(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

COURBEVOIE

(11) **N° de publication :** ***3 110 268***

(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **20 04945**

(51) Int Cl[8] : ***G 06 N 3/08*** *(2019.12), G 06 F 21/55*

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 18.05.20.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 19.11.21 Bulletin 21/46.

(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) :** *IDEMIA IDENTITY & SECURITY FRANCE Société par actions simplifiée (SAS)* — FR.

(72) **Inventeur(s) :** CHABANNE Hervé et GUIGA Linda.

(73) **Titulaire(s) :** IDEMIA IDENTITY & SECURITY FRANCE Société par actions simplifiée (SAS).

(74) **Mandataire(s) :** REGIMBEAU.

(54) Procédés d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée, et d'apprentissage de paramètres d'un deuxième réseau de neurones.

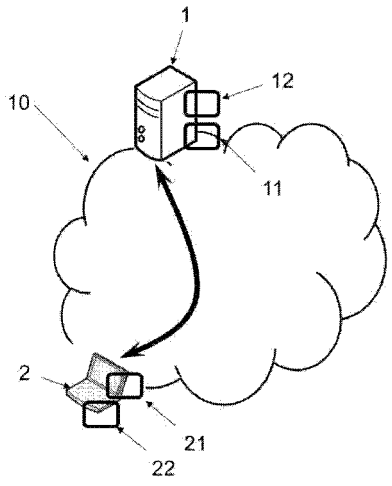
(57) La présente invention concerne un procédé d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée, le procédé étant caractérisé en ce qu'il comprend la mise en œuvre par des moyens de traitement de données (21) d'un terminal (2) d'étapes de :

(a) construction d'un deuxième réseau de neurones correspondant au premier réseau de neurones dans lequel est inséré au moins un réseau de neurones à convolution approximant la fonction identité ;

(b) utilisation du deuxième réseau de neurones sur ladite donnée d'entrée.

La présente invention concerne également un procédé d'apprentissage de paramètres du deuxième réseau de neurones

Figure pour l'abrégé : Fig. 1



FR 3 110 268 - A1

Description

Titre de l'invention : Procédés d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée, et d'apprentissage de paramètres d'un deuxième réseau de neurones

[0001] DOMAINE TECHNIQUE GENERAL

[0002] La présente invention concerne le domaine de l'intelligence artificielle, et en particulier un procédé d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée.

[0003] ETAT DE L'ART

[0004] Les réseaux de neurones (ou NN, pour neural network) sont massivement utilisés pour la classification de données.

[0005] Après une phase d'apprentissage automatique (généralement supervisé, c'est-à-dire sur une base de données de référence déjà classifiées), un réseau de neurones « apprend » et devient tout seul capable d'appliquer la même classification à des données inconnues. Plus précisément, la valeur de poids et paramètres du NN est progressivement modifiée jusqu'à ce que ce dernier soit capable de mettre en œuvre la tâche visée.

[0006] Des progrès significatifs ont été réalisés les dernières années, aussi bien sur les architectures des réseaux de neurones, que sur les techniques d'apprentissage (en particulier en apprentissage profond) ou encore sur les bases d'apprentissage (taille et qualité de ces dernières), et des tâches auparavant considérées comme impossibles sont aujourd'hui accomplies par des réseaux de neurones avec une excellente fiabilité.

[0007] Tout cela fait que les réseaux de neurones performants et leurs bases d'apprentissage ont aujourd'hui une forte valeur commerciale et sont traités comme des « secrets d'affaire » à protéger. De surcroit, beaucoup de bases de données contiennent des données potentiellement personnelles (par exemple des empreintes digitales) qui doivent rester confidentielles.

[0008] Malheureusement, ont été récemment développées des techniques de « reverse engineering » permettant à un attaquant d'extraire les paramètres et le modèle de n'importe quel réseau de neurones dès lors qu'on est capable de lui soumettre suffisamment de requêtes bien choisies, comme décrit dans le document *Cryptanalytic Extraction of Neural Network Models, Nicholas Carlini, Matthew Jagielski, Ilya Mironov https://arxiv.org/pdf/2003.04884v1.pdf.* Ainsi, même dans un fonctionnement « boite noire » dans lequel on n'aurait accès qu'aux entrées et aux sorties (par exemple via un client web) on pourrait retrouver l'intérieur du réseau.

[0009] L'idée est de constater que dans un réseau de neurones, on trouve une alternance de

couches linéaire et couches non-linéaires mettant en œuvre une fonction d'activation telle que ReLU. Cette non-linéarité entraîne des « points critiques » de saut du gradient, et on peut ainsi géométriquement définir pour chaque neurone un hyperplan de l'espace d'entrée du réseau tel que la sortie est à un point critique. Les hyperplans de la deuxième couche sont « pliés » par les hyperplans de la première couche et, ainsi de suite.

[0010] L'attaquant peut par exploration retrouver les intersections des hyperplans et progressivement tout le réseau de neurone.

[0011] Un défi supplémentaire rencontré par les réseaux de neurones est l'existence des « perturbations antagonistes », c'est-à-dire des changements imperceptibles qui lorsque appliqués sur une entrée du réseau de neurone changent significativement la sortie. On voit par exemple dans le document *A Simple Explanation for the Existence of Adversarial Examples with Small Hamming Distance par Adi Shamir, Itay Safran, Eyal Ronen, et Orr Dunkelman, https://arxiv.org/pdf/1901.10861v1.pdf* comment une perturbation antagoniste appliquée sur une image de chat peut conduire à celle-ci classifiée à tort comme une image de guacamole.

