

# AUTOMATED DESIGN USING NEURAL NETWORKS AND GRADIENT DESCENT

**Oliver Hennigh**

Mexico

Guanajuato, Gto, Mexico

loliverhennigh101@gmail.com

## ABSTRACT

In this paper, we propose a novel method that makes use of deep neural networks and gradient descent to perform automated design on complex real world problems. Our approach works by training a neural network to mimic the fitness function of the optimization task and then, using the differential nature of the neural network, we perform gradient descent to maximize the fitness. We demonstrate this methods effectiveness by designing an optimized heat sink and both 2D and 3D wing foils that maximize the lift drag ratio under steady state flow conditions. We highlight that our method has two distinct benefits. First, evaluating the neural networks prediction of fitness can be orders of magnitude faster then simulating the system of interest. Second, using gradient descent allows the design space to be searched much more efficiently.

## 1 INTRODUCTION

Automated Design is the process by which a object is designed by a computer to meet or maximize some measurable objective. This is typically performed by modeling the system and then exploring the space of designs to maximize some desired fitness function whether that be automotive car styling with low drag or an integrated circuit with small profile. The most notable historic example of this is the 2006 NASA ST5 spacecraft antenna designed by a evolutionary algorithm to create the best radiation pattern. More recently there has been (Flow Sculpter) and (optical computer). While there have been some significant sucesses in this feild the dream of true automated is still far from realized. The main challanges faced are heavy computational requirements for accuratly modeling the physical system under investigation and often exponential large search space. These two problems negatively complement eachother making the computation requirements intractable for even simple problems. For example, in the realively simple flow problems explored in this work, a heavily optimized flow solver running on modern gpus requires around 5 minutes to perform each simulation. Given that as many as 3,000 designs need to be tested to acheive reasonable performance, this results in a total computation time of 9 days on a single GPU. Increassing the resolution of the simulation or expanding the parameter space quickly make this an unrealizable problem without considerable resources.

Our approach works to solve the current problems of automated design in two ways. First, we learn a computationally effeicient representation of the physical system on a neural network. This trained network can be used evaluate the quality or fitness of the design several orders of magnitude faster. Second, we use the differentiable nature of the trained network to get a gradient on the parameter space when performing optimization. This allows significantly more efficient optimization requiring far fewer iterations then other methods. These two abilitys of our method overcome the present difficulties with automated design and allow the previous mention 9 day optimization to be run in only 10 mins. While we only look at two relatively simple problems in this work, we emphasize that the ideas behind our method are applicable to a wide variety of automated design problems.

The first problem tackled in this work is designing a simple heat sink to maximize the cooling of a heat source. The setup of our simulation is ment to somewhat minime the conditions seen in a heat sink on a computer processor. We keep this optimization problem realatively simple though and use this only as a first test and introduction to the method.

We also test our method on the significantly more difficult task of designing both 2D and 3D wing-foil with high lift drag ratios under steady state flow conditions. This problem is of tremendous importance in many engineering areas such as aeronautical, aerospace and automotive engineering. Because this is a particularly challenging problem and often times unintuitive for designers, there has been considerable work using automated design to produce optimized designs. We center much of the discussion in this paper around this problem because of its difficulty and view this as a true test our methods advantages and disadvantages.

As we will go into more detail in later sections, in order to perform our flow optimization tests we need a network that predicts the steady state flow from an objects geometry. This problem has previously been tackled here where they use a relatively simple network architecture. We found that better perform could be obtained using some of the modern network architecture developments. For this reason, in addition to presenting our novel method of design optimization, we also present this superior network for predicting steady state fluid flow with neural networks.

This work has the following contributions.

- We demonstrate a novel way to use neural networks to greatly accelerate automated design problems.
- We present this method in such a way that it can readily applied to many other automated design problems.
- We provide a new network architecture for predicting steady state fluid flow that vastly outperforms previous models.

## 2 RELATED WORK

This touches on several vastly different subjects. Because of this we have provided a brief review of the related areas.

