

딥러닝을 이용한 연안 소형 어선 주요 치수 추정 연구

장민성 · 김동준^{1*} · 자오양²

부경대학교 조선해양시스템공학과 대학원생, ¹부경대학교 조선해양시스템공학과 교수,

²부경대학교 마린융합디자인공학과 대학원생

A study on estimating the main dimensions of a small fishing boat using deep learning

Min Sung JANG, Dong-Joon KIM^{1*} and Yang ZHAO²

Student, Department of Naval Architecture and Marine Systems Engineering, Pukyong National University, Busan 48513, Korea

¹Professor, Department of Naval Architecture and Marine Systems Engineering, Pukyong National University, Busan 48513, Korea

²Student, Department of Marine design Convergence engineering, Pukyong National University, Busan 48513, Korea

The first step is to determine the principal dimensions of the design ship, such as length between perpendiculars, beam, draft and depth when accomplishing the design of a new vessel. To make this process easier, a database with a large amount of existing ship data and a regression analysis technique are needed. Recently, deep learning, a branch of artificial intelligence (AI) has been used in regression analysis. In this paper, deep learning neural networks are used for regression analysis to find the regression function between the input and output data. To find the neural network structure with the highest accuracy, the errors of neural network structures with varying the number of the layers and the nodes are compared. In this paper, Python TensorFlow Keras API and MATLAB Deep Learning Toolbox are used to build deep learning neural networks. Constructed DNN (deep neural networks) makes helpful in determining the principal dimension of the ship and saves much time in the ship design process.

Keywords: Fishing boat design, Determination of main dimension, DNN, Regression analysis

서론

최근 컴퓨터 기술의 발전에 따라 다양한 공학 분야에서 AI (Artificial intelligence) 기술이 적용되고 있으며, 국내 선박 설계분야에 있어서도 빠른 속도로 적용이 이루어지고 있다. Lee (1988)의 전문가 시스템에 대한 해설을 시작으로, 2019년 대우조선해양에서는 인공

지능을 활용한 선박영업지원 설계시스템을 개발하였다(ChosunBiz, 2019). Kim et al. (2019)은 심층 신경망 기법(Deep Neural Networks, DNN)을 이용하여 경하중량을 추정하였으며, Cho et al. (2019)은 딥러닝(Deep learning)을 이용하여 선박의 저항을 추정하였고, Shin et al. (2020)은 강화학습(Reinforcement learning)을 배

*Corresponding author: djkim@pknu.ac.kr, Tel: +82-51-629-6614, Fax: +82-51-629-6608

관경로 설계에 적용하고 있다. 최근에는 설계뿐만 아니라 운동예측, 생산가공정보 등의 분야로 점차 적용이 확대되고 있다. Lim and Jo (2020)는 바이지선의 횡요(Roll) 응답 예측에 기계 학습(Machine learning) 기법을 적용하고 있으며, Kim (2020)은 기계 학습을 이용하여 선체 외판 가공 정보를 예측하고 있다.

본 연구에서는 딥러닝의 주요 적용 분야인 회귀 분석(Regression analysis)을 선박의 주요 치수(길이, 폭, 깊이 등)를 결정하는 과정에 적용하고자 하였다. 연안 소형 어선을 대상으로 총 11,000여척의 데이터가 수집되었으며, 각 지역별 선박의 특성 등에 차이를 고려하여 부산 지역의 연안 소형 어선 1,600여척을 대상으로 연구를 수행하였다.

선박의 길이와 톤수를 입력변수로 사용하고 폭과 깊이를 출력변수로 설정하여 DNN 모델을 생성하였다. 딥러닝을 실현할 수 방법은 많이 제공되고 있는데 본 논문에서는 구글(Google)에서 제작한 오픈소스 라이브러리 텐서플로우(TensorFlow)를 이용하여 직접 코딩을 하는 방식을 택하였다. 은닉층(Hidden layer)과 은닉층의 Neuron 수(노드, Node)를 변화시켜가며 결과에 미치는 영향을 파악하고자 하였다. 또한, 비교를 위해 Matlab에서 제공하는 방법을 선택하여 같은 데이터를 사용하여 두 방법의 결과를 비교 분석하였다.

이러한 연구를 통해 머지않아 시행될 어선등록제도 변경에 따른 새로운 선형에 대한 초기설계에 소요 시간을 단축할 수 있도록 함을 목표로 하였다.