[0012] Plus précisément, dès lors qu'un attaquant a réussi à identifier le découpage en hyperplans évoqué avant, il peut déterminer un vecteur permettant, à partir d'un point de l'espace d'entrée, de franchir un hyperplan et donc de modifier la sortie.

[0013] On comprend donc qu'il est essentiel de parvenir à sécuriser les réseaux de neurones.

[0014] Une première piste est d'augmenter la taille, le nombre de couches et le nombre de paramètres du réseau de sorte à compliquer la tâche de l'attaquant. Si cela fonctionne, d'une part cela ne fait que ralentir l'attaquant et surtout cela dégrade les performances car le réseau de neurone est alors inutilement lourd et difficile à apprendre.

[0015] Une deuxième piste est de limiter le nombre d'entrées pouvant être soumises au réseau de neurones, ou du moins de détecter les séquences suspectes d'entrées. Cela n'est toutefois pas toujours applicables, puisque l'attaquant peut légalement avoir accès au réseau de neurones en ayant par exemple payé un accès sans restriction.

[0016] Ainsi, on pourrait encore améliorer la situation.

PRESENTATION DE L'INVENTION

[0017] Selon un premier aspect, la présente invention concerne un procédé d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée, le procédé étant caractérisé en ce qu'il comprend la mise en œuvre par des moyens de traitement de données d'un terminal d'étapes de :

[0018] (a) construction d'un deuxième réseau de neurones correspondant au premier réseau de neurones dans lequel est inséré au moins un réseau de neurones à convolution approximant la fonction identité ;

[0019] (b) utilisation du deuxième réseau de neurones sur ladite donnée d'entrée.

[0020] Selon d'autres caractéristiques avantageuses et non limitatives :

[0021] Ledit réseau de neurones à convolution est inséré en entrée d'une couche cible du premier réseau de neurones.

[0022] Ledit réseau de neurones à convolution présente une taille de sortie inférieure à une taille d'entrée de ladite couche cible de sorte à approximer seulement certains canaux d'entrée de cette couche cible.

[0023] L'étape (a) comprend la sélection de ladite couche cible du premier réseau de neurones parmi les couches linéaires dudit premier réseau de neurones et/ou la sélection des canaux d'entrée de ladite couche cible à approximer parmi tous les canaux d'entrée de la couche cible.

[0024] L'au moins un réseau de neurones à convolution approximant la fonction identité présente une taille de sortie égale au produit de deux entiers.

[0025] Le procédé comprend une étape (a0) préalable d'obtention des paramètres du premier réseau de neurones et de l'au moins un réseau de neurones à convolution approximant la fonction identité.

[0026] L'étape (a0) comprend l'obtention des paramètres d'un ensemble de réseau de neurones à convolution approximant la fonction identité.

[0027] L'étape (a) comprend la sélection dans ledit ensemble d'au moins un réseau de neurones à convolution approximant la fonction identité à insérer.

[0028] L'étape (a) comprend, pour chaque réseau de neurones à convolution approximant la fonction identité sélectionné, ladite sélection de ladite couche cible du premier réseau de neurones parmi les couches linéaires dudit premier réseau de neurones et/ou la sélection des canaux d'entrée de ladite couche cible à approximer parmi tous les canaux d'entrée de la couche cible.

[0029] L'étape (a) comprend en outre la sélection préalable d'un nombre de réseaux de neurones à convolution approximant la fonction identité dudit ensemble à sélectionner.

[0030] L'étape (a0) est une étape, mise en œuvre par des moyens de traitement de données d'un serveur, d'apprentissage des paramètres du premier réseau de neurones et de l'au moins un réseau de neurones à convolution approximant la fonction identité à partir d'au moins une base de données d'apprentissage.

[0031] Le premier réseau de neurones et le ou les réseaux de neurones à convolution approximant la fonction identité comprennent une alternance de couches linéaires et de couches non-linéaires à fonction d'activation.

[0032] Ladite fonction d'activation est la fonction ReLU.

[0033] Ladite couche cible est une couche linéaire du premier réseau de neurones.

[0034] l'au moins un réseau de neurones à convolution approximant la fonction identité comprend deux ou trois couches linéaires.

[0035] Les couches linéaires du réseau de neurones à convolution sont des couches de convolution à filtre par exemple de taille 5x5.

[0036] Selon un deuxième aspect est proposé un procédé d'apprentissage de paramètres d'un deuxième réseau de neurones, le procédé étant caractérisé en ce qu'il comprend la mise en œuvre par des moyens de traitement de données d'un serveur d'étapes de :

[0037] (a) construction du deuxième réseau de neurones correspondant à un premier réseau de neurones dans lequel est inséré au moins un réseau de neurones à convolution approximant la fonction identité ;

[0038] (a1) apprentissage des paramètres du deuxième réseau de neurones à partir d'une base de données d'apprentissage

[0039] Selon un troisième aspect est proposé un procédé d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée, le procédé comprenant l'apprentissage de paramètres d'un deuxième réseau de neurones conformément au précédé selon le deuxième aspect ; et la mise en œuvre par des moyens de traitement de données d'un terminal d'une étape (b) d'utilisation du deuxième réseau de neurones sur ladite donnée d'entrée.

[0040] Selon un quatrième et un cinquième aspect, l'invention concerne un produit programme d'ordinateur comprenant des instructions de code pour l'exécution d'un procédé selon le premier ou le troisième aspect d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée, ou selon le deuxième aspect d'apprentissage de paramètres d'un deuxième réseau de neurones ; et un moyen de stockage lisible par un équipement informatique sur lequel un produit programme d'ordinateur comprend des instructions de code pour l'exécution d'un procédé selon le premier ou le troisième aspect d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée, ou selon le deuxième aspect d'apprentissage de paramètres d'un deuxième réseau de neurones .

