말뚝지지력 예측을 위한 신경망 모델의 적용성 평가

Evaluation of Neural Network Model for Estimating Pile Load Capacity

이병식*

Lee, Byungsik*

Abstract

Neural network models based on deep learning algorithms are increasingly used for estimating pile load capacities as supplements of bearing capacity equations and field load tests. A series of hyperparameter tuning is required to improve the performance and reliability of developing a neural network model. In this study, the number of hidden layers and neurons, the activation functions, the optimizing algorithms of the gradient descent method, and the learning rates were tuned. The grid search method was applied for the tuning, which is a hyperparameter optimizer supplied by the developing platform. The cross-validation method was applied to enhance reliability for model validation. An appropriate number of epochs was determined using the early stopping method to prevent the overfitting of the model to the training data. The performance of the tuned optimum model evaluated for the test data set revealed that the model could estimate pile load capacities approximately with an average absolute error of 3,000 kN and a coefficient of determinant of 0.5.

Key words : Deep Learning, Pile Load Capacity, Hyperparameter Tuning, Overfitting

요 지

말뚝지지력 예측에 있어서 지지력공식 방법이나 현장재하시험의 보조방법으로서 딥러닝 알고리즘을 이용한 신경망(neural network) 모델의 적용 사례가 증대되고 있다. 신경망 모델 개발의 효율성과 신뢰도 향상을 위해서는 적절한 하이퍼파라미터 결정을 위한 조율과정이 필수적으로 요구된다. 이 논문에서는 말뚝지지력 데이터셋에 대해서 다층퍼셉트론신경망 모델의 구성을 위한 은닉층과 뉴런의 개수 및 활성화함수를 결정하고, 훈련 및 검증 시 적용할 경사하강법 최적화 알고리즘, 학습률 등에 대한 조율을 수행하였다. 수동 조율은 매우 비효율적이므로 개발 플랫폼에서 제공하는 하이퍼파라미터 옵티마이저인 grid search 방법을 적용했다. 검증의 신뢰도 향상을 위해 교차검증 방법을 적용하여 최적 조합을 결정했다. 훈련데이터에 대한 모델의 과대적합을 배제하기 위해 early stopping 방법을 적용하여 적절한 epoch 수를 검토했다. 결정한 최적모델의 기능을 테스트셋에 대해서 검증한 결과 개략적으로 예측값과 기준값 간의 최소 평균절대오차 약 3,000 kN과 결정계수 약 0.5의 수준으로 말뚝지지력을 추정할 수 있었다.

핵심용어 : 딥러닝, 말뚝지지력, 하이퍼파라미터 튜닝, 과대적합

1. 서 론

지반공학 관련 공학문제를 해결하기 위한 인공지능(artificial intelligence, AI)기법의 적용 사례가 날로 증가추세에 있다. Shahin (2013, 2015a, 2016), Juwaied (2018), Zhang et al. (2020), Ebid (2021)은 이들 사례의 기술수준을 정리하

였다. Shahin (2013)은 매우 복잡한 지반공학 문제 해결에 있어서 데이터구동 접근법(data-driven approach)으로 모델을 훈련시켜 문제를 해결할 수 있다는 점을 AI 기법의 장점으로 언급했다. 또한 Shahin (2015b, 2016)은 얕은기초와 말뚝 기초 문제에 AI 기법을 적용한 사례를 검토하였다. 이들 논문에서는 공통적으로 인공지능기법의 적용 성과를 증대

*교신저자, 정회원, 공주대학교 사회환경공학과 교수(Tel: +82-41-521-9309, Fax: +82-41-568-0287, E-mail: blee@kongju.ac.kr)
Corresponding Author, Member, Professor, Department of Civil Engineering, Kongju National University

