5710783 Project Final Report

1st Kourosh Ghahramanimarangalou
1787001
e178700@metu.edu.tr

***Abstract*—The aim of this project is to predict aerodynamic polar coefficients of any 2D airfoil-like shape. To test this idea first a small data-set of known airfoils was created and analysed. Training different MLPs and CNNs on this data and testing it on another family indicated that this data-set is not enough to represent the variations in shapes. As a result, Class Shape Transformation (CST) Methodology was used to create 16000 new and random airfoil-like shapes. Each generated airfoil shape is then analyzed in a Reynolds number range from 1 Million to 8 Million, with a delta Re of 500k, and in an angle of attack range of -5 to 15 degrees, with a delta alpha of 1 degree. The test set was created in the same manner from the known 1500 airfoils in Airfoil database. Then different MLPs and CNNs were trained on the data. Infinity norm $||L||_\infty$ and percent error were used as the metrics to investigate the performance of the model. Current best model has $||L||_\infty$ of 0.31 and 94% of the predicted aerodynamic polar fall in 8% error range. It does the calculations for 1.2 million data points in 16 seconds compared to 3 days required to run XFOIL simulations.**

***Index Terms*—Aerodynamic polar, XFOIL, MLP, CNN**

I. Introduction

Wing profiles or airfoils are tear like shapes that can be seen in cross section of aircraft and bird wings. When subjected to an upstream they will generate forces that have components parallel and perpendicular to the flow. The normal force is called lift and the parallel force in the direction of the flow is called drag. Effects of airfoil shape on the calculation of forces is represented by lift c_l and drag c_d and pitching moment c_m coefficients. Obtaining these coefficients required long experiments and measurements which was not feasible. So some methods were developed to estimate these coefficients. Panel method and computational fluid dynamics (CFD) are the two most widely used techniques to estimate these coefficients. CFD utilises Navier-Stokes equations which are the nonlinear equations describing the fluid motion. The coefficients are obtained by post processing the solutions after it converges which takes quite long time for even one configuration. It also requires a volume mesh to represent the flow continuum. The panel method on the other hand utilises the potential flow theory and is way faster than the CFD method. The down side of it is that it is not accurate and can not represent viscous flow and like CFD it iterate to get to a solution. Although our method is almost 20000 times faster than Xfoil simulations.

Airfoils are the main reason that aircraft fly, Propellers and jet engines generate thrust and wind turbines harness the wind energy. Designing and optimizing them directly effects the performance of the aforementioned systems. The coefficients for each shape is calculated through the methods described. This process has some limitations. First of all it takes quite some time and often brute force is used to find a good design. Because of the first reason often a small determined family of airfoils (for which their aerodynamic coefficients are obtained experimentally)is used in design which is not optimal. Flight conditions and requirements might need analysis of airfoils for different Reynolds Numbers (Re). Using CFD requires creating a new mesh for each airfoil and the quality of the mesh dramatically effects the results. These reasons and many more would indicate that there is need for a generic and fast method that can solve the mentioned problems.

In recent years, data driven techniques are becoming increasingly popular and deep learning techniques are being implemented in a variety of aerodynamics problems. Wallach et al [3] investigated obtaining the aerodynamic coefficients for the NACA 23012 airfoil using a Multi Layer Perceptron (MLP) neural network trained using results from Xfoil simulations at 125 data points. The aerodynamic coefficients of a transport twinjet geometry as well as a wing-body configuration were also investigated in their study. Their neural network models were able to successfully capture variations of lift and drag coefficients for the geometries investigated in their paper. Dos Santos et al [4] parameterized airfoil geometries using Sobieczky polynomial functions [5] and generated aerodynamic databases for limited angle of attack ranges, either using Xfoil or MSES depending on Mach number regime, for thousands of airfoil geometries at fixed Mach or Reynolds numbers. These datasets are then used in training MLP neural networks for prediction. Their results showed good agreements for airfoils from their validation set but they concluded that the accuracy of their predictions for lift and drag coefficients is not sufficient for the neural network model to be included in design optimization studies. Sun et al [6] investigated airfoil inverse design using neural networks. They have generated an aerodynamic database for 208 airfoils by solving full potential equation for transonic flow conditions and trained neural networks to obtain an input-output relationship between aerodynamic coefficients as input parameters and the shape of the airfoil as output, which was characterized using 11 parameters of PARSEC methodology. They concluded that neural networks can successfully be used in airfoil and wing inverse design studies with enough accuracy and efficiency. More recently Zhang et al [7] trained a convolutional neural network to predict the lift coefficient. The training is performed using an aerodynamic coefficient database that they have generated using Xfoil simulations of 133 airfoils from the UIUC airfoil database in a wide Reynolds and Mach number range. To enhance the data quality they also created flipped airfoil data using the symmetry for the lift coefficient, totaling a number of 80,000 data points for training.

