XPlane-ML - an Environment for Learning and Decision Systems for Airplane Operations

Tyler C. Staudinger
Boeing Research & Technology
1000 Redstone Gateway SW
Huntsville, AL 35808
tyler.c.staudinger@boeing.com

Zachary D. Jorgensen
Boeing Research & Technology
1000 Redstone Gateway SW
Huntsville, AL 35808
zachary.d.jorgensen@boeing.com

Dragos D. Margineantu
Boeing Research & Technology
P.O. Box 3707, M/C 42-101
Seattle, WA 98124
dragos.d.margineantu@boeing.com

Abstract

The development of machine learning approaches can benefit significantly from the availability of high quality simulation environments and from architectures that allow an easy integration of the learners and simulation environments. The main research benefits of the availability of such learning-simulation settings are the enabling of fast architecting of the learning approaches, and exercising of infrequent or anomalous (edge/corner) cases that cannot be experienced and experimented with enough in real-world settings. In particular, preliminary machine learning experiments in highly regulated and controlled environments can benefit from a simulation setting tremendously until we, researchers, become familiar with the feature space, and the basics of the application problem. One such domain is airplane operations both on airport grounds (taxiing) and the air. The X-Plane simulation environment is an ideal simulation platform for that domain - for visual perception tasks in airport environments, and for decision making tasks. This paper describes an open source architecture for learning different tasks that enable the operation of airplanes, in conjunction with X-Plane. We focused primarily on developing an architecture that works for perception tasks, but have used this architecture and we discuss possible approaches for employing it in decision making tasks. We believe that opening our setting to the AI/ML, sensing, simulation, and pilots' communities will enable formulating. advancing, understanding better and solving a significant set of learning and decision tasks in airport and flight environments.

1 Introduction

Taxiing, identifying and ollowing center lines, detecting airport specific objects, runway incursions, identifying weather conditions (such as) lighting, deciding to abort a standard maneuver such as take-off, and various other scenarios are examples of perception and decision making tasks that occur in ariplane operations and can benefit from the assistance of learned models that d etect, classify, estimate and/or decide fast. Real-world data for these tasks is available only in small amounts and therefore, a critical component for developing and testing such systems is an architecture that enables the development and testing of learning approaches, and the advancing of research on machine

Preprint. Work in progress.

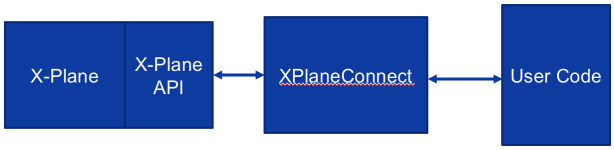

Figure 1: XPlane-ML architecture.

learning robustness and decision systems assurance questions. to be developed and robust visual detection systems, end to end control, and reinforcement learning systems.

We employed the proposed XPlane-ML environment for an initial case study: an estimator for the distance from the airplane's front gear to the center of the taxiway or runway (also referred to as the *cross track error* of the moving airplane), while taxiing. This estimator is to be employed as an input to the inner-loop controller of the airplane, for it to be able to follow the centerline while taxiing. Following the centerline of taxiways and runways is a regulatory requirement for taxiing airplanes on airport grounds.

2 XPlane-ML Overview

Figure 1 presents the architecture of the XPlane-ML. It is composed of the platform - the X-Plane airplane simulator (available from www.x-plane.com), XPlaneConnect, and the user's code - in our case a learner.

2.1 XPlaneConnect

XPlaneConnect is an open source toolbox developed by NASA, for interfacing with the X-Plane commercial simulation environment. It is available as a download from https://github.com/nasa/XPlaneConnect. XPlaneConnect allows access to all the states of the X-Plane software via functions written in C, C++, Java, MATLAB and Python, in real time.

XPlaneConnect has the role of exposing (for reading and writing) the airplane's states to the learner as follows. Each variable employed in X-Plane is accessed through *data references*, which are unique identifiers for simulator's variables. For example for the indicated airspeed, the data reference is:

```
dataref=sim/flightmodel/position/indicated_airspeed
```

A complete listing of all available data references is available in:

```
X-Plane\Resorces\plugin\dataRefs.txt
```

2.2 The Learning Compoment

Any learning algorithm(s) can be employed in the user's code. For the preliminary experiments and to demonstrate the usability of X-TaxNet , we chose MobileNetV2 [5] as the learner due to its extreme parameter efficiency lending to desirable generalization properties as well as fast inference time. The network achieves comparable performance to VGG19 on ImageNet classification with 40x fewer adjustable parameters in the network. For our task, a single 1-by-1 convolutional filter followed by a softsign activation function was used to replace the classification layers of the original architecture.

For training the regression model, a smooth L1 loss function was employed. Stochastic Gradient Decent with a learning rate of 0.001 and momentum of 0.9 was utilized along with learning rate decay which reduced the learning rate by 50% every 10 epochs. L2 regularization and standard affine data augmentations were used during the training process over 100 epoch to improve generalization.

