Vérification formelle de la sûreté d'un système contrôlé par réseaux de neurones

Arthur Clavière[1], Eric Asselin[1], Christophe Garion[2] and Claire Pagetti[3]

[1]Collins Aerospace [2]ISAE-SUPAERO [3]ONERA

Cet article est un résumé étendu de l'article "Safety Verification of Neural Network Controlled Systems", accepté à SSIV 2021 (7th International Workshop on Safety and Security of Intelligent Vehicles).

Contexte/motivation Les réseaux de neurones représentent une technologie prometteuse dans le domaine du contrôle. Parmi les applications connues, le ACAS Xu utilise des réseaux de neurones pour décider de manoeuvres d'évitement entre des drones [3]. Par rapport à un contrôleur classique basé sur des tables, le ACAS Xu a une empreinte mémoire largement réduite. Par ailleurs, afin de garantir un temps d'exécution suffisamment petit, le ACAS Xu a une architecture particulière. Au lieu d'utiliser un unique grand réseau, il utilise plusieurs petits réseaux, avec un système de switch entre ces réseaux : un seul réseau est exécuté à chaque pas de temps, ledit réseau étant choisi en fonction de la commande précédente.

Cependant, démontrer la sûreté de fonctionnement d'un tel système demeure problématique. Habituellement, cet objectif est réalisé via l'application de certains standards. Dans l'aéronautique par exemple, ces standards demandent de raffiner les exigences système en une spécification complète et correcte pour chaque composant du système. Des activités de vérification doivent ensuite être menées dans le but de démontrer que chaque composant satisfait sa spécification. Mais cette démarche s'applique difficilement dans le cas où le système intègre un réseau de neurones. Une première raison à cela est la difficulté à raffiner les exigences système en une spécification complète pour le comportement dudit réseau, la seule spécification disponible étant souvent un ensemble de données exemples, données à partir desquelles est entraîné le réseau. L'intéret d'un réseau de neurones réside d'ailleurs en partie dans sa capacité à résoudre un problème complexe, pour lequel on n'a pas de solution a priori. Autrement dit, on ne dispose pas d'une spécification pour le comportement attendu du réseau. De plus, les méthodes d'apprentissage actuelles ne garantissent pas la correction du réseau vis-à-vis d'une hypothétique spécification, comment alors démontrer qu'un réseau est correct s'il ne l'est pas ?

Plusieurs méthodes ont déjà été proposées pour résoudre ces problèmes. Parmi ces méthodes, certaines cherchent à enrichir la spécification du comportement attendu du réseau. Ces approches s'intéressent à définir et à vérifier des propriétés

d'intérêt sur les réseaux de neurones, telles la robustesse locale [4], mais ces propriétés n'offrent pas des garanties suffisantes quant à la sûreté du système intégrant ces réseaux. D'autres approches s'inspirent des travaux déjà menés pour la vérifcation de systèmes hybrides, en les adaptant au cas où la composante discrète du système est un réseau de neurones [2]. En raisonnant au niveau du système, ces méthodes permettent de s'affranchir des problèmes cités ci-dessus, essentiellement liés au raffinement des exigences au niveau des réseaux et à leur vérification. Mais ces méthodes ne permettent pas de démontrer la sûreté d'un système contrôlé par réseaux de neurones du type du ACAS Xu, notamment du fait du mécanisme de switch entre les réseaux. Nous proposons dans cet article une méthode pour vérifier la sûreté d'un tel système.

Approche En premier lieu, afin de formaliser le problème, nous introduisons un modèle pour la classe de système correspondant au ACAS Xu. Ce modèle reprend les codes d'un système en boucle fermée, avec d'une part une partie dynamique temps continu (plant) et d'autre part une partie contrôleur temps discret. La spécificité du modèle réside dans la définition du contrôleur qui consiste en un classificateur et qui d'une part intègre des réseaux de neurones et d'autre part dispose d'un état interne (commande produite au pas de temps précédent) permettant de choisir le réseau de neurones à exécuter (mécanisme de switch entre les réseaux). Comme indiqué précédemment, l'avantage d'un tel mécanisme est d'offrir un temps d'exécution réduit, aspect non négligeable dans le monde de l'embarqué. Pour la partie plant, nous faisons l'hypothèse classique d'une dynamique modélisée via une équation différentielle ordinaire avec les hypothèses de continuité habituelles, assurant l'unicité de la solution si les conditions initiales sont fixées.

La vérification de la sûreté de ce système est ensuite exprimée comme un problème de décision : décider si les états atteignables du système sur un horizon temporel fini intersectent ou pas un ensemble d'états d'erreurs (*e.g.*, l'ensemble des états correspondant à une collision dans le cas du ACAS Xu). Formulé ainsi, le problème est indécidable et ce du fait de la dynamique possiblement non-linéaire du plant. De plus le problème est rendu particulièrement complexe par l'utilisation des réseaux de neurones et le système de switch entre ces réseaux. Aussi, nous nous concentrons sur la calcul d'une sur-approximation des états atteignables du système.

Pour calculer cette sur-approximation, nous introduisons une représentation symbolique des états atteignables par le système, intégrant l'état interne du contrôleur. Il s'agit d'un liste d'états symboliques où chaque état symbolique est un doublet composé (i) d'une boîte représentant un ensemble d'états du système dynamique et (ii) de l'état interne du contrôleur c'est-à-dire la commande produite au pas de temps précédent. Cette représentation symbolique est utilisée pour approximer les états initiaux du système. A partir de cette représentation des états initiaux, les états atteignables par le système à chaque nouveau pas de temps sont cal-

culés, en utilisant la même représentation et en combinant deux méthodes : d'une part de la simulation garantie pour approximer la partie dynamique et d'autre part de l'interprétation abstraite pour approximer la partie contrôleur. La simulation garantie est réalisée à l'aide de l'outil DynIBEX [1] et l'analyse des réseaux par inteprétation abstraite est réalisée à l'aide des outils Reluval [6] ou DeepPoly [5], qui implémentent des transformateurs abstraits spécifiques aux réseaux de neurones.

Enfin, nous illustrons la faisabilitié de notre approche via le cas d'étude ACAS Xu. Pour traiter ce cas d'étude, nous introduisons une heuristique particulière visant à partitioner l'ensemble des états initiaux possibles et ainsi paralléliser la résolution du problème de vérification.

References

[1] Julien Alexandre dit Sandretto and Alexandre Chapoutot. Validated Explicit and Implicit Runge-Kutta Methods. *Reliable Computing electronic edition*, 22, July 2016.

[2] Radoslav Ivanov, James Weimer, Rajeev Alur, George J. Pappas, and Insup Lee. Verisig: verifying safety properties of hybrid systems with neural network controllers. *CoRR*, abs/1811.01828, 2018.

[3] Kyle D. Julian, Mykel J. Kochenderfer, and Michael P. Owen. Deep neural network compression for aircraft collision avoidance systems. *CoRR*, abs/1810.04240, 2018.

[4] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. *CoRR*, abs/1702.01135, 2017.

[5] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3(POPL), 2019.

[6] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Formal security analysis of neural networks using symbolic intervals. In *27th USENIX Security Symposium, USENIX Security 2018*, pages 1599–1614, 2018.