Learning-based Multirotor Control Enhancements

Eckart Cobo Briesewitz

April 16, 2024

1 Introduction

The rise in popularity of quadrotors has lead to a need in new methods for predicting their interactions with aerodynamic forces. A simple physical model of a quadrotors behaviour can not perfectly model the real movement of a quadrotor. Therefore there is a need for models that can learn these inaccuracies. In previous work researchers have been able to develop models for this very purpose. In the previous thesis from the IMRC Lab, a bachelor student designed two types of models to solve this issue. They can be divided into two categories, these being neural networks and decision trees (or rather decision tree ensembles). In this thesis we will take the two best performing models from the previous thesis and introduce them into a crazyflie's firmware to test how much performance gain there is.

2 Basic feed forward function

3 The models to be tested

As mentioned in the introduction the models can be divided into two categories each of which has it's own advantages and disadvantages that we will get into. Lets start with the neural networks.

3.1 Neural network

3.2 Decision tree ensemble

-Table comparing the models-

4 Uploading the models to the crazyflie

The crazyflie has an STM32 microcontroller which does not allow for the same flexibility one would find in the python code used to train the models, therefore uploading the models into this drone is no trivial task. In order to be able to run our models on this restrictive hardware we will have to find a way to transport the model paramenters into c code. Which is a language that runs very close to the hardware. Of course we will have to write different c-code/"python generator code" for each of two model structures. We will write a python script to translate the .pth (Neural Network from PyTorch) and .json (Decision tree ensembles from XGBoost) into compilable c code. Writing python code to generate the corresponding c code will also allow for a convenient way to transform future models into the necesary format for the crazyflie.
The bitcraze provides the crazyflie with an open source firmware. The models can of course not just be thrown into the firmware, they need to be able to read the state of the quadrotor and adjust the output of the controller with the predicted residual.

4.1 Neural Network

For the neural network we can divide the c code into some utilities that will stay the same for every model like a layer propagation function and scripts that store model specific information like weights and biases. In the utils_nn.h and utils_nn.c scripts you will find the layer function which takes the values of some previous layer, the weights and biases of the next layer and propagates the information. This is done via for loops as with the standard STM32 we can't simply paralelise the calculation like how PyTorch or other libraries for training models would do it.

Now we can take a look at the nn.h and nn.c scripts. These have been generated by a python script which takes a .pth file and reads it's contents. They contain a structure which defines the weights and biases of a neural network. They also define a convenient function called nn_forward which propagates a given input string through the network.

4.2 Decision tree ensemble

For the decision tree ensemble we have a similar system where the tree_utils.h and tree_utils.c files define a basic tree tranversal function and the generated files tree.h and tree.c are generated by the node values of the tree defined in the .json file outputed by the XGBoost library. In the tree.h and tree.c we define a tree_forward function which calculated the mean over all the trees in the ensemble.

4.3 Adding the models to the firmware

5 Testing the performance

To test the model performance we will take some predefined paths for the drone to follow and check how far it diverges from the planned path when trying to follow it. To do this we measure the drone state at each time step and compare it to the desired state. We first do this with the default controller in the crazyflie firmware and then switch to our new contoller inside of the firmware.

5.1 Neural Network

5.2 Decision tree ensemble

6 Results