하기 위해서는 모델개발에서 공학적 판단과 전문지식이 반영되어야 하며, AI 기법의 성공사례를 일반화하지 말고 일부 데이터에 특화된 모델로 다루어야 함을 언급했다. 또한 당시 수준으로는 AI 기법을 기존방법의 대체보다는 보완 방법 혹은 시간 많이 걸리는 해석 결과에 대한 빠른 검증수단으로 적용할 것을 제안하였다. Juwaied (2018)는 지반공학 문제를 해결하기 위한 인공지능 기법의 이용사례에 대한 문헌연구를 했다. 적용문제에는 현장조사, 흙의 특성과 거동, 말뚝지지력, 기초침하, 하중-침하 모델, 액상화, 비탈면안정, 흙지지구조물, 터널과 지하굴착 등의 문제 해결을 위한 AI 기법의 적용사례를 정리했다. 사례검토를 통해 AI 기법이 기존 방법과 대비하여 적어도 개략적으로 잘 작동한다고 했다. Zhang et al. (2020)은 AI기법을 소프트웨어기법(software computing method, SCM)으로 칭하고, 복잡, 비선형, 불확실하거나 완전히 이해되지 않은 복잡한 거동의 특성을 내포한 지반공학 문제를 기존 방법보다 우수하게 해결할 수 있다고 했다. 결과적으로 SCM을 기존해석 방법이나 현장계측에 대한 보조 방법으로 적용하는 것이 바람직하다고 제안한 점은 Shahin (2013, 2015a, 2016)의 결론과 크게 다르지 않았다. Ebid (2021)는 인공지능 방법이 지반공학 문제 해결에 적용된 1984년부터 2019년 사이에 발표된 626종의 문헌 사례를 정리했다. 여기서 2008년 이후에 연구사례가 급증하고 있음을 볼 수 있었다.

2008년 이후 말뚝기초 지지력에 대한 연구사례들을 조사하였다. Shahin and Jaksa (2009)는 drilled shaft 기초의 지지력을 artificial neural network (ANN)을 사용하여 추정된 결과가 콘관입시험(cone penetration test, CPT)을 기반으로 한 이론식을 이용한 결과 보다 우수함을 보였다. Ardalan et al. (2009)은 말뚝기초에 대해서 CPT 값을 특성으로 입력하는 인공신경망 모델을 사용하여 지지력을 추정했다. 예측값을 측정값과 비교한 결과 말뚝지지력 추정 방법을 일정 수준 개선하는 결과를 얻었다. Shahin (2010)은 항타말뚝과 drilled shaft 기초의 지지력을 예측하는 인공신경망 모델을 개발하고 결과를 기존 지지력 추정 공식의 결과와 비교하여 인공신경망 모델이 지지력을 가장 잘 예측함을 보였다. 하지만 개발한 모델의 적용 범위 확대를 위해서는 넓은 범위의 새로운 데이터에 대한 훈련이 필요함을 제안했다. Shahin (2013)은 말뚝의 하중-침하거동을 예측할 수 있는 recurrent neural network (RNN) 모델을 개발하였다. 모델을 적용하여 하중-침하곡선을 적절히 예측하였으며 설계 시 현장시험을 보완할 수 있는 방법으로 제안하였다. Kordjazi et al. (2014)은 support vector machine (SVM) 모델로 말뚝지지력을 추정하였다. 추정결과를 CPT를 기반으로 한 기존 방법과 비교한 결과 개발한 모델이 보다 적절함을 확인했다. Benali et al. (2017)은 사질토에 타입한 말뚝의 지지력을 추정하는 ANN 모델을 개발하고 SPT 기반 방법과 비교하여 ANN모델의 장점과 적절성을 확인하였다.

Harandizadeh (2020)은 group method of data handling (GMDH)을 결합한 neural-fuzzy (NF) system으로 말뚝지지력을 예측하였다. 최적화 알고리즘으로 particle swarm optimization (PSO)과 gravitational search algorithm (GSA)를 적용했다. 결과를 근거로 개발모델을 말뚝지지력 예측을 위한 강력한 소프트웨어로 사용할 수 있음을 제안했다. Pham et al. (2020)은 ANN과 random forest (RF) 모델을 개발하였다.