This paper presents the results of a deep learning implementation study to predict aerodynamic coefficients through training a neural network using a large database of Xfoil simulations for 16,000 different airfoil shapes in a wide range of Reynolds numbers and angles of attack. The airfoil geometries are randomly generated using Class Shape Transformation (CST) Methodology. Each generated airfoil shape is then analyzed in a Reynolds number range from 1 Million to 8 Million, with a delta Re of 500k, and in an angle of attack range of -10 degrees to 15 degrees, with a delta alpha of 1 degree. Mach number is kept constant and the n factor is fixed at 9 to simulate free transition conditions. The resulting data set, which consists of approximately 3 Million data points for lift, drag and moment coefficients each, is used to train the deep learning model. The performance of the trained network is tested with NACA 4 digit airfoils as well as with DU series airfoils, which were not used in the training process.

II. DATA

A. Airfoil Generation

A typical airfoil shape is shown in Fig. 1. I use Bernstein polynomials to decode the airfoil shapes. The multipliers of each polynomial acts as the airfoil encoding and selected randomly (normally linear regression is used to obtain the coefficients of a known airfoil). Upper and lower surfaces of the airfoil is generated using the equations in Fig.2.

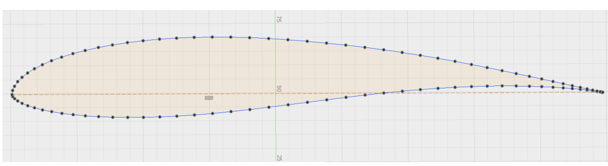

Fig. 1. Shape of an airfoil

$$\varsigma_U(\psi) = \psi^{0.5}.\,(1-\psi)^{1.0} \sum_{i=0}^{N_U} \left[A_U(i).\frac{N_U!}{i!(N_U-i)!}.\psi^i.(1-\psi)^{N_U-i}\right] + \psi.\Delta\varsigma_U$$

$$\varsigma_L(\psi) = \psi^{0.5}.\,(1-\psi)^{1.0} \sum_{i=0}^{N_L} \left[A_L(i).\frac{N_L!}{i!(N_L-i)!}.\psi^i.(1-\psi)^{N_L-i}\right] + \psi.\Delta\varsigma_L$$

Fig. 2. CST Decoder equations

the first and last weighting coefficients for the upper and lower surface A_U and A_L are known to determine the leading edge radius and the trailing edge angle, respectively. Rest of the weighting coefficients vary the thickness distribution in general [9]. Since our main objective in this study is to train a neural network for learning the correlation between the variation in shapes and their corresponding aerodynamic polar, we generate a wide range of airfoil shapes based on randomly selected values for the weighting coefficients A_U and A_L, for the order of Bernstein polynomials (N) as well as for the exponents of the class shape functions.

