222 et une couche de sortie générant des pseudo-étiquettes 1224. Un système d'apprentissage 1226, qui est par exemple mis en œuvre par du matériel et/ou par du logiciel, est par exemple agencé pour l'apprentissage du réseau 1220 en utilisant les données d'apprentissage, en fournissant des pseudo-échantillons à la couche d'entrée 1222, en recevant les pseudo-étiquettes de sortie 1224 résultantes, et en ajustant en conséquence les paramètres θ du réseau 1220. Ainsi, dans ce cas, les données d'apprentissage ne comprennent pas par exemple les pseudo-échantillons reproduits X_m' , $X_{(m+1)}'$, $X_{(m+2)}'$, etc., résultant de l'injection de chaque pseudo-valeur de données respective X_m , $X_{(m+1)}$, $X_{(m+2)}$, etc. dans le réseau 1200.
- [0114] La figure 12B représente schématiquement un système de transfert de connaissance selon encore un autre exemple de réalisation de la présente description.
- [0115] Dans l'exemple de la figure 12B, un ANN entraîné 1250 est du type de l'ANN 1100 de la figure 11, mettant en œuvre seulement une fonction auto-associative. L'ANN 1250 est représenté de manière similaire à l'ANN 1200, excepté que les sorties de pseudo-étiquettes 1204 ne sont plus présentes.
- [0116] Des données d'apprentissage générées en utilisant l'ANN 1250 sont par exemple utilisées pour l'apprentissage d'un autre ANN 1260, qui est par exemple similaire à l'ANN 1250, comprenant une couche d'entrée 1262 et une couche de sortie 1264. Un système d'apprentissage 1266, qui est par exemple mis en œuvre par du matériel et/ou par du logiciel, est par exemple agencé pour l'apprentissage du réseau 1260 en utilisant les données d'apprentissage, en fournissant des pseudo-échantillons à la couche d'entrée 1262, en recevant les données de sortie résultantes 1264, et en ajustant en

conséquence les paramètres θ du réseau 1260. Dans ce cas, les données d'apprentissage comprennent par exemple les pseudo-valeurs de données X_m , $X(m+1)$, $X(m+2)$, etc., qui ont été injectées dans le réseau 1250, et les pseudo-échantillons reproduits X_m' , $X(m+1)'$, $X(m+2)'$, etc. résultant de l'injection de chaque pseudo-valeur de données respective X_m , $X(m+1)$, $X(m+2)$, etc.

- [0117] Un avantage des modes de réalisation décrits ici est qu'on peut générer des données d'apprentissage qui capturent relativement bien des régions d'intérêt de l'espace d'entrée d'une fonction donnée, de sorte que l'apprentissage d'un ou plusieurs nouveaux réseaux peut être effectué relativement rapidement et avec précision. Par exemple, en utilisant, pour chaque injection de germe, au moins deux points, en excluant un premier point, sur une trajectoire de pseudo-échantillons générés par une réinjection dans un réseau auto-associatif entraîné, les présents inventeurs ont trouvé qu'on peut générer des données d'apprentissage particulièrement efficaces. En particulier, des données d'apprentissage pertinentes peuvent être générées dans le cas d'un classificateur en détectant le moment où une frontière de classe est traversée, et en utilisant les points de chaque côté de la frontière de classe.
- [0118] En outre, à la différence de nombreuses solutions proposées précédemment, la solution proposée ici est entièrement agnostique en ce qui concerne la relation entre le ou les ANN entraînés, et le ou les ANN cibles vers lesquels la connaissance doit être transférée.
- [0119] Divers modes de réalisation et variantes ont été décrits. La personne du métier comprendra que certaines caractéristiques de ces divers modes de réalisation et variantes pourraient être combinées, et d'autres variantes apparaîtront à la personne du métier. Par exemple, bien qu'on ait décrit des modes de réalisation dans lesquels des ANN non entraînés sont entraînés en utilisant des données d'apprentissage, il serait aussi possible de transférer un ou plusieurs des paramètres de l'ANN entraîné, comme les poids appris d'une première couche de l'ANN, vers l'ANN non entraîné afin d'accélérer le transfert de connaissance. En effet, même si les modèles des ANN entraînés et non entraînés ne sont pas identiques, il peut être possible de transférer au moins certains des paramètres.
- [0120] Enfin, la mise en œuvre pratique des modes de réalisation et variantes décrits est à la portée de la personne du métier à partir des indications fonctionnelles données ci-dessus. Par exemple, l'apprentissage d'un ANN en utilisant une technique d'apprentissage profond est bien connu de la personne du métier et n'a pas été décrit en détail.