많은 연구사례에도 불구하고 현재 수준에서 인공지능 모델을 기존 해석방법의 완전한 대체방법으로 제안한 문헌은 찾을 수 없었다. Shahin (2013)이 언급한 바와 같이 AI 방법의 성공사례는 일반화하기 보다는 일부 데이터에 특화된 모델로 다루어야 한다는 점을 반영한다면 당연한 결과라 할 수 있었다. 이 논문에서 다른 연구결과를 비롯하여 많은 연구자들이 유사한 연구를 지속적으로 수행해야 하는 이유도 여기에 있는 것으로 판단되었다. 또한 인공지능 모델의 일반화, 적합성, 훈련의 효율성 등을 확보하기 위해 하이퍼파라미터 매개변수 조절, 옵티마이저, 신경망 모델에서의 활성화 함수 선택 등이 중요한 것으로 판단되었다. 그럼에도 불구하고 하이퍼파라미터 조절에 관한 연구는 많지 않았다.

이 논문에서는 선행연구에서 말뚝지지력 추정을 위해 가장 많이 적용되었던 신경망모델인 다층퍼셉트론을 새로운 데이터셋에 대해서 훈련 및 검증하였다. 훈련 및 검증과정에서 은닉층과 은닉층의 뉴런 개수, 학습율, 활성화함수 등의 하이퍼파라미터(hyperparameter)를 교차검증을 통해 최적화하였다. 옵티마이저, epoch 수의 영향에 대해서도 검토하였다. 연구결과를 바탕으로 말뚝지지력 추정을 위한 인공지능 모델의 성능을 평가하였다.

2. 신경망 모델

2.1 모델 구조

신경망 모델은 다층퍼셉트론으로서 입력층과 출력층, 은닉층으로 구성했다. 입력층과 출력층의 뉴런의 개수는 2.2에 기술된 특성과 기준의 개수와 동일하다. 은닉층 수와 각 층의 뉴런 수는 최적값을 결정할 필요가 있었다. 뉴런의 활성화 함수로는 sigmoid, hyperbolic tangent (tanh), rectified linear unit function (ReLU) 함수들의 적용을 고려할 수 있었다.

모델 훈련에서는 오차역전파(backpropagation)알고리즘을 적용하고 경사하강법(gradient descent)으로 가중치를 변화시켜 손실이 최소화되도록 하였다. 손실함수는 평균제곱오차(mean squared error, MSE)를 적용하고, 측정항목(metrics) 함수에 평균절대오차(mean absolute error, MAE)를 적용했다. 경사하강법의 옵티마이저로는 stochastic gradient descent (SGD), adaptative gradient algorithm (AdaGrad), root mean square propagation (RMSprop), adaptative moment estimation (Adam) 등의 적용을 검토했다.

모델 개발에서 실제로 적용한 뉴런의 수, 은닉층 수, 활성

화 함수, 옵티마이저와 더불어서 학습을, epoch 등은 하이퍼파라미터 조율을 통해 결정되었다. 조율에 적용한 하이퍼파라미터 조합은 2.3절에, 조율 결과는 3.1절에서 상세히 기술하였다.

2.2 데이터

신경망 모델 개발을 위한 데이터셋으로는 Lee (2021)가 수행한 연구에서 적용한 자료를 사용하였다. Lee (2021)는 federal highway administration (FHWA)의 deep foundation load test database (DFLTD)에서 강관개관말뚝(SPO)에 대한 정적재하시험자료를 선별하여 모델개발을 위한 데이터셋으로 사용하였다. 데이터셋에 대한 주요 내용을 쉽게 참고할 수 있도록 다음과 같이 요약하였다.

2.2.1 특성과 기준

선별한 개관식 원형 강관말뚝에 대한 54개 재하시험자료를 기반으로 인공지능 모델 개발을 위한 데이터프레임을 가공하였다. 데이터프레임은 모두 29개의 인덱스에 대해 각각 54개 입력값을 가지는 구조로 만들었다. 데이터프레임은 모델훈련을 위한 데이터셋으로 사용할 수 있도록 가공하였으며 상세한 내용은 Lee (2021)에서 참조할 수 있다. 결과적으로 최대하중(MaxF), 최대변위(MaxD), 길이(Len), 직경(Dia), 항복하중(Qu), 주면에서의 평균 N값(mean_N), 선단 N값(NB)의 7개 인덱스를 가진 데이터프레임을 만들어 모델 개발에 적용하였다. 데이터프레임의 인덱스명을 MaxF 등과 같이 괄호 안의 기호로 나타냈다.

