

Levenberg-Marquardt 인공신경망 알고리즘을 이용한 지반공학문제의 적용성 검토

Application of Artificial Neural Network with Levenberg-Marquardt Algorithm in Geotechnical Engineering Problem

김영수¹⁾, Young-Su Kim, 이재호²⁾, Jae-Ho Lee, 서인식³⁾, In-Shik Seo, 김현동⁴⁾, Hyun-Dong Kim, 신지섭⁵⁾, Ji-Sub Shin, 나윤영⁶⁾, Yun-Young Na

- 1) 경북대학교 토목공학과 교수, Professor, Dept. of Civil Engineering, Kyungpook National University.
- 2) 경북대학교 토목공학과 박사 후 연구원, Post-Doctoral Researcher, Dept. of Civil Engineering, Kyungpook National University.
- 3) 경동정보대학 토목공학과 교수, Professor, Dept. of Civil Engineering, Kyungdong college of Techno-information.
- 4) 도화엔지니어링, Engineer, Dohwa Engineering.
- 5) 경북대학교 토목공학과 박사과정, Graduate Student, Dept. of Civil Engineering, Kyungpook National University.
- 6) 경북대학교 토목공학과 석사과정, Master course, Dept. of Civil Engineering, Kyungpook National University.

SYNOPSIS : Successful design, construction and maintenance of geotechnical structure in soft ground and marine clay demands prediction, control, stability estimation and monitoring of settlement with high accuracy. It is important to predict and to estimate the compression index of soil for predicting of ground settlement. Lab. and field tests have been and are indispensable tools to achieve this goal. In this paper, Artificial Neural Networks (ANNs) model with Levenberg-Marquardt Algorithm and field database were used to predict compression index of soil in Korea. Based on soil property database obtained from more than 1800 consolidation tests from soils samples, the ANNs model were proposed in this study to estimate the compression index, using multiple soil properties. The compression index from the proposed ANN models including multiple soil parameters were then compared with those from the existing empirical equations.

Key words : compression index, Artificial Neural Networks, field database, predict

1. 서 론

연약지반에 구조물을 시공하는 경우에 대상 구조물의 안정성과 침하에 대한 사전 검토가 필요하며 이를 위해서는 연약지반의 공학적 특성을 파악하는 것이 중요하다(Das, 2007년). 연약지반의 공학적 특성은 주로 현장시험이나 실내시험을 통하여 추정된다. 연약지반의 압밀특성을 규명하는 토질상수 중에서 압밀계수는 현장에서 수행되는 원위치 시험법 중에서 CPTu(cone penetration test with pore water measurement), SBPT(self-boring pressuremeter test), DMT(flat dilatometer) 등이 최근에 국내에서도 많이 제안되고 있으며, 이를 원위치시험 및 실내시험들의 상관식을 근거한 경험식들을 최근까지도 많은 국외 연구자들에 의하여 이루어지고 있다(윤길립 등, 2003년). 연약지반의 압밀 특성 값들 중 연

약지반의 최종 침하량 산정에 가장 중요한 요소인 압축지수를 현장시험으로부터 신뢰성 있게 예측할 수 있는 시험장치는 아직 개발되어 있지 못한 실정이다. 따라서, 압축지수의 산정은 기존 제안자들에 의한 경험식을 이용하거나 또는 실내시험을 수행하여 결정하여야 하는 문제가 있다. 기존 제안된 국외 경험식의 경우는 대부분 국내지반과 전혀 다른 물성치, 응력이력 그리고 흡입자 구조배열을 가지는 국외지반에 국한된 결과라고 할 수 있다(윤길립 등, 2003년). 압축지수의 산정에 있어 불확실성을 포함할 수 있는 기존 국외의 경험식을 맹목적으로 적용하는 것보다는 불확실성을 최소화하고 국내 점토의 기본 물성치들에 대한 충분한 검토에 근거하는 적합한 압축지수의 경험식을 제안하여 경제적인 설계가 이루어질 수 있도록 경험식을 제안할 필요성이 증가하고 있다(윤길립 등, 2003년). 국내 연약지반의 압축지수를 추정하는 연구는 송만순 (1988년), 윤길립 등 (2003년)이 수행하여 실제 값과의 비교를 통해 예측식의 제안 및 타당성을 검토한 바가 있다. 본 논문에서는 압밀에 대한 공학적 특성들 중에 점토의 압축성을 대변하고 있는 압축지수와 관련한 지반 물성 값들과 상관성을 보다 면밀히 조사하고자 인천지역의 약 187개 데이터에 대하여 분석하였다. 분석된 결과와 데이터베이스를 토대로 인천지역 연약지반에 적합한 새로운 압축지수에 대해 인공신경망을 이용하여 예측 식을 제안하고자 한다. 제안된 예측 식은 기존의 추정식과 비교검토를 수행하였다.