Revendications

- [Revendication 1] Procédé de génération de données d'apprentissage pour transférer de la connaissance d'un réseau neuronal artificiel entraîné (402, 1002, 1200, 1250) vers un autre réseau neuronal artificiel (406, 1210, 1220, 1260), le procédé comprenant :
- a) injecter un échantillon aléatoire (SEED) dans le réseau neuronal artificiel entraîné, le réseau neuronal artificiel entraîné étant agencé pour mettre en œuvre au moins une fonction auto-associative pour reproduire des échantillons d'entrée au niveau d'une ou plusieurs de ses sorties ;
 - b) réinjecter un pseudo-échantillon (X_m), généré sur la base de l'échantillon reproduit présent au niveau desdites une ou plusieurs sorties du réseau neuronal artificiel entraîné, dans le réseau neuronal artificiel entraîné afin de générer un nouvel échantillon reproduit (X_m') au niveau desdites une ou plusieurs sorties ; et
 - c) répéter b) une ou plusieurs fois pour générer une pluralité de pseudo-échantillons réinjectés ($X_m, X_{(m+1)}, \dots$) ;
- dans lequel les données d'apprentissage pour l'apprentissage de l'autre réseau neuronal artificiel (406, 1210, 1220, 1260) comprennent au moins deux des pseudo-échantillons réinjectés ($X_m, X_{(m+1)}, \dots$) issus du même échantillon aléatoire (SEED) et des valeurs de sortie correspondantes ($Y_m, Y_{(m+1)}, \dots, X_m', X_{(m+1)}', \dots$) générées par le réseau neuronal artificiel entraîné.
- [Revendication 2] Procédé selon la revendication 1, dans lequel le réseau neuronal artificiel entraîné (402), ou un autre réseau neuronal artificiel entraîné (1006), est agencé pour mettre en œuvre une fonction de classification, et dans lequel les valeurs de sortie correspondantes des données d'apprentissage comprennent des pseudo-étiquettes ($Y_m, Y_{(m+1)}, \dots$) générées par la fonction de classification sur la base des pseudo-échantillons réinjectés.
- [Revendication 3] Procédé selon la revendication 2, comprenant en outre une détection, sur la base des pseudo-étiquettes, du moment où une frontière (208, 210) entre deux espaces de pseudo-étiquettes est traversée entre des réinjections consécutives de deux des pseudo-échantillons, dans lequel lesdits au moins deux pseudo-échantillons réinjectés forment les données d'apprentissage comprennent au moins les deux pseudo-échantillons réinjectés consécutivement.
- [Revendication 4] Procédé selon la revendication 2 ou 3, dans lequel les pseudo-étiquettes

- sont des sorties non normalisées de la fonction de classification.
- [Revendication 5] Procédé selon l'une quelconque des revendications 1 à 4, dans lequel l'autre réseau neuronal artificiel (406) est agencé pour mettre en œuvre au moins une fonction auto-associative pour reproduire des échantillons d'entrée au niveau d'une ou plusieurs de ses sorties, et dans lequel les valeurs de sortie correspondantes des données d'apprentissage comprennent les nouveaux échantillons reproduits (X_m' , $X_{(m+1)'}$, ...) générés par la fonction auto-associative du réseau neuronal artificiel entraîné (402, 1002, 1200, 1250) sur la base des pseudo-échantillons réinjectés.
- [Revendication 6] Procédé selon l'une quelconque des revendications 1 à 5, comprenant en outre :
- d) répéter a), b) et c) au moins une fois sur la base de nouveaux échantillons aléatoires afin de générer, à chaque répétition, au moins deux autres pseudo-échantillons réinjectés formant les données d'apprentissage.
- [Revendication 7] Procédé selon l'une quelconque des revendications 1 à 6, comprenant en outre la génération de l'échantillon aléatoire sur la base d'une distribution normale ou sur la base d'une distribution uniforme ajustée.
- [Revendication 8] Procédé selon l'une quelconque des revendications 1 à 7, dans lequel la génération du pseudo-échantillon comprend l'injection de bruit dans l'échantillon reproduit présent au niveau desdites une ou plusieurs sorties du réseau neuronal artificiel entraîné.
- [Revendication 9] Procédé de transfert de connaissance à partir d'un réseau neuronal artificiel entraîné (402, 1002, 1200, 1250) vers un ou plusieurs autres réseaux neuronaux artificiels (406, 1210, 1220, 1260), le procédé comprenant :
- générer des données d'apprentissage en utilisant le procédé de l'une quelconque des revendications 1 à 8 ; et
 - entraîner l'autre réseau neuronal artificiel sur la base des données d'apprentissage générées.
- [Revendication 10] Système de génération de données d'apprentissage pour transférer de la connaissance à partir d'un réseau neuronal artificiel entraîné (402, 1002, 1200, 1250) vers un autre réseau neuronal artificiel (406, 1210, 1220, 1260), le système comprenant un générateur de données (404) agencé pour :
- a) injecter un échantillon aléatoire (SEED) dans le réseau neuronal artificiel entraîné, le réseau neuronal artificiel entraîné étant agencé pour

mettre en œuvre au moins une fonction auto-associative pour reproduire des échantillons d'entrée au niveau d'une ou plusieurs de ses sorties ;
 b) réinjecter un pseudo-échantillon (X_m), généré sur la base de l'échantillon reproduit présent au niveau desdites une ou plusieurs sorties du réseau neuronal artificiel entraîné, dans le réseau neuronal artificiel entraîné afin de générer un nouvel échantillon reproduit (X_m') au niveau desdites une ou plusieurs sorties ; et
 c) répéter b) une ou plusieurs fois pour générer une pluralité de pseudo-échantillons réinjectés ($X_m, X_{(m+1)}, \dots$) ;
 dans lequel le générateur de données (404) est en outre agencé pour générer les données d'apprentissage pour l'apprentissage de l'autre réseau neuronal artificiel (406, 1210, 1220, 1260) de manière à comprendre au moins deux des pseudo-échantillons réinjectés ($X_m, X_{(m+1)}, \dots$) issus du même échantillon aléatoire (SEED) et des valeurs de sortie correspondantes ($Y_m, Y_{(m+1)}, \dots, X_m', X_{(m+1)}', \dots$) générées par le réseau neuronal artificiel entraîné.