재료 및 방법

딥러닝의 간단한 소개

딥러닝은 기계학습의 특정한 한 분야로서 Fig. 1에서와 같이 여러 개의 연속된 층(Layer)으로 인공신경망(Artificial neural networks)을 구성하여, 데이터를 학습하는 방식이다. 인간의 신경망을 통해 정보가 전달되는 것과 유사하게 컴퓨터를 이용하여 표현한 연속된 층을 통해 정보가 흘러감으로써 데이터를 학습하고 적절한 값이 도출될 수 있게 한 것이다. 데이터로부터 모델을 만드는 데 얼마나 많은 층을 사용했는지가 그 모델의 깊이가 되며, 최근의 딥러닝 모델은 학습을 위해 수십 개, 수백 개의 연속된 층을 가지고 있다.

활성화 함수(Activation function)는 시냅스(Synapse)의 활성화 원리를 모사하여 데이터를 유효하게 전달하거나 차단하는 역할을 한다. 각 층에서 다음 층으로 전달될 때 가중치(Weight)와 편향(Bias)에 따라 연산이 진행된 후 이 값이 활성화 함수에 입력되고, 출력된 값이 다음 층으로 전달되게 된다. 활성화 함수는 DNN 모델의 학습 속도 및 학습률에 큰 영향을 미치며, 다양한 활성화 함수가 존재하며 각각 장단점이 있다. 본 논문에서는 최근 딥러닝 학습 시 많이 사용되며 학습 속도가 빠른 ReLU(Rectified Linear Unit) function을 사용하였다(Fig. 2).

손실함수(Loss function)란 신경망에서 얻어진 예측값과 결과값의 차이를 계산하는 함수이며, 신경망의 성능을 수치로 표현한 것으로 값이 클수록 신경망 성능이 나쁜 것으로 판단할 수 있다. 대표적인 손실함수로는

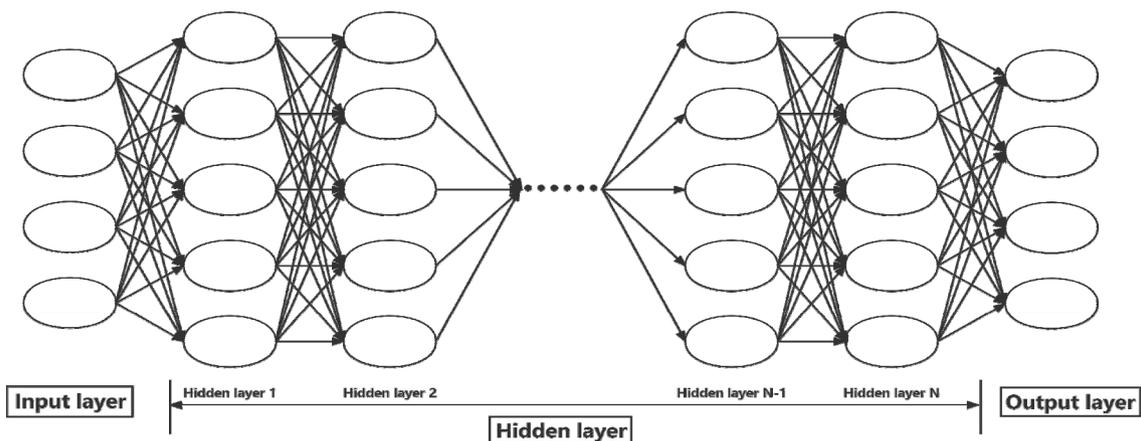


Fig. 1. Deep learning model.

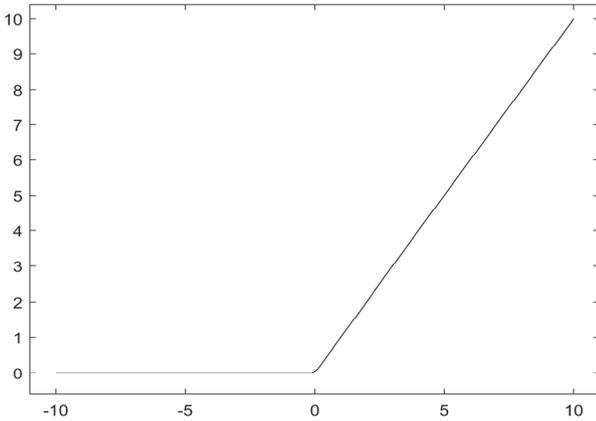


Fig. 2. ReLU function.