2. 적용된 인공신경망의 개요

2.1 인공신경망의 정의

인공신경망(artificial neural network)은 인간 두뇌의 구조와 동작방식을 모델로 하여 간단한 기능을 하는 단위 처리기 사이를 연결시킨 망구조로 뇌의 신경세포를 단순화시켜 수학적으로 모델링한 인공 신경세포들과 그들 간의 연결로 이루어져 있다 (Hecht-Nielsen, 1990년). 인공신경망을 구성하는 처리소자들은 전달함수를 가지고 있어서 시스템의 비선형 특성을 학습할 수 있고 입력과 출력공간의 사상(mapping)특성에 따라 연상기억(associative memory), 필터(filter), 변환(transformation), 인식(perception), 최적화(optimization)등의 기능을 수행할 수 있다 (Hecht-Nielsen, 1990년). 인공신경망은 입력층과 출력층을 거치면서 입력층에서 입력되는 여러 변수들 중에서 출력에 큰 영향을 미치는 변수를 학습규칙에 의하여 선택한 후 예측 안전율을 추정할 수 있다. 여기에서 사용되는 학습규칙은 일반화 델타규칙 (generalized delta rule)이며 최급하강법 (gradient descent method)에 의하여 시행되는데, 최급하강법은 다층 퍼셉트론 인공신경망을 학습시켜 모형의 최적 매개변수를 구하는 방법이다 (Rumelhard 등, 1986년). 다층 퍼셉트론은 입력층과 출력층 사이에 하나 이상의 은닉층이 존재하는 인공신경 회로망으로 그림 1과 같은 계층구조를 갖는다. 단층퍼셉트론의 경우 학습 가능한 층이 하나만 존재하기 때문에 선형 분리 가능한 문제에만 적용가능하다는 한계가 있는데, 이러한 한계를 벗어나기 위하여 다층 퍼셉트론을 이용한다. 다층 퍼셉트론(multi-layer perceptron)은 단층 퍼셉트론과 유사한 구조를 가지고 있지만 은닉층과 각 유닛의 입출력 특성을 비선형으로 함으로써 네트워크의 능력을 향상시켜 단층 퍼셉트론의 여러 가지 단점들을 극복했다(Rumelhard 등, 1986년).

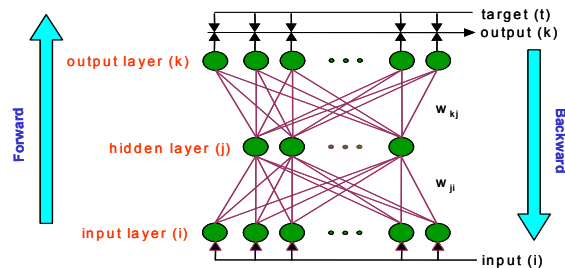


그림 1. 다층 신경망의 구조 및 학습 모식도 (Rumelhard 등, 1986년)

2.2 Levenberg-Marquardt 알고리즘 (LM)

Levenberg-Marquardt 알고리즘(LM 알고리즘)은 비선형 함수들의 제곱의 합으로 된 최소화 함수로 만들어진 Newton's method를 변형한 방법이다. 이것은 Mean Squared Error (MSE)를 performance index로 사용했을 때 인공신경망 학습에 매우 잘 맞는 것으로 알려져 있다. performance index $F(x)$ 를 최적화시키기 위한 Newton's method는 다음과 같다.

$$x_{k+1} = x_k - A_k^{-1}g_k \quad (1)$$

여기서, $A_k = \nabla^2 F(x)|_{x=x_k}$ 이고, $g_k = \nabla F(x)|_{x=x_k}$ 이다. 일반적으로 $\Delta F(x)$ 는 식 (2)로 나타낼 수 있으며, J 는 Jacobian matrix로서 식 (3)로 표현되어진다.

$$\nabla F(x) = 2J^T(x)v(x) \quad (2)$$

$$J = \begin{bmatrix} \frac{\partial \epsilon_1}{\partial w_1} & \cdots & \frac{\partial \epsilon_1}{\partial w_n} & \frac{\partial \epsilon_1}{\partial w_0} \\ \vdots & & \vdots & \vdots \\ \frac{\partial \epsilon_p}{\partial w_1} & \cdots & \frac{\partial \epsilon_p}{\partial w_n} & \frac{\partial \epsilon_p}{\partial w_0} \end{bmatrix} = \begin{bmatrix} x_1 & \cdots & x_{n_1} & 1 \\ \vdots & & \vdots & \vdots \\ x_{1p} & \cdots & x_{n_p} & 1 \end{bmatrix} \quad (3)$$

여기서, w 는 연결강도의 벡터, w_0 는 뉴런의 바이어스, ϵ 은 오차 벡터(개별적 패턴에 대한 실제값과 네트워크에 의해 계산된 값과의 차)이다. 다음으로 k, j 요소의 Hessian matrix를 구하여 보면, 식 (4), (5), (6) 그리고 (7)로 나타낼 수 있다.