데이터프레임의 인덱스에 대해서 모델 훈련 및 검증을 위한 데이터셋의 기준(target, label)을 항복하중(Qu)으로, 나머지 6개 인덱스는 특성(feature)으로 정했다. 각 특성과 기준 상호 간의 상관관계를 나타내면 Fig. 1과 같다. 이 결과에서 기준 Qu에 대한 특성 간의 상관관계는 Fig. 1의

Fig. 1. Matrix of the Pearson Correlation Coefficient between the Label and Features

3행 혹은 3열에서 볼 수 있다. Qu에 대한 상관계수는 MaxF, Dia, Len, NB, mean_N, MaxD의 순으로 0.99, 0.78, 0.56, 0.30, 0.04, -0.75로 구해져 Dia, Len, NB가 상대적으로 중요한 특성임을 인지하였다.

임의의 두 특성이 독립적 특성이 되기 위해서는 이들 특성들이 가지는 다른 특성과의 상관관계가 유사하지 않아야 한다. MaxF와 Qu의 경우에 다른 특성과의 상관관계가 거의 일치하는 관계를 나타냈다. 따라서 MaxF는 Qu에 대해 독립특성을 가지지 못한 것으로 판단되어 모델훈련의 특성에서 배제하였다. MaxD는 Qu와의 상관관계가 음수로 나타나 침하량이 클수록 지지력이 감소하는 추세를 반영하였다. MaxD의 상관계수를 절대값으로 고려한 경우 다른 특성과의 상관관계가 MaxF 및 Qu와 유사하였다. 또한 MaxD는 말뚝 재하시험 결과로부터 얻은 측정값으로서 말뚝 및 주변지반의 물리적 특성을 나타내는 다른 특성과는 차별화가 필요한 것으로 판단되어 특성에서 제외하였다. 최종적으로 모델훈련을 위한 특성으로 Dia, Len, NB, mean_N의 4개 특성을 선정하였다.

2.2.2 훈련 및 검증 데이터셋

데이터셋을 사이킷런(scikit-learn) (Pedregosa et al., 2011)의 train_test_split 함수를 사용하여 훈련셋트 75%와 테스트셋트 25%로 분리하였다. 훈련셋트는 하이퍼파라미터 조율을 위해 25%의 데이터를 검증셋트로 분리하였다.

2.2.3 정규화 데이터셋

데이터셋에서 특성의 범위와 단위가 특성별로 상이하므로 데이터를 정규화하였다. 정규화한 데이터를 사용할 경우 회귀분석에서 결정한 기술키로부터 특성이 미치는 영향을 균등하게 평가할 수 있기 때문이다.

훈련셋트에 대해서는 각 특성에 대한 평균과 표준편차를 이용하여 특성의 입력값의 분포가 평균이 0이고 표준편차 1이 되도록 정규화하였다. 테스트셋트에 대해서는 데이터셋 정규화가 훈련된 모델의 테스트에 영향을 미치지 않도록 훈련데이터의 평균과 표준편차를 이용하여 정규화하였다. 그 결과 테스트셋트는 평균이 0, 표준편차가 1에 가까운 값을 가지도록 정규화되었다. 기준은 일률적으로 1/1,000 kN의 축척으로 정규화하였다. 그 결과 기준값(Qu)의 범위는 0.55~33.61, 평균 8.178, 표준편차 8.479이었다. Figs. 2(a)~2(c)에 정규화한 훈련셋트와 테스트셋트, 기준값의 분포를 각각 나타냈다.

2.3 모델 훈련 및 검증

2.1에서 설명한 신경망 모델을 2.2의 정규화 데이터셋트에 대해 훈련 및 검증하였다. 훈련 및 검증 성과를 최적화하기 위해 사이킷런의 Grid Search 방법을 적용하여 하이퍼파라미터를 조율하였다. 이 과정에서 검증의 신뢰도 향상을 위해

Fig. 2. Distribution of the Normalized Data Set

Table 1. Variations of Hyperparameters for Grid Search

Hyperparameter	Input
Neutons in input layer	4
Neurons in output layer	1
hidden layers	1, 2, 3
Neurons in the hidden layer	8, 16, 32
Learning rate	1e-5, 1e-4, 1e-3
Activation function	ReLU, tanh
Optimizer	SGD, Adam
Number of epochs	Early Stopping
Folds in the cross validation	4

k-fold 교차검증(cross validation)을 적용했다. Grid Search에서 검토한 하이퍼파라미터를 정리하면 Table 1과 같다.