PRESENTATION DES FIGURES

[0041] D'autres caractéristiques et avantages de la présente invention apparaîtront à la lecture de la description qui va suivre d'un mode de réalisation préférentiel. Cette description sera donnée en référence aux dessins annexés dans lesquels :

[0042]
- [Fig. 1]la figure 1 est un schéma d'une architecture pour la mise en œuvre des procédés selon l'invention ;
- [Fig. 2a]la figure 2a représente schématiquement les étapes d'un premier mode de réalisation d'un procédé d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée selon l'invention ;
- [Fig. 2b]la figure 2b représente schématiquement les étapes d'un deuxième mode de réalisation d'un procédé d'utilisation sécurisée d'un premier réseau

de neurones sur une donnée d'entrée selon l'invention ;

– [Fig. 3]la figure 3 représente schématiquement un exemple d'architecture d'un deuxième réseau de neurones rencontré dans la mise en œuvre des procédés selon l'invention.

DESCRIPTION DETAILLEE

[0043] *Architecture*

[0044] Selon deux aspects complémentaires de l'invention, sont proposés :

[0045] – un procédé d'utilisation sécurisée d'un premier réseau de neurones (1e NN)

– un procédé d'apprentissage de paramètres d'un deuxième réseau de neurones (2e NN).

[0046] Ces deux types de procédés sont mis en œuvre au sein d'une architecture telle que représentée par la [fig.1], grâce à au moins un serveur 1 et un terminal 2. Le serveur 1 est l'équipement d'apprentissage (mettant en œuvre le deuxième procédé) et le terminal 2 est un équipement d'utilisation (mettant en œuvre le premier procédé). Ledit procédé d'utilisation est mis en œuvre sur une donnée d'entrée, et est par exemple une classification de la donnée d'entrée parmi plusieurs classes si c'est un NN de classification (mais cette tâche n'est pas nécessairement une classification même si c'est la plus classique).

[0047] On ne sera limité à aucun type de NN en particulier, même si typiquement il s'agit d'une alternance de couches linéaires et de couches non-linéaire à fonction d'activation ReLU (Rectified Linear Unit, i.e. Unité de Rectification Linéaire) qui est égale à $\sigma(x) = max(0, x)$. On comprend donc que chaque hyperplan correspond à l'ensemble des points de l'espace d'entrée tels qu'une sortie d'une couche linéaire est égale à zéro. On notera « ReLU NN » un tel réseau de neurones.

[0048] Dans tous les cas, chaque équipement 1, 2 est typiquement un équipement informatique distant relié à un réseau étendu 10 tel que le réseau internet pour l'échange des données. Chacun comprend des moyens de traitement de données 11, 21 de type processeur, et des moyens de stockage de données 12, 22 telle qu'une mémoire informatique, par exemple un disque dur.

[0049] Le serveur 1 stocke une base de données d'apprentissage, i.e. un ensemble de données pour lesquelles on connait déjà la sortie associée, par exemple déjà classifiées (par opposition aux données dites d'entrée que l'on cherche justement à traiter). Il peut s'agir d'une base d'apprentissage à haute valeur commerciale qu'on cherche à garder secrète.

[0050] On comprendra qu'il reste possible que les équipements 1 et 2 puissent être le même équipement, voire la base d'apprentissage être une base publique.

[0051] A noter que le présent procédé n'est pas limité à un type de NN et donc pas à une

nature particulière de données, les données d'entrée ou d'apprentissage peuvent être représentatives d'images, de sons, etc. Le 1e NN peut tout à fait être un CNN, même si l'on décrira plus loin un CNN spécialisé qu'on va utiliser dans le cadre du présent procédé.

[0052] Dans un mode de réalisation préféré il s'agit de données biométriques, les données d'entrée ou d'apprentissage étant typiquement représentatives d'images voire directement des images de traits biométriques (visages, empreintes digitales, iris, etc.), ou directement des données prétraitées issues des traits biométriques (par exemple la position de minuties dans le cas d'empreintes digitales).

[0053] *Principe*

[0054] La présente invention propose de complexifier la tâche des attaquants sans complexifier le NN grâce à des hyperplans artificiels. En d'autres termes on sécurise le NN en le rendant nettement plus robuste sans pour autant l'alourdir et dégrader ses performances.

[0055] Par commodité on nommera « premier réseau de neurones » le NN d'origine à protéger et « deuxième réseau de neurones » le NN modifié et ainsi sécurisé. A noter que, comme l'on verra plus tard, la sécurisation du 1e NN peut être faite a posteriori (une fois qu'il a été appris), ou dès l'origine (i.e. on apprend directement une version sécurisée du NN).

[0056] Plus en détails, la sécurisation du premier réseau en deuxième réseau consiste à intégrer dans son architecture au moins un réseau de neurones à convolution (CNN) approximant la fonction identité (on fera référence à ce dernier en tant que « CNN Identité » par commodité).

[0057] Ce CNN « parasite » ne modifie pas le fonctionnement du NN car ses sorties sont sensiblement égales à ses entrées. Par contre il brise la structure en hyperplans d'origine.

[0058] L'idée d'approximer la fonction identité est très originale pour un CNN, car c'est une tâche contre-nature qu'il a du mal à accomplir. Pour reformuler, on cherche toujours à ce qu'un CNN accomplisse des traitements sémantiquement complexes (comme par exemple de la segmentation d'image), et jamais une tâche aussi triviale que de reproduire sa propre entrée.

[0059] De surcroît, comme l'on verra plus loin, on peut insérer plusieurs CNN Identité dans le 1e NN, à divers endroits, impliquant certains canaux, le tout choisi le cas échéant de manière dynamique et aléatoire, ce qui ne laisse plus aucune chance à un attaquant (il faudrait un nombre inimaginable de requêtes envoyées au 2e NN pour arriver à retrouver le 1e NN d'origine sous les hyperplans artificiels).