평균 제곱 오차(Mean Squared Error, MSE), 교차 엔트로피 오차(Cross Entropy Error, CEE) 등이 있으며, 본 논문에서는 평균 제곱 오차를 사용하였다. 예측값과 목표값과 차이에 대한 제곱의 평균을 의미하며, 다음 식으로 표현된다

$$MSE = \frac{1}{n} \sum_{i=0}^n (y_i - t_i)^2$$

y_i = 예측값
 t_i = 목표값

학습률(Learning Rate)은 가중치를 업데이트하는 비율이며, 큰 값으로 설정하면 학습이 빨리 진행되는 장점이 있으나, 과도한 학습으로 적정한 값을 벗어날 우려가 있고, 너무 작은 값으로 설정하게 되면 학습 속도가 느려질 수 있다. 학습률은 보통 0.1~0.0001의 값을 사용하며, 본 논문에서는 0.0001로 모든 학습을 진행하였다.

Table 1. Deep learning network conditions

Deep learning network conditions	
Input value	Ton, Length
Output value	Breadth, Depth
Number of hidden layer	2, 4, 6, 8, 10
Number of neurons	5, 10, 20
Training end condition	If the minimum loss value cannot be updated more than 500 times
Activation function	ReLU (Rectified Linear Unit) function
Loss function	MSE (Mean Squared Error)
Learning rate	0.0001
Optimization function	Adam

학습데이터 선정

학습에 사용된 자료는 총 1,666여척이며, 선박의 요소별 관계를 Fig. 3부터 Fig. 7에 나타내었다. 길이를 기준으로 하여 폭(Fig. 3)과 톤수(Fig. 4)의 경우 추세선을 그릴 수 있을 정도로 상관관계가 있으나, 깊이(Fig. 5)의 경우 뚜렷한 추세를 파악하기 어려우며, 치수의 변화폭이 상대적으로 작고, 0.6~1.2 m의 범위 안에 대부분의 데이터들이 모여 있는 것을 확인할 수 있다.

폭을 기준으로 톤수와의 관계를 살펴보면 폭이 커질 때 톤수가 함께 커지는 것을 확인할 수 있으나(Fig. 6), 2.7 m를 넘어가면서 비례관계가 없어지는 것을 볼 수 있다. 깊이의 경우 치수의 변화폭이 상대적으로 작으므로 길이-깊이 관계와 비슷한 형태를 보인다(Fig. 7). 깊이와 톤수와 관계를 살펴보면 길이와 폭의 경우와는 달리 뚜렷한 추세를 파악하기 어렵다(Fig. 8).

텐서플로 프로그램을 이용한 딥러닝

텐서플로와 케라스(Keras)를 이용한 딥러닝

파이썬(Python)을 이용하여 딥러닝을 모델링하고, 텐서플로와 이 텐서플로 위에서 동작하는 딥러닝 프레임 워크인 케라스를 이용하여 학습을 진행하였다. 텐서플로에는 가중치와 편향, 그리고 역전파 과정이 모두 함수에 포함되어있는 최적화 함수가 존재하며 (Kim, 2020), 케라스는 파이썬으로 작성된 오픈 소스 신경망 라이브러리로서 거의 모든 종류의 딥러닝 모델을 간편하게 만들고 학습시킬 수 있는 프레임 워크이다(Park, 2018).

학습 조건

독립변수는 톤수와 길이로 하고, 학습 후 출력되는

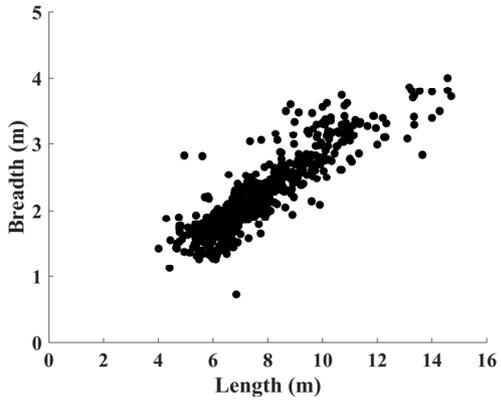


Fig. 3. Relationship between length and breadth.