Fig. 3. Grid Search Result

3. 결과 및 분석

3.1 최적모델

Grid Search 방법에서 총 108가지의 하이퍼파라미터를 조합한 각 모델에 대해 4점 교차검증을 수행하였다. 최소 훈련 손실(MSE) 조건을 만족하는 모델을 결정하기 위해 108개 모델에 대한 MSE를 Fig. 3에 비교하여 나타냈다. Fig. 3의 결과에서 손실값이 낮은 상위 20위까지의 모델에 대한 하이퍼파라미터 조합과 모델 훈련 및 검증 결과를 발췌하여 Table 2에 나타냈다.

Fig. 3과 Table 2의 결과로부터 최적 하이퍼파라미터로서 은닉층의 개수는 3, 은닉층의 뉴런수 32, 학습율 0.001, 활성화 함수 ReLU, 옵티마이저는 Adam의 조건을 적용한 모델에 대해서 테스트 손실이 최소로 발생하였다. 최소 손실이 발생한 모델을 최적모델로 정했다.

Table 2의 결과에서 최적모델의 검증셋에 대한 평균 점수(-손실)는 -37.832, 표준편차는 13.566이었다. 같은 모델에 대해서 훈련셋에 대한 평균점수는 -17.893, 표준편차는 2.407이었다. 이와 같이 모델은 검증셋보다 훈련셋에 대해서 기준값을 보다 정확하게 추정한 결과를 보였다. 따라서 모델이 다소 과대적합 되었을 가능성이 있는 것으로 판단되었다. 또한 훈련과 검증에서 얻은 점수가 모두 낮은 편이고 검증에 대해서는 표준편차가 다소 크게 나타났다. 이 결과는 모델에서 기준값을 추정할 수 있는 모델의 성능이 다소 낮다는 것을 의미하였다. 낮은 성능의 원인으로는 데이터셋 샘플 수의 부족이 가장 큰 이유인 것으로 추정되었다.

하이퍼파라미터가 모델의 훈련에 미치는 영향을 Table 2의 결과로부터 추론하여 보았다. 은닉층의 개수, 뉴런의 개수, 활성화함수 등의 하이퍼파라미터들은 20위 내의 결과에 적용된 경우가 비교적 고르게 분포되어 있어 상대적으로

Table 2. Grid Search Results Sorted by the Test Score

Rank	Hidden Layers	Neurons	Activation Function	Optimizer	Learning Rate	Mean Fit Time	Mean Test Score	Std. Test Score	Mean Train Score	Std. Train Score
1	3	32	relu	Adam	0.001	1.535	-37.832	13.566	-17.893	2.407
2	3	8	relu	SGD	0.001	1.473	-45.321	43.146	-16.852	5.704
3	3	32	relu	SGD	0.001	1.476	-45.583	32.512	-13.825	3.808
4	3	16	relu	SGD	0.001	1.469	-45.924	44.499	-14.479	2.725
5	2	32	relu	SGD	0.001	1.438	-49.462	44.299	-16.078	3.519
6	2	8	relu	SGD	0.001	1.526	-57.105	58.690	-19.461	4.197
7	3	32	tanh	Adam	0.001	1.489	-57.372	66.749	-47.848	19.142
8	3	32	tanh	SGD	0.001	1.418	-59.449	60.455	-35.987	11.278
9	2	32	tanh	SGD	0.001	1.385	-60.586	67.380	-35.821	9.785
10	1	32	relu	SGD	0.001	1.343	-63.548	79.247	-22.549	4.132
11	2	16	relu	SGD	0.001	1.386	-64.029	65.322	-17.686	3.302
12	2	32	relu	Adam	0.001	1.433	-64.140	67.960	-42.265	10.368
13	3	16	tanh	SGD	0.001	1.491	-67.167	77.562	-47.809	14.733
14	3	16	relu	Adam	0.001	1.469	-71.411	70.027	-56.335	18.380
15	1	16	relu	SGD	0.001	1.350	-72.123	80.128	-38.564	13.531
16	3	8	tanh	SGD	0.001	1.479	-73.575	88.663	-58.141	19.050
17	1	16	tanh	SGD	0.001	1.373	-76.244	91.381	-52.595	12.944
18	1	32	tanh	SGD	0.001	1.360	-76.585	93.662	-46.347	8.602
19	2	16	tanh	SGD	0.001	1.465	-77.445	89.711	-49.111	12.634
20	2	8	tanh	SGD	0.001	1.383	-78.993	94.684	-58.562	18.427

