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

(19) Organisation Mondiale de la Propriété Intellectuelle
Bureau international

(43) Date de la publication internationale
23 octobre 2025 (23.10.2025)

WIPO | PCT

(10) Numéro de publication internationale
WO 2025/219411 A1

(51) Classification internationale des brevets :
H03M 7/42 (2006.01) *H03M 7/30* (2006.01)

(21) Numéro de la demande internationale :
PCT/EP2025/060420

(22) Date de dépôt international :
15 avril 2025 (15.04.2025)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
FR2404026 18 avril 2024 (18.04.2024) FR

(71) Déposant : **ORANGE** [FR/FR] ; 111, quai du Président Roosevelt, 92130 Issy-les-Moulineaux (FR).

(72) Inventeurs : **LAMPIN, Quentin** ; ORANGE - Intellectual Property and Licensing, Orange Gardens - 44 Avenue de la République - CS 50010, 92326 Chatillon (FR). **LARUE, Guillaume** ; ORANGE - Intellectual Property and Licensing, Orange Gardens - 44 Avenue de la République - CS 50010, 92326 Chatillon (FR). **DUFRENE, Louis Adrien** ; ORANGE - Intellectual Property and Licensing, Orange Gardens - 44 Avenue de la République - CS 50010, 92326 Chatillon (FR).

(81) États désignés *(sauf indication contraire, pour tout titre de protection nationale disponible)* : AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH,

(54) Title: METHOD FOR MANAGING THE COMMUNICATION OF AN ITEM OF VECTOR DATA

(54) Titre : PROCÉDÉ DE GESTION DE LA COMMUNICATION D'UNE DONNÉE VECTORIELLE

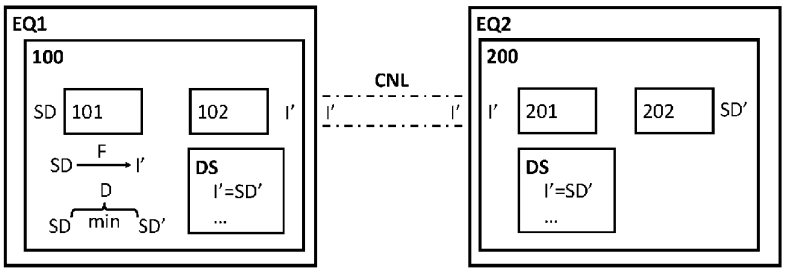


Fig. 1

(57) Abstract: The invention relates to a method for managing the communication of an item of vector data (SD), referred to as the item of data to be communicated, from a first device (EQ1) to a second device (EQ2), the method comprising: obtaining an index (I') by applying an indexing function (F) to the item of data to be communicated (SD), wherein the indexing function (F) is such that the closer two items of vector data are, as measured by a distance (D) between them, the closer the indices calculated for these two items of vector data by the function (F) are to one another, wherein the index (I') makes it possible to retrieve, in a data structure (DS), another item of vector data (SD'), referred to as the item of communicated data, wherein the data structure (DS) associates indices with items of vector data, and wherein the index (I') is capable of being transmitted from the first device (EQ1) to the second device (EQ2).

(57) Abrégé : Procédé de gestion de la communication d'une donnée vectorielle L'invention se rapporte à un procédé de gestion de la communication d'une donnée vectorielle (SD), dite donnée à communiquer, depuis un premier équipement (EQ1) à un deuxième équipement (EQ2], le procédé comprenant : obtenir un index (I') en appliquant une fonction d'indexation (F) à la donnée à communiquer (SD), la fonction d'indexation (F) étant telle que, plus deux données vectorielles sont proches selon une distance (D) mesurée entre elles, plus les index calculés pour ces deux données vectorielles par la fonction (F) sont proches entre eux, ledit index (I') permettant de retrouver dans une structure de données (DS) une autre donnée vectorielle (SD'), dite donnée communiquée, la structure de données (DS) associant des index à des données vectorielles, l'index (I') étant apte à être transmis depuis le premier équipement (EQ1) au deuxième équipement (EQ2).

WO 2025/219411 A1

[Suite sur la page suivante]

TJ, TM, TN, TR, TT, TZ, UA, UG, US, UY, UZ, VC, VN, WS, ZA, ZM, ZW.

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

Publiée:

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

— *en noir et blanc ; la demande internationale telle que déposée était en couleur ou en échelle de gris et est disponible sur PATENTSCOPE pour téléchargement.*

Description

Titre : Procédé de gestion de la communication d'une donnée vectorielle

Domaine technique

Le domaine technique est celui des télécommunications.

Plus précisément, l'invention se rapporte à un procédé de gestion de la communication d'une donnée vectorielle. La donnée vectorielle communiquée représente un contenu sémantique.

Une donnée va être communiquée par le procédé selon l'invention depuis un équipement à un autre équipement. Une telle donnée est un type de données particulier qui est utilisé dans le domaine de l'intelligence artificielle. Les capacités de l'intelligence artificielle, et en particulier de l'intelligence artificielle générative (traduction de l'anglais *generative artificial intelligence*), c'est-à-dire la branche de l'intelligence artificielle qui se concentre sur la production automatique de médias, en particulier des textes, des images ou des vidéos, sont apparues aux yeux du grand public avec la mise en ligne de Chat-GPT (produit de la société Open AI), dont la première version date de novembre 2022. D'autres programmes d'intelligence artificielle générative sont développés par les sociétés Google (produit Gemini), Anthropic (produit Claude) ou Mistral AI (produit Le Chat). Les produits nommés ici sont des agents conversationnels mais d'autres produits d'intelligence artificielle générative peuvent produire des images (produits DALL-E de la société Open AI, ou Midjourney, du laboratoire de recherche du même nom), des vidéos, des sons, mais encore des programmes d'ordinateurs ou bien des spécifications de molécules chimiques. D'autres tâches d'intelligence artificielle peuvent concerner par exemple la reconnaissance de formes, la prise de décisions, la classification, la planification ou tout autre type d'activités imitant les capacités de l'intelligence humaine.

Les programmes d'intelligence artificielle générative encodent pour traitement les données qu'ils manipulent dans un format que nous nommons ici donnée vectorielle. Nous utilisons également les termes donnée sémantique ou vecteur sémantique. Un contenu sémantique, le sens attendu ou interprété par des utilisateurs d'un texte, image ou vidéo, est donc associé à une donnée vectorielle. Une telle donnée vectorielle est un vecteur de grande dimension dans l'espace des nombres réels. Le nombre de dimensions typique pour une donnée vectorielle va de plusieurs centaines à plusieurs milliers. Le projet MTEB (acronyme de l'anglais *Massive Text Embedding Benchmark*) réalise des comparaisons de performance entre modèles utilisés par des programmes d'intelligence artificielle générative et, en mars 2024, sur les dix modèles les plus performants, sept utilisaient des vecteurs de dimension 4096 et trois de dimension 1024.

Le terme le plus souvent utilisé en langue anglaise pour parler des données vectorielles manipulées par les programmes d'intelligence artificielle générative est *embedding*. Cependant, ce terme confond la fonction qui va encoder une donnée telle qu'un texte ou une image pour traitement ultérieur, fonction

mathématique qui est un plongement (traduction de l'anglais *embedding*) et la donnée obtenue comme résultat de l'application de cette fonction. Pour éviter cette confusion entre une fonction et son résultat, nous utiliserons par la suite dans ce document en général le terme donnée vectorielle, même si on retrouve plus souvent dans la littérature, en anglais et en français, le terme *embedding*.

Les données vectorielles sont donc caractérisées par le fait qu'elles représentent un contenu sémantique sous la forme d'une donnée informatique. Il est possible de voir qu'un contenu sémantique est bien associé aux données vectorielles en s'intéressant aux transformations qui peuvent leur être appliquées. Un exemple classique en intelligence artificielle considère une donnée correspondant au concept de « France » qui peut être transformée en une autre donnée correspondant au concept de « Paris » et la même transformation, appliquée à la donnée vectorielle associée à « Allemagne », produira la donnée vectorielle associée à « Berlin » et de même pour des données « Espagne »-« Madrid », « Italie »-« Rome » etc. Cela indique qu'un contenu sémantique correspondant à un pays ou à une ville est bien associé à la donnée vectorielle correspondante et que des manipulations de la donnée vectorielle correspondent à des manipulations des concepts associés à la donnée vectorielle.

Les données vectorielles, initialement apparues dans le contexte de l'intelligence artificielle générative, ont trouvé des utilisations dans l'ensemble des programmes d'intelligence artificielle.

Comme indiqué, les données dont nous parlons sont des vecteurs de nombres réels de dimension allant de plusieurs centaines à plusieurs milliers. Ces plusieurs centaines ou milliers de nombres réels qui forment une donnée vectorielle sont en général représentés pour traitement informatique par un type de données appelé nombre flottant (traduction de l'anglais *float*) qui utilise 32 bits pour être représentés, c'est-à-dire 4 octets. Une donnée qui est un vecteur de dimension 4096 aura donc une taille mémoire approximative de 16 Ko (16 kilooctets, soit 16 kilobytes en anglais, ou 16 Kb).

On peut noter que le nombre de dimensions des données vectorielles a tendance à augmenter avec le temps, et que les modèles les plus performants ont tendance à utiliser des vecteurs de dimension plus élevée.

Certains chercheurs d'intelligence artificielle proposent d'utiliser pour représenter des concepts des vecteurs de nombres complexes. Une telle proposition, si elle était adoptée et que le nombre de dimension restait constant, impliquerait de multiplier par deux la taille des données vectorielles en remplaçant un nombre réel flottant sur 32 bits par deux nombres réels afin de représenter un nombre complexe.

Par ailleurs, on peut parler de communication de donnée dans plusieurs cas. Par exemple, il y a communication lorsqu'une donnée informatique est transmise sur un réseau informatique, qui est local à plusieurs ordinateurs, ou bien sur un réseau de télécommunications qui peut être local ou à longue distance, ou bien lorsqu'une donnée est transférée d'un support à l'autre, c'est-à-dire par exemple copiée depuis un système de fichiers d'un ordinateur vers un support mémoire connecté à l'ordinateur tel qu'une clé USB ou un disque dur externe, ou réciproquement.

Dans tous ces cas de communication, des erreurs de transmission peuvent survenir. Les données informatiques sont définies classiquement comme des suites de bits. Lors de la communication d'une donnée, un ou plusieurs bits constituant cette donnée peuvent être modifiés par le canal de transmission. Il y a dans ce cas erreur de transmission, et la donnée n'est plus transmise dans son intégrité et n'est donc pas communiquée sans erreur.

Nous venons de parler de suites de bits pour définir des données informatiques. Cependant, en toute généralité, il faut plutôt parler de suites de symboles. En effet, les bits sont des symboles qui vont par paire (0 et 1) mais les transmissions réellement effectuées au niveau physique concernent rarement de tels symboles qui vont par paires. Les symboles transmis au niveau physique vont être par exemple des ondes électromagnétiques auxquelles des modulations de changement de phase sont appliquées afin de transmettre une valeur parmi 4, 8 ou 16 valeurs. Ces techniques sont par exemple utilisées en télécommunications mobiles. Les modulations de changement de phase sont dénommées *phase-shift keying* en anglais et on retrouve les acronymes 4-PSK, 8-PSK ou 16-PSK pour décrire des modulations de changement de phase qui permettent de transmettre en une seule émission une valeur parmi 4, 8 ou 16 valeurs. Dans les transmissions par fibres optique, l'utilisation de longueurs d'onde permet là aussi de transmettre des symboles qui peuvent prendre une valeur parmi de nombreuses valeurs possibles et pas seulement des symboles tels que des bits qui prennent leurs valeurs dans une paire de valeurs possibles. Même si les symboles transmis en tant que tels ne sont pas des bits et peuvent prendre plus de deux valeurs, il existe malgré tout dans tous les cas un lien direct entre un symbole transmis et des bits. Par exemple, dans le cas de la modulation 4-PSK, les symboles transmis peuvent prendre quatre valeurs et ces quatre valeurs correspondent respectivement aux paires de bits '00', '01', '10' et '11'.

Nous parlerons donc dans ce document du concept général de suite de symboles qui peut être compris dans une lecture plus simplifiée comme étant une suite de bits.

Ces points sont bien connus de l'homme du métier et ne seront pas développés plus avant.

État de la technique

Le développement à venir de l'intelligence artificielle promet une explosion à venir de la transmission des données vectorielles représentant un contenu sémantique. En effet, on peut facilement imaginer des applications intéressantes dans lesquelles les données vectorielles seraient communiquées depuis un équipement qui les produit vers un autre équipement qui va les utiliser.

Par exemple, lors d'une communication téléphonique, un système de reconnaissance de la parole couplé à un système d'interprétation pourrait interpréter chaque phrase prononcée par un interlocuteur dans une donnée vectorielle. Le sens de la phrase prononcée par l'interlocuteur serait alors associé à la donnée vectorielle. Cette donnée serait communiquée à un autre interlocuteur qui disposerait d'un système inverse d'interprétation couplé à un système de synthèse de la parole. Ce système permettrait

de réaliser une traduction automatique de la conversation téléphonique, quelles que soient les langues utilisées par les interlocuteurs. Dans une conférence téléphonique avec des interlocuteurs parlant chacun sa langue, les interlocuteurs entendraient les interventions de chacun traduites à la volée dans sa langue.

Pour donner un autre exemple, un système d'intelligence artificielle peut détecter sur une vidéo filmée par une caméra de vidéosurveillance des événements tels qu'une attaque ou un cambriolage. Un utilisateur peut souhaiter disposer d'un tel service de vidéosurveillance sans que le flux vidéo filmant sa maison soit transmis à l'extérieur. Plutôt que de transmettre la vidéo, un système de vidéosurveillance par intelligence artificielle peut produire des données vectorielles à contenu sémantique et les communiquer. Un système entraîné pour détecter les intrusions n'aura alors besoin que de ces données vectorielles pour détecter d'éventuelles intrusions et la confidentialité des utilisateurs de la vidéosurveillance sera préservée.

Dans le domaine des télécommunications, la recherche commence à explorer le concept de réseau sémantique qui s'adapte automatiquement aux besoins des applications qui transmettent des données sur le réseau en utilisant des techniques d'intelligence artificielle. De tels réseaux sémantiques doivent s'appuyer sur des représentations généralistes de l'information qui se prêtent à des adaptations automatiques et à des optimisations. Ces représentations peuvent porter des données sémantiques de diverses natures, comme des images, du texte, voire des éléments de configurations du réseau sémantique lui-même. Un réseau sémantique s'appuie donc sur les données vectorielles sémantiques comme format de donnée élémentaire.

Ces exemples d'application possibles montrent l'intérêt général de transmettre des données vectorielles (souvent appelées *embeddings* en anglais) dans les réseaux informatiques et/ou de télécommunications ou, plus généralement, d'interconnecter des systèmes d'intelligence artificielle afin de rendre des tâches complexes.

Plus généralement, une donnée vectorielle est une représentation multimodale. Il est possible de représenter toutes sortes de données (textes, images, vidéos etc.) sous forme de données vectorielles, sachant que les contenus sémantiques de ces représentations multimodales se retrouvent associés aux données vectorielles. Les données vectorielles à contenu sémantique pourraient donc être utilisées comme format de donnée universel au sein des réseaux de télécommunications à la place des suites de bits utilisées aujourd'hui. Une telle donnée présenterait alors une granularité d'information plus fine et des possibilités d'optimisation plus poussées.

L'interconnexion d'applications d'intelligence artificielle ou le déploiement de réseaux sémantiques va impliquer une explosion de la quantité de données vectorielles à communiquer entre équipements. Ces communications de données vectorielles vont correspondre à la transmission effective de symboles entre équipements. Le problème à résoudre est donc d'arriver à communiquer des données vectorielles depuis un émetteur vers un récepteur en préservant les contenus sémantiques associés aux données vectorielles à communiquer tout en économisant le nombre de symboles effectivement transmis.

Le domaine scientifique des communications sémantiques (traduction de l'anglais *semantic communications*) s'intéresse à la problématique de ne communiquer que le contenu minimal des données nécessaires à la réalisation d'une tâche et en particulier au contenu nécessaire pour faire communiquer entre eux des systèmes d'intelligence artificielle. L'approche prévalente dans ce domaine est de ne communiquer que les données pertinentes à une tâche visée afin de limiter la taille mémoire des éléments transmis en termes de bits ou octets transmis. Dans l'exemple donné ci-dessus de vidéosurveillance, un traitement avant émission cherchera à détecter des images (ou extraits d'images) suspects de représenter une intrusion et seuls ces extraits seront communiqués, en retirant des images ou données sémantiques des éléments jugés inutiles à la tâche de vidéosurveillance. Par exemple, la couleur pourra être retirée avant émission dans cette approche.