### 2.1 NEURAL NETWORK BASED SURRAGATE MODELS

Our method uses a network to predict steady state fluid flow from boundary conditions. This idea was first presented in (Guo et al., 2016) where they showed. Our flow prediction network has several key difference to this original work. First, we heavily improve the network architecture by keeping the network all convolutional and taking advantage of both residual connections and a U-Net architecture. This proved to drastically improve accuracy while maintain fast computation. Second, it takes in the binary representation of the boundary conditions instead of the Signed distance map. We found that with our improved network architecture, we over came the issue in their work using such a representation of the boundary.

### 2.2 AUTOMATED DESIGN OPTIMIZATION OF AIRFOILS

To date, there has been substantial work in automated aerodynamical design for use in aeronautical and automotive applications. The standard work flow is to first parameterize the search space of designs then iteratively simulate and evaluate the. We use the A variety of search methods have been used with some success in the optimization process. Simulated Annealing, Genetic Algorithms and Particle swarm have all been used with varying degrees of success.

Automotive Aerodynamic Design Exploration Employing New Optimization Methodology Based on CFD

Adaptive Shape Parameterization for Aerodynamic Design

Pros and Cons of Airfoil Optimization 1

Multi-level CFD-based Airfoil Shape Optimization With Automated Low-fidelity Model Selection

A Surface Parameterization Method for Airfoil Optimization and High Lift 2D Geometries Utilizing the CST Methodology

A Universal Parametric Geometry Representation Method CST

### 3 GRADIENT DECENT ON PARAMETER SPACE

Our automated design optimization problem can be viewed in concrete terms as maximizing some desired fitness function  $F(x)$  where  $F : X \rightarrow \mathbb{R}$  for some space  $X$  of design parameters.

$$\max_{x \in X} F(x) \quad (1)$$

In some real world setting, evaluating the fitness function  $F$  can be computationally demanding as is the case with our fluid simulations. The first aspect of our work is to replace  $F$  with a computationally efficient neural network  $F_{net}$ . This can offer considerable speed improvements as we will discuss below. The second aspect of this work is the observation that  $F_{net}$  is differentiable and provided that the parameter space  $X$  is real valued we can obtain a usable gradient in the direction of maximizing the fitness. This allows gradient decent to be performed where as in most setting  $F$  is not differentiable and requires other search techniques such as simulated annealing or genetic algorithms. This allows faster optimization to be performed with fewer iterations. The requirement of  $X$  to be real valued presents some challenges though. To show case this and our solutions to them we go through the example problem of optimizing the fin height on a heat sink.

In our simple heat sink problem,  $X$  contains 15 real valued parameters between 0 and 1. Each of these parameters corresponds to the height of an aluminum fin on the heat sink as seen in the figure. In our optimization problem we assume that there is a fixed amount of aluminum and scale the total length of all the fins to meet this requirement. This presents an interesting problem of determining the optimal length each fin should have to maximize the cooling of the heat source. The simplest application of our method would be to learn a neural network to take in the 15 fin height values and output a single value corresponding to the temperature at the heat source. This approach has the drawback that if you want to add another aspect to the optimization like making sure the left side is cooler than the right side you would need to retrain the network. Another solution is to have the network again take in the fin parameters but output the full heat distribution of heat sink. This allows different quantities to be optimized but is still limiting in that our network only runs on a single parameter set up. Our solution to this problem is to train two networks. The first network,  $P_{net}^{heat}$ , takes in the fin parameters and generates a binary image corresponding to the geometry of the heat sink. We refer to this as the parameterization network. The second network,  $S_{net}^{heat}$ , predicts the steady state heat distribution from the geometry. Because the parameterization network is performing an extremely simple task and training data can be generated cheaply, we can quickly retrain  $P_{net}^{heat}$  if we want to change the parameter space. The same approach is used for the steady state flow problem and a figure depicting this can be found here. This approach allows our network to be as versatile as possible while still allowing it to be used on many design optimization tasks.