[Revendication 11] Système selon la revendication 10, comprenant en outre l'autre réseau neuronal artificiel (406, 1210, 1220, 1260), et un système d'apprentissage (1218, 1226, 1266) agencé pour entraîner l'autre réseau neuronal artificiel sur la base des données d'apprentissage.

[Revendication 12] Système selon la revendication 11, dans lequel le réseau neuronal artificiel entraîné (402, 1200), ou un autre réseau neuronal artificiel entraîné (1006), est agencé pour mettre en œuvre une fonction de classification, et dans lequel le générateur de données (404) est agencé pour générer des données d'apprentissage de manière à comprendre en outre des pseudo-étiquettes ($Y_m, Y_{(m+1)}, \dots$) générées par la fonction de classification sur la base des pseudo-échantillons réinjectés, et dans lequel l'autre réseau neuronal artificiel (1210, 1220) est capable de mettre en œuvre une fonction de classification.

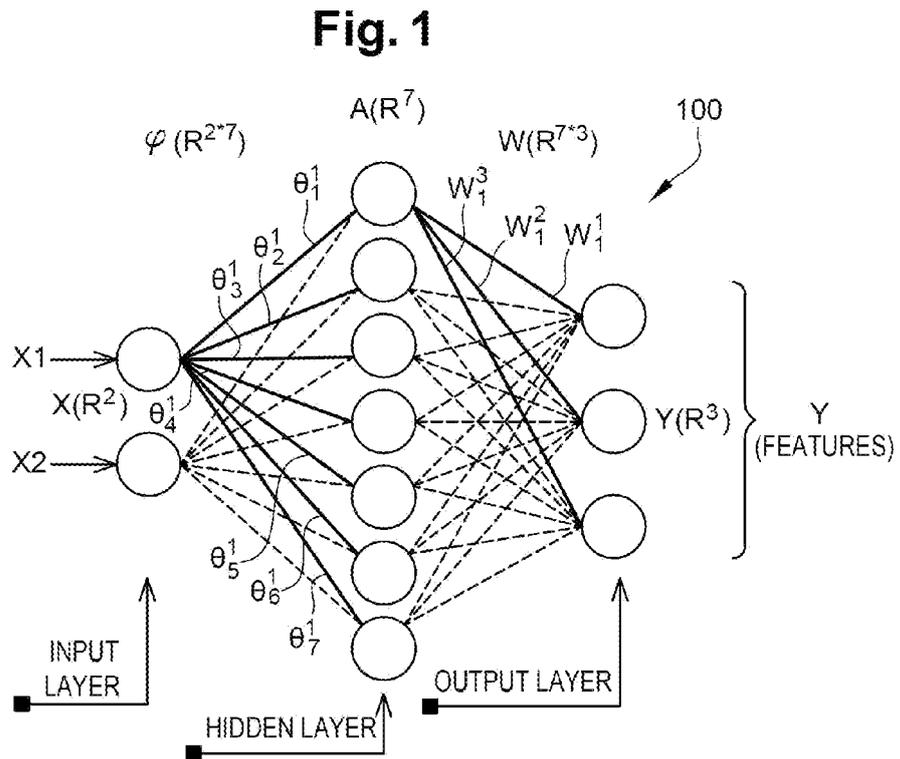
[Revendication 13] Système selon la revendication 11 ou 12, dans lequel l'autre réseau neuronal artificiel (406, 1210, 1260) est agencé pour mettre en œuvre au moins une fonction auto-associative pour reproduire des échantillons d'entrée au niveau d'une ou plusieurs de ses sorties, et dans lequel les données d'apprentissage comprennent en outre les nouveaux échantillons reproduits ($X_m', X_{(m+1)}', \dots$) générés par la fonction auto-associative du réseau neuronal artificiel entraîné (402, 1200, 1250) sur la base des pseudo-échantillons réinjectés.

[Revendication 14] Système selon l'une quelconque des revendications 10 à 13, comprenant

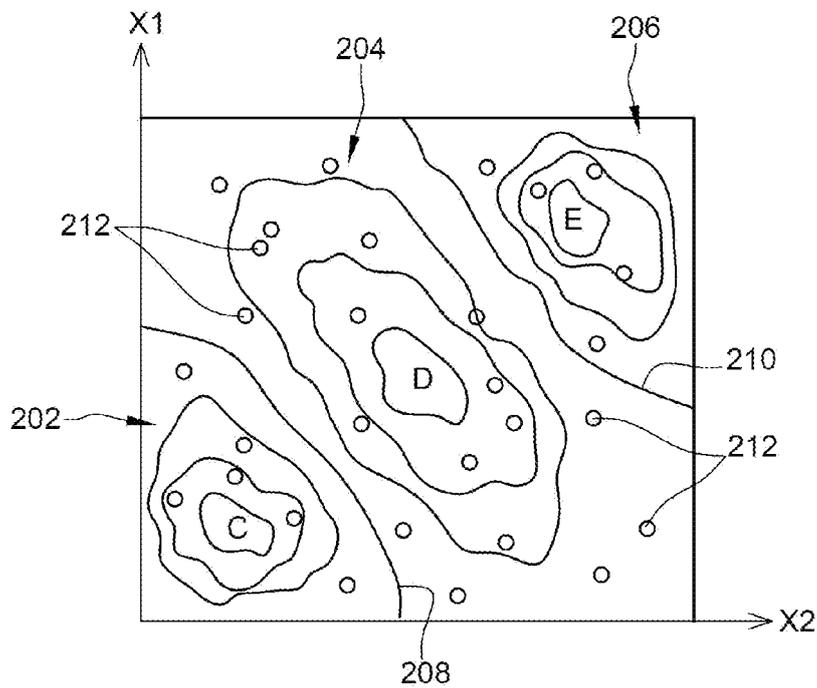
en outre un générateur de germes (408) agencé pour générer l'échantillon aléatoire sur la base d'une distribution normale ou sur la base d'une distribution uniforme ajustée.