Un inconvénient majeur de cette approche est que le traitement réalisé avant émission dépend de la tâche qui est attendue à la réception. Or, cette tâche à la réception ne peut pas être connue dans tous les cas. Connaître cette tâche à la réception implique aussi que l'opérateur de télécommunications (qui transmet les données et va souvent les préparer à l'émission) doit être informé des tâches que réalisent les applications qui utilisent son réseau. Or, les opérateurs de télécommunications sont en général agnostiques quant aux applications qui utilisent leurs réseaux. Et il serait de plus très peu pratique de développer des optimisations réseau spécifiques à chaque tâche de traitement attendu. Pour ces raisons, une telle approche ne pourrait s'appliquer que dans le cas très spécifique d'une application dédiée entre un émetteur et un récepteur qui seraient reliés entre eux par un réseau privé dans lequel des optimisations spécifiques seraient implémentées pour optimiser la tâche d'intelligence artificielle visée. Une telle approche ne pourra pas « passer à l'échelle » et se généraliser aux innombrables usages attendus en intelligence artificielle et plus généralement à tous les usages dans lesquels on cherche à adapter les données transmises en fonction du traitement qui sera effectué par l'application réceptrice.

L'invention vient améliorer la situation.

Exposé de l'invention

Selon un premier aspect fonctionnel, l'invention a trait à un procédé de gestion de la communication d'une donnée vectorielle, dite donnée à communiquer, depuis un premier équipement à un deuxième équipement, le procédé comprenant :

– obtenir un index, dit index obtenu, en appliquant une fonction d'indexation à la donnée à communiquer, la fonction d'indexation calculant un index à partir d'une donnée vectorielle, la fonction d'indexation étant telle que, plus deux données vectorielles sont proches selon une distance mesurée entre elles, plus les index calculés pour ces deux données vectorielles par la fonction sont proches entre eux, ledit index obtenu permettant de retrouver dans une structure de données une autre donnée vectorielle, dite donnée communiquée, la structure de données associant des index à des données vectorielles, l'index obtenu étant apte à être transmis depuis le premier équipement au deuxième équipement.

Dans un mode de réalisation, le procédé comprend de surcroît :
– transmettre l'index obtenu depuis le premier équipement au deuxième équipement.

Grâce à l'invention, le nombre des symboles à transmettre pour communiquer une donnée vectorielle est très fortement réduite. En effet, au lieu de transmettre les symboles ou bits encodant la donnée vectorielle tels quels, le procédé obtient puis, dans certains modes, transmet un index correspondant en général à un élément d'une structure de données. Ce n'est pas nécessairement la même entité qui obtient l'index que celle qui le transmet. Cet index permet de retrouver dans la structure de données une donnée vectorielle. Si on suppose que la donnée vectorielle à communiquer est un vecteur de dimension 4096 de nombres réels, et que ces nombres réels sont représentés par des flottants de 32 bits, sa taille est approximativement de 16 kilooctets. Si la structure de données comprend 65 536 éléments, chacun de ces éléments pourra disposer d'un index différent si ceux-ci sont des suites de 16 bits (en effet, 2 à la puissance 16 vaut 65 536). Au lieu de transmettre 16 kilooctets, le procédé va donc transmettre 16 bits, soit 2 octets. La réduction de taille de la donnée transmise est ici très spectaculaire.

La structure de données peut être de tout type à partir du moment qu'elle permet de retrouver une donnée vectorielle à partir d'un index. Il peut s'agir par exemple d'une table clés-valeurs dans laquelle les index jouent le rôle de clés et les valeurs sont des données vectorielles particulières. On parle aussi de dictionnaire. Mais la structure peut aussi être une base de données. Ou bien la structure peut organiser les index selon un arbre de préfixes et les données enregistrées dans la structure seront alors des feuilles de l'arbre en question.

Le procédé comprend l'obtention d'un premier index à partir de la donnée vectorielle qui est à communiquer. La structure de données comprend un nombre fini d'éléments alors que l'espace possible des données vectorielles est très grand, voire infini. Par exemple, si les données à communiquer sont des vecteurs de nombres réels de dimension 4096, elles appartiennent à l'espace vectoriel R^{4096}. En réalité, l'espace des données vectorielles est fini puisqu'il s'agit de données informatiques qui sont des ensembles de nombres flottants codés sur 32 bits. Cependant, il est raisonnable de considérer l'espace possible des données vectorielles comme infini en première approximation. Il y a donc très peu de chance que la donnée à communiquer, qui appartient à un espace infini, soit présente dans la structure de données qui est un ensemble fini, de taille raisonnable.

L'index obtenu est apte à être transmis depuis le premier équipement au deuxième équipement. A la place de la donnée à communiquer, le procédé transmet (ou fait transmettre par une autre entité) un index qui permet de retrouver une autre donnée, qui sera donc la donnée effectivement communiquée. Pour que le contenu sémantique associé à la donnée à communiquer soit préservé, l'index obtenu devra permettre de retrouver dans la structure de données une donnée vectorielle dont le contenu sémantique associé est très proche, voire le plus proche possible, de celui de la donnée à communiquer. L'index est donc obtenu en appliquant une fonction d'indexation à la donnée à communiquer, fonction qui préserve la distance entre données vectorielles. L'index est donc choisi en fonction d'une distance mesurée entre la distance à communiquer et la distance communiquée. En général, l'index sera choisi

pour minimiser la distance mesurée entre la donnée à communiquer et la donnée effectivement communiquée.

Deux cas sont alors possibles :

- Ou bien l'index obtenu est exactement un index associé à une donnée sémantique dans la structure de données. Dans ce cas, si on suppose que l'index est transmis sans erreur de transmission, il permettra de retrouver directement une donnée sémantique dans la structure de données, qui est alors la donnée communiquée.
- Ou bien l'index obtenu en appliquant la fonction d'indexation à la donnée à communiquer n'est pas un index associé à une donnée sémantique dans la structure de données. Dans ce cas, il peut quand même être transmis directement et il faudra, à la réception, chercher dans la structure de données un index proche qui donnera accès à une donnée sémantique. Il est aussi possible de chercher, avant transmission, un index proche de l'index obtenu, index associé à une donnée vectorielle qui sera transmis.

Dans tous les cas, l'index obtenu permet de retrouver, directement ou indirectement, une donnée vectorielle dans la structure de données, et c'est cette donnée vectorielle qui sera considérée comme étant la donnée communiquée.

On peut noter que l'index obtenu n'est pas toujours associé directement à une donnée sémantique dans la structure de données mais permet de retrouver une donnée sémantique dans la structure de données. De plus, des erreurs de transmission ajoutées par le canal de communication peuvent faire en sorte qu'un index obtenu qui était directement associé dans la structure de données à une donnée vectorielle soit transformé en un index qui ne corresponde plus directement à une donnée vectorielle. Il faut donc dans tous les cas prévoir à la réception qu'il puisse être nécessaire de chercher dans la structure de données un index proche de l'index reçu qui corresponde directement à une donnée vectorielle dans la structure de données. La façon dont l'index est obtenu doit chercher à ce que l'index obtenu, auquel des erreurs de transmission sont rajoutées, permette de retrouver dans la structure de données une donnée vectorielle qui soit proche, selon la distance mesurée, de la donnée à communiquer.

La structure de données réalise une quantification de l'espace vectoriel dans lequel les données vectorielles prennent des valeurs. Au lieu d'une infinité possible de valeurs, un ensemble fini et restreint (mettons qu'il comprenne 65 536 éléments) est défini qui donnera des représentants pour les données à communiquer. Ces représentants sont choisis en fonction d'une distance mesurée entre données vectorielles. Deux choix sont possibles concernant la taille des index en regard du nombre d'éléments présents dans la structure de données :

- Ou bien la taille des index est telle que le nombre d'index possibles correspond exactement au nombre d'éléments dans la structure de données. Dans l'exemple ci-dessus de 65 356 éléments, un index de 16 bits donnera exactement ce nombre. Chaque obtention d'index à partir d'une donnée sémantique à communiquer donnera donc un index présent dans la

structure de données, associé à une donnée communiquée, l'index étant choisi selon la distance entre la donnée à communiquer et celle communiquée. Cet index sera transmis, avec éventuellement des erreurs de transmission, qui donneront un autre index qui sera lui aussi présent dans la structure de données. En supposant que ces erreurs sont peu nombreuses, l'index reçu permettra de retrouver dans la structure de données une donnée vectorielle reçue qui sera également à une distance convenable de la donnée communiquée et de la donnée à communiquer.

- Ou bien la taille des index est plus grande que la taille qui donne un nombre d'index possibles égal au nombre d'éléments dans la structure de données. Dans l'exemple de 65 356 éléments, on pourrait choisir un index de 20 bits qui prend plus d'un million de valeurs possibles. Dans ce cas, il y a beaucoup plus d'index possibles que de valeurs dans la structure de données. L'index obtenu pourra être exactement l'index d'une donnée présente dans la structure, ou bien un index proche. Des erreurs de transmission pourront faire dériver l'index reçu par rapport à l'index transmis. Mais comme il y a beaucoup plus de nombres de valeurs d'index possibles que d'éléments présents dans la structure de données, il est probable que l'index reçu ne corresponde à aucun élément dans la structure de données mais que l'index le plus proche de l'index reçu soit bien celui transmis au départ.

Autrement dit, la longueur des index et la taille de la structure de données peuvent être telles que le nombre de valeurs possibles des index est égal au nombre d'éléments dans la structure de données ou bien que le nombre de valeurs possibles des index soit supérieur au nombre d'éléments dans la structure de données. La transmission de l'index permettra en réception de retrouver dans la structure une donnée vectorielle qui sera alors considérée comme étant la donnée vectorielle communiquée. Quand le nombre de valeurs possibles des index est plus grand que le nombre d'éléments dans la structure de données et que l'index reçu se trouve en dehors de la structure de données, c'est l'index le plus proche de l'index reçu qui permettra de retrouver une donnée vectorielle qui sera utilisé.

Dans le premier cas, l'avantage est de transmettre un index d'une taille minimale par rapport au nombre d'élément présents dans la structure de données. Dans le deuxième cas, l'avantage est d'être résistant à des erreurs de transmission de l'index.

L'index est une suite de symboles, appartenant à un ensemble de symboles. Dans certains modes de réalisation, ces symboles sont des bits, c'est-à-dire qu'ils peuvent prendre deux valeurs possibles représentées en général par '0' et '1'. Dans d'autres modes de réalisation, ces symboles correspondent à des grandeurs qui seront celles utilisées dans la transmission physique utilisée par le procédé. Par exemple, comme expliqué ci-dessus, l'index calculé peut être une suite de symboles qui correspondent à des modulations de changements de phase parmi quatre changements de phase possible, c'est-à-dire des symboles 4-PSK. Dans ce cas, les index seraient des suites de valeurs choisies parmi les quatre valeurs possibles 4-PSK. Une suite de 8 symboles 4-PSK prendra alors une valeur parmi 65 536 possibles (en effet, 4 à la puissance 8 vaut 65 536) et un dictionnaire de 65 536 points pourra être indexé par des index de longueur 8 formés par des symboles 4-PSK à la place des index de longueur

16 bits. Par la suite, nous choisirons le plus souvent des exemples dans lesquels les index sont des suites de bits mais il est clair que ces exemples ne sont que des exemples parmi d'autres.

La transmission de l'index obtenu peut être réalisée en transmettant directement la suite de symboles formant l'index. Dans d'autres modes de réalisation, la suite de symboles formant l'index obtenu est traduite dans un format adapté à la transmission puis transmise, ce qui correspond à la transmission de l'index. Dans des modes de réalisation, c'est l'entité qui obtient l'index qui va transmettre l'index obtenu. Dans d'autres modes de réalisation, l'index apte à être transmis est confié à une autre entité qui réalisera effectivement la transmission.

On a vu que la réduction de taille de la donnée communiquée est très forte, puisqu'au lieu de transmettre les symboles ou les bits formant la donnée tels quels, on transmet un index qui permet de retrouver une donnée vectorielle. Soit l'index obtenu est associé à une donnée vectorielle dans la structure de données, et la donnée associée est alors retrouvée directement. Soit l'index obtenu n'est pas associé à une donnée vectorielle, mais un index proche de l'index obtenu, associé à une donnée vectorielle, pourra être obtenu à partir du premier index, si bien qu'une donnée vectorielle est alors retrouvée indirectement. Un avantage du procédé est que cette réduction est obtenue sans aucune prise en compte du traitement ultérieur qui va être effectué sur la donnée vectorielle communiquée. Le procédé de transmission est agnostique vis-à-vis des applications d'intelligence artificielle ou autres utilisant la donnée vectorielle. Cette caractéristique est particulièrement appréciable pour un opérateur de télécommunications qui veut proposer des services de communication de données généralistes, à toutes sortes de clients possibles.

L'index transmis permet de retrouver une donnée vectorielle présente dans la structure de données. L'index est choisi en utilisant une fonction d'indexation qui prend en compte une distance mesurée entre cette donnée vectorielle et la donnée vectorielle initiale à communiquer. La distance entre données vectorielles peut être mesurée par une distance au sens mathématique du terme sur l'ensemble auquel appartient la donnée communiquée. Par exemple, la donnée vectorielle est un vecteur de 4096 réels et la distance sera mesurée par une distance sur l'ensemble vectoriel R^{4096}. Si la donnée vectorielle est un vecteur de 1024 complexes, la distance sera mesurée par une distance sur l'ensemble vectoriel C^{1024}. Comme ce qui est transmis à la place de la donnée est un index correspondant à une autre donnée vectorielle, index calculé par une fonction d'indexation qui prend en compte la distance entre données vectorielles dans l'ensemble auquel les données appartiennent, la distorsion causée par la transmission de l'index et par la récupération d'une donnée présente dans la structure de données au lieu de la donnée initiale à communiquer elle-même, cette distorsion est minimisée.

Selon une variante de l'invention, la structure de données comprend une donnée vectorielle associée à toutes les valeurs possibles d'un index d'une longueur donnée.

De cette manière, lorsque la fonction d'indexation calcule un index à partir de la donnée vectorielle à communiquer, le procédé est assuré de trouver dans la structure de données l'index résultat de ce

calcul. Cet index peut alors être transmis en étant assuré qu'une donnée vectorielle lui est bien associé dans la structure de données.

Selon une variante de l'invention, le procédé comprend transmettre l'index calculé depuis le premier équipement au deuxième équipement.

Dans cette variante, l'index calculé par la fonction d'indexation à partir de la donnée vectorielle à communiquer est transmis même s'il n'est pas associé à une donnée vectorielle dans la structure de données. Ce sera à la réception que l'index reçu permettra de trouver dans la structure de données une donnée vectorielle associée. L'index reçu pourra être l'index calculé qui a été transmis s'il n'y a pas d'erreurs de transmission. Il n'y a donc pas forcément une donnée vectorielle associée à l'index reçu dans la structure de données mais il sera possible de trouver l'index le plus proche de l'index reçu pour lequel une donnée vectorielle est associée. L'index reçu peut être différent de l'index calculé qui a été transmis à la suite d'une ou plusieurs erreurs de transmission. Il sera là aussi possible d'obtenir un index associé à une donnée vectorielle dans la structure de données en cherchant l'index qui convient le plus proche de l'index reçu.

Selon un premier mode de réalisation de cet aspect fonctionnel de l'invention, la distance mesurée entre la donnée à communiquer et la donnée communiquée est fonction de la proximité entre les contenus sémantiques associés aux deux données vectorielles à communiquer et communiquée.

Le choix de la distance utilisée pour mesurer la distance entre données vectorielles est fait de telle manière à correspondre à l'interprétation ultérieure du contenu sémantique des données, c'est-à-dire qu'une distance minimale entre données vectorielles correspond à une distance minimale entre significations interprétées par les utilisateurs et les applications qui manipulent les données vectorielles. La distance mesurée est donc fonction de la proximité entre les contenus sémantiques associés aux données vectorielles ou plus précisément est calculée par une fonction préservant la proximité entre les contenus sémantiques associés aux données vectorielles. Cette fonction peut être une distance au sens mathématique du terme sur l'espace vectoriel dans lequel les données prennent des valeurs. Le fait que la fonction qui mesure la distance entre données vectorielles préserve la proximité entre contenus sémantiques associés aux données et que l'index est choisi en fonction de cette distance mesurée entre la donnée à communiquer et celle effectivement communiquée va assurer que le contenu sémantique associé à la donnée vectorielle à communiquer ne sera pas trop distordu par la distorsion qui consiste à communiquer une autre donnée vectorielle, correspondant à l'index obtenu, à la place de la donnée à communiquer.

Le contenu sémantique associé à une donnée vectorielle est celui qui est présent à l'origine de la production d'une donnée pour une utilisation ultérieure. Il peut s'agir du sens d'un texte, d'une image, d'une vidéo, d'un son tel que ce contenu sera interprété par des utilisateurs ou des applications d'intelligence artificielle. Un tel contenu sémantique peut être dit encodé dans la donnée vectorielle, ou présent dans la donnée vectorielle, ou intégré dans la donnée vectorielle, ou représenté par la donnée vectorielle. Nous utilisons ici le terme associé à la donnée vectorielle.

Selon un deuxième mode de réalisation de cet aspect fonctionnel de l'invention, qui pourra être mis en œuvre alternativement ou cumulativement avec le mode précédent, l'index correspondant à la donnée à communiquer est obtenu en cherchant dans la structure de données la donnée vectorielle la plus proche de la donnée à transmettre selon la distance mesurée, et l'index transmis est l'index associé à ladite donnée vectorielle la plus proche.