[0060] *Procédé*

[0061] Selon un premier aspect, est proposé en référence à la [fig.2a] un premier mode de

réalisation du procédé d'utilisation sécurisée du 1e NN sur une donnée d'entrée, mis en œuvre par les moyens de traitement de données 21 du terminal 2.

[0062] Le procédé commence avantageusement par une étape « préparatoire » (a0) d'obtention des paramètres du 1e NN et d'au moins un CNN Identité, préférentiellement une pluralité de CNN Identité, notamment de diverses architectures, de diverses tailles d'entrée et sortie, appris sur des bases différentes, etc., de sorte à définir un ensemble si possible varié de CNN Identité, on verra cela plus en détail plus loin.

[0063] Cette étape (a0) peut être une étape d'apprentissage de chacun des réseaux sur une base d'apprentissage dédiée, en particulier du 1e NN, préférentiellement mise en œuvre par les moyens de traitement de données 11 du serveur 1 à cet effet, mais on comprendra que les réseaux (en particuliers les CNN Identité) pourraient être pré-existants et pris en l'état. En tout état de cause, le ou les CNN Identité peuvent être appris en particulier sur n'importe quelle base d'images publiques, voire même sur des données aléatoires (pas besoin qu'elles soient annotées car on suppose que l'entrée est aussi la sortie attendue). On verra plus loin un mode de réalisation alternatif dans lequel il n'y a pas cette étape (a0).

[0064] Dans une étape principale (a), on construit ledit 2e NN correspondant au 1e NN dans lequel est inséré au moins un réseau de neurones à convolution approximant la fonction identité, en particulier un ou plusieurs CNN Identité sélectionné(s). En d'autres termes, l'étape (a) est une étape d'insertion du ou des CNN Identité dans le 1e NN. S'il y a plusieurs CNN Identité sélectionnés, ils peuvent être insérés chacun à la suite.

[0065] A ce titre, l'étape (a) comprend avantageusement la sélection d'un ou plusieurs CNN Identité dans ledit ensemble de CNN Identité, par exemple aléatoirement. D'autres « paramètres d'insertion » peuvent être sélectionnés, notamment une position dans le 1e NN (couche cible) et/ou des canaux d'une couche cible du 1e NN, voir plus loin. En tout état de cause il reste possible que l'ensemble de CNN Identité ne contienne qu'un seul CNN de sorte qu'il n'y a pas besoin de sélection, voire même que le CNN Identité soit appris à la volée.

[0066] Par insertion, on entend l'ajout des couches du CNN Identité en amont de la couche « cible » du 1e NN de sorte que l'entrée de cette couche soit au moins en partie la sortie du CNN Identité. En d'autres termes, le CNN Identité « intercepte » tout ou partie de l'entrée de la couche cible pour la remplacer par sa sortie.

[0067] On comprend que comme le CNN Identité approxime la fonction identité, ses sorties sont sensiblement identiques à ses entrées de sorte que les données que reçoit la couche cible sont sensiblement identiques à celles interceptées.

[0068] La couche cible est préférentiellement une couche linéaire (et pas une couche non-linéaire à fonction d'activation par exemple), de sorte le CNN Identité est inséré en

entrée d'une couche linéaire du 1e NN.

[0069] Avantageusement, le CNN Identité présente une taille de sortie inférieure à une taille d'entrée de ladite couche linéaire de sorte à approximer seulement certains canaux d'entrée de cette couche linéaire (i.e. pas tous). Par taille d'entrée/sortie, on entend le nombre de canaux d'entrée/sortie.

[0070] C'est ce que l'on voit dans l'exemple de la [fig.3], qui représente un 1e NN à trois couches linéaires (dont une couche cachée centrale), dans lequel un CNN Identité est disposé en entrée de la deuxième couche.

[0071] On voit que la première couche présente huit canaux d'entrée (taille 8), alors que le CNN identité présente seulement quatre canaux d'entrée/sortie (par définition un CNN approximant la fonction identité à les mêmes dimensions d'entrée et de sortie). Ainsi, sur les huit canaux d'entrée de la première couche linéaire, seuls quatre sont approximés, et les quatre autres sont tel quels. Le fait de n'affecter que certains des canaux (i.e. pas tous les neurones) a trois avantages : le CNN peut être plus petit et donc impliquer moins de calculs lors de l'exécution, le fait d'affecter seulement partiellement une couche génère des perturbations surprenante pour un attaquant, et on peut même disposer plusieurs CNN sous une couche et donc augmenter encore davantage les perturbations des hyperplans pour un attaquant.

[0072] L'étape (a) peut également comprendre comme expliqué la sélection de la couche cible et/ou des canaux d'entrée de la couche linéaire cible affectés par le CNN Identité (le cas échéant préalablement sélectionné). Par exemple, dans la figure 3 il s'agit des canaux 1, 2, 3 et 4, mais on aurait pu prendre n'importe quel ensemble de quatre canaux parmi les huit, par exemple les canaux 1, 3, 5 et 7.

[0073] Cette sélection peut à nouveau être faite au hasard et dynamiquement, i.e. on tire des nouveaux canaux à chaque nouvelle requête d'utilisation du 1e NN. En pratique la sélection peut être faite selon le protocole suivant (chaque étape étant optionnelle – chaque choix peut être aléatoire ou prédéterminé) :

[0074]
1. on choisit un nombre de CNN Identité à insérer ;
2. on tire autant de CNN Identité que ce nombre dans l'ensemble des CNN Identité (avec ou sans remise) ;
3. pour chaque CNN Identité tiré on choisit une couche cible à affecter (i.e. en entrée de laquelle le CNN sera inséré) parmi les couches linéaires du 1e NN ;
4. pour chaque CNN Identité tiré on choisit autant de canaux d'entrée de la couche cible associée que le nombre de canaux d'entrée/sortie de ce CNN Identité.