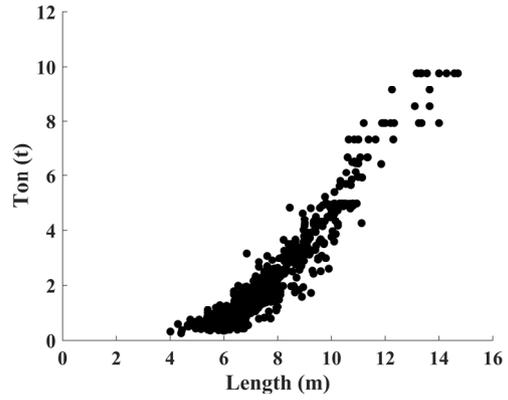


Fig. 4. Relationship between length and ton.

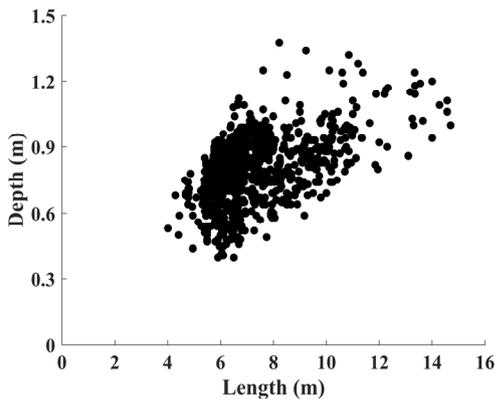


Fig. 5. Relationship between length and depth.

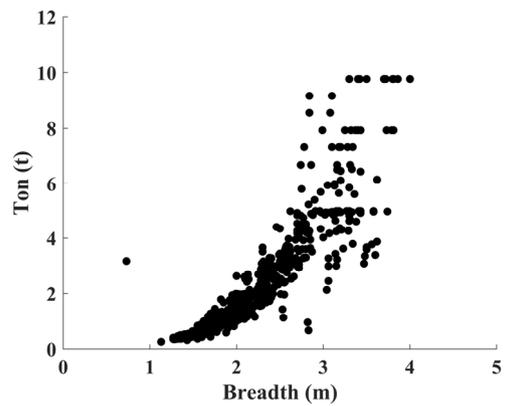


Fig. 6. Relationship between breadth and ton.

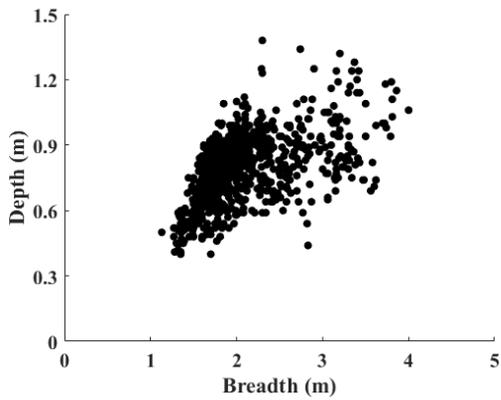


Fig. 7. Relationship between breadth and depth.

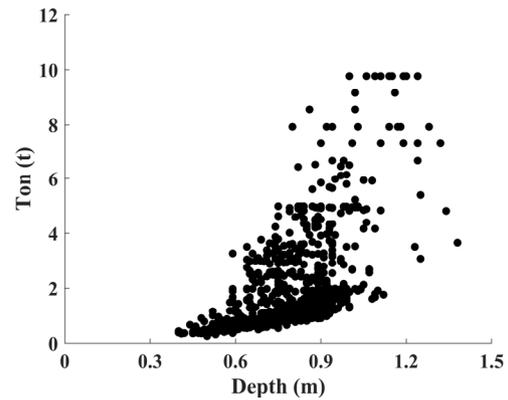


Fig. 8. Relationship between depth and ton.