$$[\nabla^2 F(x)]_{k,j} = \frac{\partial^2 F(x)}{\partial x_k \partial x_j} = 2 \sum_{i=1}^N \left\{ \frac{\partial v_i(x)}{\partial x_k} \frac{\partial v_i(x)}{\partial x_j} + v_i(x) \frac{\partial^2 v_i(x)}{\partial x_k \partial x_j} \right\} \quad (4)$$

$$\nabla^2 F(x) = 2J^T(x)J(x) + 2S(x) \quad (5)$$

$$S(x) = \sum_{i=1}^N v_i(x) \nabla^2 v_i(x) \quad (6)$$

$$\nabla^2 F(x) \cong 2J^T(x)J(x) \quad (7)$$

식 (7)와 식 (2)를 식(1)에 대입하면 Gauss-Newton method를 식 (10)과 같이 구할 수 있다.

$$x_{k+1} = x_k - [2J^T(x_k)J(x_k)]^{-1}2J^T(x_k)v(x_k) = x_k - [J^T(x_k)J(x_k)]^{-1}J^T(x_k)v(x_k) \quad (10)$$

Newton's method에 비해 Gauss-Newton method가 더 나은 점은 2차미분을 계산할 필요가 없다는 것이다. 그러나 Gauss-Newton method에 한 가지 문제점이 있다. 그것은 $H=J^TJ$ 의 역행렬을 구하기가 어렵다는 것이다. 이것은 다음의 간략화된 Hessian 매트릭스를 수정해서 해결할 수 있다.

$$G = H + \mu I \quad (11)$$

위 매트릭스의 역행렬을 구하기 위해, H 의 고유값과 고유벡터를 $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ 와 $\{z_1, z_2, \dots, z_n\}$ 라 가정하면 식 (12)로 나타낼 수 있다.

$$Gz_i = [H + \mu I]z_i = Hz_i + \mu z_i = \lambda_i z_i + \mu z_i = (\lambda_i + \mu)z_i \quad (12)$$

고유벡터 G 는 고유벡터 H 와 같아지고, 고유벡터 G 는 $(\lambda_i + \mu)$ 가 된다. G 는 모든 i 에 대해 $(\lambda_i + \mu) > 0$ 이 될 때까지 μ 가 증가함에 의해 양의 값이 될 수 있다. 그러므로 매트릭스는 역행렬이 된다. 결과적으로 LM 알고리즘은 식 (13) 또는 (14)로 나타낼 수 있다.

$$x_{k+1} = x_k - [J^T(x_k)J(x_k) + \mu_k I]^{-1} J^T(x_k)v(x_k) \quad (13)$$

$$\Delta x_k = -[J^T(x_k)J(x_k) + \mu_k I]^{-1} J^T(x_k)v(x_k) \quad (14)$$

이 알고리즘은 μ_k 가 증가함에 따라 작은 학습율을 가지는 steepest descent 알고리즘에 접근할 수 있는 매우 유용한 방법이다. LM 알고리즘은 비선형 함수들의 제곱의 합으로 된 최소화 함수로 만들어진 Newton's method를 변형한 방법이다. 이것은 Mean Squared Error (MSE)를 performance index $F(x)$ 로 사용했을 때 인공신경망 학습에 매우 잘 맞는 것으로 알려졌다. Levenberg-Marquardt 알고리즘을 multilayer network의 훈련에 적용하는 방법은 다음과 같다. multilayer network 훈련에서 performance index는 mean squared error이다. 각 output이 같은 확률로 일어난다고 가정하면, mean squared error는 훈련세트의 Q 번째 output의 squared error의 합에 비례하며, 이것은 식 (15)와 같이 표현되어진다.

$$F(x) = \sum_{q=1}^Q (t_q - a_q)^T (t_q - a_q) = \sum_{q=1}^Q e_q^T e_q = \sum_{q=1}^Q \sum_{j=1}^{S^M} (e_{j,q})^2 = \sum_{i=1}^N (v_i)^2 \quad (15)$$

여기서, $e_{j,q}$ 는 q 번째 입력값과 출력값의 쌍에 대한 j 번째 항의 오차이다. Levenberg-Marquardt 알고리즘에서 중요한 단계는 Jacobian matrix를 계산하는 것이다. 이것을 계산해내기 위해서는 역전파 알고리즘의 변환이 사용된다. network의 연결강도와 bias에 대해 일반적인 역전파 알고리즘에서는 squared error의 미분을 계산한다. Jacobian matrix를 만들기 위해서 squared error의 미분대신에 error의 미분을 계산해야한다. error 벡터는 식 (16), parameter 벡터는 식 (17)로 나타낸다.

$$V^T = [v_1 \ v_2 \ \dots \ v_N] = [e_{1,1} \ e_{2,1} \ \dots \ e_{S^M,1} \ e_{1,2} \ \dots \ e_{S^M,Q}] \quad (16)$$

$$X^T = [x_1 \ x_2 \ \dots \ x_n] = [w_{1,1}^1 \ w_{1,2}^1 \ \dots \ w_{S^1,R}^1 \ b_1^1 \ \dots \ b_{S^1}^1 \ w_{1,1}^2 \ \dots \ b_{S^M}^M] \quad (17)$$