[0075] En ce qui concerne le point 3., il est à noter que deux CNN Identité peuvent être choisis comme affectant la même couche cible : soit les canaux concernés sont distincts et il n'y a aucun problème, soit au moins un canal se recoupe et dans ce cas-là

on peut soit décider que ce n'est pas souhaitable (et on recommencer le tirage), soit accepter qu'un CNN identité soit en amont de l'autre : un canal peut ainsi être approximé deux fois de suite avant d'arriver en entrée de la couche cible.

[0076] En ce qui concerne le point 4., il est à noter que les CNN Identité sont typiquement des réseaux travaillant sur des images, i.e. des objets bidimensionnels (des « rectangles »), et donc présentant un nombre de canaux d'entrée/sortie égale au produit de deux entiers, i.e. de la forme a*b, où a et b sont des entiers chacun supérieur ou égal à deux, et même préférentiellement des « carrés » de dimension a^2. On peut tout à fait imaginer utiliser des CNN travaillant sur des objets tridimensionnels et donc présentant un nombre de canaux d'entrée/sortie égale au produit de trois entiers, i.e. de la forme a*b*c, etc. Dans l'exemple de la figure 3, on a un CNN Identité travaillant sur des images 2x2, et donc à quatre canaux d'entrée/sortie.

[0077] A noter enfin que les actions de sélection et de construction peuvent être partiellement imbriquées (et donc mise en œuvre en même temps) : s'il y a plusieurs CNN identité à insérer, on peut déterminer les paramètres d'insertion pour le premier, l'insérer, déterminer les paramètres d'insertion du deuxième, l'insérer, etc. De plus, comme expliqué la sélection de la couche cible et/ou des canaux peut être faite à la volée dans l'étape (a).

[0078] A l'issue de l'étape (a) on suppose que le 2^e NN est construit. Alors, dans une étape (b) ce 2^e NN peut être utilisé sur ladite donnée d'entrée, i.e. on applique le 2^e NN à la donnée d'entrée et on obtient une donnée de sortie qui peut être fournie à l'utilisateur du terminal 2 sans aucun risque de pouvoir remonter au 1^e NN.

[0079] *CNN Identité*

[0080] Des CNN de faible taille uniquement constitués d'une alternance de couches de convolution (couches linéaires) et couches non-linéaires à fonction d'activation telle que ReLU donnent de très bons résultats à la fois en qualité d'approximation de l'identité et en complexification des hyperplans sans alourdir le 1^e NN pour autant.

[0081] Par exemple le CNN Identité peut comprendre seulement deux ou trois couches de convolution (même si on ne sera limité à aucune architecture).

[0082] Selon un mode de réalisation particulièrement préféré on peut prendre un CNN Identité d'entrée/sortie carrée de taille jusqu'à 16x16 avec deux ou trois couches de convolution à filtres de taille 5x5.

[0083] *Tests*

[0084] Des tests ont été faits en prenant comme 1^e NN un ReLU NN à trois couches cachées de type réseau entièrement connecté (FCN, Fully-connected network), les couches cachées ayant respectivement 512, 512 et 32 canaux d'entrée, ce FCN étant utilisé pour la reconnaissance de chiffres manuscrits (classification d'images d'entrée de taille quelconque). Ce 1^e NN peut être entraîné pour cette tâche sur la base d'apprentissage

MNIST (Mixed National Institute of Standards and Technology) et montre alors un taux de classification correct de 97.9%

[0085] Le CNN Identité évoqué avant de taille d'entrée 16x16 (256 canaux) peut être entraîné sur 10000 images aléatoires, et on obtient une erreur absolue moyenne entre l'entrée et la sortie de 0.0913%.

[0086] L'insertion de ce CNN Identité sur 256 des 512 canaux d'entrée de la première ou deuxième couche cachée du 1e NN ne montre aucune baisse du taux de classification correct pour le 2e NN.

[0087] *Apprentissage a posteriori*

[0088] Alternativement à l'obtention préalable des paramètres du 1e NN et du ou des CNN Identité, on peut directement commencer par l'étape (a) de construction du 2e NN à partir de modèles des 1e NN et CNN Identité, le cas échéant en mettant en œuvre les sélections évoquées avant pour déterminer l'architecture du 2e NN, et ensuite seulement on apprend les paramètres du 2e NN sur la base d'apprentissage du 1e NN (par exemple la base NIST évoquée ci-avant). Il s'agit du mode de réalisation illustré par la [fig.2b], on comprend que la construction et l'apprentissage sont cette fois mis en œuvre du côté du serveur 1.

[0089] Cela évite d'avoir à apprendre séparément les paramètres du CNN Identité puisque ses paramètres propres sont automatiquement appris en même temps que ceux du reste du NN.

[0090] Les résultats sont équivalents, le seul désagrément est qu'il n'est pas possible de « reconstruire » dynamiquement le 2e NN à chaque requête car ce serait trop long de remettre en œuvre un apprentissage à chaque fois.

[0091] Ainsi, selon un deuxième aspect, l'invention concerne un procédé d'apprentissage d'un deuxième réseau de neurones, mis en œuvre par les moyens de traitement de données 11 du serveur 1, comprenant à nouveau l'étape (a) de construction du deuxième réseau de neurones correspondant à un premier réseau de neurones dans lequel est inséré au moins un réseau de neurones à convolution approximant la fonction identité ; puis une étape (a1) d'apprentissage des paramètres du deuxième réseau de neurones à partir d'une base de données publique d'apprentissage.