Grâce à ce mode de réalisation, une manière simple de procéder au calcul de l'index est donnée. Les données vectorielles présentes dans la structure de données ont été choisies à l'avance pour réaliser une quantification efficace de l'espace dans lequel se trouvent les données à transmettre. Des index sont associés aux données vectorielles présentes dans la structure de données et ont été calculés au préalable. La donnée vectorielle présente dans la structure de données la plus proche de la donnée vectorielle à transmettre est déterminée, la proximité entre les données vectorielles étant donnée par la distance mesurée entre données vectorielles. En général, cette distance mesurée est telle qu'une distance minimale entre données correspond à une différence minimale dans les interprétations sémantiques ultérieures. En particulier, la distance peut être mesurée par une fonction préservant la proximité entre les contenus sémantiques associés aux données vectorielles. L'index associé à la donnée la plus proche est retrouvé dans la structure de données et c'est cet index qui sera transmis par la suite.

De cette manière, il n'est pas nécessaire de calculer pour chaque donnée vectorielle à transmettre un index mais simplement de retrouver dans la structure de données la donnée vectorielle la plus proche de la donnée à transmettre.

Selon l'invention, l'index est obtenu en utilisant une fonction d'indexation choisie de telle manière que, pour deux données vectorielles distinctes, plus ces données vectorielles sont proches selon la distance mesurée entre elles, plus les index calculés pour ces deux données vectorielles par la fonction d'indexation sont proches entre eux.

De cette manière, le procédé de communication d'une donnée sémantique selon l'invention résiste aux éventuelles erreurs de transmission des symboles qui forment l'index calculé. En effet, si un ou plusieurs symboles, en faible quantité, qui forment l'index transmis à la place de la donnée vectorielle, sont changés durant la transmission, l'index reçu finalement sera quand même proche de l'index transmis initialement car le nombre d'erreurs de transmission est faible en général. En effet, si les erreurs de transmission deviennent trop grandes, la transmission s'arrête. Le point dans le dictionnaire qui correspond à l'index reçu sera donc proche du point qui correspond à l'index initialement transmis en vertu du choix de la fonction d'indexation. La distorsion due à l'erreur de transmission d'un ou plusieurs symboles de l'index, qui s'ajoute à la distorsion du remplacement de la donnée vectorielle à communiquer par une autre donnée présente dans la structure de données, sera donc là aussi minimisée.

On peut noter que si un index reçu, à cause d'erreurs de transmission, ne correspond pas à un index présent dans la structure de données, le récepteur de l'index va chercher un index proche de l'index

reçu qui correspond bien à un élément de la structure. Cette recherche pourra permettre de revenir sur l'index qui était initialement transmis pour peu que le nombre d'erreurs de transmission soit assez faible. Et si l'index retrouvé ainsi est quand même différent de l'index initial, il reste quand même proche de l'index initial et le choix de la fonction d'indexation assure alors une proximité entre la donnée sémantique correspondant à l'index reçu et la donnée sémantique correspondant à l'index initial.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, la distance entre données vectorielles est mesurée en utilisant la similarité cosinus.

La similarité cosinus est une grandeur connue dans le domaine de l'intelligence artificielle pour mesurer la similarité de données présentant un contenu sémantique entre elles. La similarité cosinus de deux vecteurs est calculée en divisant le produit scalaire des deux vecteurs par le produit des normes des deux vecteurs. Cette grandeur vaut entre -1 et 1, et va valoir -1 pour des vecteurs opposés, 0 pour des vecteurs orthogonaux et 1 pour des vecteurs qui ont la même direction, et qui sont donc égaux à un facteur proportionnel près.

Il est possible de calculer une distance entre vecteurs à partir de la similarité cosinus, par exemple en ôtant la similarité cosinus de 1. Une telle distance varie entre 0 (pour des vecteurs ayant la même direction) et 2 (pour des vecteurs opposés). D'autres formules permettant d'obtenir une distance à partir de la similarité cosinus sont possibles.

Une telle distance entre vecteurs a l'avantage de ne pas prendre en compte la grandeur des vecteurs mais seulement leur direction. Une telle distance est adaptée au calcul de proximité sémantique entre données vectorielles intégrant un contenu sémantique. Elle préserve la proximité entre les contenus sémantiques associés aux données vectorielles dont la distance est mesurée avec une similarité cosinus.

Dans d'autres modes de réalisation, la distance entre deux données vectorielles est mesurée avec la distance euclidienne.

La distance euclidienne bien connue dans les espaces vectoriels donne une alternative de réalisation par rapport aux distances utilisant la similarité cosinus. La proximité selon la distance euclidienne correspond à une certaine proximité entre contenus sémantiques associés aux données vectorielles.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, la proximité entre deux index est définie par la distance de Hamming.

La distance de Hamming est une distance définie entre suites de bits qui est calculée en comptant le nombre de bits distincts à une position donnée entre deux suites de bits. Par exemple, la distance de Hamming entre l'octet 00000000 et l'octet 00010001 est de deux, car, aux quatrième et huitième positions, les bits sont différents. La notion de distance de Hamming peut être étendue pour s'appliquer à des suites de symboles comme le sont les index calculés. Cette distance donne une manière pratique

et efficace de calculer la proximité entre les index calculés et utilisés dans le procédé. La proximité entre index mesurée selon la distance de Hamming correspond bien à la notion de chercher à minimiser la distorsion causée par les erreurs de transmission. En effet, elle est directement associée au nombre d'erreurs de transmission subie par une suite de bits ou de symboles dans un canal de communication.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, la fonction d'indexation est choisie de telle manière que, pour deux données vectorielles distinctes, si la distance mesurée entre ces deux données vectorielles est inférieure à un seuil donné, alors la probabilité que les index calculés par la fonction d'indexation pour ces deux données vectorielles soient les mêmes est supérieure à une probabilité donnée.

Grâce à ce mode de réalisation, le lien entre la distance mesurée entre deux données vectorielles et la proximité entre les index calculés pour ces deux données est affiné. Dans une bulle autour d'une donnée vectorielle, c'est-à-dire pour l'ensemble des données vectorielles qui sont à une distance inférieure à un seuil de la donnée initiale, la probabilité que ces données aient un même index calculé est supérieure à une probabilité donnée. De cette manière, plus les données sont proches entre elles, plus leurs index sont proches. Les distances entre données vectorielles et proximités entre index peuvent être calculées en utilisant des similarités cosinus d'une part, des distances de Hamming d'autre part, ou tout autre mode adapté de calcul de distances entre données vectorielles et index.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, la fonction d'indexation est une fonction de hachage sensible à la localité.

Une fonction de hachage sensible à la localité (traduction de l'anglais *locality sensitive hashing*, pour laquelle on peut utiliser l'acronyme LSH) est une fonction, qui appartient à une famille de fonctions, qui calcule des index pour des points d'un espace vectoriel donné et qui vérifie les deux propriétés suivantes, à savoir que, d'une part, si la distance entre deux points est inférieure à un seuil, la probabilité que les index calculés par cette fonction pour les deux points soit égaux est supérieure à une première probabilité donnée et que, d'autre part, si la distance entre deux points est supérieure à un autre seuil, multiple du premier, la probabilité que les index calculés pour les deux points soient égaux est inférieure à une deuxième probabilité donnée, très inférieure à la première. Les fonctions de ce type qui indexent les points d'un espace vectoriel donné répondent bien à ce qui est souhaitable pour l'invention, à savoir que des données vectorielles proches auront des index proches, voire confondus, et que des données vectorielles éloignées auront au contraire des index éloignés, sachant que les données vectorielles sont des points d'un espace vectoriel donné.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, la fonction d'indexation est une fonction de hachage sensible à la localité pour la similarité cosinus.

Dans ce mode de réalisation, la famille de fonctions de hachage sensibles à la localité est déterminée pour une distance donnée entre données vectorielles et cette distance utilise la similarité cosinus.

L'avantage de ce mode est de réaliser une indexation adaptée à cette distance qui est adaptée à la proximité des contenus sémantiques associés aux données vectorielles.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le procédé comprend une étape préalable de fragmentation de la donnée vectorielles à communiquer en plusieurs fragments et les étapes ultérieures du procédé s'appliquent aux dits plusieurs fragments.

Jusqu'à présent, le procédé selon l'invention traitait une donnée qui est un point dans un espace vectoriel de grande dimension (jusqu'à 4096, voire plus) d'un seul bloc. Mais il peut être avantageux de procéder à une fragmentation de la donnée vectorielle à communiquer et d'appliquer les différentes étapes du procédé à des fragments de données. Par exemple, une donnée qui est un vecteur de 4096 réels pourrait être divisé en 1024 fragments de dimension 4 ou en 512 fragments de dimension 8.

Un premier avantage est de faciliter l'échange de structures de données dans certains modes de réalisation du procédé. En effet, la structure de données doit être connue en émission et en réception et, dans certains cas, elle sera transmise avant la mise en place du procédé. Si on reprend notre exemple d'une structure de données comprenant 65 536 éléments, et que les éléments sont des données dans l'espace vectoriel R^{4096}, alors chaque élément a une taille approximative de 16 kilooctets, et la structure de données a une taille approximative de 1 gigaoctet. L'échange d'une telle structure de données entre émetteur et récepteur devient prohibitif. Si on fragmente la donnée sémantique en fragments de dimension 4, et qu'on garde la même taille pour la structure de données, les 65 536 éléments ne pèseront plus que 4 fois 32 bits (soit 4 réels encodés en *float32*), c'est-à-dire que la structure aura une taille approximative de 128 bits fois 65 536, soit 1 mégaoctet. La structure de données qui doit être stockée et éventuellement échangée a donc diminué de taille par un facteur 1000.

Par ailleurs, la fragmentation permet d'améliorer la précision des données communiquées, il est vrai aux dépens du nombre de symboles ou de bits transmis formant les index. Si on reprend l'exemple d'une donnée sémantique appartenant à l'espace vectoriel R^{4096} fragmentée en 1024 fragments de dimension 4, et que la structure de données utilisée compte 65 536 éléments de dimension 4, ces éléments serviront à quantifier des fragments de dimension 4 et non plus une donnée complète de dimension 4096. La donnée communiquée est alors bien plus précise, au détriment de la taille mémoire demandée par la transmission des index, puisqu'il faudra transmettre 1024 index de taille 16 bits et non plus un seul index de taille 16 bits. Le gain de transmission d'un index de taille 16 bits à la place d'une donnée vectorielle de dimension 4 (soit 4 réels encodés sur 32 bits, donc 128 bits) n'est donc plus que d'un facteur 8. La fragmentation permet donc de choisir entre une communication très économe (un index de 16 bits à la place d'une donnée de taille approximative 16 kilooctets, soit un facteur de gain de 1024) ou une communication beaucoup plus précise (65 536 éléments pour couvrir un espace de dimension 4 au lieu d'un espace de dimension 4096) avec un facteur de gain seulement de 8.

Il est possible de mélanger ces effets. Alors qu'une structure de données de taille 65 536 éléments peut être vu comme appropriée pour des données dans l'espace vectoriel R^{4096}, la fragmentation dans un

espace de dimension 4 peut permettre d'avoir des structures de données de taille 1024. Les index pour cette taille de structure seront alors de longueur 10 bits. Le facteur de gain par rapport à 4 réels encodés sur 32 bits est alors supérieur à 12. La précision donnée par 1024 éléments peut être bien suffisante pour représenter des fragments de donnée vectorielle de dimension 4.

Ces exemples numériques ne sont bien sûr que des modes de réalisation parmi d'autres.

Selon un mode de réalisation qui pourra être mis en œuvre cumulativement avec le mode précédent, la taille des fragments dans lesquels la donnée vectorielle à communiquer est fragmentée est choisie en fonction de la taille attendue des paquets de données pouvant être transmis entre le premier équipement et le deuxième équipement.

Dans ce mode de réalisation, la fragmentation permet d'adapter la taille des index transmis à la taille attendue des paquets qui peuvent être transmis par le réseau. Il est en effet avantageux de ne pas découper une donnée en plusieurs paquets. La taille des paquets qui seront transmis au travers du canal de communication entre les premiers et deuxièmes équipements est dénommée unité de transmission maximale (traduction de l'anglais *maximal transmission unit*, pour lequel on trouve l'acronyme MTU). En fonction du MTU attendu sur la transmission, une fragmentation donnée pourra être définie qui évitera que les index transmis soient découpés en plusieurs paquets.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, la structure de données est définie au préalable en fonction du format de la donnée vectorielle à communiquer.

Le fonctionnement du procédé implique que la structure de données soit disponible à la réception pour retrouver une donnée vectorielle correspondant à l'index reçu, qui sera proche de la donnée à communiquer par vertu de la distance mesurée entre données vectorielles et de la fonction d'indexation choisie. Il est possible de définir à l'avance une structure de données comprenant des données vectorielles représentatives des données à communiquer. Ces données vectorielles représentatives auront le même format que la donnée à communiquer et seront par exemple de la même dimension.

Une manière de procéder est de définir à l'avance une grille de points (les données vectorielles présentes dans la structure de données) qui couvre de façon appropriée l'espace vectoriel dans lequel les données à communiquer prendront des valeurs. Cette grille pourra être plus dense aux endroits de l'espace vectoriel dans lequel des données vectorielles à communiquer sont attendues afin d'assurer une représentation plus fine dans les endroits d'intérêt. On peut noter, qu'en général, les données vectorielles utilisées ou produites par des applications d'intelligence artificielle ont une grandeur réduite ; la valeur des réels composant une donnée vectorielle à communiquer est le plus souvent comprise entre -100 et 100. Il sera donc avantageux que les données vectorielles composant la structure de données soient majoritairement autour de 0, avec éventuellement des valeurs plus grandes pour capturer certaines données exceptionnelles.

Une telle structure de données pourra donc être définie à l'avance pour un format donné de données vectorielles. Les applications manipulant des données vectorielles d'un format donné pourront alors avoir en mémoire la structure de données correspondant à ce format précis. Le procédé pourra être alors utilisé pour toute tâche de communication de données vectorielles présentant un contenu sémantique, en utilisant la structure de données associée au format, et les applications réceptrices pourront disposer d'une information quant au format de données vectorielles qui leur permettra de retrouver les données vectorielles proches associées à un index reçu, ces données proches étant présentes dans la structure de données.

Cette structure de données définie pour un format précis de données vectorielles peut aussi être utilisé pour réaliser un stockage efficace de donnée. Un index est calculé pour une donnée vectorielle à mémoriser, qui correspond dans la structure de données à une donnée proche de la donnée vectorielle à mémoriser. C'est cet index qui est sauvegardé et qui, à la lecture ultérieure, donnera accès à la donnée proche de la donnée vectorielle à mémoriser. D'éventuelles erreurs de sauvegarde de l'index seront minimisées, comme pour les erreurs de transmission, par la nature de la fonction d'indexation choisie. Et la distorsion entre la donnée vectorielle à mémoriser et celle correspondant à l'index effectivement mémorisé sera également minimisée par le choix de la distance mesurée entre données vectorielles qui préserve la proximité entre contenus sémantiques associés aux données vectorielles.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, la structure de données est obtenue par apprentissage automatique en fonction des données vectorielles déjà communiquées et/ou destinées à être communiquées.

Dans ce mode de réalisation, la structure de données est apprise, par un procédé d'apprentissage automatique, afin de s'adapter au mieux à l'ensemble des données vectorielles déjà communiquées ou bien destinées à être communiquées. Plusieurs techniques d'apprentissage peuvent être utilisées, par exemple supervisé ou non, ou bien utilisant ou non une fonction de récompense dans le cas d'un apprentissage par renforcement. Une technique d'apprentissage automatique permettra d'avoir une plus grande densité de données représentatives dans la portion de l'espace vectoriel dans laquelle un plus grand nombre de données vectorielles à communiquer est attendu. Pour une même taille de la structure de données, ce mode de réalisation aura donc l'avantage d'une plus grande proximité des données communiquées avec les données à communiquer initiales puisque la grille de données communiquées est plus dense là où il en sera davantage besoin, alors qu'une structure de données générique, définie à l'avance, sera moins précise.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'apprentissage automatique de la structure de données utilise la méthode des k-moyennes où k est le nombre de données vectorielles présentes dans la structure de données.

Lorsque le mode de réalisation du procédé comprend un apprentissage automatique de la structure de données, l'objectif de l'apprentissage automatique est de réaliser un regroupement (traduction de l'anglais *clustering*) d'un ensemble de données communiquées ou à communiquer, dit ensemble

d'apprentissage, autour d'un nombre réduit de données vectorielles représentatives qui seront enregistrées dans la structure de données. Un tel regroupement peut être effectué grâce à la méthode des k-moyennes, où k est la taille de la structure de données. Grâce à ce mode de réalisation, le regroupement est uniforme, au sens que les données appartenant à l'ensemble d'apprentissage sont bien réparties de manière égalitaire en groupes dont le centre est une donnée vectorielle représentative qui se retrouvera dans la structure de données pour représenter toutes les données vectorielles du même regroupement.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, la fonction d'indexation est définie au préalable en fonction du format de la donnée vectorielle à communiquer.

