A deep artificial neural network-assisted genetic-algorithm method to optimize a slotted hydrofoil

Mahya Hajihassanpour

Abstract

A deep learning approach is designed to act as a surrogate model in the process of the genetic algorithm-based optimization of a slotted hydrofoil. The designed deep Artificial Neural Network (ANN) has a multilayer perceptron-based architecture that composes input, fully connected, leaky ReLU, batch normalization and regression layers. The accuracy of this deep ANN is assessed for a dummy multi-input multi-output problem and its efficiency is verified. This verified deep ANN is employed in the process of optimization providing promising outcomes.

Deep ANN

Having inputs and target outputs data, in the process of the ANN, the training dataset and test ones should be first determined. The training dataset is created by 80% of data while the remaining 20% of data is used to verify the trained network. The preprocessing on the training dataset should be done before training a deep ANN. This preprocessing includes the normalization of data to the interval of [0,1]. This normalization increases the accuracy and speed of the ANN.

The first layer of the designed ANN is an input layer with two features. This layer is connected to a triple-layer that will be repeated N_r times. The first layer in this triple layer is a fully-connected layer with N_n neurons. The next one is the leaky Rectified Linear Unit (ReLU) that can be efficiently employed for the regression jobs. The last one in the triple-layer layer is the batch normalization layer aiming to reduce errors of the ANN and increase its accuracy by retaining numbers throughout the ANN on the same scale. After repeating this triple layer N_r times, then a fully connected layer is used to give out the data to the last layer which is the regression layer. A schematic of the designed deep ANN is shown in Fig. 1.

These training data will enter the network and optimization is performed to properly adjust the weights and biases as the learnable parameters of the ANN. This optimizer tries to minimize a loss function indicating the differences between the ANN outputs and the target outputs. The optimizer is the Adam method. The number of epochs is 250 and the learning rate equals 0.08. Our study shows that these number of epochs and learning rate are appropriate choices leading to accurate and fast ANN.

Finally, the accuracy of the trained deep ANN is assessed by evaluating the differences between the outputs of the deep ANN and the target outputs based on the L_2-norm error. This error is evaluated for both the training dataset and testing one to assess overfitting and underfitting problems.

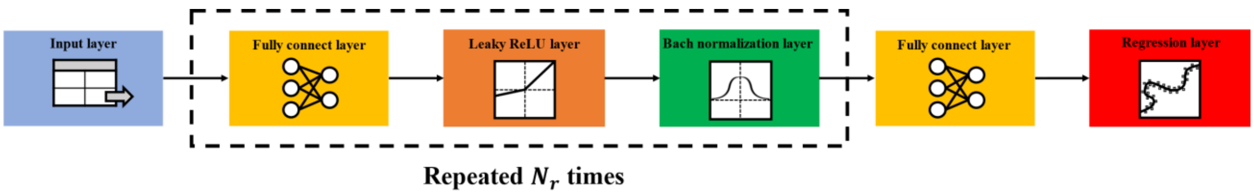

Figure 1 Architecture of the designed ANN network

Dummy problem

The inputs and the outputs of this problem are as follow:

	Symbols and equations	Interval
Input #1	X_1	[-5,5]
Input #2	X_2	[-5,5]
Output #1	$Y_1 = X_1$	
Output #2	$Y_1 = X_1X_1X_1 + X_2X_2X_2 + X_1X_2$	

The first function is linear, however, the second one is highly non-linear and the designed deep ANN should be able to capture both of them that might be hard by a single design (see Fig. 2). The data consist of 1000 samples generated uniformly by a pseudorandom number generator in the prescribed intervals. Then the ANN is trained and tested as discussed in the previous section. The number of replication of the triple-layer is $N_r = 4$ and their number of neurons is $N_n = 60$. The errors obtained are given in Table 1 that are quite in an acceptable range.

Table 1 Errors for the dummy problem.

	Training dataset	Testing dataset
$L_2 - norm\ error\ (Y_1)$	0.0092	0.0090
$L_2 - norm\ error\ (Y_2)$	0.0082	0.0082