Table 2. Result of deep learning (TensorFlow)

No. of layer	No. of neurons	Minimum MSE		Stop epoch
		Train	Validation	
2	5	0.026222	0.034501	1078
	10	0.016015	0.019135	3930
	20	0.014519	0.018541	3261
4	5	0.021019	0.023506	1505
	10	0.015183	0.018715	3101
	20	0.014573	0.018381	1497
6	5	0.015436	0.019280	2941
	10	0.015439	0.018848	3286
	20	0.015070	0.018461	1538
8	5	0.019186	0.021830	2516
	10	0.014037	0.018803	4202
	20	0.013675	0.018042	3458
10	5	0.020198	0.022083	5876
	10	0.016084	0.019697	1907
	20	0.013436	0.017923	1853

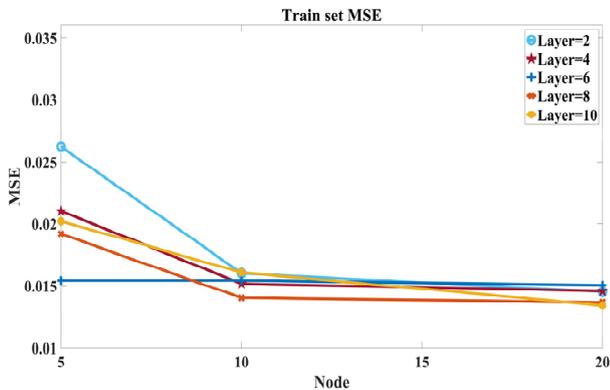


Fig. 9. Train MSE for each number of layer.

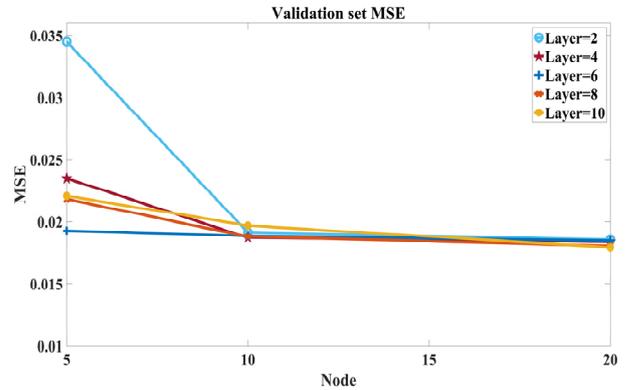


Fig. 10. Validation MSE for each number of layer.

종속변수는 폭과 깊이로 설정하였다. 훈련 집합 (Training set)으로는 전체 데이터의 70%, 검증 집합 (Validation set)으로는 30%를 설정하였다. 학습조건은 Table 1과 같이 은닉층의 수를 2, 4, 6, 8, 10로 증가시키면서 각 은닉층의 노드 수는 5, 10, 20의 세 가지 경우에 대해 학습을 진행하였다. 기본 학습 횟수(Epoch)는 10,000회로 지정하되, 학습을 진행하면서 계속 갱신되는 Validation MSE 값이 500회 이상 최저값을 갱신하지 못하면 학습이 종료되도록 설정하였다. 또한, 활성화 함수는 Relu를, 손실함수는 MSE를 사용하고, 학습률은 0.0001로 고정한 뒤 학습을 진행하였다. 학습된 결과는

Table 2와 Fig. 9, 10에 정리하였다.

Fig. 9에 의하면 노드 수가 증가하면 오차는 감소하고, 층 수가 증가하면 노드 수가 적은 부분에서 오차가 역시 감소함을 알 수 있다. 하지만 층 수가 10개 이상이 되면 층 수의 증가는 오차에 크게 영향을 미치지 않았다. Fig. 10의 검증 집합 결과에서도 같은 경향을 보이고 있다.

노드 수의 증가와 오차 감소의 관계를 파악해보기 위하여 Table 1과 동일한 조건에서 노드 수를 128, 256, 512로 증가시켜 계산한 결과를 Table 3에 보이고 있다. 하지만 Table 3에 의하면 노드 수의 증가에도 불구하고

Table 3. Result of increasing the number of Neurons

No. of hidden layer	No. of neurons (128)		No. of neurons (256)		No. of neurons (512)	
	Train MSE	Val MSE	Train MSE	Val MSE	Train MSE	Val MSE
2	0.0163	0.0149	0.0171	0.0442	0.0164	0.0191
4	0.0150	0.0147	0.0148	0.0163	0.0152	0.0176
6	0.0149	0.0175	0.0146	0.0159	0.0123	0.0165
8	0.0146	0.0151	0.0145	0.0167	0.0139	0.0180
10	0.0151	0.0160	0.0143	0.0148	0.0119	0.0189

오차는 줄어들지 않고 있으며, 일정한 경향도 보이지 않고 진동하는 모습을 보인다. 이는 층 수와 노드 수가 일정한 정도에 이르면 그 이후에는 수를 증가시켜도 계산 시간만 늘어날 뿐 결과의 개선에는 영향을 미치지 않는다는 것을 알 수 있다.