[Revendication 15] Système selon l'une quelconque des revendications 10 à 14, dans lequel le générateur de données (404) est agencé pour générer le pseudo-échantillon en injectant du bruit dans l'échantillon reproduit présent au niveau desdites une ou plusieurs sorties du réseau neuronal artificiel entraîné.

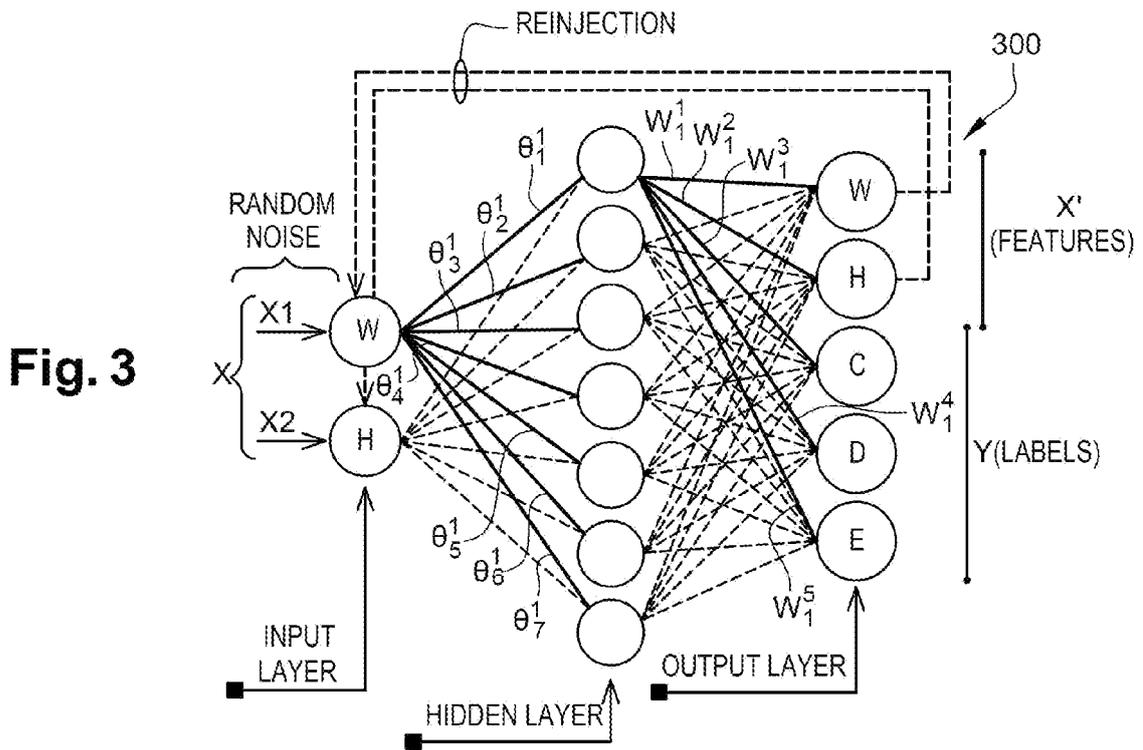
[Fig. 1]



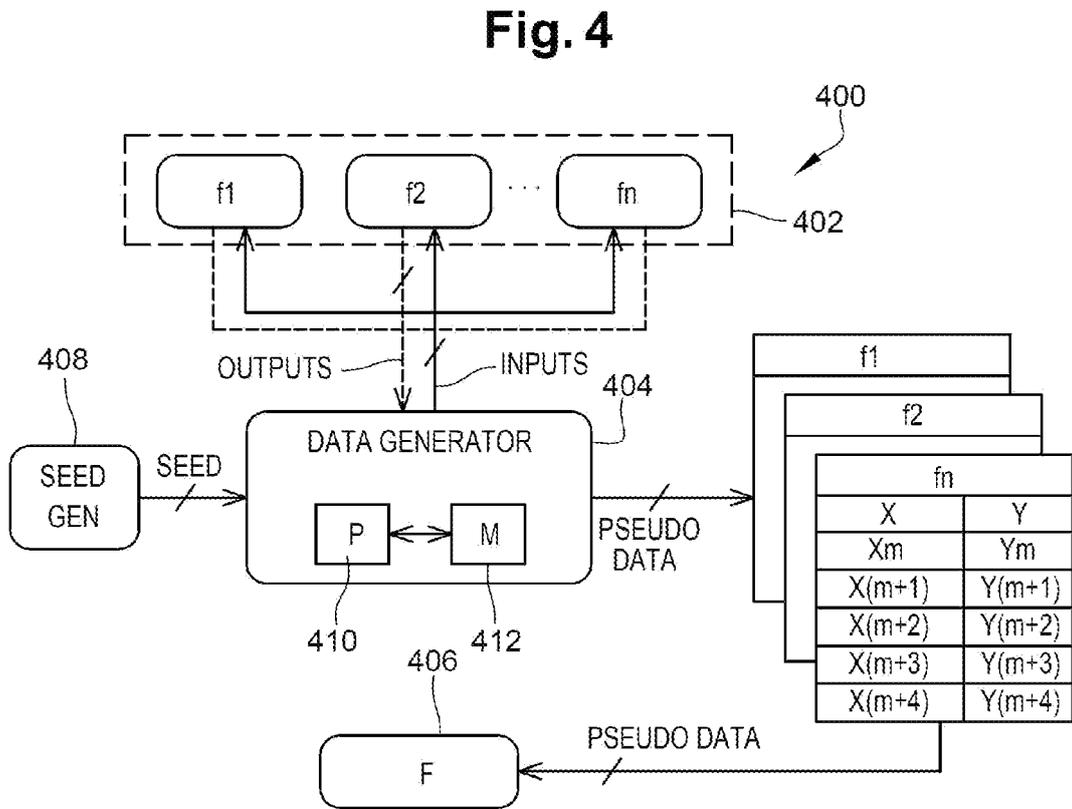
[Fig. 2]