Dans ce mode de réalisation, de manière similaire à ce qui a été vu précédemment pour la structure de données, c'est la fonction d'indexation qui est définie au préalable en fonction du format de la donnée vectorielle à communiquer. Pour un espace vectoriel donné dans lequel les données à communiquer vont prendre des valeurs, une fonction d'indexation disposant des propriétés attendues peut être définie une fois pour toutes. Cette fonction pourra alors être utilisée par défaut pour les données vectorielles du format attendu. Ce mode de réalisation a l'avantage, quand il est utilisé en commun avec le mode similaire relatif à la structure de données, de permettre des communications efficaces, sans traitement préalable, des données vectorielles d'un format donné. Ces communications peuvent également correspondre à des sauvegardes comme on l'a déjà vu.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, la fonction d'indexation est obtenue par apprentissage automatique en fonction des données vectorielles déjà communiquées et/ou destinées à être communiquées.

Ce mode de réalisation a l'avantage de permettre d'utiliser une fonction d'indexation particulièrement adaptée aux données vectorielles à communiquer. Par exemple, comme pour l'apprentissage de la structure de données, la fonction d'indexation apprise pourra être plus discriminante dans les zones de l'espace vectoriel où la densité de données vectorielles à communiquer est plus forte.

Certaines fonctions sont particulièrement adaptées à l'apprentissage automatique. C'est le cas des fonctions de hachage sensibles à la localité (fonctions LSH). Ces fonctions sont définies selon des hyperplans qui découpent en deux parties l'espace vectoriel dans lequel les données à communiquer prennent des valeurs. Une partie, située d'un côté de l'hyperplan, correspondra à la valeur 0 pour un bit de rang donné dans l'index calculé alors que l'autre partie, située de l'autre côté, correspondra à la valeur 1 pour le bit de même rang. Pour une longueur L d'index donné, un nombre égal d'hyperplans est fixé qui va découper l'espace vectoriel en 2 puissance L, si on considère que l'index est une suite de bits. Des calculs similaires peuvent être faits quand l'index est une suite d'autres symboles, en ramenant ces autres symboles à des bits.

Le placement de ces hyperplans, qui définissent une fonction d'indexation LSH, peut être obtenu par apprentissage selon la densité des données à transmettre dans l'espace vectoriel, afin de réaliser les découpes les plus adaptées.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le procédé comprend la transmission de valeurs de paramètres utiles à la communication de données vectorielles selon le procédé.

Le procédé selon l'invention fonctionne en supposant que l'entité qui reçoit l'index transmis connaisse au minimum la structure de données pour retrouver une donnée vectorielle correspondant à l'index reçu. D'autres valeurs de paramètres utiles au fonctionnement du procédé peuvent devoir être connues par l'entité qui se trouve en réception, par exemple la taille des fragments si la donnée vectorielle a été découpée en plusieurs fragments avant l'application du procédé.

Par exemple, quand la structure de données et la fonction d'indexation sont définies au préalable, comme on l'a vu faire dans plusieurs modes de réalisation, il suffit à l'émetteur de placer dans sa transmission de l'index obtenu des références à la structure de données et à la fonction qu'il va utiliser. Ces références peuvent être par exemple placées dans un paquet préambule à la transmission ou dans un en-tête de paquet de données. Le récepteur recevra ces valeurs de paramètres qui font référence à des fonctions d'indexation ou structures de données connues. Il saura alors quelle structure de données utiliser pour retrouver la donnée vectorielle proche de la donnée à transmettre grâce à l'index reçu. De telles structures de données et ensembles de paramètres pourraient être définis au préalable et standardisés pour des formats de données vectorielles utilisés de façon généralisée, tels ceux qui sont présentés dans la comparaison MTEB. La communication d'une donnée vectorielle selon le procédé comprendrait alors dans ce mode de réalisation la transmission d'une information de référence permettant à l'équipement recevant l'index associé à la donnée vectorielle communiquée quels sont les paramètres utilisés pour la communication.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le procédé comprend une phase préalable de négociation de valeurs de paramètres utiles à la communication de données vectorielles selon le procédé, lesdits paramètres pouvant être choisis dans la liste suivante non exhaustive :

1. Dimension de la donnée vectorielle ;
2. Dimension d'un fragment de la donnée vectorielle ;
3. Type de donnée utilisé comme type de base par une donnée vectorielle ;
4. Format de la donnée vectorielle ;
5. Identifiant du modèle produisant la donnée vectorielle ;
6. Taille du dictionnaire ;
7. Contenu du dictionnaire ;

8. Taille des index du dictionnaire ;
9. Symboles utilisés par les index ;
10. Fonction d'indexation utilisée.

Grâce à ce mode de réalisation, des possibilités plus poussées de communication entre l'émetteur et le récepteur sont mises en œuvre. On a vu dans le mode de réalisation précédent que des valeurs utiles à l'exécution du paramètre pouvaient être transmises, par exemple dans un en-tête de paquet, et permettre de retrouver des éléments déjà définis, comme la structure de données utilisée en réception pour retrouver la donnée vectorielle proche de la donnée à communiquer. Mais ce mode précédent ne suffit pas quand, par exemple, la structure de données et/ou la fonction d'indexation sont apprises par apprentissage automatique de façon spécialisée pour des données à communiquer. Le récepteur ne peut pas dans ce cas connaître *a priori* les éléments nécessaires au fonctionnement du procédé.

Dans ce mode de réalisation, une phase de négociation préalable à la communication permet à l'émetteur et au récepteur de se mettre d'accord sur l'ensemble des paramètres utiles à la communication de données vectorielles selon le procédé. Ce mode peut être combiné avec le précédent : par exemple, une première transmission peut donner les références à un codec qui va être utilisé, puis une négociation permet à l'émetteur et au récepteur de se mettre d'accord sur une fragmentation, qui peut dépendre du MTU attendu entre le premier équipement et le second équipement. Ce mode de réalisation propose une liste non exhaustive des éléments qui peuvent être négociés durant cette étape préalable de négociation.

Selon un deuxième aspect fonctionnel, l'invention a trait à un procédé de gestion de la réception d'une donnée vectorielle caractérisé en ce qu'il comprend la réception d'un index, dit index reçu, suivie par la lecture d'une donnée vectorielle dans une structure de données associant des index à des données vectorielles en fonction de l'index reçu.

Ce deuxième aspect fonctionnel de l'invention correspond à la réception de l'index transmis à la place de la donnée vectorielle à communiquer. On parle ici d'index reçu, car il est possible que des erreurs de transmission aient modifié l'index durant la transmission et que l'index reçu ne soit pas égal à l'index transmis. Une fois l'index reçu, il est possible de retrouver dans la structure de données une donnée vectorielle à partir de l'index reçu. Cette donnée vectorielle lue sera considérée comme la donnée vectorielle devant être communiquée au départ.

Selon un premier mode de réalisation de ce deuxième aspect fonctionnel, le procédé comprend, si l'index reçu ne correspond à aucun index présent dans la structure de données, une recherche dans la structure de données de l'index le plus proche de l'index reçu et la donnée vectorielle lue dans la structure de données est celle qui correspond au dit index le plus proche.

Selon un autre mode de réalisation de ce deuxième aspect fonctionnel, qui pourra être mis en œuvre alternativement ou cumulativement avec le mode précédent, la proximité entre l'index reçu et les index présents dans la structure de données est définie par la distance de Hamming.

Comme déjà indiqué précédemment, les erreurs de transmission peuvent faire que l'index reçu soit différent de l'index transmis et en particulier, que l'index reçu ne soit pas présent dans la structure de données. Dans ce cas, un mode de réalisation avantageux est de chercher l'index le plus proche qui soit présent dans la structure de données afin de bien obtenir pour finir une donnée vectorielle.

Si l'on récapitule, le procédé selon l'invention peut accumuler les distorsions suivantes dans la transmission d'une donnée vectorielle :

- En général, la donnée vectorielle à communiquer n'est pas présente dans la structure de données. L'index calculé va correspondre directement ou indirectement à une donnée vectorielle proche présente dans la structure de données, appelée donnée vectorielle communiquée.
    - Dans certains cas, il y a autant de données vectorielles dans la structure de données que de valeurs d'index possibles. L'index calculé sera dans ce cas associé directement à une donnée vectorielle.
    - Dans d'autres cas, il y a moins de données vectorielles dans la structure de données que de valeurs d'index possibles. Alors, l'index calculé n'est pas nécessairement associé directement à une donnée vectorielle présente dans la structure de données. Mais il est possible de trouver dans la structure de données l'index le plus proche de l'index calculé. Une donnée vectorielle associée à cet index le plus proche peut alors être trouvée dans la structure de données.
    - Dans tous les cas, l'index calculé correspond donc directement ou indirectement à une donnée vectorielle présente dans la structure de données. La fonction d'indexation et la proximité entre index sont définies pour que cette donnée vectorielle transmise soit proche de la donnée à transmettre selon une distance mesurée qui préserve les contenus sémantiques associés aux données vectorielles, mais il s'agit quand même ici d'une première distorsion.
- L'index calculé est transmis qu'il soit associé ou pas à une donnée sémantique dans la structure de données. Il peut être modifié par le canal de transmission et l'index reçu peut être différent de l'index transmis.
    - Dans certains cas, il y a autant de données vectorielles dans la structure de données que de valeurs d'index possibles. L'index reçu sera dans ce cas associé directement à une donnée vectorielle, qui pourra être différente de la donnée vectorielle communiquée. Il s'agit ici d'une deuxième distorsion.
    - Dans d'autres cas, il y a moins de données vectorielles dans la structure de données que de valeurs d'index possibles. Alors, l'index reçu n'est pas nécessairement associé directement à une donnée vectorielle présente dans la structure de données. Mais il est possible de trouver dans la structure de données l'index le plus proche de l'index reçu. Cet index le plus proche peut être l'index transmis : dans ce cas, on retombe sur l'index qu'on voulait transmettre et la distorsion due aux erreurs de transmission est annulée. Ou bien l'index le plus proche est différent de l'index transmis : dans ce cas,

une autre donnée vectorielle que la donnée communiquée est associée, et on a ici une deuxième distorsion.

Le procédé selon l'invention doit donc permettre de choisir avec attention la fonction d'indexation, la proximité entre index, le fait que la distance mesurée entre données vectorielles préserve bien les contenus sémantiques associés aux données vectorielles, le nombre de données représentatives présentes dans la structure de données, la concentration de ces données représentatives dans les zones les plus riches en données à communiquer de l'espace vectoriel dans lequel les données prennent des valeurs et tout autre paramètre utile afin de minimiser ces distorsions. Comme indiqué, le choix de la similarité cosinus, de la distance de Hamming et de fonctions de hachage sensibles à la localité pour la similarité cosinus sont des exemples de paramètres de fonctionnement du procédé selon l'invention qui donnent de premiers bons résultats dans nos expériences, mais le procédé selon l'invention n'est pas limité à ces exemples de réalisations et couvre l'utilisation de nombreux autres paramètres possibles.

Selon un autre mode de réalisation du deuxième aspect fonctionnel de l'invention, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le procédé comprend, si l'index reçu correspond à un fragment d'une donnée vectorielle, la reconstitution d'une donnée vectorielle complète à partir des fragments de données vectorielles correspondant aux index reçus.

Grâce à ce mode de réalisation, le procédé selon l'invention permet de gérer la fragmentation des données vectorielles communiquées. Il faut simplement s'assurer de reconstituer la donnée vectorielle complète à partir des fragments de données vectorielles correspondants aux différents index reçus. Cette reconstitution permet de compléter le fonctionnement du procédé selon l'invention.

Selon un premier aspect matériel, l'invention se rapporte à un gestionnaire de communication apte à gérer la communication d'une donnée vectorielle, dite donnée à communiquer, depuis un premier équipement à un deuxième équipement, le gestionnaire de communication comprenant :

- Un module d'obtention d'index, le module d'obtention étant apte à obtenir un index, dit index obtenu, en appliquant une fonction d'indexation à la donnée à communiquer, la fonction d'indexation calculant un index à partir d'une donnée vectorielle, la fonction d'indexation étant telle que, plus deux données vectorielles sont proches selon une distance mesurée entre elles, plus les index calculés pour ces deux données vectorielles par la fonction d'indexation sont proches entre eux, ledit index obtenu permettant de retrouver dans une structure de données une autre donnée vectorielle, dite donnée communiquée, la structure de données associant des index à des données vectorielles, le module d'obtention d'index étant apte à fournir un index à un transmetteur d'index apte à transmettre un index depuis le premier équipement au deuxième équipement.

Dans un mode de réalisation, le gestionnaire de communication comprend de surcroît un transmetteur de l'index obtenu depuis le premier équipement au deuxième équipement.

Le gestionnaire de communication peut également être dénommé comme une entité de gestion apte à gérer la communication d'une donnée.

Selon un premier mode de réalisation de cet aspect matériel, l'invention se rapporte à un équipement de télécommunications comprenant un gestionnaire de communication apte à gérer la communication d'une donnée vectorielle selon l'invention.

Un tel équipement de télécommunications peut être par exemple un routeur, ou bien une passerelle domestique ou dans un environnement professionnel, ou bien une carte réseau présente dans un équipement informatique tel qu'un ordinateur, ou bien un terminal mobile, ou bien un objet communicant tel qu'une caméra ou autre objet produisant des données vectorielles pour des applications d'intelligence artificielle ou pas. En général, un gestionnaire de communication selon l'invention peut être embarqué dans tout type d'objet ou d'équipement produisant des données vectorielles à destination d'un récepteur distant ou bien ayant à réaliser la communication de telles données à destination d'un autre équipement dans un réseau informatique ou de télécommunications.

Selon un deuxième aspect matériel, l'invention se rapporte à un gestionnaire de réception de données vectorielles comprenant :

- Un module de réception d'un index, dit index reçu ;
- Un lecteur d'une donnée vectorielle dans une structure de données associant des index à des données vectorielles en fonction de l'index reçu.

Le gestionnaire de réception peut également être dénommé comme une entité de gestion apte à gérer la réception d'une donnée.

Selon un premier mode de réalisation de cet aspect matériel, l'invention se rapporte à un équipement de télécommunications comprenant un gestionnaire de réception apte à gérer la réception d'une donnée vectorielle selon l'invention.

Un tel équipement de télécommunications peut être par exemple un routeur, ou bien une passerelle domestique ou dans un environnement professionnel, ou bien une carte réseau présente dans un équipement informatique tel qu'un ordinateur, ou bien un terminal mobile, ou bien un équipement gérant la réception de données dans un système d'informatique en nuage. En général, un gestionnaire de réception selon l'invention peut être embarqué dans tout type d'objet ou d'équipement recevant des données vectorielles pour sauvegarde ou traitement par des applications d'intelligence artificielle ou autres ou bien ayant à réaliser la réception de telles données dans un réseau informatique ou de télécommunications.

Dans des modes de réalisation, les gestionnaires de communication et de réception d'une donnée vectorielle sont intégrées dans un seul équipement. Dans certains de ces modes de réalisation, les équipements en question peuvent être des routeurs réalisant à la fois la communication vers d'autres équipements et la réception en provenance d'autres équipements de données vectorielles. Dans

d'autres modes de réalisation, les gestionnaires de communication et de réception correspondent à des composants distincts d'un même équipement. Le gestionnaire de communication peut correspondre à un composant produisant des données vectorielles et le gestionnaire de réception à un composant de mémorisation et de lecture des données vectorielles produites. Le canal de transmission entre les équipements dans ce cas est un bus logiciel qui relie entre eux les deux composants d'un même équipement.

Selon un autre aspect matériel, l'invention a trait à un programme d'ordinateur apte à être mis en œuvre par un gestionnaire de communication, le programme comprenant des instructions de code qui, lorsqu'il est exécuté par un processeur, réalise les étapes du procédé de gestion de la communication d'une donnée vectorielle défini ci-dessus.

Selon un autre aspect matériel, l'invention a trait à un programme d'ordinateur apte à être mis en œuvre par un gestionnaire de réception, le programme comprenant des instructions de code qui, lorsqu'il est exécuté par un processeur, réalise les étapes du procédé de gestion de la réception d'une donnée vectorielle défini ci-dessus.

Enfin, selon un autre aspect matériel, l'invention a trait à des supports de données sur lesquels sont enregistrés des programmes d'ordinateurs comprenant des séquences d'instructions pour la mise en œuvre des procédés de gestion de la communication et de la réception d'une donnée vectorielle définis ci-dessus.

Les supports de données peuvent être n'importe quelle entité ou dispositif capable de stocker les programmes. Par exemple, les supports peuvent comporter un moyen de stockage, tel qu'une ROM, par exemple un CD ROM ou une ROM de circuit microélectronique, ou encore un moyen d'enregistrement magnétique tel qu'un disque dur. D'autre part, les supports peuvent être des supports transmissibles tels qu'un signal électrique ou optique, qui peuvent être acheminés via un câble électrique ou optique, par radio ou par d'autres moyens. Les programmes selon l'invention peuvent être en particulier téléchargés sur un réseau de type Internet. Alternativement, le support d'informations peut être un circuit intégré dans lequel le programme est incorporé, le circuit étant adapté pour exécuter ou pour être utilisé dans l'exécution du procédé en question. Le programme selon l'invention peut utiliser tout type de technologie informatique en termes de langages de programmation compilés, interprétés, ou une combinaison des deux ainsi qu'en termes de systèmes d'exploitation.