The random selection process is performed using a uniform distribution in order to sample a stochastic range of shapes. The order of Bernstien polynomials are randomly selected from integers between 2 and 10, and the class function exponents are selected from the ranges 0.45 to 0.55 and 0.95 to 1, instead of using constant values of 0.5 and 1. The first weighting function that determines the leading edge radius is selected within a range of 0.01 to 0.05, and last weighting function for the trailing edge angle is selected from integers between -10 to 20. The rest of the weighting coefficients are selected within a range of -0.1 to 0.6 for the upper surface and -0.6 to 0.1 for the lower surface. The upper and lower boundaries for these ranges were selected such that the upper and lower surfaces will not cross and intersect with each other. The selection of these ranges allowed us to create a wide range of airfoil shapes, some of which actually can be "impractical" for aerodynamic applications. However these wide range of airfoils provided large enough shape variations for the neural network to establish a good link between these shape variations and aerodynamic coefficients. The "impractical" airfoils were intentionally not excluded from the Xfoil calculations, as long as Xfoil provided converged results for the simulations.

Using the methodology described above we have generated 16000 airfoil shapes and analyzed all of them using Xfoil. Simulations for each airfoil were done in a Reynolds number range from 1 Million to 8 Million, every 500k, and in an angle of attack range of -10 degrees to 15 degrees with a step of 1 degree. As expected some runs at certain airfoil, Reynolds number and angle of attack combinations did not converge. Using the converged results we obtained 2,256,624 data points in total for training (total number of converged airfoil, Reynolds number and angle of attack cases). Of course for each of these cases the data-set includes the aerodynamic coefficients CL, CD and CM.

B. Analysis

For the analysis I used modified panel method in Xfoil program. I have prepared a file manager to run the generated airfoils in Aerospace Engineering's cluster on 192 cores. Except the angle of attacks where Xfoil might not converge I obtained a data-set of approximately 8000 (airfoils) x 20 (angles of attack) x 16 (Re) x 1 (N_{crit}) which is roughly 2.5 million data examples for 256000 runs.

C. Training/ Test sets

Training set is created from the random airfoils as mentioned before. For the test set I have used the 350 NACA-4 airfoils and 10 wind turbine airfoils -which are considered as hard examples to solve- the inputs were generated for Re 500k - 8M with increments 50k (one 10'th of the increments used for the training set). The alpha and (N_{crit}) values were the same as the training data.

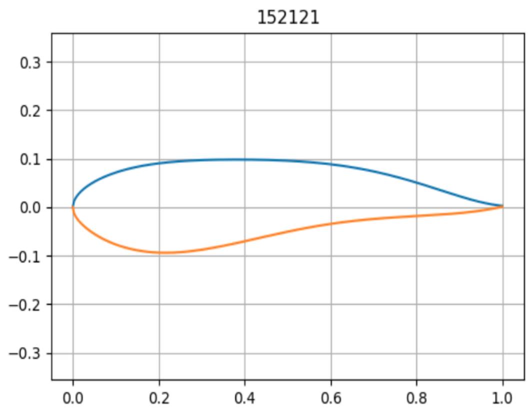

Fig. 3. A random airfoil selected.

D. Network inputs

Fig.3. shows a random airfoil selected from the data-set.

In order to get better simulation results the half of the points were generated in between the [0, 0.15] range in the longitudinal direction. But when creating the inputs a cubic spline was used to generate new points on a uniform grid. Furthermore I flatten the y values of the airfoil to get a vector and then append the normalized Re and alpha values to it.

E. Output

Initially I wanted to predict all the polar values but after training the first network with reasonable results I saw that the error for the drag coefficient is higher than the others so I decided to work on the drag coefficient alone. Fig.4. shows the predictions of the initial network.

