

Arturo Sangalli : l'essor discret du calcul souple

Nous croyons tous utiliser des systèmes électroniques et informatiques basés sur des mathématiques exactes. C'est faux. Le flou s'est introduit sans bruit dans notre vie de tous les jours. Réseaux de neurones, algorithmes génétiques et logique floue sont les avenues d'un nouveau monde mathématique qui ouvre des paysages auparavant inaccessibles.

Arturo Sangalli est docteur en mathématiques de l'Université de Montréal. Après des recherches post-doctorales à l'École polytechnique de Zurich, il a enseigné dans différentes universités du Canada, d'Europe et du Moyen-Orient. Il est professeur au Collège régional Champlain de Lennoxville. Collaborateur au *New Scientist*, il a obtenu en 1996 le prix de l'auteur de l'année décerné par l'ACFAS (Association canadienne française pour l'avancement des sciences). Son ouvrage « Eloge du flou » a reçu le Prix du meilleur livre en sciences de l'informatique de l'Association des éditeurs américains.

La Recherche : l'intelligence artificielle est-elle en train de connaître une révolution du flou ?

Arturo Sangalli : une évolution, plutôt, car les nouveaux systèmes ne vont pas remplacer les anciens. Ils apportent des compétences supplémentaires. En gros, on peut dire qu'il existe trois types de nouveaux systèmes, qui ont en commun de s'inspirer des mécanismes de la nature. C'est là que réside la révolution : dans le concept. Les algorithmes génétiques tirent leur principe de la théorie de l'évolution naturelle, les réseaux de neurones essaient de modéliser le fonctionnement des assemblages de neurones biologiques, tandis que la logique floue cherche à imiter la souplesse du langage et du raisonnement humains.

En quoi consiste la logique floue, et à quoi peut-elle servir ?

La logique floue sert à piloter des systèmes de contrôle, celui d'un appareil photo, d'une machine à laver ou d'une usine de purification d'eau par exemple. Mais au lieu d'utiliser des règles quantitatives, comme le fait la logique classique, elle utilise des règles « floues », ce qui veut dire deux choses : que les données ne doivent pas être quantitatives et que les règles ne sont pas basées sur un modèle mathématique. Son créateur, Lotfi Zadeh, la qualifiait de « méthode pour calculer avec des mots ». Par exemple, « si la température est élevée et la densité de l'image faible, alors la charge électrique du tambour doit être haute » est l'une des neuf règles « linguistiques » qui contrôlent le fonctionnement d'une photocopieuse à logique floue. Il s'agit donc de faire réaliser une tâche au système, en sachant quel résultat on veut obtenir, mais sans pouvoir décrire précisément la fonction qui relie ce résultat aux données, et sans que les données soient forcément quantitatives.

En quoi est-il important de pouvoir traiter des données « floues » ou « de façon floue » ?

Parce que souvent nous n'avons pas le choix. Si je vous demande de me décrire un visage, vous n'allez pas pouvoir me fournir des mesures anthropométriques, vous direz « visage rond », « nez fin » ou « menton pointu ». Ce sont des données qualitatives, qu'un système de contrôle classique ne peut pas comprendre. Et même s'il s'agit d'apprécier une grandeur quantitative, vous aimeriez pouvoir la contrôler sans passer par une mesure : vous savez qu'il fait « un peu trop chaud », ou que votre photocopie est « un peu trop grande », mais vous pouvez difficilement fournir un chiffre. Par ailleurs, il est utile d'appliquer un traitement flou, même à des données quantitatives, pour adoucir les transitions entre celles-ci. Prenons le cas de l'iris électronique d'un appareil photo.

Avec un contrôleur classique, on a une règle du type : si la luminosité dépasse 1000 lumens, alors le temps de pose est d'autant. En-dessous de 1000 lumens, il est d'autant. Ce qui veut dire que 10 lumens ou 900 lumens sont traités de la même façon. Avec un contrôleur flou, on peut utiliser plusieurs zones dans l'image et plusieurs zones de valeur de luminosité, sans pour autant les définir numériquement, et puis on fait jouer en même temps les différentes règles associées à ces différentes zones. Le temps de pose final est une sorte de moyenne pondérée de toutes ces règles, qui ne se trouve dans aucune d'entre elles en particulier.