Brève description des figures

L'invention sera mieux comprise à la lecture de la description qui suit, donnée à titre d'exemple, et faite en référence aux dessins annexés sur lesquels :

[Fig 1] représente un gestionnaire de communication et un gestionnaire de réception d'une donnée vectorielle, comprises dans deux équipements, illustrant un exemple de réalisation de l'invention.

[Fig 2] illustre un exemple d'étapes mises en œuvre dans le cadre d'un mode de réalisation de l'invention.

[Fig 3] illustre des exemples de distorsions possibles subies par une donnée vectorielle lors de sa communication et de sa réception.

[Fig 4] illustre des résultats d'expérimentation obtenus lors du développement des procédés selon l'invention.

Description détaillée

La figure 1 représente un gestionnaire de communication 100 gérant la communication d'une donnée vectorielle SD depuis un équipement EQ1 vers un équipement EQ2, à travers un canal de communication CNL. Un gestionnaire de réception 200 gère la réception d'une donnée vectorielle SD'.

Dans l'exemple de réalisation présenté dans la figure 1, l'entité de gestion 100 comprend les modules suivants :

- Un module 101 d'obtention d'un index I', le module d'obtention 101 étant apte à obtenir un index I' en appliquant une fonction d'indexation F à la donnée à communiquer SD, la fonction d'indexation F calculant un index à partir d'une donnée vectorielle, la fonction d'indexation F étant telle que, plus deux données vectorielles sont proches selon une distance D mesurée entre elles, plus les index calculés pour ces deux données vectorielles par la fonction F sont proches entre eux, ledit index I' permettant de retrouver dans une structure de données DS une autre donnée vectorielle SD', dite donnée communiquée, la structure de données DS associant des index à des données vectorielles ;
- Un transmetteur 102 de transmission de l'index I' obtenu au travers d'un canal de communication CNL entre le premier équipement EQ1 et le deuxième équipement EQ2.

Dans l'exemple de réalisation présenté dans la figure 1, l'index I' est obtenu à partir de la donnée sémantique SD en utilisant à un moment une fonction d'indexation F, la fonction d'indexation F étant telle que, plus deux données vectorielles sont proches selon la distance D mesurée entre elles, plus les index calculés pour ces deux données vectorielles par la fonction F sont proches entre eux.

Dans l'exemple de réalisation présenté dans la figure 1, l'application de la fonction F à la donnée vectorielle SD produit directement un index I' tel qu'il existe une donnée vectorielle SD' associée à l'index I' dans la structure de données DS. Dans d'autres exemples de réalisation, la fonction d'indexation F calcule à partir de la donnée SD un index I qui n'est pas présent dans la structure de données DS et un index I' proche de l'index I est obtenu, une donnée vectorielle étant associée à I' dans la structure DS. Dans d'autres exemples encore, la fonction d'indexation calcule à partir de la donnée vectorielle SD un indice I et cet indice est transmis même s'il n'est pas présent dans la structure de

données DS. C'est à la réception que l'indice le plus proche présent dans la structure de données sera recherché afin de trouver une donnée vectorielle communiquée correspondant à l'index transmis. Dans tous les cas, l'index I' obtenu puis transmis permet de retrouver directement ou indirectement une donnée vectorielle SD' dans la structure de données DS, qui sera considérée comme étant la donnée vectorielle communiquée à la place de la donnée SD à communiquer.

Dans l'exemple de réalisation présenté dans la figure 1, l'entité de gestion 200 comprend :

- Un module 201 de réception d'un index I', dit index reçu ;
- Un lecteur 202 d'une donnée vectorielle SD' dans une structure de données DS associant des index à des données vectorielles en fonction de l'index reçu I'.

Au lieu de transmettre directement les symboles formant la donnée vectorielle SD, l'entité de gestion 100 obtient un index I' en utilisant une fonction d'indexation F et le transmet, ce qui économise fortement la taille de données à transmettre.

La transmission des symboles formant l'index I' se fait dans l'exemple de la figure 1 à travers un canal de communication CNL. Des erreurs de transmission peuvent avoir lieu durant la transmission. Cependant, dans l'exemple présenté dans la figure 1, l'index reçu I' est égal à l'index calculé I'. On verra en figure 3 des exemples dans lesquels des erreurs de transmission apparaissent. L'entité de gestion 200 peut lire dans une structure de données DS une donnée vectorielle SD' correspondant à l'index reçu I'. Cette donnée vectorielle SD' sera ensuite utilisée dans des applications ultérieures d'intelligence artificielle ou autres à la place de la donnée à transmettre SD. La donnée vectorielle SD' a donc été communiquée depuis l'équipement EQ1 vers l'équipement EQ2 à la place de la donnée vectorielle SD qui était la donnée à communiquer.

Dans l'exemple présenté dans la figure 1, la seule distorsion causée par l'utilisation du procédé selon l'invention correspond à l'utilisation de l'index calculé I', associé à la donnée vectorielle SD', présent dans la structure de données DS. Nous présentons en figure 3 l'ensemble des distorsions pouvant être introduites par le procédé selon l'invention.

Le gestionnaire de communication 100 compris dans l'équipement EQ1 présente l'architecture matérielle d'un ordinateur conventionnel. Il comporte notamment un processeur, une mémoire vive de type RAM et une mémoire morte telle qu'une mémoire de type Flash, ROM, (non représentés sur la figure) ainsi que des dispositifs d'entrée-sortie tels que, dans certains cas, des claviers et/ou écrans (non représentés sur la figure), et des ports réseaux permettant de communiquer avec d'autres entités et serveurs grâce à un réseau de communication tel qu'Internet non représenté sur la figure.

Le gestionnaire de réception 200 compris dans l'équipement EQ2 présente également l'architecture matérielle d'un ordinateur conventionnel. Il comporte notamment un processeur, une mémoire vive de type RAM et une mémoire morte telle qu'une mémoire de type Flash, ROM, (non représentés sur la figure) ainsi que des dispositifs d'entrée-sortie tels que, dans certains cas, des claviers et/ou écrans

(non représentés sur la figure), et des ports réseaux permettant de communiquer avec d'autres entités et serveurs grâce à un réseau de communication tel qu'Internet non représenté sur la figure.

Les gestionnaires 100 et 200 peuvent être inclus par exemple dans des équipements de télécommunications. De tels équipements EQ1, EQ2 peuvent être des routeurs, des passerelles domestiques ou déployées dans des environnements professionnels. Les entités de gestion 100 et 200 peuvent être incluses dans des cartes réseau présentes dans des équipements informatiques EQ1, EQ2 tels que des ordinateurs ou bien des équipements EQ1, EQ2 gérant la réception ou l'émission de données dans des systèmes informatiques en nuage. Les gestionnaires 100 et 200 peuvent également être inclus dans des terminaux mobiles EQ1, EQ2 où ils serviront respectivement à la communication ou à la réception de données vectorielles à travers des réseaux de télécommunications sans fil 3G, 4G, 5G, 6G ou autres auxquels sont connectés les terminaux mobiles EQ1, EQ2. En général, un gestionnaire 100 ou 200 selon l'invention peut être embarqué dans tout type d'objet ou d'équipement EQ1, EQ2 produisant des données vectorielles SD pour sauvegarde ou traitement par des applications d'intelligence artificielle ou autres ou bien ayant à réaliser la réception de telles données dans un réseau informatique ou de télécommunications.

Dans un autre domaine technique, les gestionnaires 100 et 200 peuvent être inclus dans des capteurs d'images et de sons et dans tous types d'équipements EQ1, EQ2, par exemple des moyens de transport ou dans des équipements EQ1, EQ2 robotiques. Des capteurs d'images ou de sons peuvent encoder des données vectorielles à partir des images ou sons captés. De tels capteurs (caméras, micros) peuvent inclure un gestionnaire 100 de communication d'une donnée vectorielle. Des gestionnaires 200 peuvent être déployés dans des serveurs de traitement qui vont recevoir les données vectorielles transmises et agir à partir d'elles. Ces gestionnaires 200 peuvent par exemple se retrouver dans des serveurs embarqués dans des moyens de transport où ils serviront à l'interprétation de données transmises par les capteurs par l'intermédiaire des gestionnaires 100 de la communication de données vectorielles. Cette organisation des gestionnaires 100 et 200 peut être utile dans le cadre des moyens de transport autonomes, ou des robots, dans lesquels des capteurs traitent des images et sons captés, produisent des données vectorielles, et ces données vectorielles sont adressées pour interprétation et action conséquentes à des serveurs de contrôle. Dans ce cadre, les gestionnaires 100 et 200 peuvent être inclus dans un même équipement.

Dans un autre domaine, les gestionnaires 100 et 200 peuvent appartenir à un même équipement et gérer la mémorisation d'une donnée vectorielle. Dans ce cadre, le gestionnaire 100 obtient un index I' à partir d'une donnée vectorielle à mémoriser SD et c'est cet index I' qui est mémorisé. Le gestionnaire 200 récupère l'index I' mémorisé et lit la donnée vectorielle SD' correspondante dans la structure de données DS. La donnée vectorielle SD' est donc finalement la donnée qui a été mémorisée à la place de la donnée vectorielle SD à mémoriser.

Le canal de communication CNL est une liaison entre les gestionnaires 100 et 200. Le canal CNL peut être un segment d'un réseau de télécommunications ou bien au contraire correspondre à l'enchaînement de plusieurs réseaux de communication permettant de relier le gestionnaire 100 au

gestionnaire 200. Le canal CNL peut aussi être une liaison à l'intérieur d'un équipement (moyen de transport autonome, robot, équipement informatique en général) entre des capteurs (caméras, micros) et des serveurs de traitement ou bien entre un composant produisant des données sémantiques et les mémorisant et un composant les récupérant. Dans ces cas, le canal CNL peut être un bus de données à l'intérieur d'un équipement informatique. Les technologies de communications déployées dans le canal CNL peuvent être des technologies filaires, optiques, satellitaires ou sans fil ou une combinaison de ces différentes technologies.

Le canal CNL peut réaliser une liaison directe entre le gestionnaire 100 et le gestionnaire 200. Par exemple, le gestionnaire 100 peut être inclus dans une carte réseau d'un ordinateur personnel ; le gestionnaire 200 peut être inclus dans une passerelle domestique déployant un réseau local avec la technologie WiFi ; et le canal CNL peut être dans ce cas la liaison WiFi entre l'ordinateur incluant le gestionnaire 100 et la passerelle incluant le gestionnaire 200. Ou bien, le gestionnaire 100 peut être inclus dans un téléphone mobile utilisant la téléphonie sans fil de technologie 3G, 4G, 5G, 6G ou autre et le gestionnaire 200 peut être inclus dans une station de base déployant la technologie correspondante et le canal CNL sera la liaison sans fil entre le téléphone mobile incluant le gestionnaire 100 et la station de base incluant le gestionnaire 200. Ou bien le gestionnaire 100 est associé à une caméra, un micro ou un système radar ou lidar et adresse des données vectorielles au gestionnaire 200 inclus dans un serveur de traitement embarqué dans un véhicule autonome ou un robot par l'intermédiaire d'une liaison interne en câble coaxial ou fibre optique, qui forme le canal CNL. Ou bien le canal CNL est un bus logiciel ou de données reliant des composants d'un équipement informatique.

Dans d'autres exemples, le canal CNL va réaliser une liaison indirecte entre le gestionnaire 100 et le gestionnaire 200. Par exemple, les gestionnaires 100 et 200 peuvent être inclus dans des routeurs EQ1, EQ2 présents dans des réseaux à commutation de paquets, qui peuvent être par exemple des routeurs IP (acronyme de l'anglais *Internet Protocol*). Le canal CNL pourra être dans cet exemple un chemin selon le protocole IP entre les deux routeurs EQ1, EQ2 qui incluent les gestionnaires 100 et 200. Dans cet exemple de réalisation, le canal CNL pourra donc emprunter plusieurs segments de réseaux IP et plusieurs routeurs IP et sera alors une liaison indirecte entre les deux gestionnaires 100 et 200.

Le canal CNL peut être vu, dans des modes de réalisation, comme transmettant des suites de bits, c'est-à-dire des suites de symboles binaires 0 ou 1 entre les deux équipements EQ1, EQ2 incluant les deux gestionnaires 100 et 200. Mais au niveau de la transmission physique, le canal CNL va transmettre des symboles physiques qui peuvent prendre plus de valeurs que seulement deux valeurs. Par exemple, dans le domaine des communications sans fil, les symboles transmis par le canal CNL peuvent être des modulations de phase portées par une onde hertzienne, et ces modulations de phase peuvent prendre jusqu'à seize ou plus valeurs distinctes. De même, dans le domaine des liaisons par fibre optique ou support cuivre, les symboles transmis par le canal CNL seront des variations d'ondes électro-magnétiques portées par les supports physiques.

L'index calculé I' qui est transmis par le gestionnaire 100 à travers le canal CNL sera donc dans certains modes de réalisation une suite de bits 0 ou 1 mais sera, dans d'autres modes de réalisation, une suite de symboles correspondant aux symboles pouvant être transmis dans le canal CNL.

Le canal CNL introduit des erreurs dans la transmission. L'index reçu I' est donc, en toute généralité, différent de l'index calculé I' qui a été transmis. Dans l'exemple présenté dans la figure 1, ces deux index sont identiques.

Le gestionnaire 200 dispose d'une structure de données DS qui permet de faire correspondre à l'index reçu I' une donnée vectorielle SD'. Cette donnée vectorielle SD' est alors mise à disposition par le gestionnaire 200 et utilisée par des applications d'intelligence artificielle ou autres à la place de la donnée vectorielle SD initiale. Au lieu de communiquer la donnée vectorielle SD complète en transmettant l'ensemble des symboles ou bits la constituant, le gestionnaire 100 n'a eu à transmettre que l'index I', ce qui représente une grande économie de taille de donnée transmise.

La structure de données DS peut être de tout type. Il peut s'agir par exemple d'une table clé-valeurs, ou d'un dictionnaire, associant des index I, I' à des données vectorielles respectives SD, SD'. La structure de données DS peut aussi être une base de données où les index I, I' sont associés aux données vectorielles respectives SD, SD'. La structure de données DS peut être sauvegardée en mémoire vive ou morte par l'entité de gestion 200 ou bien requêtée à la demande lors d'une réception d'un index I' pour retrouver la donnée vectorielle associée SD'.

Les données vectorielles SD, SD' sont des données obtenues en transformant des données telles que des textes ou des images dans un modèle. Le contenu sémantique des données textuelles ou imagées est alors associé aux données vectorielles obtenues. Le fait que les données vectorielles SD, SD' préservent les contenus sémantiques de données initiales qui leur sont associées les rend particulièrement adaptées aux applications d'intelligence artificielle générative ou autres. Les données vectorielles SD, SD' suivent en général un format donné qui va leur permettre d'encoder, d'intégrer ou de représenter plus ou moins d'informations sémantiques. Des formats possibles sont recensés sur le site MTEB (acronyme de l'anglais *Massive Text Embedding Benchmark*) pour des données vectorielles SD, SD' qui représentent des textes et leurs performances dans différentes tâches d'intelligence artificielle sont présentées. Un format performant est par exemple le format Mistral-7B, de la société Mistral AI. Les données vectorielles SD, SD' dans ce format sont des vecteurs de 4096 réels, c'est-à-dire des éléments de l'espace vectoriel R^{4096}. De façon équivalente, on peut dire que les données vectorielles SD, SD' sont des points appartenant à l'espace cartésien de même dimension R^{4096}. D'autres formats peuvent être utilisés, par exemple de dimension 1024.

La taille d'une donnée vectorielle SD selon le format Mistral-7B est donc de 4096 fois les 32 bits permettant d'encoder un nombre réel dans le format *float32*. Si on remplace la communication d'une donnée vectorielle SD par la transmission d'un index de longueur 32 bits, le gain en taille de donnée transmise est alors un facteur 4096. D'autres gains plus faibles ou plus forts peuvent être obtenus selon

les choix d'indexation retenus durant la mise en œuvre d'un mode de réalisation de l'invention possible parmi tout un ensemble de modes de réalisation possibles.

La transmission d'un index I' à la place de la transmission des symboles ou bits formant la donnée vectorielle SD, puis les erreurs de transmission causées par le canal CNL, et enfin la lecture d'une donnée SD' correspondant à l'index reçu I' impliquent des distorsions par rapport à la donnée vectorielle à communiquer SD. Ces distorsions sont présentées plus en détail dans la figure 3. Dans l'exemple de la figure 1, une fonction d'indexation F est choisie de manière à rendre minimales ces distorsions selon une distance D mesurée entre données vectorielles. Dans la figure 1, ceci est représenté par une distance D minimale entre la donnée vectorielle SD et la donnée vectorielle SD'.