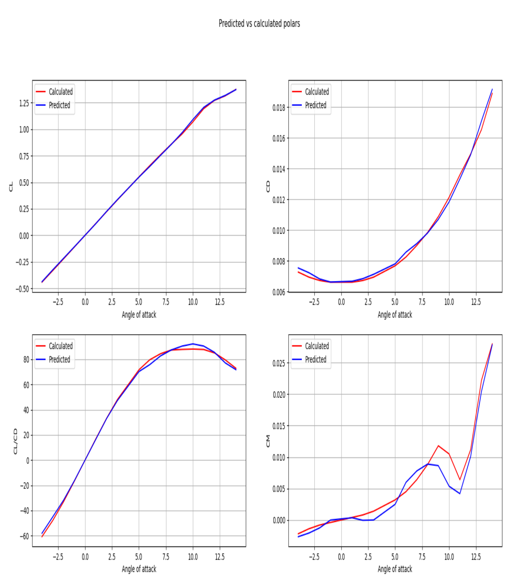

Fig. 4. Output of network vs ground truth for a range of inputs

The network presented here is trained only for drag coefficients. The output values are scaled by a factor of 5.

III. Tested Ideas

A. Activation Function Selection

The output values (aerodynamic polar) are signed real numbers. ReLU activation function was not used here for two reasons, firstly because when ReLU is applied as an activation function in a layer, neurons become strictly positive but the problem demanded both negative and positive results (It can be used in the hidden layers) and secondly because its gradient is zero for negative inputs. Leaky ReLU was selected for this problem and when I compared the results of the same network architecture with ReLU activation functions I noticed the loss of the network with leaky ReLU dropped further.

B. Initial Multi Layer Perceptron

The first architecture that I tested had 4 hidden layers were each consist of 512, 256, 200, 128 neurons respectively, and four outputs Fig.4. shows the network results on a random airfoil. The network performed very well for one of the outputs and badly for other two (pitching moment and drag coefficient). The pitching moment coefficient is not an important parameter concerning the aerodynamic performance of the airfoil, whereas the drag coefficient is the most import parameter defining the efficiency of the airfoil. After consulting with my supervisor I decided to split it and work on drag coefficients where the network performed poorly.

Some small modifications to the network increased the performance although it was not significant.
At that point I decided to try some other architectures and see if I am able to predict the aerodynamic coefficients better. As I stated to test deeper networks the the training and test errors both started to increase. To overcome that I have tested many architectures with skip connections (multiplying the skip connection with different activation functions even with periodic functions, used identity skip connections as well as training the weights of the connection), Tested a network with two loss functions where one of them was applied randomly to one of the layers at each iteration -dropped the error approximately to the same order as the residual type architectures- (still working on it). All the networks that I have tested showed slight improvement on the results.

C. Convolutional neural network

While using MLPs we thought of the airfoil shape as some distribution (the lateral values of the airfoil do no exit 0.35 range and the length is 1 unit, so one can put upper and lower surfaces besides each other and treat them as some distribution) but actually when solving for aerodynamics of the airfoil we use the slope of two consecutive points. So one might think that using 1D convolution would incorporate the same effect and lead to better results. My experiments on the data using 1D convolutions on the data or applying 2D convolution on the results of the first 1D convolution (treating the channel as height of the 2D convolution input) were unsuccessful. Both couldn't decrease the training loss to less than 10E-4 in 400 iterations. And because testing them took significantly longer I gave up the idea for the time being.

IV. Final Model

The final architecture that was selected for this problem (and provided) consists of 20 hidden layers with 500 neurons each, where skip connections go over two fully connected layers. The training for this network was stopped at loss 67e-8. The results of the network on the NACA-4 family and wind turbine airfoils is presented in Fig.5 and 6.

92% of the predictions for NACA-4 family drag coefficients have less than 5% error and 97% of them are in the 10% error bound. The network takes about 16s to predict 1.2M data points of the NACA-4 test set and when we test the network for the wind turbine airfoils we observe that the 78% less of the predictions have than 5% error and 97% of them are within 10% error bound and the whole run takes about 0.3s.