[0092] Dans un troisième aspect de l'invention, on peut utiliser ce procédé d'apprentissage comme partie d'un procédé d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée (comme le procédé selon le premier aspect), en rajoutant la même étape (b) d'utilisation du 2e NN sur ladite donnée d'entrée (mise en œuvre cette fois par les moyens de traitement de données 21 du terminal 1), i.e. on applique le 2e NN à la donnée d'entrée et on obtient une donnée de sortie qui peut être fournie à l'utilisateur du terminal 2 sans aucun risque de pouvoir remonter au 1e NN.

[0093] *Produit programme d'ordinateur*

[0094] Selon un quatrième et un cinquième aspects, l'invention concerne un produit programme d'ordinateur comprenant des instructions de code pour l'exécution (en particulier sur les moyens de traitement de données 11, 21 du serveur 1 ou du terminal 2) d'un procédé selon le premier ou le troisième aspect de l'invention d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée ou un procédé selon le deuxième aspect de l'invention d'apprentissage de paramètres d'un deuxième réseau de neurones, ainsi que des moyens de stockage lisibles par un équipement informatique (une mémoire 12, 22 du serveur 1 ou du terminal 2) sur lequel on trouve ce produit programme d'ordinateur.

Revendications

[Revendication 1] Procédé d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée, le procédé étant caractérisé en ce qu'il comprend la mise en œuvre par des moyens de traitement de données (21) d'un terminal (2) d'étapes de :
(a) construction d'un deuxième réseau de neurones correspondant au premier réseau de neurones dans lequel est inséré au moins un réseau de neurones à convolution approximant la fonction identité ;
(b) utilisation du deuxième réseau de neurones sur ladite donnée d'entrée.

[Revendication 2] Procédé selon la revendication 1, dans lequel ledit réseau de neurones à convolution est inséré en entrée d'une couche cible du premier réseau de neurones et présente une taille de sortie inférieure à une taille d'entrée de ladite couche cible de sorte à approximer seulement certains canaux d'entrée de cette couche cible.

[Revendication 3] Procédé selon la revendication 2, dans lequel l'étape (a) comprend la sélection de ladite couche cible du premier réseau de neurones parmi les couches linéaires dudit premier réseau de neurones et/ou la sélection des canaux d'entrée de ladite couche cible à approximer parmi tous les canaux d'entrée de la couche cible.

[Revendication 4] Procédé selon l'une des revendications 1 à 3, dans lequel l'au moins un réseau de neurones à convolution approximant la fonction identité présente une taille de sortie égale au produit de deux entiers.

[Revendication 5] Procédé selon l'une des revendications 1 à 4, comprenant une étape (a0) préalable d'obtention des paramètres du premier réseau de neurones et de l'au moins un réseau de neurones à convolution approximant la fonction identité.

[Revendication 6] Procédé selon la revendication 5, dans lequel l'étape (a0) comprend l'obtention des paramètres d'un ensemble de réseau de neurones à convolution approximant la fonction identité, l'étape (a) comprenant la sélection dans ledit ensemble d'au moins un réseau de neurones à convolution approximant la fonction identité à insérer.

[Revendication 7] Procédé selon les revendications 3 et 6 en combinaison, dans lequel l'étape (a) comprend, pour chaque réseau de neurones à convolution approximant la fonction identité sélectionné, ladite sélection de ladite couche cible du premier réseau de neurones parmi les couches linéaires dudit premier réseau de neurones et/ou la sélection des canaux d'entrée

de ladite couche cible à approximer parmi tous les canaux d'entrée de la couche cible.

[Revendication 8] Procédé selon l'une des revendications 6 et 7, dans lequel l'étape (a) comprend en outre la sélection préalable d'un nombre de réseaux de neurones à convolution approximant la fonction identité dudit ensemble à sélectionner.

[Revendication 9] Procédé selon l'une des revendications 5 à 8, dans lequel l'étape (a0) est une étape mise en œuvre par des moyens de traitement de données (11) d'un serveur (1) d'apprentissage des paramètres du premier réseau de neurones et de l'au moins un réseau de neurones à convolution approximant la fonction identité à partir d'au moins une base de données d'apprentissage.

[Revendication 10] Procédé selon l'une des revendications 1 à 9, dans lequel le premier réseau de neurones et le ou les réseaux de neurones à convolution approximant la fonction identité comprennent une alternance de couches linéaires et de couches non-linéaires à fonction d'activation telle que la fonction ReLU.

[Revendication 11] Procédé selon les revendications 2 et 10 en combinaison, dans lequel ladite couche cible est une couche linéaire.

[Revendication 12] Procédé selon l'une des revendications 10 et 11, dans lequel l'au moins un réseau de neurones à convolution approximant la fonction identité comprend deux ou trois couches linéaires, qui sont des couches de convolution à filtre par exemple de taille 5x5.