La distance D est mesurée par une fonction mesurant la proximité entre deux données vectorielles SD et SD'. Cette fonction peut être une distance, au sens mathématique du terme, dans l'espace vectoriel auquel appartiennent les données vectorielles SD, SD'. Dans des modes de réalisation, la distance D peut être par exemple la distance euclidienne entre points de l'espace cartésien associé à l'espace vectoriel des données vectorielles SD, SD'. Dans d'autres modes de réalisation, la distance D peut être obtenue en calculant la similarité cosinus entre vecteurs de l'espace vectoriel des données SD, SD'. D'autres distances peuvent être définies qui seront spécifiques aux tâches ultérieures qui vont utiliser les données vectorielles SD, SD' afin que la proximité entre données vectorielles SD, SD' mesurée par la distance D soit en relation directe avec la proximité des contenus sémantiques associés aux données elles-mêmes dans la tâche ultérieure qui les utilise.

Dans des modes de réalisation, la distance D peut être apprise à partir d'un ensemble d'apprentissage représentatif des données vectorielles SD, SD' qui devront être transmises. En effet, si les données vectorielles SD à transmettre sont plutôt concentrées dans une zone précise de l'espace vectoriel auquel elles appartiennent, il peut être pertinent que la fonction mesurant la distance D soit plus discriminante dans cette zone, pour prendre en compte des différences entre données concentrées dans cette zone, et moins discriminante en dehors de cette zone puisqu'on s'attend à y trouver moins de données vectorielles SD.

Nous rappelons que l'index I' est choisi en fonction d'une distance D mesurée entre la donnée vectorielle à communiquer SD et celle communiquée SD' et que, dans des modes de réalisation, cette distance D mesurée est calculée par une fonction préservant la proximité sémantique entre les contenus sémantiques associés aux deux données vectorielles SD, SD'.

Dans l'exemple représenté dans la figure 1, la fonction F d'indexation est choisie de façon à assurer une distorsion minimale entre la donnée vectorielle à transmettre SD et la donnée transmise SD'. De plus, selon l'invention, la fonction d'indexation F est choisie de telle manière que pour deux données vectorielles SD, SD' distinctes, plus ces données SD, SD' sont proches entre elles selon la distance D mesurée entre elles, plus les index calculés I, I' pour ces deux données vectorielles SD, SD' sont proches entre eux. La proximité entre index calculés peut être déterminée par la distance de Hamming ou par une autre méthode de calcul appropriée. Un intérêt de ce choix est d'assurer une certaine

robustesse du procédé en présence d'erreurs de transmission dans le canal CNL qui affectent l'index transmis.

La fonction F calcule un index de taille limitée à au maximum quelques bits (ou symboles dans le cas où le canal CNL transmet sur sa couche physique des symboles différents) à partir d'une donnée vectorielle formée de plusieurs milliers de réels. On utilisera par exemple des index de longueur 10 bits, ce qui donne 1024 valeurs d'index possibles, ou 16 bits, ce qui donne 65 536 valeurs d'index possibles. Ces index sont associés dans une structure de données DS à des données vectorielles SD, SD'. Au vu de la taille en nombre de bits des données vectorielles (16 kilooctets pour une donnée vectorielle SD de taille 4096 réels encodés en *float 32*), si la structure de données DS comprend 65 536 données associées à toutes les valeurs possibles prises par un index de longueur 16 bits, elle aura une taille de 1 gigaoctet, ce qui pourrait être vu comme une taille maximale.

Il est donc clair que le nombre de valeurs possibles pour les index est très petit par rapport au nombre de valeurs possibles pour les données vectorielles.

Par ailleurs, il n'est pas obligatoire qu'il existe dans la structure de données DS une donnée vectorielle associée à toutes les valeurs possibles d'index. L'index calculé I' à partir de la donnée vectorielle SD peut donc être obtenu de deux manières :

- Ou bien un premier index I est initialement calculé à partir de la donnée vectorielle SD en utilisant la fonction d'indexation F, mais cet index n'est pas présent dans la structure de données DS. Deux cas peuvent alors se présenter selon les modes de réalisation :
    - Le procédé peut chercher dans la structure de données DS un index I' proche de l'index initialement calculé I et c'est cet index I' qui sera l'index obtenu par le module 101 et transmis finalement par le module 102 dans le canal CNL. La proximité entre index pourra par exemple être défini par la distance de Hamming et devra en tout cas assurer la proximité entre la donnée à transmettre SD et la donnée transmise SD' associée à l'index calculé I' dans la structure de données DS.
    - Ou bien l'index I obtenu à partir de la donnée vectorielle SD n'est pas présent dans la structure de données DS mais est cependant transmis par le module 102 dans le canal CNL. C'est l'entité de gestion 200 de la réception qui devra chercher à partir de l'index reçu (I, dans le cas où le canal CNL n'introduit pas d'erreurs de transmission) un index proche présent dans la structure de données DS, ce qui permettra de lire dans la structure de données DS une donnée vectorielle associée à cet index proche.
- Ou bien l'index obtenu à partir de la donnée vectorielle SD est présent dans la structure de données DS, par exemple parce que celle-ci couvre l'ensemble de toutes les valeurs que peut prendre un index de longueur donnée. Dans ce cas, le procédé transmet l'index obtenu I' mais celui-ci correspond, selon toute probabilité, dans la structure de données DS, à une donnée vectorielle SD' différente de SD. La fonction d'indexation F ou un autre

mode d'obtention de l'index I' devra alors assurer la proximité entre la donnée vectorielle SD à communiquer et la donnée vectorielle SD' effectivement communiquée par l'intermédiaire de l'index obtenu I'.

Dans des modes de réalisation, la fonction d'indexation F est choisie de telle manière que, pour deux données vectorielles SD, SD' distinctes, si la distance donnée D entre ces deux données SD, SD' est inférieure à un seuil donné, alors la probabilité que les index calculés I, I' par la fonction d'indexation F pour ces deux données vectorielles SD, SD' soient les mêmes est supérieure à une probabilité donnée. Le choix d'une fonction d'indexation F disposant de cette propriété permet de renforcer les chances que deux données vectorielles proches selon la distance D mesurée entre elles auront le même index calculé.

Une fonction d'indexation F disposant de ces propriétés est donnée par la famille des fonctions de hachage sensibles à la localité (dites fonctions LSH, de l'acronyme anglais *locality-sensitive hashing*). Ces fonctions produisent des index proches entre eux, voire égaux, pour des éléments de départ proches entre eux pour une distance D. D'autres exemples de fonctions disposant de cette propriété peuvent être utilisés dans différents modes de réalisation.

La fonction d'indexation F choisie peut être une fonction LSH pour la similarité cosinus. Dans d'autres modes de réalisation, il pourra s'agir d'une fonction LSH pour la distance euclidienne.

Dans des modes de réalisation, le calcul de l'index I' est réalisé en cherchant dans la structure de données DS la donnée vectorielle SD' la plus proche de la donnée à transmettre SD selon la distance D mesurée. De cette manière, il est assuré que la donnée transmise SD' par l'intermédiaire de l'index calculé I' est bien la donnée vectorielle SD' la plus proche parmi les données possibles présentes dans la structure de données DS.

Ici, les index présents dans la structure de donnée DS ont été obtenus au préalable pour un ensemble de données vectorielles présentes dans la structure de données DS qui réalisent une couverture appropriée des valeurs qui peuvent être prises par les données vectorielles SD à communiquer dans l'espace vectoriel auquel elles appartiennent. De cette manière, il n'est pas nécessaire de calculer pour chaque donnée vectorielle SD à transmettre un index I' mais simplement de retrouver dans la structure de données DS la donnée vectorielle SD' la plus proche de la donnée à communiquer SD selon la distance D, puis de transmettre cet index I' trouvé dans la structure de données DS.

La structure de données DS réalise donc une quantification de l'espace vectoriel dans lequel les données vectorielles SD, SD' à communiquer peuvent prendre des valeurs. Cette quantification permet de transmettre, à la place des bits ou symboles constituant une donnée vectorielle SD complète, un index calculé I' par la fonction F, index qui est associé dans la structure de données DS à une donnée vectorielle SD' proche de la donnée vectorielle SD selon la distance D.

En général, les procédés selon l'invention fonctionnent quand la structure de données DS est connue à la fois par les gestionnaires de la communication 100 et de la réception 200. Ce n'est pas une obligation :

le gestionnaire 100 de la communication peut se contenter de calculer un index I en utilisant la fonction d'indexation F pour une donnée vectorielle SD et le transmettre, sans considération de la présence de l'index I dans une structure de données DS, et c'est à la réception que le gestionnaire 200 fera le rapprochement entre l'index reçu (I, si le canal CNL n'introduit pas d'erreurs de transmission, ou I' s'il y a eu des erreurs de transmission) et ceux présents dans la structure de données DS.

Dans des exemples de réalisation, le procédé de gestion de la communication d'une donnée vectorielle SD par le gestionnaire 100 comprend une étape de fragmentation.

On a vu que la structure de données DS réalise une quantification de l'espace vectoriel dans lequel les données vectorielles SD, SD' prennent des valeurs. Mais la taille de la structure de données DS est limitée. On a vu par exemple qu'une structure de données DS associant 65 536 données vectorielles SD, SD' d'une taille de 4096 réels encodés sur 32 bits aurait une taille globale proche du gigaoctet. Une telle taille serait probablement maximale. Il peut donc être intéressant plutôt de fragmenter les données vectorielles SD à communiquer, par exemple en fragments de dimensions 4, 16, 64 etc. La structure de données DS utilisée alors associerait à des index des fragments de données vectorielles de la dimension donnée. Les mêmes principes quant au calcul d'indexation relativement à la distance donnée D s'appliquent alors mais dans l'espace vectoriel des fragments, de dimension réduite par rapport à l'espace vectoriel complet des données vectorielles. L'index I' obtenu est alors un index permettant de retrouver dans la structure de données DS un fragment de dimension donnée. Dans des exemples de réalisation associés, le procédé de réception d'une donnée vectorielle par le gestionnaire de réception 200 comprend alors une étape de défragmentation. Le gestionnaire de réception 200 reçoit dans ces modes de réalisation plusieurs index correspondants aux différents fragments, index transmis par le gestionnaire de communication 100. Après avoir retrouvé dans la structure de donnée DS les fragments correspondant aux index reçus, le gestionnaire de réception 200 reconstruit une donnée vectorielle complète en combinant les différents fragments retrouvés.

Dans des modes de réalisation, la structure de données DS est définie au préalable en fonction du format de la donnée vectorielle SD à transmettre.

Pour un format utilisé de façon fréquente dans des applications d'intelligence artificielle, il peut être intéressant de définir à l'avance une ou des structures de données DS qui réalisent une quantification adaptée de l'espace vectoriel correspondant aux données vectorielles à transmettre. Pour un format donné, plusieurs structures de données DS différentes peuvent exister qui seront plus ou moins adaptées aux applications utilisant les données vectorielles communiquées. Des données vectorielles correspondant à des contenus visuels peuvent être par exemple davantage concentrées dans une zone précise de l'espace vectoriel où elles prennent des valeurs, différentes des zones où l'on retrouve des données vectorielles correspondant à des contenus textuels, et ce pour un même format de données vectorielles. Les quantifications intéressantes, et les structures de données DS correspondantes à ces deux tâches seront donc distinctes.

Pour un format donné de données vectorielles à communiquer, la structure de données DS peut être définie en utilisant un algorithme d'apprentissage automatique. En effet, la meilleure manière de concentrer la quantification réalisée par la structure de données DS aux endroits où elle est utile est de réaliser un apprentissage automatique à partir d'un corpus de données vectorielles. Ce corpus peut être représentatif d'une tâche précise utilisant les données vectorielles à transmettre, ou bien être générique pour tous types de données vectorielles utilisant un format donné.

Dans le cas d'un apprentissage automatique de la structure de données DS, un algorithme d'apprentissage adapté est la méthode des k-moyennes où k est le nombre de données vectorielles présentes dans la structure de données. Ici, l'apprentissage automatique va réaliser un regroupement (traduction de l'anglais *clustering*) d'un ensemble de données SD, SD' communiquées ou à communiquer, qui forment un ensemble d'apprentissage, autour d'un nombre réduit de données vectorielles représentatives qui seront enregistrées dans la structure de données DS. Un tel regroupement peut être effectué grâce à la méthode des k-moyennes, où k est la taille de la structure de données. Grâce à ce mode de réalisation, le regroupement est uniforme, au sens que les données appartenant à l'ensemble d'apprentissage sont bien réparties de manière égalitaire en groupes dont le centre est une donnée vectorielle représentative qui se retrouvera dans la structure de données DS pour représenter toutes les données vectorielles du même regroupement.

De manière associée, une fonction d'indexation peut être apprise qui indexera chaque donnée vectorielle présente dans la structure de données DS de façon que les données proches selon la distance donnée D présentes dans la structure de données DS seront associées à des index proches entre eux, par exemple selon la distance de Hamming entre index.

Dans des modes de réalisation, la fonction d'indexation F est définie au préalable pour un format donné de données vectorielles ou bien apprise selon un corpus d'apprentissage. Le corpus d'apprentissage peut être générique, pour toutes les tâches attendues des données vectorielles dans un format donné, ou bien spécifique aux données vectorielles à communiquer pour une tâche donnée.

Certaines fonctions sont particulièrement adaptées à l'apprentissage automatique. C'est le cas des fonctions de hachage sensibles à la localité (fonctions LSH). Ces fonctions sont définies selon des hyperplans qui découpent l'espace vectoriel dans lequel les données vectorielles prennent des valeurs en deux parties. Une partie, située d'un côté de l'hyperplan, correspondra à la valeur 0 pour un bit de rang donné dans l'index calculé alors que l'autre partie, située de l'autre côté, correspondra à la valeur 1 pour le bit de même rang. Pour une longueur L d'index donné, un nombre égal d'hyperplans est fixé qui va découper l'espace vectoriel en 2 puissance L, si on considère que l'index est une suite de bits. Des calculs similaires peuvent être faits quand l'index est une suite d'autres symboles, en ramenant ces autres symboles à des bits.

Le placement de ces hyperplans, qui définissent une fonction d'indexation LSH, peut être obtenu par apprentissage selon la densité des données à communiquer dans l'espace vectoriel, afin de réaliser les découpes les plus adaptées.

Pour améliorer le fonctionnement de l'invention dans le mode de réalisation présenté dans la figure 1, il est préférable que la fonction d'indexation F et la structure de données DS soient connues à la fois par le gestionnaire de la communication 100 et par le gestionnaire de la réception 200. Une manière de procéder est par exemple, pour le gestionnaire 100, de transmettre, dans un préambule à la transmission qui peut se trouver dans un en-tête de paquets ou dans un datagramme prévu pour ce rôle, des références à une fonction d'indexation F et/ou une structure de données DS, références qui sont connues par le gestionnaire 200.

Dans d'autres modes de réalisation, le gestionnaire de communication 100 pourra transmettre dans un échange préalable la structure de données DS elle-même. Dans ce cas, la taille de la structure de données DS (et donc le nombre de données vectorielles présentes dans la structure de données DS, réalisant la quantification de l'espace vectoriel des données vectorielles à communiquer) est à prendre en compte dans l'évaluation des performances des procédés selon l'invention. Une transmission préalable d'une structure de données DS adaptée aux données vectorielles à communiquer ultérieurement devra être rentabilisée par des communications nombreuses de données vectorielles.

Des modes de réalisation de l'invention comprennent la négociation de valeurs de paramètres utiles à la transmission de données vectorielles pour l'invention. Grâce à ces modes de réalisation, des possibilités plus poussées de communication entre le gestionnaire de communication 100 et le gestionnaire 200 de réception sont mises en œuvre. On a vu précédemment que des valeurs utiles à l'exécution du procédé pouvaient être transmises, par exemple dans un en-tête de paquet, et permettre de retrouver des éléments déjà définis, comme la structure de données DS utilisée en réception pour retrouver la donnée vectorielle SD' proche de la donnée à communiquer SD. Mais ce mode précédent ne suffit pas quand, par exemple, la structure de données DS et/ou la fonction d'indexation F sont apprises par apprentissage automatique. Le récepteur ne peut pas dans ce cas connaître *a priori* les éléments utiles au fonctionnement du procédé.

Dans ce mode de réalisation, une phase de négociation préalable à la transmission du ou des index permet de se mettre d'accord sur l'ensemble des paramètres utiles à la communication de données vectorielles selon le procédé. Ce mode peut être combiné avec le précédent : par exemple, une première transmission peut donner les références à une structure de données DS, puis une négociation permet à l'émetteur et au récepteur de se mettre d'accord sur un autre paramètre tel que la taille des paquets échangés.

La figure 2, quant à elle, illustre un exemple d'étapes mises en œuvre dans le cadre d'un mode de réalisation de l'invention.

Dans cet exemple, les gestionnaires 100 et 200 réalisent une phase de négociation permettant de mettre en place le transfert de données vectorielles depuis le gestionnaire de communication 100 vers le gestionnaire de réception 200 à travers un canal CNL non représenté sur la figure 2. Pour cela, les

gestionnaires 100 et 200 échangent plusieurs messages transmis à travers le même canal CNL ou par un autre moyen.