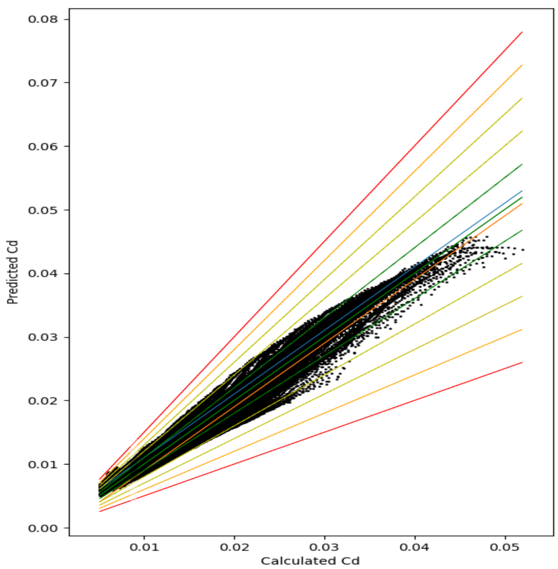

Fig. 5. Output of network vs Xfoil analysis of NACA-4 airfoils.

V. Conclusion

The error of the results mainly comes from the fact that the XFOIL analysis error increases with increase in the alpha, and the fact that some of the training data did not converge during the XFOIL simulations. Aside from that the results obtained are satisfactory and significantly better than the previous studies.

VI. References

References

[1] Drela, M., "XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils," in Low Reynolds Number Aerodynamics, Berlin, Heidelberg, 1989.

[2] Drela, M, "A User's Guide to MSES 3.05," MIT Department of Aeronautics and Astronautics, Massachusetts, 2007.

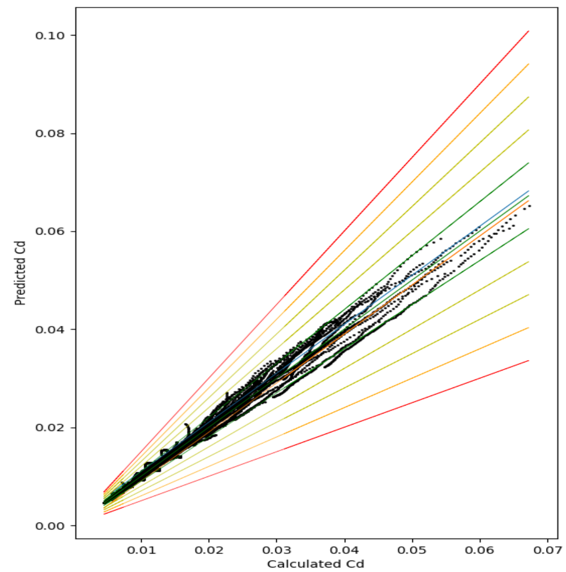

Fig. 6. Output of network vs Xfoil analysis of Wind Turbine Airfoils.

[3] Wallach , R., Mattos, B. S., Roberta, G. d., and Curvo, M., "Aerodynamic Coefficient Prediction of a General Transport Aircraft Using Neural Network," in 44th AIAA Aerospace Sciences Meeting and Exhibit, Nevada, 2006.

[4] dos Santos, M. C., de Mattos, B. S., and Girardi, R. d., "Aerodynamic Coefficient Prediction of Airfoils Using Neural Networks," in 46th AIAA Aerospace Sciences Meeting and Exhibit, Nevada, 2008.

[5] Sobieczky, H., "Parametric Airfoils and Wings," Notes on Numerical Fluid Mechanics, vol. 68, no. Vieweg, pp. 71-88, 1998.

[6] Sun , G., Sun, Y., and Wang, S., "Artificial neural network based inverse design: Airfoils and wings," Aerospace Science and Technology, pp. 415-428, 2015. doi: 10.1016/j.ast.2015.01.030

[7] Zhang, Y., Sung , W., Mavris, D., "Application of Convolutional Neural Network to Predict Airfoil Lift Coefficient," in AIAA SciTech Forum, Kissimmee, Florida, 2018. doi: 10.2514/6.2018-1903

[8] KKulfan B.M., "Universal Parametric Geometry Representation Method," Journal of Aircraft, Vol. 45, No.1, January-February 2008.

[9] Oğuz, K., "Aerodynamic Optimization of HAWT Blades by Using CST Method, BEM Theory and Genetic Algorithm," MSc Thesis, Middle East Technical University, 2019.