La logique floue fonctionne avec des règles connues, mais fournit des résultats qui ne sont pas prévisibles ?

Exactement. Parce que les règles correspondent à un savoir empirique et non à un modèle mathématique. Prenons comme exemple une tâche impossible à programmer de manière classique : maintenir un bâton en équilibre vertical. Un enfant le fait sans problème sur son doigt. Un robot qui utiliserait un modèle mathématique prendrait des heures pour calculer sa position suivante. Takeshi Yamakawa a résolu le problème au moyen de sept règles floues du type : « si la perche s'incline légèrement en s'éloignant de vous et tombe lentement, bougez vers l'avant mais pas trop vite » (1). Le système fonctionne très bien, mais à la différence d'un système classique, on ne peut pas « prouver » sur le papier qu'il fonctionne. Il faut le faire fonctionner pour évaluer sa performance, et si l'on change une règle, il faut l'essayer pour voir ce que ça donne. Aucun calcul n'est possible. Le résultat n'est plus traçable. C'est d'ailleurs une caractéristique commune aux trois approches dont je parle ici. Elles se comportent comme des boîtes noires, là où l'informatique classique est une boîte transparente.

Comment fonctionnent les réseaux de neurones ?

Précisons que ce sont des algorithmes et non des structures particulières qui imiteraient nos cerveau. Mais ces algorithmes sont « incomplets » dans le sens où certains paramètres ne sont pas définis. On laisse au système le soin de les fixer lui-même, au fil des répétitions de la tâche qu'on lui assigne. Cette « auto-définition » du système lui donne deux atouts majeurs : la capacité d'apprendre et celle de généraliser. Pour reconnaître des chiffres et des lettres écrits à la main, par exemple, il faut en avoir vu beaucoup et se faire une idée générale de ce qu'est un 4 ou un B. Il est impossible de donner une définition du 4 qui permet de reconnaître tous les 4. Le réseau va « apprendre » ce qu'est un 4 par l'expérience. On lui fournit des milliers de 4 et de « non-4 », et il ajuste ses paramètres internes de manière à maximiser la discrimination entre les deux catégories. Après cela, il sera capable de généraliser, c'est-à-dire de fournir une réponse adéquate pour un stimulus auquel il n'a jamais été confronté. Par l'exposition à un nombre limité d'exemples, on forme un système capable de reconnaître des chiffres manuscrits, du vin frelaté, des virus informatiques ou l'improbabilité des achats d'un client donné. Comme dans le cas précédent, il est impossible de savoir comment le réseau a produit sa réponse. Mais on constate qu'il s'en sort bien.

Si le réseau s'auto-configère en permanence, peut-il arriver qu'on obtienne des réponses différentes à la même question, si on la repose après un certain laps de temps ?

Ce serait le cas si on laissait le réseau continuer à modifier ses paramètres. Mais on évite justement ce genre de « surentraînement » qui peut être gênant. On gèle les paramètres au terme de la période d'apprentissage, quand on estime que le réseau est performant. En revanche, l'inconstance des réponses est tout-à-fait courante avec les algorithmes génétiques qui, eux, incorporent un élément de hasard, à l'image de l'évolution naturelle. Ce sont des programmes qui « se reproduisent », en croisant leur matériel avec d'autres, qui sont soumis à des mutations aléatoires, et qui sont sélectionnés en fonction de leurs résultats. Au fil des générations, la qualité des solutions fournies par ces algorithmes s'améliorent. On peut fixer arbitrairement le niveau de qualité que l'on veut atteindre. Cette approche convient particulièrement pour les problèmes qui ont un nombre énorme de solutions acceptables mais dont la solution optimale est incalculable, par exemple les problèmes de trajectoires de robots, de conception de réseaux de communication

ou de transport. Il existe une infinité de chemins possibles et on veut trouver en un temps raisonnable un chemin qui soit performant, mais pas nécessairement le meilleur.

Et c'est l'ensemble de ces techniques que vous rangez sous le vocable « calcul souple » ?