영향이 크지 않은 것으로 판단되었다.

옵티마이저의 경우에는 SGD 방법이 상위 순위에 많이 나타나는 추세를 보였지만 최소 손실은 Adam 방법에서 주로 얻을 수 있었다. 학습율은 0.001이 지배적이었다. 학습율이 낮아지는 경우에 기울기의 변화가 심한 작은 미소 구간으로 해가 수렴하는 현상이 발생하여 보다 큰 손실이 발생하는 것으로 추정되었다.

해석에 소요되는 평균훈련시간(mean fit time)은 하이터파라미터 조건에 대해서 크게 변동하지는 않았다. 하지만 작은 차이에도 불구하고 데이터 샘플 수가 증가하는 경우에 전체 계산시간이 함께 증가하게 될 것이므로 향후 옵티마이저의 영향에 대한 상세한 검증이 필요한 것으로 판단되었다. 108 개 조합에 대한 전체 계산시간은 항상 일정하지는 않았지만 인텔 Core(TM) i7-8700 CPU @3.20GHz 컴퓨터에서 약 700초 정도 소요되었다.

3.2 최적모델 평가

Grid Search에서 결정된 최적 하이터파라미터를 적용한 최적모델의 모델 구조와 층별 매개변수 수를 Figs. 4(a)와 4(b)에 각각 보였다. 이 모델에 대해서 학습율 0.001, 활성화 함수 ReLU, 옵티마이저는 Adam의 조건을 적용하여 훈련한 모델의 적합성을 검토하였다. 훈련된 모델의 훈련셋트에

(a) Model structure

Model: "sequential_1322"

Layer (type)	Output Shape	Param #
dense_3771 (Dense)	(None, 32)	160
dense_3772 (Dense)	(None, 32)	1056
dense_3773 (Dense)	(None, 32)	1056
dense_3774 (Dense)	(None, 1)	33

Total params: 2,305
Trainable params: 2,305
Non-trainable params: 0

(b) Parameters of each layers

Fig. 4. Model Structure and Parameters for Each Layer (LeNail, 2019)

Fig. 5. Distribution of Error for the Best Model

Fig. 6. Comparison between the Predicted and the Measured Samples

대한 MSE는 14.509, MAE는 2.796이었다. 테스트셋트에 대해 검증한 결과 MSE는 26.375, MAE는 3.510이었다. 이들 결과에 대해서 기준값 예측을 위한 모델의 성능 수준을 평가하고 과대적합의 가능성을 검토하였다.

모델로부터 구한 예측값의 신뢰도를 보기 위해 모든 데이터셋트에 대해서 예측값을 기준값과 비교한 결과를 검토하였다. Fig. 5에 예측값과 기준값 간의 오차 분포를 나타냈다. Fig. 6에는 훈련셋트와 테스트셋트를 구분하여 예측값과 기준값 간의 오차를 각 샘플에 대해서 나타냈다. Fig. 7에는 예측값과 기준값 간의 상관관계를 나타냈다.