**Fig. 2**

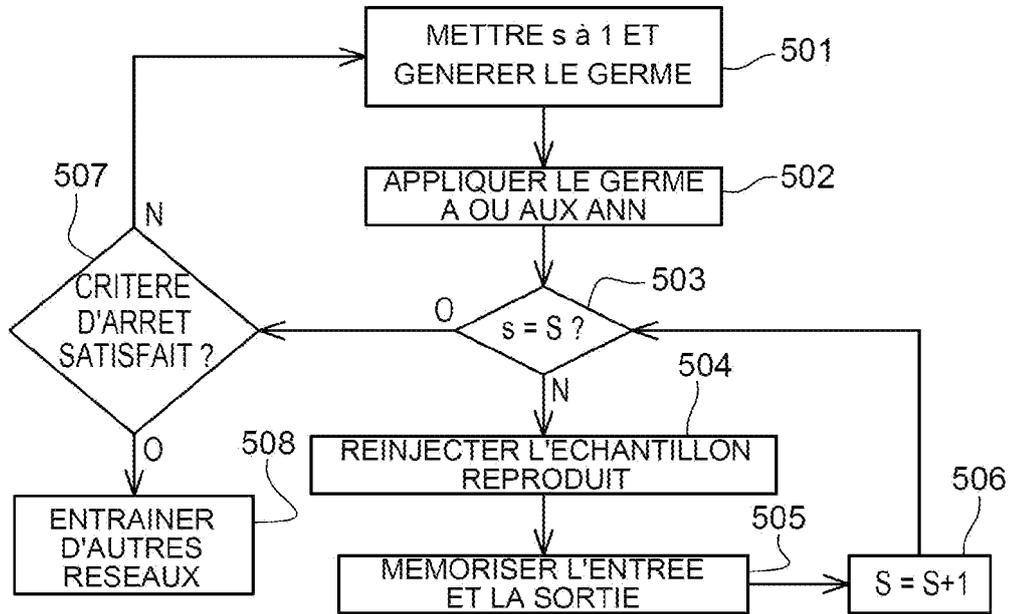
[Fig. 3]



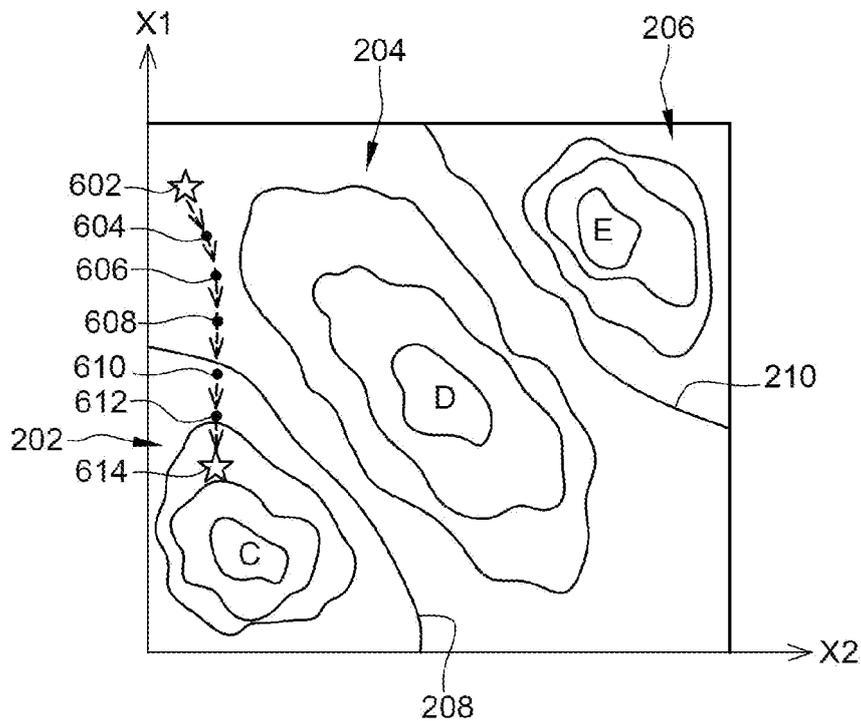
[Fig. 4]