Matlab 프로그램을 이용한 딥러닝

Matlab은 미국 Mathworks회사에서 개발한 소프트웨어로 여러 공학 분야에서 응용되고 있으며, 딥러닝 분야에서 활용도가 매우 높다고 할 수 있다. 이 소프트웨어

에는 딥러닝 분야 Toolbox가 있으며, Toolbox 내부에는 이미 작성되어있는 함수도 많아서 직접 호출하면 쉽게 이용할 수 있는 장점이 있다.

본 논문에서는 케라스를 이용해서 직접 코딩한 딥러닝 모델이 적절한 것인지 판단하기 위하여 Matlab을 이용하여 계산을 수행하였다. Table 4에 사용한 Matlab structure data를 수록하였으며, Table 5와 Fig. 11, 12에 결과를 보여주고 있다.

Matlab에서는 최적화 함수(Optimization function)로 Adam을 제공하지 않아서, Table 4와 같이 Matlab에서

Table 4. Matlab structure data

Deep learning network structure	
Number of hidden layer	2, 4, 6, 8, 10
Number of neurons	5, 10, 20
Activation function	ReLU function
Learning rate	0.0001
Optimization function	LM

Table 5. Result of deep learning (Matlab)

No. of layer	No. of neurons	Minimum MSE		Stop epoch
		Train set	Validation set	
2	5	0.016925	0.028446	18
	10	0.013988	0.019093	15
	20	0.012389	0.019440	27
4	5	0.019867	0.020553	17
	10	0.013661	0.017382	17
	20	0.011411	0.012556	12
6	5	0.015436	0.019280	18
	10	0.015439	0.018848	19
	20	0.012240	0.014179	18
8	5	0.015504	0.016421	23
	10	0.012941	0.015693	32
	20	0.011833	0.019158	23
10	5	0.019448	0.019052	49
	10	0.016129	0.014814	37
	20	0.012581	0.015199	24

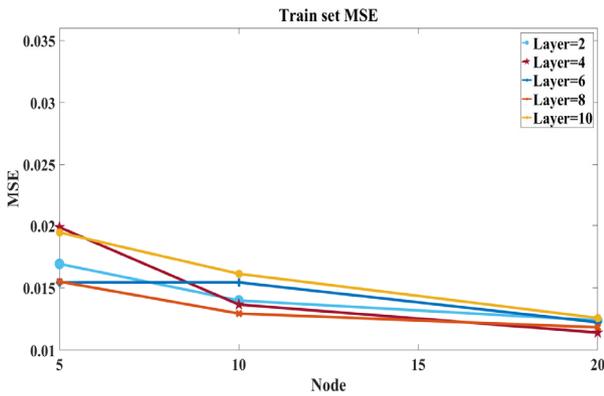


Fig. 11. Train MSE for each number of layer.

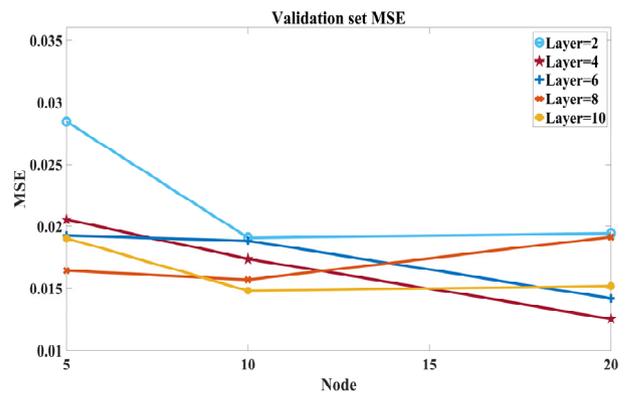


Fig. 12. Validation MSE for each number of layer.