Le premier message REQ est adressé par le gestionnaire 100 au gestionnaire 200 pour indiquer son intention de communiquer des données vectorielles et proposer des paramètres permettant de réaliser ses communications de données au mieux. De tels paramètres peuvent indiquer le format des données vectorielles à communiquer, ce qui peut correspondre à un nom de modèle connu. Un autre paramètre peut indiquer l'intention de fragmenter les données ainsi que la taille des fragments obtenus après fragmentation. Un autre paramètre peut être une référence à la structure de données qui sera utilisée pour quantifier les fragments, c'est-à-dire le dictionnaire qui regroupe des valeurs associées à des index. Ce dictionnaire peut être connu à la fois par le gestionnaire 100 et le gestionnaire 200 et déjà présent en mémoire. Il peut être par exemple connu comme réalisant une bonne quantification pour les fragments d'une taille donnée pour des données vectorielles d'un format donné. Le gestionnaire de communication 100 peut aussi indiquer sa volonté de transmettre un dictionnaire non connu par le gestionnaire de réception 200 mais particulièrement adapté à la communication à venir. Le message REQ peut également comprendre des références à la fonction d'indexation F qui sera utilisée dans des modes de réalisation qui, là encore, peut être une fonction déjà connue ou une fonction apprise pour les besoins de la communication de données vectorielles. Un autre paramètre utile à transmettre est la longueur des index qui seront transmis à la place des valeurs de fragments qui seront connus par ailleurs grâce à la structure de données. Tout autre type de paramètre utile à la communication peut être proposé dans le message REQ adressé par le gestionnaire de communication 100 au gestionnaire de réception 200.

Le gestionnaire 200 peut répondre par un message ANS qui indique les valeurs de paramètres qu'il est disposé à utiliser pour la communication. Par exemple, si le gestionnaire de communication 100 propose une structure de données adaptée à une certaine taille de fragments, mais que le gestionnaire de réception 200 n'a pas en mémoire ce dictionnaire, il peut proposer une autre fragmentation correspondant à un dictionnaire dont il dispose. Plusieurs échanges de messages REQ-ANS peuvent avoir lieu entre les gestionnaires 100 et 200, formant une négociation des paramètres utiles à la communication de données vectorielles entre eux.

Le gestionnaire de communication 100 clôt la phase de négociation proprement dite en adressant un message SET qui transmet les valeurs proprement dites sur lesquels les gestionnaires 100 et 200 viennent de s'accorder. Par exemple, le contenu du dictionnaire pourra être transmis ici alors que précédemment les gestionnaires 100 et 200 ont négocié la taille du dictionnaire qu'ils allaient utiliser.

Un message d'acquittement ACK peut être émis par le gestionnaire de réception 200 vers le gestionnaire de communication 100 pour indiquer que le gestionnaire 200 a reçu l'ensemble des valeurs utiles à la communication de données vectorielles et est prêt à recevoir des index transmis par le gestionnaire de communication 100. Cet échange de messages correspond à un mode de réalisation possible d'une phase de négociation des valeurs de paramètres utiles à la communication de données vectorielles selon l'invention, mais d'autres modes de réalisation sont possibles.

Dans d'autres modes de réalisation, les valeurs de paramètre utiles à la communication ne sont pas négociées entre les gestionnaires 100 et 200 mais sont incluses dans un datagramme faisant partie de la transmission d'index. Ce mode peut être utilisé pour des communications de données vectorielles exprimées dans des formats standard pour lesquels des structures de données réalisant des quantifications appropriées sont connues et maintenues en mémoire par l'ensemble des gestionnaires 100 et 200 réalisant les procédés selon l'invention.

Le gestionnaire de communication 100 commence la communication en réalisant une fragmentation FRAG de la donnée vectorielle à communiquer. Cette opération consiste à découper la donnée vectorielle à communiquer en plusieurs fragments selon une taille donnée qui a pu être négociée ou bien qui correspond à un mode de communication connu pour un format de donnée vectorielle donné. Par exemple, une donnée vectorielle exprimée comme vecteur à 4096 dimensions peut être découpée en 512 fragments de dimension 8. Les opérations ultérieures s'appliqueront à chacun de ces fragments individuellement. L'intérêt d'une telle découpe est par exemple de réduire la taille de la structure de données à transmettre ou à sauvegarder en mémoire, puisque les éléments présents dans cette structure de données ne seront plus des données vectorielles de dimension 4096 mais des fragments de dimension 8.

Le gestionnaire de communication 100 poursuit, dans cet exemple de réalisation, par une opération de quantification QNT. Cette opération peut consister à chercher dans la structure de données le fragment le plus proche selon une distance donnée d'un fragment de la donnée à communiquer. L'opération d'indexation IND consiste alors à trouver dans la structure de données l'index correspondant à ce fragment le plus proche. Dans d'autres modes de réalisation, un index est calculé directement pour le fragment de la donnée à communiquer et l'opération de quantification QNT correspond à la construction de la structure de données utilisée par les gestionnaires 100 et 200.

Le gestionnaire de communication 100 poursuit par une opération de transmission TRS qui comprend l'envoi d'un message comprenant l'index I trouvé dans la structure de données qui a été négociée, index I correspondant à un fragment présent dans la structure de données proche, selon la distance donnée, du fragment de la donnée à communiquer. Ces opérations de quantification QNT, d'indexation IND et de transmission TRS sont répétées pour l'ensemble des fragments correspondant à la donnée vectorielle à communiquer qui ont été obtenus par l'opération de fragmentation FRG.

Le gestionnaire de réception 200 va alors procéder à une succession d'opérations de réception RCP d'index I transmis et de lecture RDG dans la structure de données de fragments correspondant aux index I reçus. Une opération de défragmentation DFRG permet alors de reconstruire la donnée vectorielle à communiquer à partir des fragments lus qui correspondent aux index reçus. Cette donnée vectorielle reconstruite sera celle qui sera utilisée dans des applications d'intelligence artificielle à la place de la donnée vectorielle à communiquer. Les opérations de quantification QNT et/ou d'indexation IND assurent que les index obtenus sont choisis en fonction d'une distance mesurée entre la donnée à communiquer et celle reconstruite à partir des fragments correspondant aux index reçus.

La figure 3, quant à elle, illustre des exemples de distorsions possibles subies par une donnée vectorielle lors de sa communication et de sa réception.

Dans l'exemple de la figure 3, pour mieux visualiser les distorsions introduites par les procédés de gestion de la communication et de la réception selon l'invention, on suppose que les données vectorielles considérées sont des vecteurs dans un espace de dimension 3 représenté par le repère X, Y, Z situé à l'origine de ces vecteurs. Cet exemple n'est pas réaliste car les données vectorielles ou les fragments qui vont être communiqués seront souvent de dimension plus grande, et qui est une puissance de deux, mais les principes restent les mêmes dans l'espace vectoriel concerné.

L'objectif est de communiquer la donnée vectorielle SD.

On suppose que, dans cet exemple, le procédé utilise une fonction d'indexation F et que la fonction d'indexation F calcule directement à partir de SD un index I, mais que cet index I n'est pas présent dans la structure de données DS. Dans certains modes de réalisation, le procédé ne va pas transmettre directement l'index I mais va chercher un index proche I', qui est bien associé à une donnée vectorielle SD' dans la structure de données DS. Dans d'autres modes, l'index I calculé par la fonction F sera considéré comme l'index obtenu et directement transmis. L'index I' est transmis dans le canal CNL mais celui-ci introduit des erreurs de transmission. L'index reçu est alors l'index I'', que l'on suppose ici ne pas être présent dans la structure de données DS. Le procédé selon l'invention va alors chercher un index I''' proche de l'index I'' tel que I''' soit présent dans la structure de données DS. La donnée vectorielle SD''' associée à l'index I''' dans la structure de données DS va alors être utilisée dans diverses applications à la place de la donnée initiale SD. L'accumulation de distorsions fait que la donnée utilisée SD''' n'est pas la donnée initiale SD, mais ces distorsions sont réduites au maximum par des choix adaptés des index en fonction d'une distance D mesurée entre données vectorielles, de la fonction d'indexation F, de la fonction calculant la distance D mesurée entre données vectorielles, et de la quantification par la structure de données DS de l'espace vectoriel XYZ dans lequel se trouvent les données vectorielles.

Dans l'exemple précédent, les index I et I'' ne sont pas présents dans la structure de données DS. Dans certains modes de réalisation, la structure de données DS comprend une association index-donnée vectorielle pour toutes les valeurs possibles des index. Dans ce cas, l'index I directement calculé à partir de la donnée SD sera transmis directement par le canal CNL. Et la donnée vectorielle qui sera utilisée finalement sera directement celle associée à l'index I'' reçu, après que les erreurs de transmission du canal CNL aient transformé l'index I en I''.

Par ailleurs, il est important de comprendre que, même si toutes les valeurs possibles d'index sont présentes dans la structure de données DS, la donnée vectorielle associée à un index calculé dans la structure de données DS ne sera pas en général la donnée vectorielle pour laquelle l'index a été calculé. En effet, il y a un nombre fini de valeurs possibles d'index alors que les données vectorielles, qui sont

des vecteurs dans un espace vectoriel de réels, peuvent prendre une infinité de valeurs. Un index calculé I sera donc le même pour tout un ensemble de données vectorielles alors qu'une seule donnée vectorielle sera associée à cet index dans la structure de données DS. Pour améliorer l'efficacité du procédé selon l'invention, les données vectorielles qui sont envoyées vers un même index par la fonction d'indexation F doivent être proches entre elles selon la distance D mesurée entre elles. Ces données proches entre elles forment alors une zone distincte dans l'espace vectoriel. La donnée vectorielle qui sera choisie comme étant associée à l'index calculé pour les données vectorielles de cette zone peut être par exemple la donnée située au barycentre de cette zone pour améliorer la représentativité des données vectorielles qui ont un même index calculé associé à une donnée vectorielle unique.

La figure 4, quant à elle, illustre des résultats d'expérimentation obtenus lors du développement des procédés selon l'invention.

Dans cette expérimentation, nous avons voulu montrer dans quelle mesure les distorsions introduites par les procédés de gestion de la communication et de la réception de données vectorielles selon l'invention impactaient les tâches d'intelligence artificielle dans lesquelles les données vectorielles communiquées sont utilisées.

La communauté de recherche Hugging Face (accessible par le site Internet http://huggingface.co) propose des exemples de tâches, modèles, ensembles de données permettant de réaliser des comparaisons d'algorithmes dans le domaine de l'intelligence artificielle. Nous avons considéré pour nos expériences l'ensemble de données *banking77* (accessible à l'adresse https://huggingface.co/datasets/banking77). Cet ensemble de données correspond à plusieurs dizaines de milliers de phrases (en anglais) correspondant à 77 demandes possibles dans le domaine bancaire, comme la demande d'une opposition en cas de vol d'une carte bancaire, ou bien la demande du solde d'un compte, ou d'autres exemples.

Nous avons converti cet ensemble de données textuelles en données vectorielles selon le modèle Mistral-7B d'une taille de 4096 réels encodés en *float32*. Les données vectorielles représentent donc un contenu sémantique qui est celui des textes initiaux. Ces contenus sémantiques associés aux données vectorielles devront être préservés au maximum malgré les distorsions introduites par les procédés selon l'invention permettant de communiquer des données vectorielles.

Nous avons pris une partie de ces données vectorielles converties (et la classification connue des contenus sémantiques qui leur sont associés) pour entraîner un réseau de neurones à effectuer la classification d'une donnée vectorielle dans l'une des 77 catégories de demandes du domaine bancaire. Il s'agit donc d'une tâche de classification classique d'application de l'intelligence artificielle, et qui utilise des données vectorielles pour ses manipulations.

Les performances du réseau de neurones entraîné, mesurées sur d'autres données vectorielles que celles utilisées pour l'entraînement du réseau de neurones, sont présentées dans la figure 3 par la ligne en tirets, étiquetée *nominal model*, proche du sommet. Elles sont supérieures à 90% de classification correcte. *A contrario*, les performances aléatoires correspondent à un succès de 1/77. Elles sont représentées par la ligne pointillée, étiquetée *random model*, proche du zéro (en effet, 1/77 = 1.3%).

Entre ces deux extrêmes, nous trouvons trois courbes qui représentent les résultats de tâches de classification qui sont obtenues après avoir utilisé trois manières distinctes de communiquer des données vectorielles avec les procédés selon l'invention en présence d'un canal CNL introduisant de plus en plus de perturbations.

Pour obtenir les courbes en question, nous avons simulé la communication de données vectorielles entre deux équipements EQ1, EQ2 dans un canal CNL et soumis les données communiquées au réseau de neurones que nous avons entraîné. La différence avec la ligne *nominal model* montre donc l'impact de la communication des données vectorielles selon le procédé sur la tâche de classification réalisée par le réseau de neurones. Le canal CNL est simulé comme ayant un taux d'erreurs variable dans la transmission de bits, taux qui est dénommé BER (acronyme de l'anglais *bit error ratio*). L'échelle horizontale est logarithmique et montre des simulations pour un canal CNL ayant un taux d'erreur qui varie de 10^{-3}, c'est-à-dire une chance sur 1000 d'erreur de transmission d'un bit jusqu'à plus que 10^{-1}, c'est-à-dire plus qu'une chance sur dix d'erreur de transmission, en passant par 10^{-2} (une chance sur 100 d'erreur de transmission).

Dans les trois exemples de communication simulés, nous avons choisi de fragmenter les données vectorielles en fragments de dimension 8. Une donnée vectorielle qui comprend initialement 4096 réels va donc être divisée en 512 fragments qui comprennent 8 réels encodés en *float32*, soit 512 fragments de 8 * 32 = 256 bits.

Nous avons ensuite réalisé des quantifications adaptées aux fragments de données vectorielles de dimension 8. Pour cela, nous avons pris un autre corpus proposé par la communauté de recherche Hugging Face, à savoir le corpus Locutusque UltraTextbooks 2.0, accessible sur le lien http://huggingface.co/datasets/Locutusque/UltraTextbooks-2.0, et réalisé des apprentissages en utilisant un algorithme de regroupement. Il est à noter que ce corpus de données textuelles n'a rien à voir avec le domaine bancaire. Les quantifications apprises ici pourraient donc être utilisées pour transférer des données vectorielles représentant des textes en dehors du domaine bancaire de façon probablement aussi efficace.

Nous avons réalisé à partir de ce corpus trois quantifications pour apprendre 2, puis 16, puis 1024 données vectorielles qui servent à quantifier des fragments de dimension 8. Les 2 fragments de données vectorielles quantifiées peuvent ensuite être représentées par un index de longueur 1 bit ; les 16 fragments de données vectorielles peuvent être représentées par un index de longueur 4 bits ; et les 1024 fragments de données vectorielles par un index de longueur 10 bits.

Une fonction d'indexation est ensuite utilisée pour calculer un index de longueur 1, 4 ou 10 bits. Les index calculés sont transférés et soumis à des taux d'erreurs variables simulés. Les fragments correspondant aux index reçus, ayant subi les erreurs de transmission, sont récupérés, puis une donnée vectorielle complète est construite à partir des fragments et est soumise au réseau de neurones entraîné, et on mesure la performance de la classification. Les performances de classification se situent bien sûr entre la droite *nominal model* (classification sans transmission de donnée sémantique) et la droite *random model* (classification aléatoire).

On voit que, pour des index de longueur 4 ou 10 bits, les performances de classification sont très proches de la performance nominale jusqu'à un taux d'erreur de 10^{-2}. Or, l'utilisation de ces index correspond à une diminution de la taille des données transmises de 8 fois 32 = 216 bits à 4 ou 10 bits. On a donc ici un facteur de diminution de 20 au moins et jusqu'à 54 pour l'index de longueur 4 bits si on avait choisi de communiquer les données vectorielles en transmettant directement les bits qui représentent les 512 fragments de 8 réels encodés sur 32 bits. Cette diminution très forte se fait avec une perte minime au niveau de la tâche de classification. Les performances restent au-dessus de 80% jusqu'à un taux d'erreur assez fort de 10^{-1}.

La quantification sur 1 bit a des performances maximales un peu en-dessous de 80%. Ces résultats ne sont donc pas très bons, mais pas négligeables non plus au vu d'un facteur de réduction de la taille des données transmises de 216.

Finalement, quand le BER augmente jusqu'à s'approcher d'un taux d'une erreur sur 2, les performances de classification chutent et rejoignent les performances d'une classification aléatoire.

Ces expériences montrent donc bien l'intérêt de l'invention qui offre une réduction du nombre de bits transmis par rapport à une transmission directe des bits formant les données vectorielles communiquées par un facteur très appréciable et une baisse faible des performances de classification, même en présence d'erreurs de transmission à un taux significatif.

Signalons enfin ici que, dans le présent texte, le terme « module » peut correspondre aussi bien à un composant logiciel qu'à un composant matériel ou un ensemble de composants matériels et logiciels, un composant logiciel correspondant lui-même à un ou plusieurs programmes ou sous-programmes d'ordinateur ou de manière plus générale à tout élément d'un programme apte à mettre en œuvre une fonction ou un ensemble de fonctions telles que décrites pour les modules concernés. De la même manière, un composant matériel correspond à tout élément d'un ensemble matériel (ou hardware) apte à mettre en œuvre une fonction ou un ensemble de fonctions pour le module concerné (circuit intégré, carte à puce, carte à mémoire, etc.).