[Revendication 13] Procédé d'apprentissage des paramètres d'un deuxième réseau de neurones, le procédé étant caractérisé en ce qu'il comprend la mise en œuvre par des moyens de traitement de données (11) d'un serveur (1) d'étapes de :

(a) construction du deuxième réseau de neurones correspondant à un premier réseau de neurones dans lequel est inséré au moins un réseau de neurones à convolution approximant la fonction identité ;

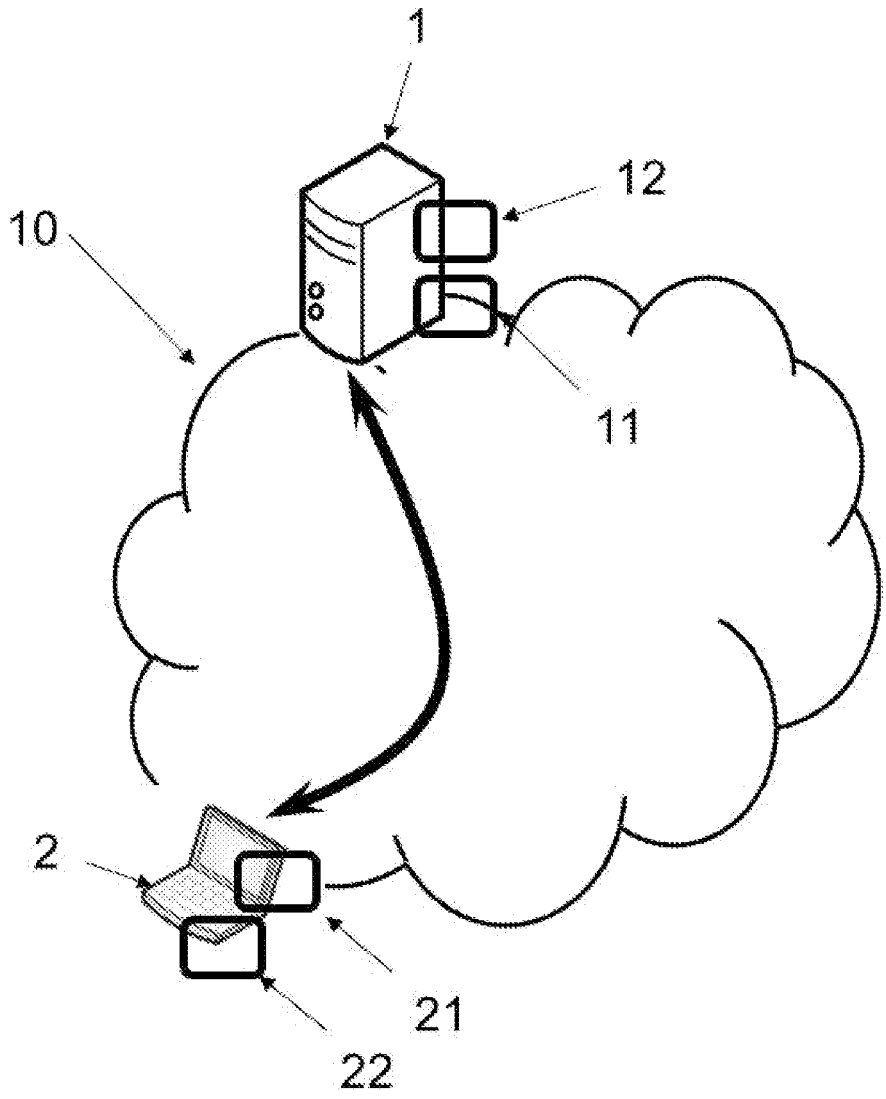
(a1) apprentissage des paramètres du deuxième réseau de neurones à partir d'une base de données d'apprentissage

[Revendication 14] Procédé d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée, le procédé comprenant l'apprentissage de paramètres d'un deuxième réseau de neurones conformément au précédé selon la revendication 13 ; et la mise en œuvre par des moyens de traitement de données (21) d'un terminal (2) d'une étape (b) d'utilisation du deuxième réseau de neurones sur ladite donnée d'entrée.

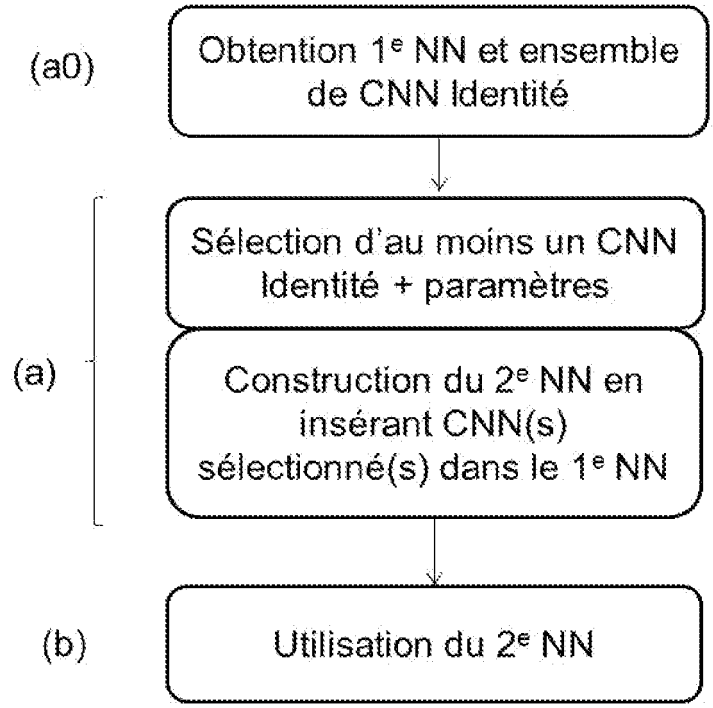
[Revendication 15] Produit programme d'ordinateur comprenant des instructions de code pour l'exécution d'un procédé selon l'une des revendications 1 à 14 d'apprentissage de paramètres d'un deuxième réseau de neurones, ou d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée, lorsque ledit programme est exécuté par un ordinateur.

[Revendication 16] Moyen de stockage lisible par un équipement informatique sur lequel un produit programme d'ordinateur comprend des instructions de code pour l'exécution d'un procédé selon l'une des revendications 1 à 14 d'apprentissage de paramètres d'un deuxième réseau de neurones, ou d'utilisation sécurisée d'un premier réseau de neurones sur une donnée d'entrée.