Fig. 5의 결과에서 훈련셋트와 테스트셋트에 대한 오차분포가 유사하게 나타났다. 과대적합이 발생하는 경우에 테스트셋트에 대한 오차분포가 훈련셋트의 경우보다 넓게 나타날 것이다. Fig. 5에서는 유사한 분포를 보이므로 과대적합의 가능성이 낮은 것으로 판단되었다. Fig. 6의 결과에서 각 샘플에 대한 오차를 시각적으로 확인할 수 있었다. 각 샘플에 대한 예측값은 과대적합이 발생하는 경우에 훈련데이터에 대해서만 매우 정확한 예측을 하게 되나 이 결과에서는 그러한 현상을 볼 수 없었다. 테스트셋트에 대한 예측값도 기준값을 훈련셋트의 경우와 유사한 수준으로 예측한 것으로 나타났다. 이와 같은 결과는 예측값과 기준값의 일치성을 비교한 Fig. 7의 결과에서도 확인할 수 있었다. 예측값과 기준값 간의 상관관계(r)는 훈련셋트에 대해서 0.905, 테스트

Fig. 7. Correlation between the Predicted and the Measured Samples

Fig. 8. Case Requiring the Early Stopping

셋트에 대해서 0.721 수준의 일치성을 보였다.

모델의 기준값 예측 성능을 평가하기 위해 Fig. 7의 결과에 대해서 구한 결정계수는 훈련셋트에 대해서 0.786, 테스트셋트에 대해서 0.466을 얻었다. 이와 같이 훈련셋트에 대해서는 유용한 수준의 결정계수를 얻었다. 하지만 테스트셋트에 대해서는 상대적으로 낮은 값을 얻은 이유는 데이터셋트의 샘플 수 부족이 원인으로 추정되었다.

샘플 수가 적은 경우에 과대적합의 문제가 발생할 가능성이 높아 early stopping을 적용한 훈련 및 검증을 수행하였다. 훈련에서 발생할 수 있는 과대적합의 가능성을 예시하기 위해 모델 훈련 시 epoch에 따른 손실의 변화를 Fig. 8에 보였다. Fig. 8의 결과에서 6번째 epoch을 지나서 계속 훈련할 경우 훈련셋트에 과대적합될 가능성이 있음을 확인하였다.

4. 결 론

신경망 알고리즘을 사용하여 말뚝의 지지력을 추정하는 모델을 개발하고 적용성을 평가한 연구 결과 다음과 같은 결론을 도출할 수 있었다.

- 1) Grid Search 방법을 적용하여 은닉층의 개수, 뉴론의

개수, 활성화함수, 학습율, 옵티마이저 등의 하이퍼파라미터에 대해서 최소 손실을 얻을 수 있는 조합을 결정하였다. 하이퍼파라미터의 다양한 조합 조건들에 대해서 손실이 작은 순으로 정렬한 경우에 학습율 0.001 인 경우가 상위 순위를 지배적으로 차지했다. 옵티마이저의 경우에는 SGD 방법이 상위 순위에 많이 나타나는 추세를 보였지만 최소 손실은 Adam 방법에서 얻을 수 있었다. 다른 조건들은 상위 순위에 비교적 고르게 분포하여 큰 변화가 없었다. 하지만 이러한 결과는 데이터의 영향에 크게 의존하므로 새로운 데이터에 대해서는 별도의 조율 수행이 필요함을 제안할 수 있었다.

- 2) 데이터수가 적은 경우에 훈련데이터에 대한 모델의 과대적합을 방지해야 하므로 early stopping 방법을 적용하여 훈련했다. 최적모델에 대한 훈련결과 MSE로 평가한 손실은 훈련셋트에 대해서 14.509, 테스트셋트에 대해서 26.375였다. 훈련셋트에 대한 손실이 테스트셋트에 비해 작게 나타났지만 차이가 크지 않아 과대적합이 배제된 것으로 평가할 수 있었다.
- 3) 기준값 예측에 대한 모델의 적합성을 평가하기 위해서 결정계수를 구한 결과 훈련셋트와 테스트셋트에 대해서 각각 0.786, 0.466의 값을 얻었다. 테스트셋트에 대한 결정계수 값이 낮은 편이므로 기준값 예측을 위한 모델의 적합성이 다소 부족한 것으로 판단되었다. 테스트셋트에 대한 결정계수는 훈련데이터의 샘플 수를 증가시켜 개선할 수 있을 것으로 판단되었다.
- 4) 결론적으로 이 연구에서 개발한 AI 모델은 말뚝지지력을 절대오차 약 3,000 kN, 결정계수 약 0.5 수준으로 예측 가능한 것으로 나타났다. 연구결과에 근거하여 말뚝지지력 예측을 위한 AI 모델의 적용성을 확인할 수 있었다.