Revendications

1. Procédé de gestion de la communication d'une donnée vectorielle (SD), dite donnée à communiquer, depuis un premier équipement (EQ1) à un deuxième équipement (EQ2), le procédé comprenant :
– obtenir un index (I') en appliquant une fonction d'indexation (F) à la donnée à communiquer (SD), la fonction d'indexation (F) calculant un index à partir d'une donnée vectorielle, la fonction d'indexation (F) étant telle que, plus deux données vectorielles sont proches selon une distance (D) mesurée entre elles, plus les index calculés pour ces deux données vectorielles par la fonction (F) sont proches entre eux, ledit index (I') permettant de retrouver dans une structure de données (DS) une autre donnée vectorielle (SD'), dite donnée communiquée, la structure de données (DS) associant des index à des données vectorielles, l'index obtenu (I') étant apte à être transmis depuis le premier équipement (EQ1) au deuxième équipement (EQ2).

2. Procédé de gestion selon la revendication 1 caractérisé en ce que la distance (D) mesurée entre la donnée à communiquer (SD) et la donnée communiquée (SD') est fonction de la proximité entre les contenus sémantiques associés aux deux données vectorielles (SD, SD').

3. Procédé de gestion selon l'une des revendications 1 ou 2 caractérisé en ce que l'index (I') est obtenu en cherchant dans la structure de données (DS) la donnée vectorielle (SD') la plus proche de la donnée à transmettre (SD) selon la distance (D) mesurée, et l'index transmis est l'index (I') associé à ladite donnée vectorielle (SD') dans la structure de données (DS).

4. Procédé de gestion selon l'une des revendications 1 à 3 caractérisé en ce que la distance (D) entre données vectorielles est mesurée en utilisant la similarité cosinus.

5. Procédé de gestion selon l'une des revendications 1 à 4 caractérisé en ce que la fonction d'indexation (F) est une fonction de hachage sensible à la localité.

6. Procédé de gestion selon l'une des revendications 1 à 5 caractérisé en ce que le procédé comprend une étape préalable de fragmentation de la donnée à communiquer (SD) en plusieurs fragments et les étapes ultérieures du procédé s'appliquent aux dits plusieurs fragments.

7. Procédé de gestion selon l'une des revendications 1 à 6 caractérisé en ce que la structure de données (DS) est définie au préalable en fonction du format de la donnée à communiquer (SD).

8. Procédé de gestion selon l'une des revendications 1 à 7 caractérisé en ce que la structure de données (DS) est obtenue par apprentissage automatique en fonction des données vectorielles déjà communiquées et/ou destinées à être communiquées.

9. Procédé de gestion selon la revendication 8 caractérisé en ce que l'apprentissage automatique de la structure de données (DS) utilise la méthode des k-moyennes où k est le nombre de données vectorielles présentes dans la structure de données (DS).

10. Procédé de gestion de la réception d'une donnée vectorielle (SD') caractérisé en ce qu'il comprend la réception d'un index (I'), dit index reçu, suivie par la lecture d'une donnée vectorielle (SD') dans une structure de données (DS) associant des index à des données vectorielles en fonction de l'index reçu (I').

11. Procédé de gestion selon la revendication 10 caractérisé en ce que le procédé comprend, si l'index reçu (I') ne correspond à aucun index présent dans la structure de données (DS), une recherche dans la structure de données (DS) de l'index le plus proche de l'index reçu (I') et en ce que la donnée vectorielle (SD') lue dans la structure de données (DS) est celle qui est associée au dit index le plus proche.

12. Procédé de gestion selon l'une des revendications 10 ou 11 caractérisé en ce que le procédé comprend, si l'index reçu (I') correspond à un fragment d'une donnée vectorielle, la reconstitution d'une donnée vectorielle complète à partir des fragments de données vectorielles correspondant aux index reçus.

13. Gestionnaire de communication (100) apte à gérer la communication d'une donnée vectorielle (SD), dite donnée à communiquer, depuis un premier équipement à un deuxième équipement, le gestionnaire de communication comprenant :
– un module (101) d'obtention d'index (I'), le module d'obtention étant apte à obtenir un index (I') en appliquant une fonction d'indexation (F) à la donnée à communiquer (SD), la fonction d'indexation (F) calculant un index à partir d'une donnée vectorielle, la fonction d'indexation (F) étant telle que, plus deux données vectorielles sont proches selon une distance (D) mesurée entre elles, plus les index calculés pour ces deux données vectorielles par la fonction (F) sont proches entre eux, ledit index (I') permettant de retrouver dans une structure de données (DS) une autre donnée vectorielle (SD'), dite donnée communiquée, la structure de données (DS) associant des index à des données vectorielles, le module (101) d'obtention d'index étant apte à fournir un index à un transmetteur d'index apte à transmettre un index depuis le premier équipement (EQ1) au deuxième équipement (EQ2).

14. Equipement (EQ1) de télécommunications comprenant un gestionnaire de communication (100) apte à gérer la communication d'une donnée vectorielle (SD) selon la revendication 13.

15. Gestionnaire de réception (200) de données vectorielles (SD') comprenant :

Un module (201) de réception d'un index (I'), dit index reçu ;

Un lecteur (202) d'une donnée vectorielle (SD') dans une structure de données (DS) associant des index à des données vectorielles en fonction de l'index reçu (I').

16. Equipement (EQ2) de télécommunications comprenant une entité de gestion (200) apte à gérer la réception d'une donnée vectorielle (SD') selon la revendication 15.

17. Programme d'ordinateur apte à être mis en œuvre par un gestionnaire de communication (100), le programme comprenant des instructions de code qui, lorsqu'il est exécuté par un processeur, réalise les étapes du procédé de gestion défini dans la revendication 1.

18. Programme d'ordinateur apte à être mis en œuvre par un gestionnaire de réception (200), le programme comprenant des instructions de code qui, lorsqu'il est exécuté par un processeur, réalise les étapes du procédé de gestion défini dans la revendication 10.

[Fig 1]

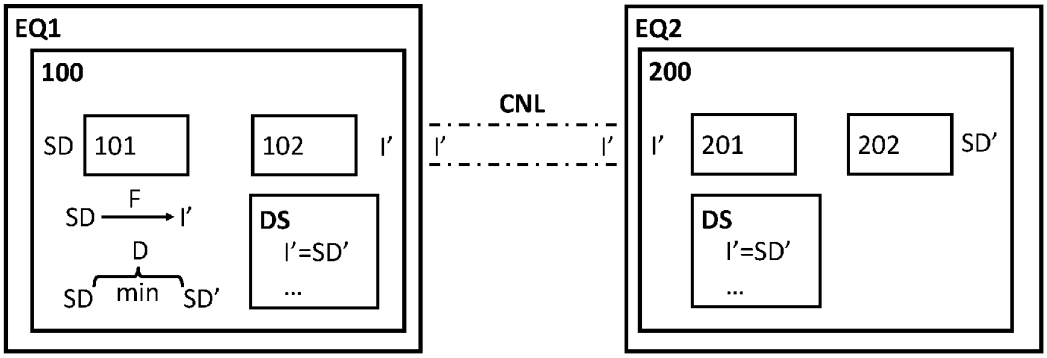


Fig. 1

[Fig 2]

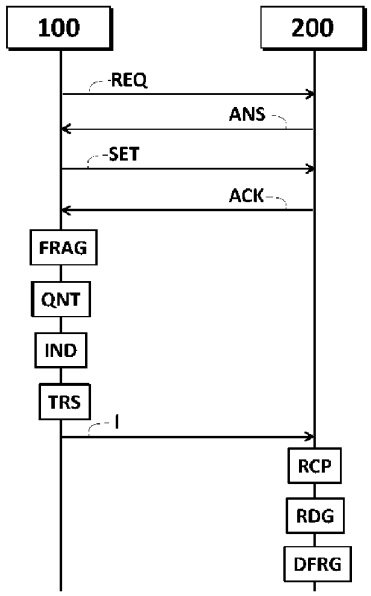


Fig. 2

[Fig 3]

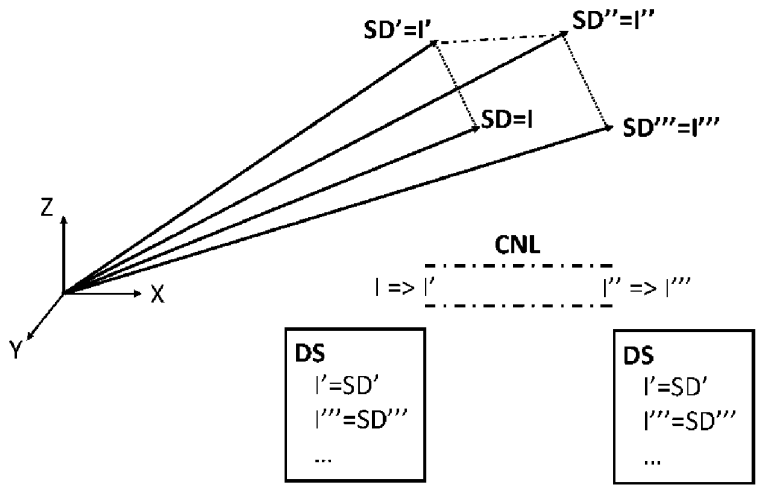


Fig. 3

[Fig 4]

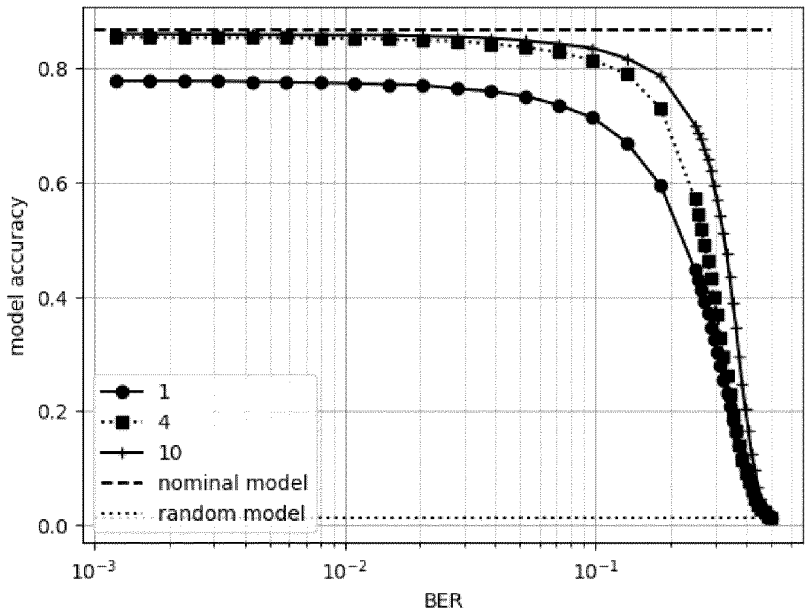


Fig. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/EP2025/060420

A. CLASSIFICATION OF SUBJECT MATTER

H03M 7/42(2006.01)i; *H03M 7/30*(2006.01)n

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H03M

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y A	WO 2004070540 A2 (HARRIS CORP [US]) 19 August 2004 (2004-08-19) the whole document	1-3,6,7,10,12-18 4,5,8,9 11
X	SEYED BAHRAM ZAHIR AZAMI, GANG FENG. "Robust vector quantizer design using self-organizing neural networks" *SIGNAL PROCESSING, ELSEVIER, AMSTERDAM, NL*, Vol. 80, No. 7, 01 July 2000 (2000-07-01), pages 1289-1298 DOI: 10.1016/S0165-1684(00)00036-0 ISSN: 0165-1684, XP004200942 abstract	1
Y	YIFAN WANG. "A Survey on Efficient Processing of Similarity Queries over Neural Embeddings" *ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853,* 17 April 2022 (2022-04-17), XP091205050 abstract section 3.1.1	4,5

☑ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 June 2025	**03 July 2025**
Name and mailing address of the ISA/EP **European Patent Office** **p.b. 5818, Patentlaan 2, 2280 HV Rijswijk** **Netherlands (Kingdom of the)** Telephone No. **(+31-70)340-2040** Facsimile No. **(+31-70)340-3016**	Authorized officer **Scappazzoni, E** Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/EP2025/060420

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	NEMATI MAHYAR ET AL. "VQ-VAE Empowered Wireless Communication for Joint Source-Channel Coding and Beyond" *GLOBECOM 2023 - 2023 IEEE GLOBAL COMMUNICATIONS CONFERENCE, IEEE,* 04 December 2023 (2023-12-04), pages 3155-3160 DOI: 10.1109/GLOBECOM54140.2023.10437659 XP034556252 the whole document	8
Y	HU QIYU ET AL. "Robust Semantic Communications With Masked VQ-VAE Enabled Codebook" *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE SERVICE CENTER, PISCATAWAY, NJ, US,* Vol. 22, No. 12, 12 April 2023 (2023-04-12), pages 8707-8722, [retrieved on 2023-04-12] DOI: 10.1109/TWC.2023.3265201 ISSN: 1536-1276, XP011955724 the whole document	8
Y	YIBO YANG ET AL. "An Introduction to Neural Data Compression" *ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853,* 18 November 2022 (2022-11-18), XP091373037 the whole document	9
A	FU QIFAN ET AL. "Vector Quantized Semantic Communication System" *IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE, PISCATAWAY, NJ, USA,* Vol. 12, No. 6, 09 March 2023 (2023-03-09), pages 982-986, [retrieved on 2023-03-10] DOI: 10.1109/LWC.2023.3255221 ISSN: 2162-2337, XP011942426 the whole document	1-18

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/EP2025/060420

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
WO 2004070540 A2	19 August 2004	EP 1595248 A2	16 November 2005
		IL 169946 A	30 November 2010
		US 2004153318 A1	05 August 2004
		WO 2004070540 A2	19 August 2004
		ZA 200506129 B	29 November 2006

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale n°
PCT/EP2025/060420

A. CLASSEMENT DE L'OBJET DE LA DEMANDE

INV. H03M7/42
ADD. H03M7/30

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

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
H03M

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

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

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X Y A	WO 2004/070540 A2 (HARRIS CORP [US]) 19 août 2004 (2004-08-19) le document en entier	1-3,6,7, 10,12-18 4,5,8,9 11

X	SEYED BAHRAM ZAHIR AZAMI, GANG FENG: "Robust vector quantizer design using self-organizing neural networks", SIGNAL PROCESSING, ELSEVIER, AMSTERDAM, NL, vol. 80, no. 7, 1 juillet 2000 (2000-07-01), pages 1289-1298, XP004200942, ISSN: 0165-1684, DOI: 10.1016/S0165-1684(00)00036-0 abrégé	1

	-/--	

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

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

* Catégories spéciales de documents cités:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent
"E" document antérieur, mais publié à la date de dépôt international ou après cette date
"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)
"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens
"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée
"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention
"X" document particulièrement pertinent;; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément
"Y" document particulièrement pertinent;; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier
"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée	Date d'expédition du présent rapport de recherche internationale
17 juin 2025	03/07/2025

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

1

Formulaire PCT/ISA/210 (deuxième feuille) (avril 2005)

C(suite). DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
Y	YIFAN WANG: "A Survey on Efficient Processing of Similarity Queries over Neural Embeddings", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 17 avril 2022 (2022-04-17), XP091205050, abrégé Section 3.1.1	4,5
Y	NEMATI MAHYAR ET AL: "VQ-VAE Empowered Wireless Communication for Joint Source-Channel Coding and Beyond", GLOBECOM 2023 - 2023 IEEE GLOBAL COMMUNICATIONS CONFERENCE, IEEE, 4 décembre 2023 (2023-12-04), pages 3155-3160, XP034556252, DOI: 10.1109/GLOBECOM54140.2023.10437659 le document en entier	8
Y	HU QIYU ET AL: "Robust Semantic Communications With Masked VQ-VAE Enabled Codebook", IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 22, no. 12, 12 avril 2023 (2023-04-12), pages 8707-8722, XP011955724, ISSN: 1536-1276, DOI: 10.1109/TWC.2023.3265201 [extrait le 2023-04-12] le document en entier	8
Y	YIBO YANG ET AL: "An Introduction to Neural Data Compression", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 18 novembre 2022 (2022-11-18), XP091373037, le document en entier	9
A	FU QIFAN ET AL: "Vector Quantized Semantic Communication System", IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE, PISCATAWAY, NJ, USA, vol. 12, no. 6, 9 mars 2023 (2023-03-09), pages 982-986, XP011942426, ISSN: 2162-2337, DOI: 10.1109/LWC.2023.3255221 [extrait le 2023-03-10] le document en entier	1-18

1

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Demande internationale n°
PCT/EP2025/060420

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
WO 2004070540 A2	19-08-2004	EP 1595248 A2	16-11-2005
		IL 169946 A	30-11-2010
		US 2004153318 A1	05-08-2004
		WO 2004070540 A2	19-08-2004
		ZA 200506129 B	29-11-2006