Machine learning based crash-allowed area detection for autonomous UAV

Adrien Chan-Hon-Tong

Machine learning based crash-allowed area detection for autonomous UAV

Adrien CHAN-HON-TONG

November 19, 2019

Abstract

There is today a clear discordance of the community about the relevancy and the safety of embedding computer vision deep learning algorithm in autonomous vehicle.

Yet, this paper presents a very simple and pragmatic autonomous vehicle use case where computer vision deep learning algorithm are both relevant and safe.

1 Introduction

Machine learning is the only conceptual way to deal with unformulated problems.

Of course, machine learning tools can be used on other contexts: deep networks have been applied in Go game [17], in control task [9], and, to replace classical anti collision airborne system (ACAS) [8]. But, these examples have almost nothing to do with machine learning, because, they concern somehow formalized problems.

Typically, [8] presents a code which could prove or disprove that a given deep network implements correctly ACAS specifications. This result is very interesting as it is a pioneer result. But, from industrial point of view, a simple (yet large) tabular can also correctly implements an ACAS, so using a deep network is not unavoidable. And, from machine learning point of view, this result is irrelevant because machine learning problems never come with specifications.

Indeed, the downside of handling problems with no specification is that there is (and will) no way to prove that a machine learning system behaves correctly on a specific but unformulated problem [18].

Because of this impossibility to specify the tackled problem (and, thus, the impossibility to have proof of correctness), DO-178 rules seems to forbid learning based system to be embedded in autonomous vehicle to perform critical function.

This point of view is also strengthen by the fact that most machine learning problem can be tackled using other way. Typically for autonomous driving, obstacle detection can be tackled by radar or lidar offering more dependable data than passive optical detection. It can be more expensive to use lidar,

it can be frustrating to stop in front of plastic bag on which the car could safely move, but, a lidar brings the guarantee to not miss obstacles. And, such guarantee will never be offered by passive optical detection.

However, this paper presents a very simple and pragmatic autonomous plane/UAV use case in which perception is unavoidable (i.e. radar and/or lidar does not trivialise the problem), and, which is directly linked which safety issue. **This use case is machine learning based crash-allowed area detection for autonomous UAV/plane**.

It will be more precisely presented in next section, before discussion.

2 Crash-allowed area detection

2.1 When machine learning helps safety

UAV are today allowed to fly according to a set of rules including maintaining visual line of sight with the pilot. Yet, simultaneous failure of remote control system, and, GPS return to base system will let the platform into an unstable state irremediably leading to a crash (if no recovery happens).

So, without autonomous perception, this hardware failure leads to an uncontrolled crash, even if flying capacity are not affected. Inversely, if the UAV has perception capacity, the UAV can try to reach crash-allowed areas in complete autonomous way. Yet, such controlled crash capacity requires the ability to decide crash-allowed areas.

And, this paper states that crash-allowed area detection is structurally a machine learning problem. First, it can not be trivialized using dedicated sensors (e.g. lidar or radar) because crash-allowed area detection relies on characterizing a signal not just receiving it. Then, there is no formal description of the task.

In addition, this is an interesting use case from safety point of view: even if the crash-allowed area detector is not perfect, it can hardly lead to worse output than an uncontrolled crash ! So, such module of crash control is clearly a tool to increase safety: here machine learning does not replace a safe way by a doubtful one. Here machine learning add a quite safe way.

Precisely, this task assumes a sensor providing an input datum x which should be characterized into crash-forbidden area (car, person, industrial stations, gaz stations, not-metal-roof building) or crash-allowed area (road, field, container, metal-roof building, cleared tree, cleared swimming pool). Given, data $x_1, ..., x_N$, a person can easily provide the desired classification $y_1, ..., y_N$. Yet, nobody can provide the mathematical function $y(x)$. So, given a deep network f, there is no way to compute the measure of the set on which $f \neq y$. But still, it is easy to count the number of samples i for which $f(x_i) \neq y_i$.

