Hyperparameters: Nr. of epochs, learning rate, batch size, num workers, input size, output size

To implement: Test results, linear regression (for Präsi, Doku)

1. **Machine-Learning-Enabled Foil Design Assistant** ++

Forward and Reverse Design (Optimization) -> Foil Design Assistant Software

Reverse: XFOIL for coefficients

17 Parameters, 13 control points to describe the foil shape

XFOIL: 200Points, cubic spline interpolation with the remaining CL and CD values, 9 alphas, Re=8x10E6

UIUC database of foil designs (1500 foil profiles encoded via point sets of varying lengths) Dataset Vectors

Regression ANN models, Multivariate Linear Regression (MLR), feedforward Artificial

Neural Networks trained with Levenberg–Marquardt (LM) back-propagation, and LM with Bayesian Regularization, Layers: Input 17-32 Parameters, Hidden (20-30 Neurons), Output (9-19 values), hyperbolic tangent sigmoid activation functions

1. **Multi-Objective Optimization of Low Reynolds Number Airfoil Using Convolutional Neural Network and Non-Dominated Sorting Genetic Algorithm** ++

Direct problem: airfoil shape -> coefficients, CNN, “classical validation case for low Reynolds number airfoil data is E387 at a Reynolds number of 2.0E5”, alpha=4.5°, XFOIL + CFD comparison for coefficients, 15 low Re-Numbers, Images (greyscale 160x160 were used) for airfoil shape, normalization: max, min values), Relu act funct

CNN structure: 5 conv-pooling pairs, 2 fully connected layers, 1. 256 neurons, 2. 128 neurons, output layer: 48 neurons-16 for each coefficient

Optimisers: SGD (different parameters: converges after 400 epochs), adaptive learning rate: AdaGrad (lr0.1), RMSdrop (lr0.0001), **Adam (lr0.0001)->chose**

1. **Machine learning in aerodynamic shape optimization** 0

Overview and comparison over optimization and ML methods

1. **A reinforcement learning approach to airfoil shape optimization** 0

Reinforcement learning coupled with deep NN

1. **An inverse design method for supercritical airfoil** +++ (further literature links)

Wall Ma -> Airfoil shape (Ma also generative generated)

Mapping model: GAN -> input 256 neurons, 2 hidden layers each 64 neurons, outputlayer 255 neurons; hyperbolic tangent (tanh) as activation function; input/output tensors normalized; Pytorch; test/train datasets: 10/90; Adam Optimizer; lr0.0003; batch size 256; 2000 epochs

1. **High Reynolds number airfoil turbulence modeling method based on machine learning technique** +

Turbulence Modeling

MLP (Multi-Layer Perceptron), Pytorch, Adam, batch size 128, lr0.003, training epoch 300, error order 10E-5

1. **Artificial neural network based inverse design Airfoils and wings** +++

Inverse Design of airfoil and wing shape: different Parameters; Comparison of 3 ANNs: Backpropagation, Radial Basis Function and General Regression Neural Network

BP configuration (choice):

learning algorithm Learngdm

input neuron transferring function tan sig

output neuron transferring function purelin

error range of network training10−5

training step0.1

top limit of steps1000

hidden layers7

RBF configuration:

hidden layer activation function Gaussian nucleus function

radius base function expansion speed 0.8

hidden layer nodes as many as input layer nodes numbers

GRNN configuration:

expansion const0.1 0.2, in order to seek accurate solution

interval 0.1

Airfoil reverse design: Ma=0.705, Re=23,000,000, alpha 2.53deg; Database: 208 airfoils

1. **Neural networks based airfoil generation for a given Cp using Bezier–PARSEC** +++

Cp-distribution -> optimized airfoil shapes

Feedforward backpropagation neural network (chosen), generalized regression network and a radial basis network in comparison

Tables with NN-Parameters

Bezier PARSEC parametrization of airfoils

700 airfoils

Epoch=250-300 for FFBP

1. **Inverse Design of Airfoil Using a Deep ConvolutionalNeural Network** +++

Pressure distribution -> airfoil shape

Xfoil for Cp-Data: Re=10000, alpha=3, 1343 airfoils

CNN: train/test split: 80/20; Adam; lr5x10E-4; minibatch size=64; epoch=380; input: 144x144x2, 216x216x2, 144x144x1, output 70; up to 5 convolution layers, up to 3 fully connected layers each 100 neurons