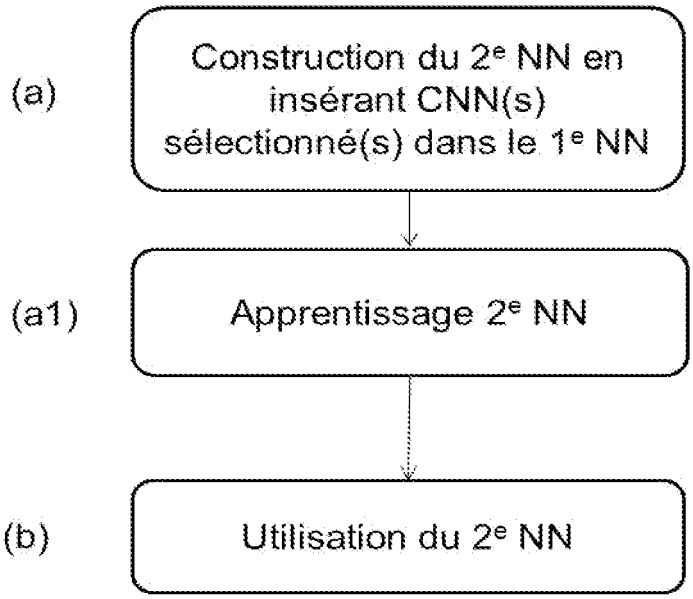
[Fig. 1]


1
12
10
11
2
21
22

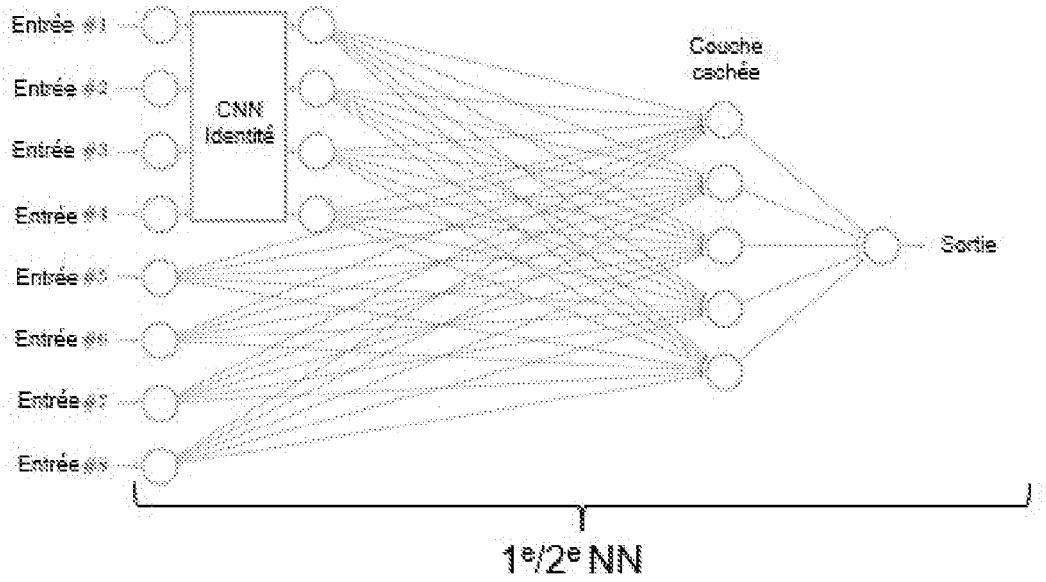
[Fig. 2a]


(a0)
Obtention 1e NN et ensemble de CNN Identité
(a)
Sélection d'au moins un CNN Identité + paramètres
Construction du 2e NN en insérant CNN(s) sélectionné(s) dans le 1e NN
(b)
Utilisation du 2e NN

[Fig. 2b]


(a)
Construction du 2e NN en insérant CNN(s) sélectionné(s) dans le 1e NN
(a1)
Apprentissage 2e NN
(b)
Utilisation du 2e NN

[Fig. 3]


Entrée #1
Entrée #2
Entrée #3
Entrée #4
Entrée #5
Entrée #6
Entrée #7
Entrée #8
CNN Identité
Couche cachée
Sortie
1e/2e NN

RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

RAPPORT DE RECHERCHE PRÉLIMINAIRE

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

N° d'enregistrement national

FA 882692
FR 2004945

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	MATHIAS LECUYER ET AL: "Certified Robustness to Adversarial Examples with Differential Privacy", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 9 février 2018 (2018-02-09), XP081235595, * abrégé * * chapitre I.; page 1 - page 2 * * Chapitres III. - IV.; page 4 - page 11; figures 1-3; tableaux I-III *	1-11, 13-16	G06N3/08 G06F21/55
X	US 2019/095629 A1 (LEE TAESUNG [US] ET AL) 28 mars 2019 (2019-03-28) * abrégé * * alinéa [0001] - alinéa [0004] * * alinéa [0018] - alinéa [0034] * * alinéa [0047] - alinéa [0058]; figures 1A, 1B *	1-16	
A	ATIN SOOD ET AL: "NeuNetS: An Automated Synthesis Engine for Neural Network Design", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 17 janvier 2019 (2019-01-17), XP081005604, * abrégé * * Chapitre 4.2; page 9 - page 10 *	2	DOMAINES TECHNIQUES RECHERCHÉS (IPC) G06N G06F

Date d'achèvement de la recherche	Examinateur
24 février 2021	Hasnas, Sergiu

CATÉGORIE DES DOCUMENTS CITÉS

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

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

EPO FORM 1503 12.99 (P04C14)

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 2004945 FA 882692

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.
Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du 24-02-2021
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 2019095629 A1	28-03-2019	AUCUN	

EPO FORM P0465