제공하는 LM을 사용하여 계산하였고, 이에 따라 Table 5의 정지 Epoch 수가 적어진 것을 알 수 있다. 계산 결과를 보면 케라스를 이용해 직접 코딩해서 얻었던 결과와 유사하다는 것을 알 수 있다. 오차가 소수점 3째자리에 차이 나는 것으로 보아 두 방법의 계산 정도는 유사하며, 앞에서 얻었던 노드 수와 층 수의 증가에 따른 결과 개선 경향도 일치하는 것으로 판단된다.

데이터 특성 분석을 통한 DNN 모델 개선

다중 출력 모델의 학습에서 각 출력값의 최소 MSE 값의 합이 작아지는 방향으로 학습이 진행된다. 학습을 위한 데이터의 특성이 폭의 변화 범위는 1 m에서 4 m이고, 깊이에 대한 변화 범위는 0.3 m에서 1.5 m이다. 같은 1 m의 변화가 발생한다고 가정하면, 폭과 깊이 두 요소에서의 변화율은 서로 다르게 나타날 것이다. 따라서 학습 시 각 요소에 맞는 가중치를 설정하여 학습한다면 결과가 좋게 나타날 것이라고 추론할 수 있다.

폭과 깊이에 적절한 가중치를 반영하기 위하여, 먼저 두 요소 사이의 평균 비율을 측정하고, 이에 맞추어 가중치를 결정한다. 폭/깊이에 대한 비율의 평균값이 2.52대 1이므로 가중치를 2.52로 하여 폭과 깊이의 비율 1:1에

서 0.4:1로 변경하여 학습과정을 수행하였다.

결과 및 고찰

결과분석

학습 후 가장 효과가 좋은 딥러닝 구조를 선택하여 Table 6과 같이 모델을 구축하였으며, 이를 이용하여 Test용으로 남겨둔 20척의 데이터를 예측하여 Table 7과 같이 결과를 비교하였다.

예측된 결과를 보면 폭의 경우 MSE는 0.0279, 평균오

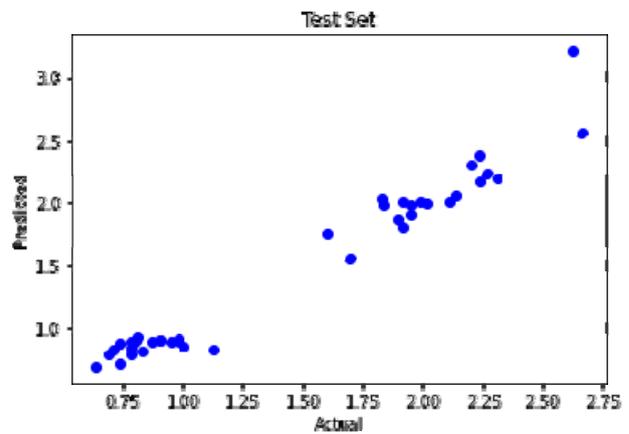


Fig. 13. Accuracy of Test Set.

Table 6. Final TensorFlow network conditions

TensorFlow Data	
Number of hidden layer	11
Number of neurons	512
Learning rate	0.0001
Activation function	ReLU function
Optimization function	Adam

Table 7. The deep learning prediction results of TensorFlow

Input Length	Input Ton	Output Breadth			Output Depth		
		Actual	Prediction	Error	Actual	Prediction	Error
10.7	4.98	2.62	3.2086	0.2247	0.98	0.9069	0.0746
4.91	0.58	1.70	1.5551	0.0852	0.64	0.6831	0.0673
5.61	1.11	1.84	1.9769	0.0744	0.83	0.8071	0.0276
6.89	1.96	2.20	2.3012	0.0460	0.79	0.8752	0.1078
6.92	1.33	2.14	2.0469	0.0435	0.78	0.7891	0.0117
6.31	1.13	1.95	1.8951	0.0282	0.78	0.8306	0.0649
6.59	1.38	1.92	1.9930	0.0380	0.90	0.8953	0.0052
6.30	1.94	2.31	2.1875	0.0530	1.13	0.8224	0.2722
8.65	2.99	2.66	2.5591	0.0379	0.69	0.7808	0.1316
6.15	1.16	1.90	1.8610	0.0205	0.81	0.9236	0.1402
6.57	1.44	1.99	2.0030	0.0065	0.80	0.8935	0.1169
7.48	1.92	2.26	2.2328	0.0120	0.90	0.8967	0.0037
5.76	0.94	1.92	1.7934	0.0659	0.71	0.8243	0.1610
6.75	1.38	1.83	2.0219	0.1049	0.78	0.8816	0.1303
6.34	1.36	2.02	1.9886	0.0155	0.87	0.8842	0.0163
5.40	0.71	1.6	1.7479	0.0924	0.74	0.7064	0.0454
6.89	1.75	2.24	2.1698	0.0313	0.95	0.8852	0.0682
6.53	1.30	1.95	1.9705	0.0105	0.98	0.8825	0.0995
6.45	1.46	2.11	1.9999	0.0522	0.74	0.8702	0.1759
7.14	2.17	2.23	2.3851	0.0696	1.00	0.8530	0.1470
		Mean Error		0.0556	Mean Error		0.0934
		MSE	0.0279		MSE	0.0109	