Up until now we have not discussed

#### 3.1 FLOW PREDICTION NETWORK

The core component of our method is being able to emulate the physics simulation with a neural network. For the steady state flow problem there has already been work doing just this found here. As mentioned above, we have made some improvements to this network architecture design. A figure illustrating our network can be found here. This model resembles the u-network architecture seen here with a series skip after each down sample. The advantages of this style of network are its high performance on image to image type tasks, trainability on small datasets, and fast evaluation time in comparison to networks with large fully connected layers. The trainability on small datasets makes this particularly effective on predicting steady state flow because generating simulation data to train on is time consuming. Our network is able to train on relatively small datasets of only 4,000 flow simulation in comparison to the 100,000 required in previous work predicting steady state flow. Other modifications we have used are the use of gated residual blocks that allow the gradient to be propagated extremely efficiently and heavily lowers training time.

## 4 EXPERIMENTS

In the following sections we subject our method and model to a variety of test. The goal the goal being first to determine how accurately and fast our model can predict steady state flow. In particular, what its accuracy in predicting values such as drag and lift as these are important quantities in our optimization. Second, we investigate how effective our gradient descent optimization is. This line of tests compares our method to other non gradient based search techniques and illustrates what is happening in the optimization processes looks like. For example, what does the gradient surface look like.

#### 4.1 HEAT SINK OPTIMIZATION

## 4.2 FLOW PREDICTION ACCURACY

Steady state heat diffusion proved to be an extremely simple problem for our network however steady state fluid

As mentioned before, our model significantly out performs priour work predicting steady state fluid flow.

```
./test/figs/generated_flow_difference.jpeg
```

Figure 1: Flow

```
./test/figs/flow_accuracy_2d.jpeg
```

Figure 2: Flow

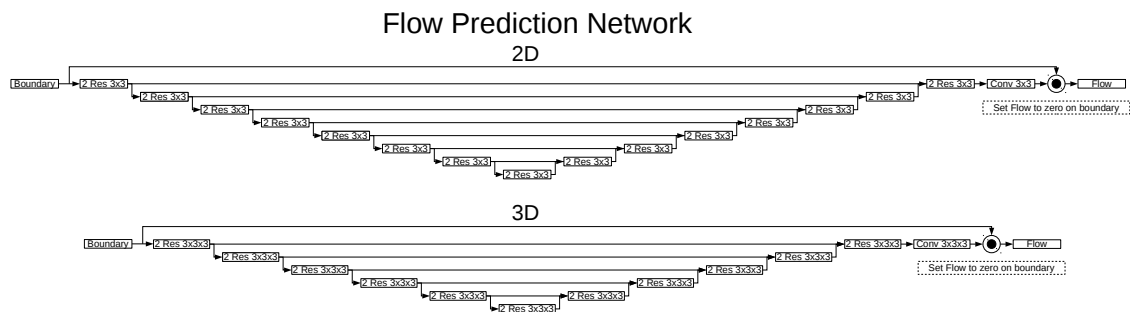


Figure 3: Flow

### 4.3 AUTOMATED DESIGN OF 2D AND 3D AIRFOILS

## 5 CONCLUSION

In this work we hav

## 6 SAMPLE TABLES

## ACKNOWLEDGMENTS

This work was made possible through the <http://aigrant.org> run by Nat Friedman and Daniel Gross. This work would not have been possible without this very generous support.

Table 1: Sample table title

Batch Size	1	2	4	8	16
Flow Net 512 <sup>2</sup>	0.150 sec	0.101 sec	0.077 sec	0.065 sec	0.058 sec
Param Net 512 <sup>2</sup>	0.083 sec	0.045 sec	0.026 sec	0.015 sec	0.011 sec
Learn Step 512 <sup>2</sup>	0.494 sec	0.345 sec	0.270 sec	0.231 sec	Nan
Flow Net 144 <sup>3</sup>	0.826 sec	0.686 sec	0.627 sec	0.623 sec	Nan
Param Net 144 <sup>3</sup>	0.195 sec	0.144 sec	0.119 sec	0.106 sec	0.093 sec
Learn Step 144 <sup>3</sup>	3.781 sec	Nan	Nan	Nan	Nan

## REFERENCES

Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 481–490. ACM, 2016.