Oui, je trouve que le mot flou n'est pas le plus indiqué, parce qu'il a une connotation négative. Si votre pensée est floue, on n'en fera pas grand cas. Dans le cas de l'informatique, le flou concerne la nature des données, ou des règles, ou des configurations, mais le processus consiste toujours en une série de calculs qui débouchent sur un résultat précis. La méthode est floue, le résultat est net. Donc je préfère parler de souplesse. Ce mot traduit bien l'opposition avec l'approche classique. Le calcul classique utilise un programme préétabli, basé sur une logique bivalente, qui s'exécute de manière séquentielle avec des données chiffrées. Les mêmes données fournissent toujours les mêmes résultats et on pourrait les recalculer à la main (à condition d'avoir le temps). Le calcul souple peut s'auto-programmer, utiliser une logique polyvalente, incorporer des éléments aléatoires, traiter des données ambiguës, calculer en parallèle et donner des réponses approximatives. Il est impossible de savoir ce qu'il va faire sans le mettre à exécution. En dépit de ces différences, le calcul souple utilise principalement le même matériel que le calcul classique, à savoir des circuits électroniques. Mais ils les utilisent différemment.

Si le calcul souple présente tant de possibilités, comment se fait-il que nous n'en soyons pas plus entourés ?

Tout d'abord, nous en sommes entourés, mais nous ne le savons pas. Le calcul souple a été intégré dans beaucoup d'appareils électroniques, systèmes de contrôles ou systèmes experts, mais nous n'en sommes pas nécessairement informés parce que ce n'est pas une notion très pertinente pour le consommateur. L'argument de vente, c'est la performance de l'appareil, peu importe qu'elle repose sur une logique classique ou sur une logique floue. Je connais quelques cas d'annonces publicitaires qui font explicitement allusion à un système de contrôle flou. En 1998, on a vanté les mérites de la nouvelle Coccinelle de Volkswagen, dont la boîte de vitesse contenait une composante floue lui permettant de s'adapter au style du conducteur. Mais de manière générale, le flou est très discret. Ce ne fut pas le cas au Japon, où il a fait un véritable tabac. D'une part, il correspondait bien à certaines caractéristiques de la culture japonaise, notamment son goût pour la nuance. D'autre part, les ingénieurs japonais se sont emparés des idées théoriques de Zadeh pour en faire la base d'un développement autonome, qui ne serait pas copié sur l'Occident. C'était presque une croisade. Mais aujourd'hui, l'engouement est un peu retombé. Je crois que l'optimisme a été exagéré. On a cru que tout allait pour voir se résoudre grâce au calcul souple, mais celui-ci a aussi ses limites. Il est en train de prendre sa juste place. S'il nous entoure, comme je l'ai dit, c'est de manière très diluée. Il est incorporé dans des architectures qui sont pour l'essentiel toujours basées sur le calcul classique.

Quelle est la raison de cet impact limité ?

Il y a le fait que ces différentes techniques se montrent efficaces pour traiter certains types de problèmes seulement. Un réseau de neurones sera très performant pour faire de la reconnaissance de formes ou pour apprendre à un robot à se déplacer dans un environnement physique, mais ce serait idiot de lui demander de gérer une banque de données ou un programme de traitement de texte. Les techniques de calcul flou permettent des applications que l'informatique classique ne permet pas, mais qui restent encore relativement limitées. Le calcul classique est et restera longtemps le pilier de notre informatique. Et c'est bien normal. Il a fait ses preuves. On le maîtrise bien. Donc, chaque fois qu'on aborde un problème, on va d'abord tenter de le modéliser complètement pour pouvoir le traiter « à l'ancienne ». Si on peut établir la fonction mathématique qui relie les inputs aux outputs, alors on peut tout calculer exactement et c'est tout de même l'idéal. Mais quand on ne peut pas établir cette fonction, alors on peut essayer les techniques souples : partir d'une description linguistique, remplacer la fonction par un réseau de neurones qui la simule... Ainsi, le calcul souple entre progressivement dans les mœurs en s'occupant des problèmes qu'il résout mieux, ou que lui seul résout. Mais cela ne se passe pas en termes de

révolution. Il rejoint l'ensemble des outils que les ingénieurs possèdent. C'est une technique parmi d'autres. Et l'avenir est certainement dans la collaboration, l'hybridation.