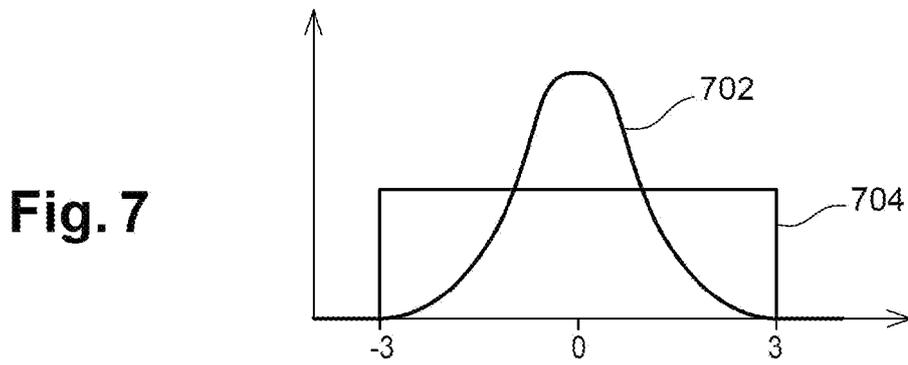
[Fig. 5]

Fig. 5

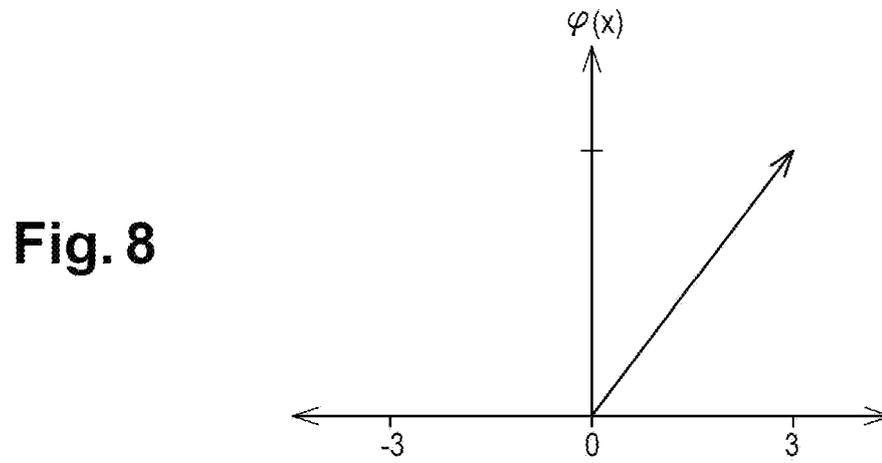
[Fig. 6]

**Fig. 6**

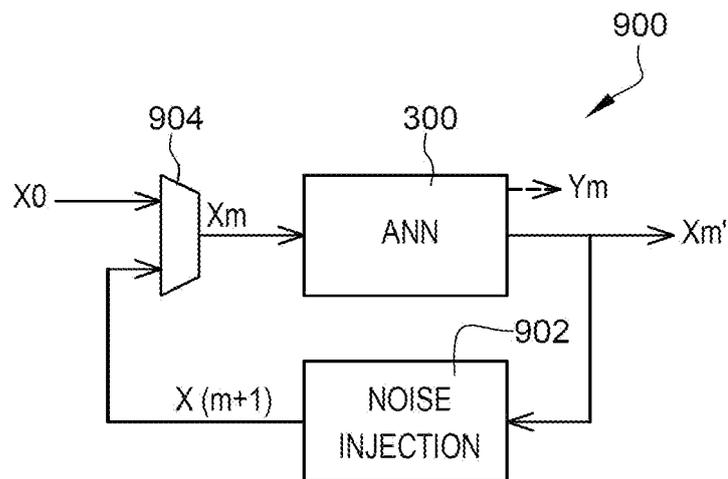
[Fig. 7]



[Fig. 8]

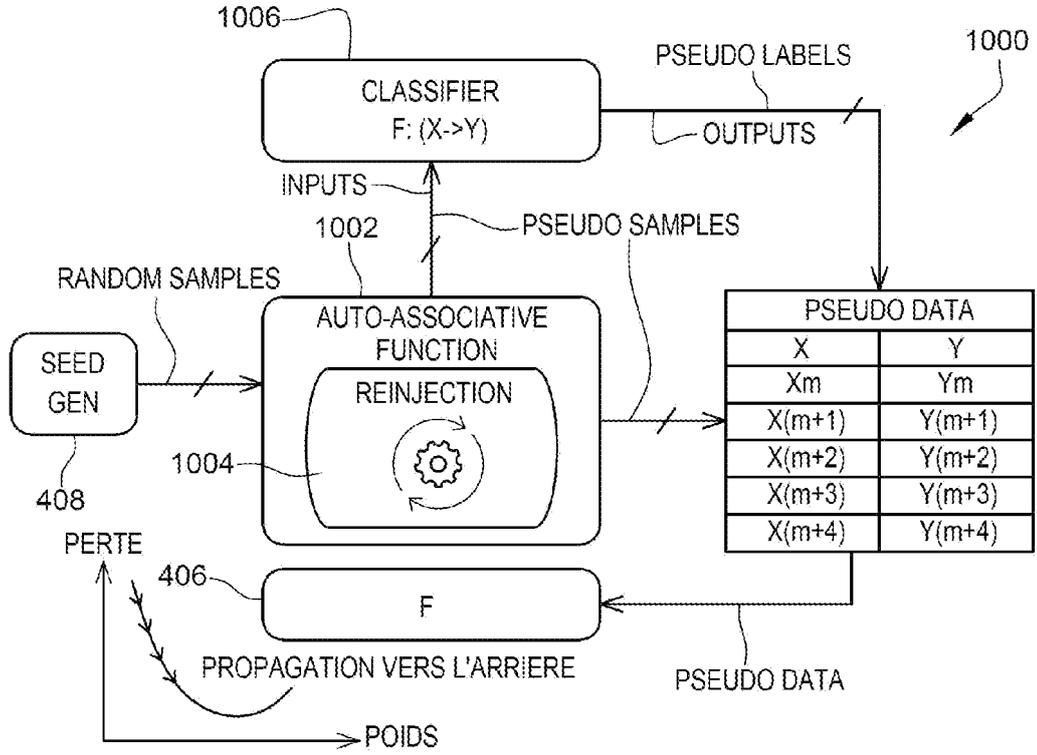


[Fig. 9]