여기서 $N=Q \times S^M$ 이고, $n=S^1(R+1) + S^2(S^1+1) + \dots + S^M(S^{M-1}+1)$ 이다. 식 (16)과 (17)를 식 (3)에 대입하면 multilayer network의 Jacobian matrix는 식 (18)과 같다.

$$J(x) = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_{1,1}^1} & \frac{\partial e_{1,1}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{1,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,1}}{\partial b_1^1} & \dots \\ \frac{\partial e_{2,1}}{\partial w_{1,1}^1} & \frac{\partial e_{2,1}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{2,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{2,1}}{\partial b_1^1} & \dots \\ \vdots & \vdots & & \vdots & \vdots & \\ \frac{\partial e_{S^M,1}}{\partial w_{1,1}^1} & \frac{\partial e_{S^M,1}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{S^M,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{S^M,1}}{\partial b_1^1} & \dots \\ \frac{\partial e_{1,2}}{\partial w_{1,1}^1} & \frac{\partial e_{1,2}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{1,2}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,2}}{\partial b_1^1} & \dots \\ \vdots & \vdots & & \vdots & \vdots & \end{bmatrix} \quad (18)$$

위의 Jacobian matrix 각 항은 역전과 알고리즘을 간단히 수정함으로써 계산될 수 있다. 기본적인 역전과 알고리즘의 각 항들의 계산은 식 (19)와 같다.

$$\frac{\partial \hat{F}(X)}{\partial x_l} = \frac{\partial e_q^T e_q}{\partial x_l} \quad (19)$$

Levenberg-Marquardt 알고리즘에 필요한 Jacobian matrix의 각 항들은 식 (20)과 같이 나타내어지며, 역전과 알고리즘의 미분은 식 (21)과 같이 계산된다.

$$[J]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial x_l} \quad (20)$$

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m} \quad (21)$$

여기서 오른쪽의 첫 번째 항은 sensitivity로써 정의된다. 역전과 알고리즘은 출력층부터 입력층으로 거꾸로 진행되면서 sensitivity가 계산된다. Jacobian matrix의 각 항도 같은 개념으로 다음과 같은 식의 Marquardt sensitivity를 사용해 계산될 수 있다.

$$\tilde{s}_{i,h}^m = \frac{\partial v_h}{\partial n_{i,q}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \quad (22)$$

Jacobian matrix의 각 항은 식 (23)으로 계산되며, bias는 식 (23)로부터 식 (24)로 나타내어 진다.

$$[J]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial w_{i,j}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \times \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = \tilde{s}_{i,h}^m \times \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = \tilde{s}_{i,h}^m \times a_{j,q}^{m-1} \quad (23)$$

$$[J]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial b_i^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \times \frac{\partial n_{i,q}^m}{\partial b_i^m} = \tilde{s}_{i,h}^m \times \frac{\partial n_{i,q}^m}{\partial b_i^m} = \tilde{s}_{i,h}^m \times a_{j,q}^{m-1} \quad (24)$$

Marquardt sensitivity는 출력층을 제외하고는 역전과 알고리즘의 sensitivity와 비슷한 방법으로 계산된다. 출력층의 Marquardt sensitivity는 다음과 같다.

$$\tilde{s}_{i,h}^M = \frac{\partial v_h}{\partial n_{i,q}^M} = \frac{\partial e_{k,q}}{\partial n_{i,q}^M} = \frac{\partial (t_{k,q} - a_{k,q}^M)}{\partial n_{i,q}^M} = \frac{\partial a_{k,q}^M}{\partial n_{i,q}^M} = \begin{cases} -f^M(n_{i,q}^M) & \text{for } i = k \\ 0 & \text{for } i \neq k \end{cases} \quad (25)$$

그러므로 입력 P_p 가 network에 적용되어서 출력 a_q^M 가 나올 때, Levenberg-Marquardt 알고리즘은 식 (26)으로 초기화된다.

$$\tilde{S}_q^M = -\dot{F}^M(n_q^M) \quad (26)$$

\tilde{S}_q^M matrix의 열은 Jacobian matrix의 한 행을 생성하기 위해 network를 통해 역전과 되어져야만 한

다. 생성된 행은 식 (27)을 이용해 역전파되어진다. 각 층의 전체적인 Marquardt sensitivity matrix는 입력에 대해 계산되어진 matrix를 붙이면서 만들어진다. 식 (28)을 이것을 나타낸 것이다.

$$\tilde{S}_q^m = \dot{F}^m(n_q^m)(W^{m+1})^T \tilde{S}_q^{m+1} \quad (27)$$

$$\tilde{S}^m = [\tilde{S}_1^m | \tilde{S}_2^m | \dots | \tilde{S}_Q^m] \quad (28)$$

sensitivity가 역전파된 후에, Jacobian matrix는 식(23)와 식(24)을 사용해 계산된다. LM 알고리즘은 다음과 같이 요약할 수 있으며, 그림 2는 LM 알고리즘의 순서도를 나타낸 것이다.

