Machine Learning Engineer Nanodegree

Capstone Proposal

Martin Simon
March 30th, 2017

Proposal

Domain Background

Finding optimal solutions in Mechanical Engineering requires thorough theoretical understanding of physics and practical experience in the domain to find good solutions. However, even skilled professionals experiment with a number of theories before finding the most suitable solution. Machine Learning algorithms can be used to find hidden patterns and automate the engineering process. Genetic Algorithms have already proven useful - proposing technical solutions, which humans would probably not find with human intuition. A great example of this is the design of a NASA satellite antenna: https://ti.arc.nasa.gov/m/pub-archive/1244h/1244%20(Hornby).pdf (https://ti.arc.nasa.gov/m/pub-archive/1244h/1244%20(Hornby).pdf). An examples of applying Machine Learning is Simulation Data Mining (http://www.joerg.frochte.de/field-of-interest/simulation-data-mining/). Handing tasks that can be automated and are based on a lot of "trial and error" is a significant improvement of an engineers workflow. However, physics simulation driven policies can act in other applications as well, such as real-time decision making (http://groups.csail.mit.edu/robotics-center/public_papers/Tedrake09.pdf).

I am currently designing a Blended Wing Body Unmanned Aerial Vehicle (UAV and am interested in experimenting with Machine Learning algorithms to help the solvers make "intelligent" decisions to speed up the development process. The policy will be using open-source softwares to generate geometry and get evaluations for the proposed geometries: XFLR5 (http://www.xflr5.com/xflr5.htm), XFoil (http://web.mit.edu/drela/Public/web/xfoil/), OpenVSP (http://openvsp.org/) and AeroPython (http://lorenabarba.com/blog/announcing-aeropython/).

Doing initial research, there is a limiting factor when solving a multi-objective optimization problem with a high number of input parameters. The described methods for finding a global minima by using genetic algorithms or brute data mining rely heavily on unexpensive evaluations. Once evaluation computational expense goes up (e.g Computational Fluid Dynamics based aerodynamics analysis can take several hours and even days) the total number of evaluations become limited. Deep neural nets based surrogate modelling has been used to substitute these expensive evaluations. Although very promising (https://www.autodeskresearch.com/publications/convolutional-neural-networks-steady-flow-approximation), they are not yet widely used nor approved. Another approach considered to solve this issue was to create a Deep Q Learning based policy, which could learn physics intuition similarly to a human engineer with low-cost evaluations and Radial Basis Functions based surrogate models. This initial intuition would be gradually enhanced with computationally more costly simulations. Unfortunately while doing preliminary experiments to support this proposal, similar limitations to attaining the intuition were found to a paper (https://arxiv.org/pdf/1611.01843v1.pdf) that pursued it using DeepMind.

Therefore a combined method of Bayesian Optimization and surrogate modelling is used to help give an approximation of the Pareto front with fewer expensive objective function evaluations. This set of non-dominated solutions is a useful tool to guide decision making process of the UAV

Problem Statement

Apply Bayesian Optimization to find Pareto-optimal solutions for a multiobjective problem: design a Blended Wing Body Unmanned Aerial Vehicle. The optimal solution is determined by evaluating the UAV-s Coefficient of Lift, Drag and stability factors.

Datasets and Inputs

The preliminary geometric model is built by myself, making some broad assumptions about where the useful model should be. Some geometric limits will be set to make sure it is manufacturable and can carry electronics. The initial mutations of the model will be random. The preliminary input model will be based on airfoil sections, thus giving the policy the option to remember what worked on one section and use that knowledge to make decisions on the next.

For modelling and analysis open-source airfoil/wing platforms will be used, such as XFoil, XFLR5 and OpenVSP.

Solution Statement

The solution is to minimize the required number of evaluations to find a global Pareto optimal solution matrix. The matrix consists of non-dominated objective values (e.g Coefficient of Lift, Coefficient of Drag) combined with decision space values (e.g airfoil geometry, angle of attack). To do so after initial decision space parameters are evaluated the following datapoints evaluated will be chosen by combining entropy based probability (e.g Gaussian) functions and objective space optimization.

Benchmark Model

The benchmark model is the human engineer who tinkers the initial geometric model of the UAV to have the best attributes of the stability, drag, lift and weight. These values will be collected before the ML-based policy is initiated.

Evaluation Metrics

Evaluation metrics are stability, drag, lift and weight values. These metrics are highly conflictive and do not have a linear trade-off. The stability for example depends on induced drag effects and the lift distribution over the airfoil. In theory a reasonably good stability and lift can be attained, while having low drag at a given environmental condition. On the other hand: the weight of the aircraft is directly influenced by the area of the airfoil. In general, for an optimal low-speed airfoil it is typical to have effects to extend laminar flow at least to 30% of the chord and to move the turbulent transition slowly to aft. Also, the leading edge negative pressure peaks should be softened by a large leading edge radius and a thicker aft camber on the positive pressure side to increase lift and stability (source) (https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19740003708.pdf). The metrics are calculated by applying Finitie Element based methods to numerically find approximations of the values. The three most common methods on solving them are Boundary Element Method, Vortex Panel Method and Computational Fluid Dynamics (CFD) with Finite Control Volumes. While the first two exploit interpolation of a 2D airfoil evaluations, the third finds to some extent the equilibrium of the Navier-Stokes equations. Although, it should be mentioned that even very exact (and computationally costly) evaluations using CFD is still an approximation and is not prone to converge due to singularity mishaps. Because of these reasons initially the metrics considered will be attained from methods based on 2D approaches. If possible a 3D CFD case will also be reviewed, but this would not directly describe the quality of the optimizinf algorithm.

Project Design

Finding an aerodynamically optimized body for the UAV begins with constructing an initial model and sets of constraints by the human engineer. The policy is given a set of parameters it can modify and is expected to learn which combinations will give the best effect when constrained by payload size, weight, UAV range and endurance (time) expectance, the stall margin (m/s), minimum airspeed (m/s), available power etc.

The project workflow:

1. Set up aerodynamic analysis programs and make a decision which of them are most beneficial for the project.
2. Build a simple Python-API to mediate with the selected programs.
3. Create a set of constraints and initial model.
4. Decide or randomize initial model parameters within constraints.
5. Fine-tune the set-up of the Multi-objective Pareto optimality system.
6. Set up Bayesian Optimization algorithm based acquisition function.
7. Create systems for surrogate models to work back from the points proposed by the acquisition function.
8. Train final policy and evaluate results.

```
In [ ]:
```