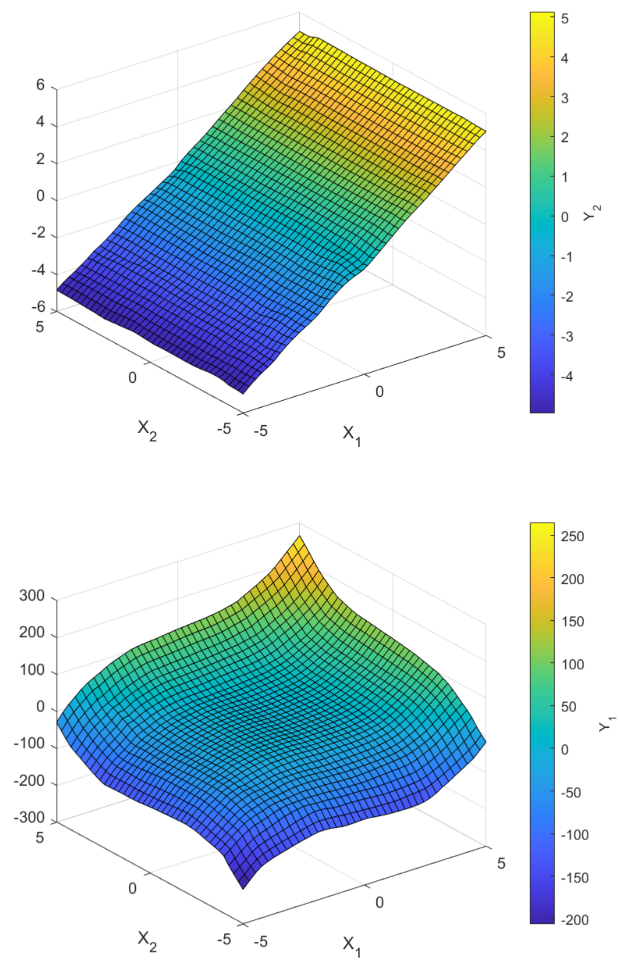

Figure 2 Response surfaces predicted by the designed deep ANN for the dummy problem.

Hydrofoil optimization

The verified deep ANN architecture is also trained to be used as a surrogate model in the process of the genetic-algorithm based optimization. Here, the dataset has 40 data only got from a physical solver which is a finite volume discretization of the governing equations of the cavitating fluid flows. The number of repeated triple-layers is $N_r = 4$ and their number of neurons is $N_n = 30$. Here, a fewer number of neurons is set to avoid the overfitting problem. Figure 3 shows the surrogate function obtained by the deep ANN and the location of the optimized points in the Pareto front. There is an acceptable agreement between these results and the ones obtained by applying the Gaussian process regression method (see the previous document). The Pareto front is also given in Table 2, so it can be seen that it is possible to reduce the cavity pocket size without a reduction in the lift coefficient.

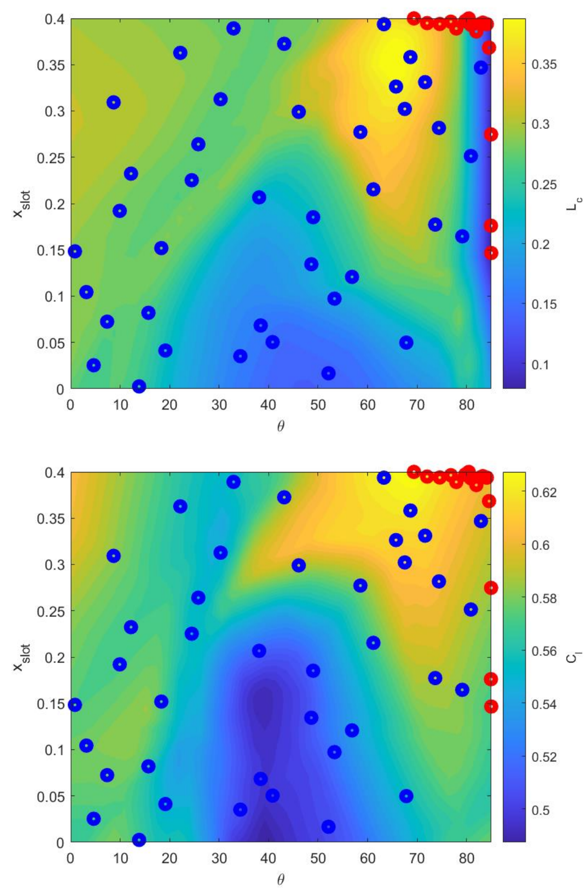

Figure 3 Contours of the lift coefficient and the cavity length obtained by the deep ANN. Training plus testing data (blue circle scatters); Optimized data from the Pareto front (red circle scatters).

Table 1 Pareto front points (GA output)

θ	x_{slot}	L_c	C_l	$\overline{L_c} - L_c$	$C_l - \bar{C}_l$	
84.99279	0.146602	0.0777	0.571752	0.133	-0.03775	
84.93054	0.176084	0.084683	0.575269	0.126017	-0.03423	
69.36935	0.399959	0.358936	0.631202	-0.14824	0.021702	
80.45111	0.399908	0.237484	0.610132	-0.02678	0.000632	
79.98436	0.396634	0.258858	0.610417	-0.04816	0.000917	
84.96125	0.274696	0.085912	0.586315	0.124788	-0.02318	
84.99279	0.146602	0.0777	0.571752	0.133	-0.03775	
76.8342	0.39641	0.313268	0.616631	-0.10257	0.007131	
84.0977	0.393764	0.115442	0.602298	0.095258	-0.0072	Selected
82.82821	0.394195	0.153016	0.604608	0.057684	-0.00489	
79.74103	0.397206	0.271288	0.611016	-0.06059	0.001516	
81.97769	0.385811	0.177898	0.604981	0.032802	-0.00452	
74.58137	0.393952	0.332981	0.620582	-0.12228	0.011082	
72.03045	0.394894	0.350353	0.625765	-0.13965	0.016265	
84.54465	0.368411	0.10137	0.59807	0.10933	-0.01143	
83.28995	0.395358	0.139394	0.603947	0.071306	-0.00555	
77.92486	0.389058	0.302342	0.613032	-0.09164	0.003532	
80.86997	0.393003	0.21275	0.607931	-0.00205	-0.00157	