1. network에 모든 입력을 적용하여 그에 따른 network의 출력과 error $e_q = t_p - a_q^M$ 를 계산한다. 식 (15)을 이용해 모든 입력에 대한 squared error의 합 $F(x)$ 를 계산한다.
2. 식 (18)의 Jacobian matrix를 계산하다. 식 (26)로 초기화 시킨 후, 식 (27)을 이용해 sensitivity를 계산한다. 식 (28)를 이용해 Marquardt sensitivity에 각각의 matrix를 붙인다. 식 (23)와 (24)으로 Jacobian matrix의 각 항을 계산한다.
3. 식 (14)를 풀어 Δx_k 를 구한다.
4. $x_k + \Delta x_k$ 를 이용해 squared error의 합을 재계산한다. 새로운 squared error의 합이 1단계에서 구한 값보다 더 작다면, μ 에 ν 를 나누고 $x_{k+1} = x_k + \Delta x_k$ 로 하고 1단계로 다시 돌아간다. squared error의 합이 줄어들지 않는다면, μ 에 ν 를 곱하고 3단계로 돌아간다. Levenberg-Marquardt 알고리즘은 식 (2)의 평균이 미리 계산된 값보다 더 작거나 squared error의 합이 목표 error보다 작아진다면 수렴한다고 가정한다.

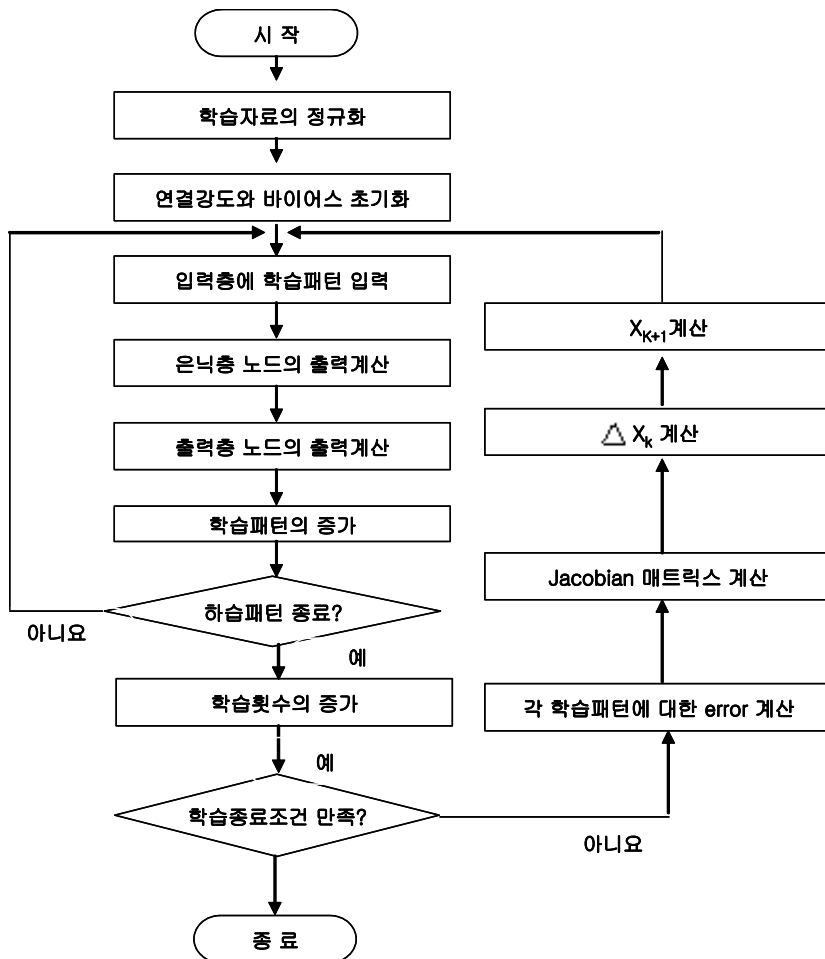


그림 2. LM 알고리즘의 계산 흐름도

3. 압축지수 추정방법

3.1 단일 지반변수 모델에 의한 압축지수 추정식

점토지반의 압축지수 추정을 위하여 제안되어진 경험식들 중에서 기본 물성치의 단일 지반변수를 적용하고 있는 경험식들을 요약하면 다음의 표 1과 같다. 국외 여러학자들의 연구결과에 의하면, 압축지수에 직접적인 영향을 줄 수 있는 단일 지반변수는 액성한계(w_l), 자연함수비(w_n), 현장 간극비(e_0), 그리고 소성지수(PI)로 규정하고 있으며 이들과 압축지수와의 관계는 모두 선형식으로 제안하고 있다.