References

- Ardalan, H., Eslami, A., and Nariman-Zadeh, N. (2009). Piles shaft capacity from CPT and CPTu data by polynomial neural networks and genetic algorithms. *Computers and Geotechnics*, Vol. 36, No. 4, pp. 616-625.
- Benali, A., Boukhatem, B., Hussien, M.N., Nechnech, A., and Karray, M. (2017). Prediction of axial capacity of piles driven in non-cohesive soils based on neural networks approach. *Journal of Civil Engineering and Management*, Vol. 23, No. 3, pp. 393-408.
- Ebid, A.M. (2021). 35 Years of (AI) in Geotechnical Engineering: State of the Art. *Geotechnical and Geological Engineering*, Vol. 39, No. 2, pp. 637-690.
- Harandizadeh, H. (2020). Developing a new hybrid soft computing technique in predicting ultimate pile bearing capacity using cone penetration test data, *Artificial Intelligence for Engineering Design. Analysis and Manufacturing*, Vol. 34, No. 1, pp. 114-126.
- Juwaied, N.S. (2018). Applications of artificial intelligence in geotechnical engineering. *ARP Journal of Engineering and Applied Sciences*, Vol. 13, No. 8, pp. 2764-2785.
- Kordjazi, A., Pooya Nejad, F., and Jaksa, M.B. (2014). Prediction of ultimate axial load-carrying capacity of piles using a support vector machine based on CPT data. *Computers and Geotechnics*, Vol. 55, pp. 91-102.
- Lee, B.S. (2021). A Study of Machine Learning Models to Estimate a Pile Load Capacity. *Journal of Korea Academia-Industrial cooperation Society*, under review.
- LeNail, A. (2019). NN-SVG: Publication-Ready Neural Network Architecture Schematics. *Journal of Open Source Software*, Vol. 33, No. 4, 747. Retrieved from <https://doi.org/10.21105/joss.00747>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830.
- Pham, T.A., Ly, H.-B., Tran, V.Q., Giap, L.V., Vu, H.-L.T., and Duong, H.-A.T. (2020). Prediction of Pile Axial Bearing Capacity Using Artificial Neural Network and Random Forest. *Applied Sciences*, Vol. 10, No. 5, 1871. Retrieved from <https://www.mdpi.com/2076-3417/10/5/1871>
- Shahin, M.A. (2010). Intelligent computing for modeling axial capacity of pile foundations. *Canadian Geotechnical Journal*, Vol. 47, No. 2, pp. 230-243.
- Shahin, M.A. (2013). 8 - Artificial Intelligence in Geotechnical Engineering: Applications, Modeling Aspects, and Future Directions. In X.-S. Yang, A.H. Gandomi, S. Talatahari, and A.H. Alavi, (Eds.), *Metaheuristics in Water, Geotechnical and Transport Engineering* (pp. 169-204). Oxford: Elsevier.
- Shahin, M.A. (2015a). Use of evolutionary computing for modelling some complex problems in geotechnical engineering. *Geomechanics and Geoengineering*, Vol. 10, No. 2, pp. 109-125.
- Shahin, M.A. (2015b). A review of artificial intelligence applications in shallow foundations. *International Journal of Geotechnical Engineering*, Vol. 9, No. 1, pp. 49-60.
- Shahin, M.A. (2016). State-of-the-art review of some artificial

intelligence applications in pile foundations. *Geoscience Frontiers*, Vol. 7, No. 1, pp. 33-44.

Shahin, M.A., and Jaksa, M.B. (2009). Intelligent computing for predicting axial capacity of drilled shafts. *International Foundation Congress and Equipment Expo (IFCEE'09)*, Orlando, Florida, pp. 26-33.

Zhang, W., Zhang, R., Wu, C., Goh, A.T.C., Lacasse, S., Liu, Z., and Liu, H. (2020). State-of-the-art review

of soft computing applications in underground excavations. *Geoscience Frontiers*, Vol. 11, No. 4, pp. 1095-1106.

Received	July 8, 2021
Revised	July 9, 2021
Accepted	July 30, 2021