L'autre facteur, c'est qu'il faut un progrès concomitant de la technologie, en particulier dans la vitesse de calcul et la capacité de mémoire des ordinateurs. Pour construire des robots mobiles, comme le chien Leibo par exemple, il faut parvenir à des réponses en temps réel, ce qui suppose des processeurs de capacité gigantesque. Certaines potentialités du calcul souple sont ainsi limitées par les capacités techniques.

Est-ce qu'il n'y a pas un problème de confiance ? Qui oserait monter dans un avion dont le pilote automatique repose entièrement sur une logique floue, c'est-à-dire dire par essence imprévisible ?

Quand on dit imprévisible, on veut dire qu'il est impossible de tracer le calcul que l'ordinateur va faire, et par conséquent de prévoir le résultat précis. On ne veut pas dire que l'avion risque de faire un looping ! Il est évident que le résultat sera fiable, dans des limites statistiques que l'on peut moduler en multipliant les tests. Je vous donne un exemple. La logique floue permet d'encoder des règles empiriques, donc des savoir-faire ou des expertises que l'on ne pourrait pas modéliser mathématiquement. On peut notamment l'utiliser pour le diagnostic médical. Prenons des urologues qui procèdent à des examens prostatiques par ultrasons et qui formulent leur expertise en termes linguistiques. On encode toutes ces « règles » de diagnostic, venant de différents experts, dans un seul système à logique floue. Ensuite, celui-ci est confronté à des données et il produit « son » diagnostic en utilisant toutes les règles qu'il a reçues. Le taux d'erreur de la machine s'avère être de 2% alors que le taux d'erreur humain est de 5% en moyenne (2). Donc je ne crois pas que le problème de la confiance soit le grand problème du calcul souple. C'est l'expérience qui détermine si on va faire confiance ou non. Aujourd'hui, les opérations bancaires sont électroniques, le vote est électronique. Le public fait de plus en plus confiance aux machines, sans se poser beaucoup de questions. Et sûrement pas de savoir si le calcul est flou ou classique. D'ailleurs, vous faites déjà confiance au pilote automatique, sans savoir comment il fonctionne. Peut-être a-t-il des composantes floues...

Les gens qui développent le calcul souple sont-ils les mêmes que ceux qui travaillent sur le calcul classique ?

Non. Le calcul souple est surtout porté par les ingénieurs. Il y a très peu de théoriciens, de chercheurs dans le milieu universitaire. Le calcul souple est une science appliquée. Les ingénieurs veulent des résultats, ils sont réalistes et peu intéressés par les questions d'élégance théorique ou pire encore, d'implications philosophiques. En outre, ils sont très peu préoccupés de rédiger des articles – et d'ailleurs le contexte souvent industriel dans lequel ils travaillent les empêche de communiquer leurs travaux. C'est sans doute l'une des raisons pour lesquelles le calcul souple fait peu de bruit. Les chercheurs théoriciens, plus nombreux en calcul classique, sont, eux, avides de publier. Mais il est amusant de savoir que cela n'a pas toujours été la tradition en mathématiques. Les mathématiciens de l'époque des lumières, par exemple, gardaient jalousement les méthodes de calcul qu'ils avaient mises au point. Ils montraient le résultat, mais jamais la technique – exactement comme les magiciens ! Heureusement, les choses ont beaucoup changé. Le partage des connaissances scientifiques est extraordinaire. On publie tout et tout est accessible à tout le monde. Sans droits d'auteur. Imaginez s'il fallait payer une redevance chaque fois qu'on utilise un théorème en classe !

Propos recueillis par Elisa Brune

Yamakawa T., « Stabilization of an inverted pendulum by a high-speed fuzzy logic controller hardware system », Fuzzy sets and systems, 32 (1989)

Fujioka T., Arita S., Saito T., et al, Transrectal ultrasonography of prostatic cancer : application of a new diagnostic item and fuzzy inference, Japanese journal of medical ultrasonic 17, 1990

Pour en savoir plus :

Sangalli A., Eloge du flou, Les Presse de l'Université de Montréal, 2001
Kosko B., Fuzzy thinking, Flamingo, 1994