표 1. 단일 지반변수를 포함하는 기존의 경험적 제안식들

제안자	제안식	비고
For $C_C = f(w_l)$		
Terzaghi 와 Peck (1967년)	$C_C = 0.009(w_l - 10)$	Normally consolidated clays
Azzouz 등 (1976년)	$C_C = 0.006(w_l - 9)$	All clays
Mayne (1980년)	$C_C = (w_l - 13)/109$	All clays
For $C_C = f(w_n)$		
Azzouz 등 (1976년)	$C_C = 0.01(w_n - 5)$	All clays
Koppula (1981년)	$C_C = 0.01w_n$	
Herrero (1983년)	$C_C = 0.01(w_n - 7.549)$	
For $C_C = f(e_0)$		
Nishida (1956년)	$C_C = 0.54(e_0 - 0.35)$	All clays
Cozzolino (1961년)	$C_C = 0.43(e_0 - 0.25)$	Brazilian clay
Sowers (1970년)	$C_C = 0.75(e_0 - 0.50)$	Clays with low plasticity
For $C_C = f(PI)$		
Nacci 등 (1975년)	$C_C = 0.02 + 0.014PI$	North Atlantic clay
Nakase 등 (1988년)	$C_C = 0.046 + 0.0104PI$	Bset for $PI < 50\%$

3.2 복수 지반변수 모델에 의한 압축지수 추정식

점토지반의 압축지수 추정을 위하여 제안되어진 경험식들 중에는 표 1과 같이 기본 물성치의 단일 지반변수를 적용하는 경우 외에 2개 이상의 복합 지반변수를 고려하는 경험식들도 제안되고 있으며 이를 요약하면 표 2와 같다.

표 2. 복합 지반변수를 포함하는 기존의 경험적 제안식들

제안자	제안식	비고
Rendon-Herrero (1983년)	$C_C = 0.141 G_S^{1.2} (1 + e_0) / G_S^{2.38}$	All clays
Azzouz 등 (1976년)	$C_C = 0.37(e_0 + 0.003w_l + 0.0004w_n - 0.34)$	678 data points
Koppula (1981년)	$C_C = -0.0997 + 0.009w_l + 0.0014PI + 0.0036w_n + 0.1165e_0 + 0.0025C_p$	109 data points
Carrier (1985년)	$C_C = 0.329[w_n G_S - 0.027w_p + 0.0133PI(1.192 + C_p/PI)]$	All clays
Koppula (1986년)	$C_C = 0.009w_n + 0.005w_l$	All clays
Al-Khafaji 와 Andersland (1992년)	$C_C = -0.156 + 0.411e_0 + 0.00058w_l$	72 data points

4. 압축지수와 관련된 데이터베이스의 특성 분석

연구대상 지구에서 압축지수에 영향을 미치는 다양한 인자들과의 상관성은 그림 3와 같다. 그림 3에서 N치, 200체 통과율에 의한 압축지수의 변화가 상당히 이산분포를 나타내고 있어 이들과 압축지수의 상관성은 높지 않은 것을 알 수가 있었다. 그림 3에서 초기 간극비, 비중, 자연함수비, 액성한계와 압축지수는 어느 정도 상관성이 있음을 알 수 있다.

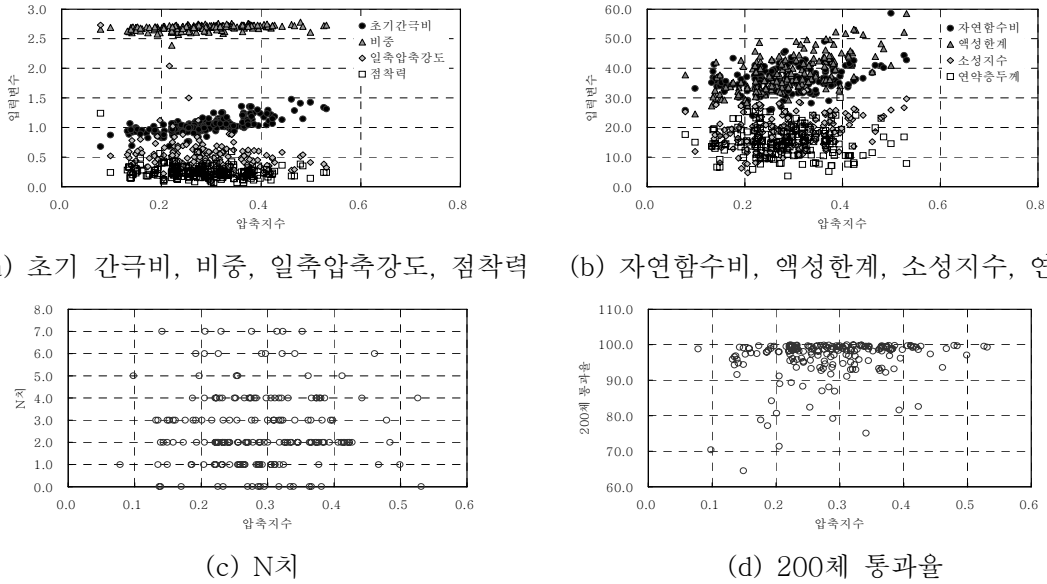


그림 3. 입력변수와 압축지수의 관계

5. 인공지능망을 이용한 압축지수의 추정

5.1 입력변수의 선정과 신경망의 설계

신경망의 처리과정은 데이터의 확보 후 각 모델에 적합한 영향인자의 파악이 선행되어야 한다.