So, it is easy to learn a deep network f using a training set, and, to evaluate empirically the offered performance (using a disjoint testing set and other good evaluation rules). This way, it seems relevant to integrate such algorithm it into a classical safety analysis.

2.2 Implementation

The simplest form of machine learning based crash-allowed area detection for autonomous UAV can be implemented as semantic segmentation problem in remote sensing image.

Hopefully there is a very large literature about semantic segmentation problem in remote sensing image. One pioneer work is [11]. This work has quickly be extended [13, 1] using designed network like UNET [15]. Such map can then be post processed to recover instance [2]. Literature trends are to perform instance detection simultaneously [6].

There is also a very large number of datasets for such task including

- Data Fusion Contest - GRSS - IEEE (typically the 2015 one)
- 2D Semantic Labeling - ISPRS dataset (Potsdam and Vaihingen)

These datasets can be useful to empirically evaluate such pipeline.

Just to support the discussion, a naive pipeline has been designed on Potsdam using a UNET network using organizer train/test splitting. Empirical pixel-wise accuracy measured on 10 runs is 86,26% with variance of 2%. Converted into a crash allowed/forbidden area, this pipeline which is a basic baseline already leads to an empirical pixel-wise accuracy of 97% ! Using latest state of the art algorithm will probably increase this last result (already very high).

This result highlights the relevancy of using machine learning based crash-allowed area detector instead of accepting uncontrolled crash.

3 Discussion

This paper clearly calls for allowing machine learning module to perform critical function in embedded platform for specific use cases like for machine learning based crash-allowed area detection for autonomous UAV/plane.

Now, given the discordance of the community about the safety issue of such embedding, this paper argues this claim by reviewing several possible objections (of course this list may be not exhaustive).

3.1 Interpretation of probability

Probabilities are completely acceptable in a safety analysis (indeed, safety objective is even given in term of probability: 10^{-9} critical issue per hour for DAL-A module).

Yet, probability of breakdown of one hardware component is slightly different from probability that a machine learning algorithm fails on datum x. Indeed, in case of machine learning, the algorithm is deterministic (after training), so, if it fails on a datum x, then it will always fail on this datum. Indeed, machine learning performance is about computing the measure (let say the volume to give an image) of the part of input space on which the predicted and desired

functions are not equal. This is often expressed in term of probability, but, it is more about measure.

This point may be a real issue for daily function. Typically, daily computer vision based landing could be problematic because if it does not work on an airport x, there would be an hazard at each landing on this airport, even if it works on all other airport in the world (so measured performance is still high).

Now, in the offered use case, this is not a problem because machine learning system is used only on probabilistic emergency context. So, the system may never work on the airport x, but, anyway, the system would be activated on this airport only with already low probability, because, the hardware failure leading to the activation is independent from the airport.

Typically, such kind of context is already accepted in certification process: a simultaneous breakdown of several CPU may or may not crash the plane depending on the piece of code being processed during the breakdown. But, the same combination breakdown - piece of code always crashes the plane. Hopefully, as the breakdown of a CPU is independent from the processed piece of code, the CPU breakdown probability is (somehow) multiplied by the probability of processing specific part of the code.

So, this reasoning stand for machine learning. The only difference is that *probability* of processing specific part of the code can be estimated exactly (at least with uniform prior), and, probability of hardware failure can be decreased by redundancy. While, for machine learning, the *probability* of failure can only be estimated empirically using testing data (and a set of good practices) i.e. like hardware failure but without redundancy option.

3.2 Unfairness

An other objection it about the risk of having the crash-allowed area detector performing less efficiently than uncontrolled crash. Of course, in average, it is hard to be worse than random ! Now, the answer is less clear on particular instance because the distribution of situation in which the detector is worse than random could be unfair.

Obviously, if the system prefers to crash into a low rent housing than into a expensive house, then this raise ethical issue.

Such issue should, of course, be tackled during evaluation (it is a part of good evaluation practices). But, again, this is just a matter of correct evaluation. Yet, a crash controlled system can not perform worse than an uncontrolled one (under correct evaluation process).