2.3 Training and the Learned Model

XPlane-ML is set up to be employed by the user as follows: the images (screenshots) are taken from the airplane (X-Plane) via XPlaneConnect, at a rate of 50 Hz. The sky in the images is cropped for

Figure 2: Preliminary test results for estimating the cross track error using MobileNetV2 in the XPlane-ML architecture. The test data included images collected while taxiing on a different set of taxiways than during training.

each of the images followed by a resizing to 240-by-320 pixels, to maintain the aspect ratio. The images are then normalized to a range of -1 to 1 and fed into the network.

The output of the model will then be used as the cross track error estimate to be fed into the classical inner loop controller to enforce the desired center seeking behavior. The controller output is then to the XPlaneConnect interface and the state of the airplane's control surfaces are adjusted.

2.4 Sample Results

XPlane-ML is usable and configurable for training, evaluating and analyzing classifiers, detectors, estimators, or decision modules in conjunction with X-Plane. Figure 2 presents preliminary results of using MobileNetV2 as the learner for the estimation of the cross track error. The training data (500 samples) was collected on one set of taxiways while the test data was collected on a different set of taxiways of the same airport. For generating the training data for this sample task we needed to sufficiently cover the space of heading angles and cross track errors (driving straight down the runway would obviously not be sufficient). The training set for this sample experiment was collected in different lighting contrast and brightness conditions. Testing was performed in low light (cloudy) conditions.

Figure 3 includes several screenshots of XPlane-ML set for employing the learner as a cross track error estimator.

3 Summary

We discuss and propose an open sourced environment XPlane-ML that was designed for accelerating the research on learning components in the operation of airplanes on airport grounds. One case study is presented and discussed: estimating the cross track error (the distance from the taxi centerline) by using a deep learning approach based on MobileNetV2.

Aside of learning tasks on images, XPlane-ML can be used for training/testing learners on estimating/setting parameters of taxi and flight phases and also for computing optimal sequences of decisions. We have employed it for learning (Deep Q-learning) parameters of airplanes during the takeoff, climb and descent phases. The XPlane-ML setting is currently employed as a testbed for assessing the robustness of the learning components on the DARPA Assured Autonomy program (https://www.darpa.mil/program/assured-autonomy).

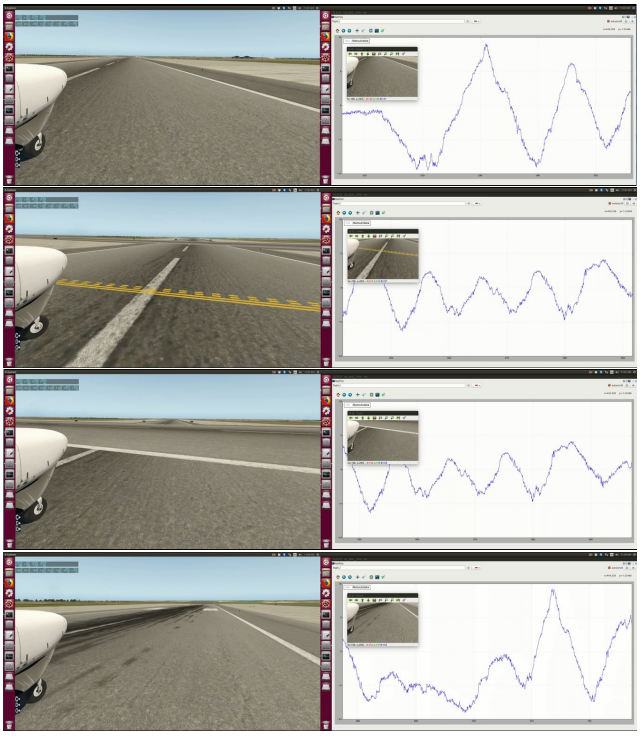

Figure 3: XPlane-ML using a trained MobileNetV2 as the estimator for the cross track error (right panel) while taxiing based on input from the X-Plane simulation environment (left panel)

The XPlane-ML will be open sourced by mid-November 2018 at least ten days before the start of the NIPS conference. The open sourcing of XPlane-ML is currently under an approval process by The Boeing Company.

References

[1] Abadi, M. et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[2] Bengio. Y. (2009). Learning Deep Architectures for AI. Foundations and Trends in Machine Learning 2 (1), 1–127.

[3] Changpinyo, S., Sandler, M. & Zhmoginov, A. (2017). The power of sparsity in convolutional neural networks. CoRR, arXiv:1702.06257.

[4] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[5] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv:1801.04381.

[6] Thrun, S. (1996). Is learning the n-th thing any easier than learning the first?. In Touretzky, D., Mozer, M., & Hasselmo, M. (Eds.), Advances in Neural Information Processing Systems 8 (NIPS'95), pp. 640–646. MIT Press.

[7] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research 11, pp. 3371–3408.