표 3. 학습, 예측시 입력변수와 출력변수의 범위

입력변수와 출력변수	학습	예측
보링깊이 (m)	1.5 ~ 18	1.5 ~ 10.5
초기간극비 e_0	0.677 ~ 1.476	0.827 ~ 1.298
자연 함수비 w_n (%)	23.9 ~ 58.6	27.4 ~ 49.1
비중 G_s	2.39 ~ 2.78	2.58 ~ 2.75
액성한계 w_l (%)	24.6 ~ 58.5	27.3 ~ 52.3
소성지수 PI (%)	4.7 ~ 32.6	8.1 ~ 27.3
#200 통과율 C_p (%)	64.5 ~ 100	77.2 ~ 100
2 μ m 통과율 (%)	8 ~ 50	11 ~ 42
일축압축강도 q_u (kg/cm ²)	0.09 ~ 2.73	0.14 ~ 1.02
UU test의 점착력 c_{uu} (kg/cm ²)	0.07 ~ 3.01	0.07 ~ 0.43
N치	1 ~ 7	1 ~ 6
연약층 두께 (m)	3.6 ~ 30.1	5.1 ~ 25.4
압축지수 C_c	0.078 ~ 0.531	0.132 ~ 0.47

이러한 영향인자는 신경망의 입력변수가 되어 학습이 이루어지며 예측시 발생된 오차의 크기로 신뢰성을 검증하게 된다. 따라서, 신경망의 예측능력은 출력값에 영향을 미치는 가장 적합한 입력변수의 선택에 큰 영향을 받는데, 신경망의 학습에 있어 입력변수가 많아지면 학습시간은 증가하지만 비선형 거동에 대한 보다 정확한 거동예측이 가능하다. 하지만 입력자료 내 거동에 영향을 미치는 중요도가 낮은 자료가 많이 포함되어 있다면 입력자료 내 중요인자의 영향이 상대적으로 감소함으로써 입력변수들과 출력값의 상호관계가 제대로 정립되지 않아 예측능력은 현저히 감소하게 되며, 수렴속도 또한 현저히 저하하게 된다. 본 연구에서는 인천 00지구의 자료를 사용하였으며, 그 중 학습자료는 136개, 예측자료는 51개를 사용하였다. 위의 복합지반변수 모델의 변수들인 액성한계(w_f), 자연 함수비(w_n), 초기 간극비(e_0), 비중(G_s), #200 통과율(C_p), 소성지수(PI)와 연약지반의 특성을 나타내기 위해 보링 깊이와 2 μ m 통과율, 일축압축강도(q_u), UU test의 점착력(c_{uu}), 표준관입시험의 N치, 연약층 두께를 입력변수로 선택하였다. 표 3에서 학습, 예측시 12개 입력변수와 출력변수의 범위를 나타내었다. 신경망의 설계에서 은닉층은 하나로 하였고, 은닉층 노드수는 입력층 노드수의 1~3배를 사용하였다. 입력층 노드수가 12개 이므로, 은닉층 노드수는 12, 16, 20, 24, 28, 32, 36개로 변화시켜 학습과 예측을 시행하였다. 그림 4는 최적 신경망 모델을 나타낸 것이다. 학습율과 λ 는 0.01, 0.1, 0.2을 사용하였다. 그리고 학습종료 조건은 앞에서 2만번을 기준하였지만, 여기서는 보통 사용하는 방법으로 error가 수렴범위가 들어오거나 반복횟수가 일정한 횟수에 도달하면 종료하도록 하였다. 그래서 error의 수렴범위는 0.1, 반복횟수는 40만번으로 하여 학습을 수행하였다.

5.2 LM알고리즘에 의한 압축지수 학습 결과

(1) 최적신경망

LM 알고리즘의 압축지수 추정에 대한 최적신경망은 입력층 노드수 12, 은닉층 노드수 24, 출력층 노드수 1, λ 0.01임을 알 수 있다. 제안된 최적신경망모델의 상관계수는 0.8089, RMSE는 0.0485로 나타났다. 그림 4에서는 LM 알고리즘의 최적신경망 구조를 나타내었다. 그림 5은 실제 계측값과 LM 알고리즘의 예측값을 비교한 그래프이다. 그림 6에서는 LM 알고리즘의 반복횟수에 따른 Error의 변화를 나타내었다. 표 5에서 LM 알고리즘 최적신경망의 연결강도와 바이어스 값을 나타내었다.