3.3 Instability

A very large academic effort has recently put emphasis on adversarial example [10, 4, 3, 14, 16, 12, 5]. Yet, does adversarial examples are an issue for crash-allowed area detection use case ?

No: again, here, the machine learning system is activated only under several hardware failure. So either, the probability of meeting an adversarial is not low,

and in this case, there will be such example in training data (natural training will be an adversarial training for free) and in the testing data, ensuring a correct estimation of the level of performance. Or either, the probability of meeting an adversarial is low, but, then the probability to meet and adversarial exactly during the hardware failure that activate the learning system is very low.

So, in the specific use case of crash-allowed area detection, it seems even not mandatory to required strong defence [19, 20], or, to consider performance under stress like in [7] (even if requiring both could be considered).

In fact, for the use case of crash-allowed area detection, the real issue with adversarial examples is the confusion that adversarial attack create in the community: certifying a neural network has nothing to do with proving that it does not admit adversarial failure. Proving that a neural network does not admit adversarial failure can be interesting (should be more precisely defined because any continuous function from $[-1,1]^D$ a compact to $[-1,1]$ will have a 0-surface), but it say nothing about the probability of failure of the network (it can not as the specification of the network are even unknown).

3.4 Incompatibility with the law

Despite previous reasoning, it seems clear that such system is today incompatible with the law at least the DO-178. Indeed, the DO-178 requires that software should cause no hazard. Dramatic hazard should only be due to hardware failure. Obviously, crash-allowed area detector is a software, but, failure in this software may lead to dramatic hazard. Indeed, in worst case, the software select the only dangerous area of crash leading to a worse situation than uncontrolled crash.

Yet, this system is incompatible with **current** law. Now, in the very specific use case considered in this paper, the is no good reason to forbid such software. First, machine learning system activate only under hardware failure (remote control plus GPS failure) - this way combined hardware-software failure probability can be interpreted as real probability. And, then, there is no alternative between machine learning based crash or uncontrolled crash. Currently, there is always alternative, but, here alternative means **no UAV**, because as soon as there are UAV, remote control plus GPS failure will happens. And, already today, middle weight UAV (sufficient to cause causality in case of uncontrolled crash) are allowed to plane.

3.5 Risk increasing

An other objection could be that if allowed such system could lead to a risk increase because constructors may integrate the gain provided by such system to allow higher risk on the hardware.

Typically, today, the probability of remote control failure is $10^{-\alpha}$, GPS failure is $10^{-\beta}$ and probability of causality during uncontrolled crash is $10^{-\gamma}$ with $\alpha+\beta+\gamma > 10$. But, tomorrow, by allowing machine learning based crash-allowed area detection with failure probability of $10^{-\phi}$, constructors could use

lower standard remote controller and GPS sensor with failure probability of $10^{-\psi}$ and $10^{-\varphi}$ such that $\alpha + \beta + \gamma = \phi + \varphi + \psi$ with $\phi > \gamma$, and, hence $\alpha + \beta < \psi + \varphi$. Indeed, if there is more doubt about the estimation of ϕ than about the estimation of $\alpha+\beta$. One could feel that the safety level has decreased.

Yet, in reality, the estimation of ϕ should be more accurate than the estimation of $\alpha, \beta, \gamma, \varphi + \psi$: failure probability on hardware is estimated in empirical fashion, but, with a lower control of the experiment than during a data acquisition campaign like for ϕ. And, γ is very coarsely estimated. So, the level of safety would instead be estimated more accurately.

Now, the question of the absolute level (10^{-9} for DAL-A) is out of the scope of this paper, but, there is no control on γ (which is just linked to the population density) while ϕ can be increased with machine learning advances (and can hardly be lower than γ). So, globally, it is a safety increment and not a safety issue.

3.6 Evaluation

So, the claim of this paper is that machine learning can be embedded and considered into safety analysis in some situation. Yet, the requirement is to be able to measure the average performance of the system i.e. the probability of failure - probability regarding the native distribution of the data.