Fig. 9

[Fig. 10]

Fig. 10



[Fig. 11]

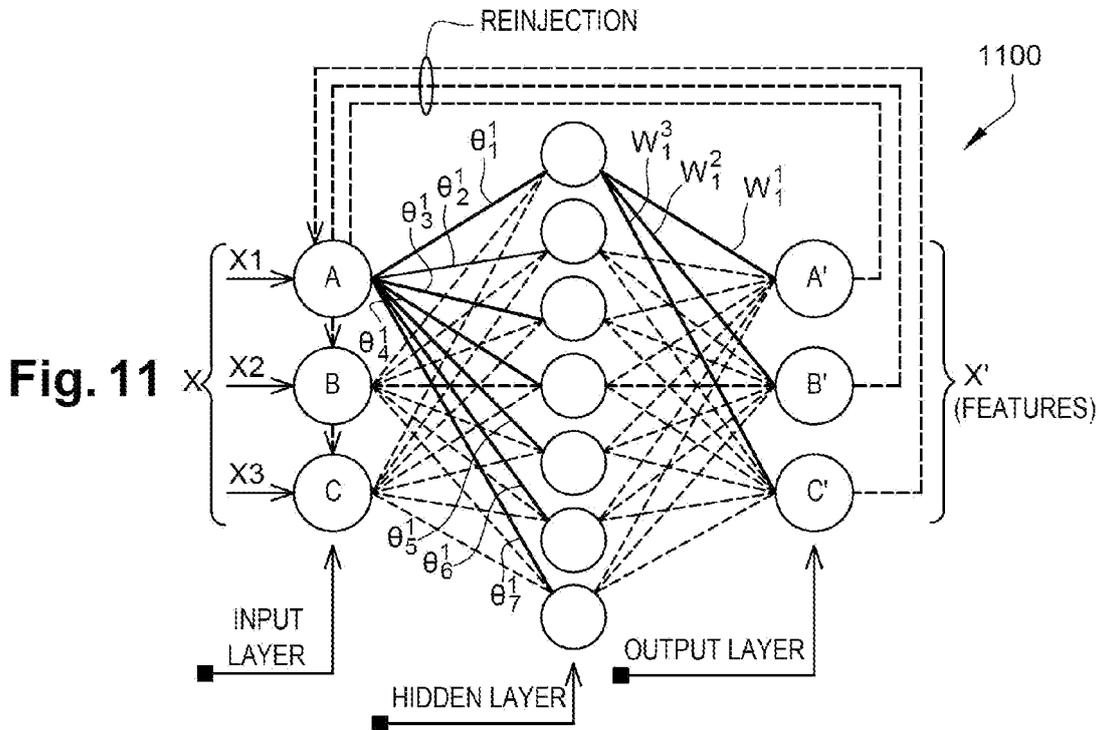
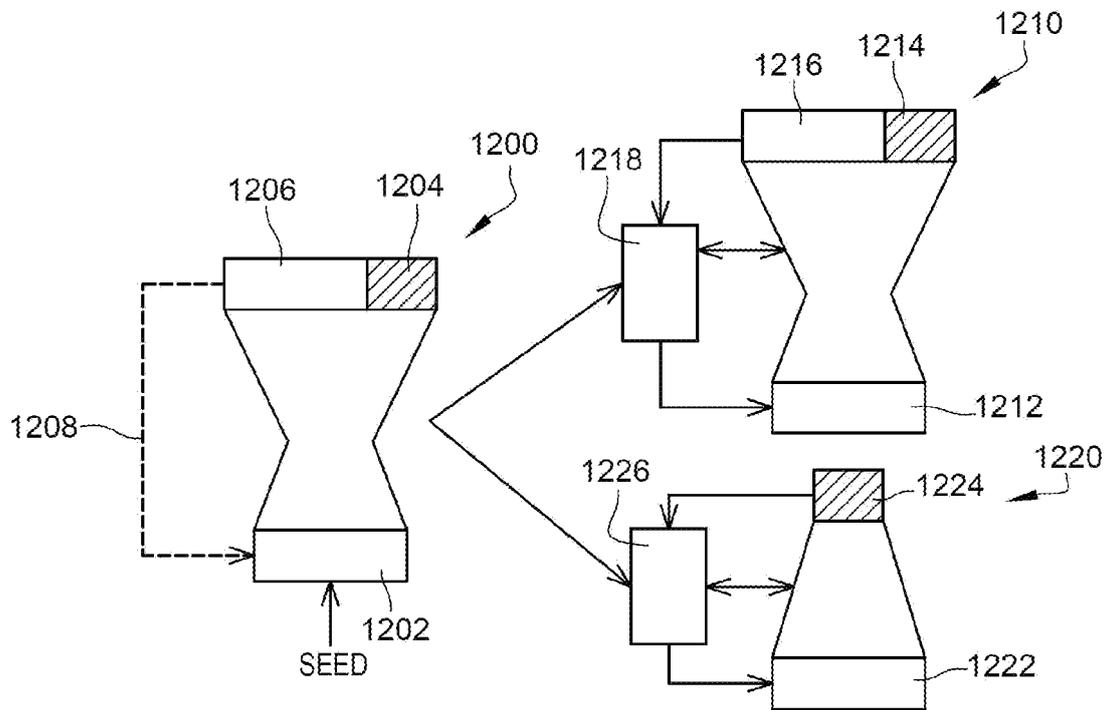
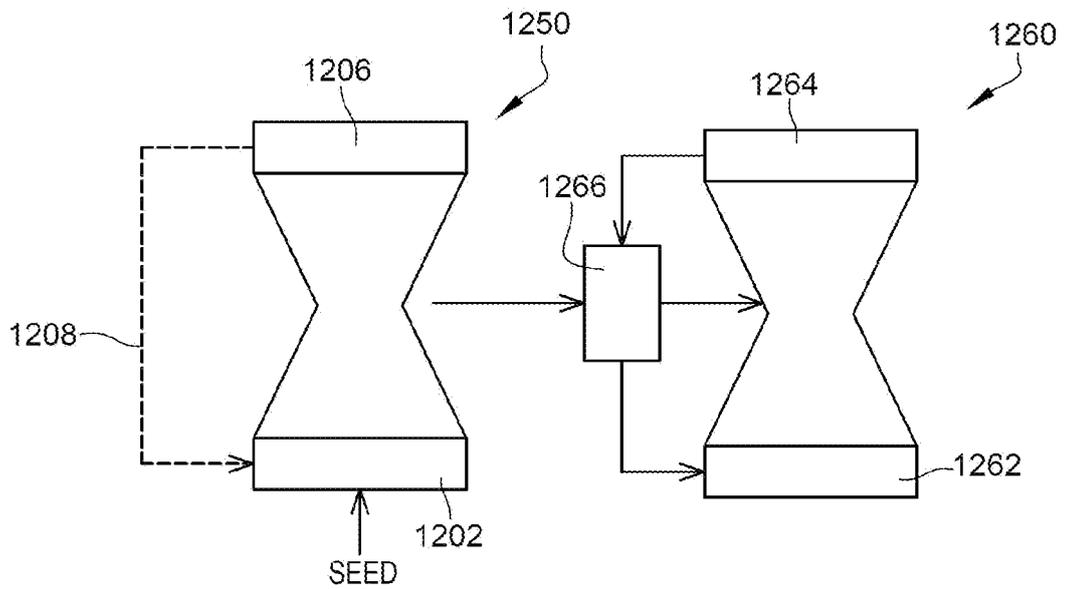