차(Mean error)는 5.56%, 깊이의 경우 MSE는 0.0109, 평균오차는 9.34%를 보였다. 여기서 오차는 실제값과 예측값의 차이를 실제값으로 나눈 값을 의미한다. 깊이의 경우 오차가 큰 것은 Fig. 5, 7, 8에서 볼 수 있듯이 깊이는 다른 모든 변수와의 관계에서 추세선을 얻기가 어려웠던 점에 기인한다. Fig. 13에서 폭과 깊이에 대해 실제값과 예측값을 확인할 수 있다.

결론

본 논문은 선박의 초기설계 단계에서 제원을 결정할 때 기존 실적선을 통하여 진행되었던 추정방식에서 더 나아가 딥러닝을 통하여 빠르고 간편하게 얻는 방법을 찾고자 하였다. 딥러닝을 이용하여 MSE 최소값이 새롭게 갱신되지 않을 때까지 학습을 진행하였다. 은닉층과 노드의 숫자를 변경해 가면서 학습을 진행하여 가장 MSE값이 작은 경우를 찾아내었다. 얻어진 회귀분석모델을 검증하기 위하여 최종 Test 집합에 적용하여 만족할 만한 결과를 확인하였다.

다중출력의 경우 가중치 설정을 통하여 정교하게 학

습조건을 지정해야 원하는 목표값에 더욱 근접한 결과를 얻을 수 있음을 알게 되었다. 따라서 현재 보유하고 있는 데이터들의 관계를 더 파악하고 각 항목 간의 관계를 고려한 뒤 추가 연구가 필요하다고 생각된다.

사사

이 논문은 부경대학교 자율창의학술연구비(2021년)에 의하여 연구되었음.

References

Cho YI, Oh MJ, Seok YS, Lee SJ, and Roh MI. 2019. Resistance estimation of a ship in the initial hull design using deep learning. *Korean Journal of Computational Design and Engineering* 24, 203-210. <https://doi.org/10.7315/CDE.2019.203>.

ChosunBiz. 2019. https://biz.chosun.com/site/data/html_dir/2019/03/04/2019030401387.html. Accessed 4 Mar 2019.

Kim BS. 2020. Development of a prediction system for shell plate forming information based on machine learning. Ph.

- D. Thesis, Seoul National University, Korea, 113.
- Kim GY, Ban IJ, Park BC, Oh SJ, Lim CO, and Shin SC. 2019. Estimation of lightweight in the initial design of ships using deep neural networks. *Journal of Korean Institute of Intelligent Systems* 29, 416-423.
- Kim HH. 2020. Get started! TensorFlow 2.0 programming, Wikibooks, 1-484.
- Lee DK. 1988. On the application of artificial Intelligence to ship design. *Bulletin of the Society of Naval Architects of Korea* 25, 56-62.
- Lim JH and Jo HJ 2020. Prediction of barge ship roll response amplitude operator using machine learning techniques. *Journal of Ocean Engineering and Technology* 34, 167-179. <https://doi.org/10.26748/KSOE.2019.107>.
- Park HS translation. 2018. Deep learning with Python by Francois Chollet. *Gilbut*, 1-476.
- Shin DS, Park BC, Lim CO, Oh SJ, Kim GY, and Shin SC. 2020. Pipe routing using reinforcement learning on initial design stage. *Journal of the Society of Naval Architects of Korea* 57, 191-197. <https://doi.org/10.3744/SSNAK.2020.57.4.191>.
-
2022. 07. 11 Received
2022. 07. 21 Revised
2022. 08. 05 Accepted