Formally, in binary classification (which is the basic machine learning problem), one can build a system f which takes input $x \in \Omega$ and produces estimated class $f(x)$. The true class of x is $y(x) \in 0, 1$ but y is not known or decidable. Yet, one is able to evaluate y (for example by using human annotation) on sample. One is also able to drawn samples according to the native probability of the problem P. So, the real probability of failure e is

$$e = \int_{\Omega} |f(x) - y(x)| P(x) \mathrm{d}x \tag{1}$$

And, as one can drawn samples according to P and evaluate y on samples, then, one can approximate e using

$$\frac{1}{N} \sum_{n \in [1,N], x_1, \ldots, x_N \sim P} |f(x_n) - y(x_n)| \underset{N \to \infty}{\to} \int_{\Omega} |f(x) - y(x)| P(x) \mathrm{d}x \tag{2}$$

Issues with this empirical way to measure e are

- the measure has a variance with is problematic to measure small probability - yet this issue is a classical statistical problem with a large literature
- contrary to other probabilities appearing in a safety analysis, the probability P can change during the life of the system - this is an important machine learning issue to be able to mitigate this point

For this last point, let remark that our goal is to estimate

$$e' = \max_{Q \text{ realistically close of } P} \int_{\Omega} |f(x) - y(x)| Q(x) \mathrm{d}x \tag{3}$$

For example, it is know that performance under adversarial example attack is just $\max_{Q \text{ realistically close of } P} \int_{\Omega} |f(x) - y(x)| Q(x) \mathrm{d}x$ with Q close to P for a Earth mover's distance i.e. for each x such that $Q(x)$ is high, there is x' very close of x such that $P(x')$ is high i.e. x is an adversarial attack around x.

It is known that deep learning performance may strongly drop under such attack - it means that this issue of computing e' the extended probability of failure is important.

Yet, this is not as crucial as making impossible to embed machine learning in critical application (in particular in case where machine learning is used only under real random failure).

4 Conclusion

Probably, deep learning will not be massively present in autonomous vehicle, even more, to implement daily critical function. Yet, depending on the use case, incorporating system trained with deep learning into critical autonomous vehicle can be useful for safety consideration.

Typically, the paper consider that machine learning based crash-allowed area detection for autonomous UAV could be a very relevant system for UAV where the double remote control failure and GPS failure will otherwise conduct to a uncontrolled crash.

So in this specific use case, machine learning can improve safety instead being a safety issue. And, one could hope that community may use such use cases to close the gap between safety community and machine learning community.

This paper is the point of view of the author, and, should not be considered as the official position of ONERA on this issue.

References

[1] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In *Asian conference on computer vision*, pages 180–196. Springer, 2016.

[2] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images. *Remote Sensing*, 9(4):368, 2017.

[3] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th*

ACM Workshop on Artificial Intelligence and Security, pages 3–14. ACM, 2017.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[5] Pascale Gourdeau, Varun Kanade, Marta Kwiatkowska, and James Worrell. On the hardness of robust classification. *arXiv preprint arXiv:1909.05822*, 2019.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[7] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[8] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[9] Byoung S Kim and Anthony J Calise. Nonlinear flight control using neural networks. *Journal of Guidance, Control, and Dynamics*, 20(1):26–33, 1997.

[10] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[11] Adrien Lagrange, Bertrand Le Saux, Anne Beaupere, Alexandre Boulch, Adrien Chan-Hon-Tong, Stéphane Herbin, Hicham Randrianarivo, and Marin Ferecatu. Benchmarking classification of earth-observation data: From learning explicit features to convolutional networks. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 4173–4176. IEEE, 2015.

[12] Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4536–4543, 2019.

[13] Dimitrios Marmanis, Jan D Wegner, Silvano Galliani, Konrad Schindler, Mihai Datcu, and Uwe Stilla. Semantic segmentation of aerial images with an ensemble of cnns. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:473, 2016.

[14] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. ACM, 2017.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[16] Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? *arXiv preprint arXiv:1809.02104*, 2018.

[17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[18] David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.

[19] Eric Wong and J Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*, 2017.

[20] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.