Fig. 11

[Fig. 12A]

Fig. 12A

[Fig. 12B]

Fig. 12B

**RAPPORT DE RECHERCHE
PRÉLIMINAIRE**

établi sur la base des dernières revendications
déposées avant le commencement de la recherche

N° d'enregistrement
national

FA 882289
FR 2003326

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	ANS B ET AL: "Neural networks with a self-refreshing memory: Knowledge transfer in sequential learning tasks without catastrophic forgetting", CONNECTION SCIENCE, vol. 12, no. 1, mars 2000 (2000-03), pages 1-19, XP055762088, GB ISSN: 0954-0091, DOI: 10.1080/095400900116177 * section 2.1; figure 1 *	1-15	G06N3/02 G06N3/08
A	ANS B ET AL: "Self-refreshing memory in artificial neural networks: learning temporal sequences without catastrophic forgetting", CONNECTION SCIENCE, vol. 16, no. 2, juin 2004 (2004-06), pages 71-99, XP055762094, GB ISSN: 0954-0091, DOI: 10.1080/09540090412331271199 * sections 2.3 et 2.4; figure 1 *	1-15	DOMAINES TECHNIQUES RECHERCHÉS (IPC)
A	Tianqi Chen ET AL: "Net2Net: accelerating learning via knowledge transfer", arXiv:1511.05641v4, 23 avril 2016 (2016-04-23), XP055349756, Extrait de l'Internet: URL:https://arxiv.org/abs/1511.05641v4 [extrait le 2017-02-27] * section 1 *	1-15	G06N
----- -/--			
Date d'achèvement de la recherche		Examineur	
21 décembre 2020		Domingo Vecchioni, M	
CATÉGORIE DES DOCUMENTS CITÉS		T : théorie ou principe à la base de l'invention	
X : particulièrement pertinent à lui seul		E : document de brevet bénéficiant d'une date antérieure	
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie		à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.	
A : arrière-plan technologique		D : cité dans la demande	
O : divulgation non-écrite		L : cité pour d'autres raisons	
P : document intercalaire		& : membre de la même famille, document correspondant	

**RAPPORT DE RECHERCHE
PRÉLIMINAIRE**

établi sur la base des dernières revendications
déposées avant le commencement de la recherche

N° d'enregistrement
national

FA 882289
FR 2003326

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
T	Solinas M et al: "Generalization of iterative sampling in autoencoders", HAL archives-ouvertes.fr, cea-02917445, 30 novembre 2020 (2020-11-30), XP055762074, Extrait de l'Internet: URL:https://hal-cea.archives-ouvertes.fr/c ea-02917445/document [extrait le 2020-12-21] * le document en entier * -----	1-15	
			DOMAINES TECHNIQUES RECHERCHÉS (IPC)
		Date d'achèvement de la recherche	Examineur
		21 décembre 2020	Domingo Vecchioni, M
CATÉGORIE DES DOCUMENTS CITÉS		T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant	
X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire			

1

EPO FORM 1503 12.99 (P04C14)