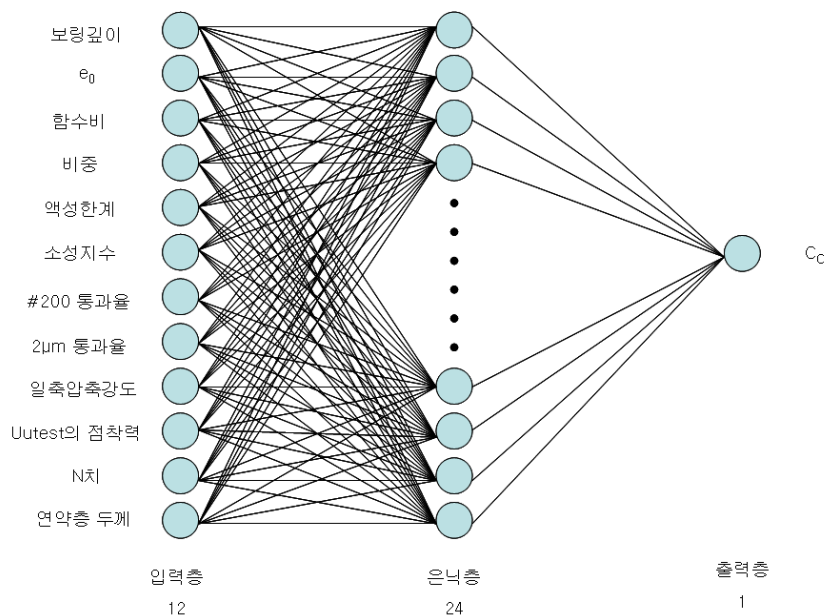


그림 4. LM 알고리즘의 최적신경망 구조

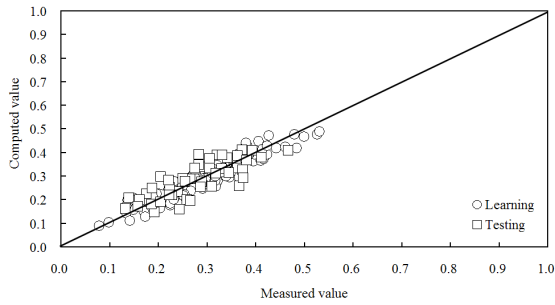


그림 5. 실제값과 예측값의 비교 - LM 알고리즘

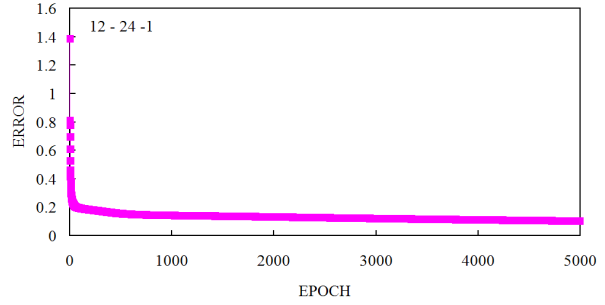


그림 6. 반복횟수에 따른 Error의 변화 - LM 알고리즘

표 5. 최적신경망의 연결강도와 바이어스 값 - LM 알고리즘

입력층 \ 은닉층	1	2	3	...	10	11	12	출력층
1	2.143	-0.837	0.617	...	0.769	0.518	1.998	-3.898
2	-0.792	2.948	-1.082	...	-0.258	-1.27	1.919	3.601
3	0.744	-0.68	-1.003	...	-0.767	2.894	-1.248	-2.651
.
.
.
22	0.721	-4.436	3.442	...	0.065	-4.322	-0.813	-4.48
23	-1.642	2.195	0.939	...	-1.068	-0.711	-2.415	-3.652
24	0.749	-0.961	0.375	...	-0.29	0.378	1.683	1.833

바이어스 값	1	2	3	...	22	23	24
은닉층	-1.086	0.816	0.248	...	-0.359	0.821	0.065
출력층	-0.79	-	-	...	-	-	-

5.3 개발된 인공지능망과 기존 경험식의 비교 검토

표 6에서는 위의 7가지 추정식과 LM 알고리즘을 상관계수와 RMSE로 비교하여 보았다. 단일 지반변수를 사용한 추정식은 Nishida (1956년)의 식이 학습과 예측자료에서 상관계수가 가장 높은 것으로 나타났지만, RMSE는 Terzaghi 와 Peck (1967년)의 식이 가장 좋은 값을 나타내는 것을 알 수 있다. 복합 지반변수를 사용한 추정식은 Azzouz 등 (1976년)의 식이 학습과 예측자료 모두에서 상관계수가 가장 높은 것으로 나타났다.

표 6. 각 방법별 학습과 예측의 상관계수와 RMSE

분류	방법	학습		예측	
		상관계수	RMSE	상관계수	RMSE
단일 지반변수	Terzaghi 와 Peck (1967년)	0.6643	0.0743	0.5795	0.0724
	Koppula (1981년)	0.3780	0.1103	0.2283	0.1112
	Nishida (1956년)	0.7275	0.1045	0.7984	0.0990
	Nacci 등 (1975년)	0.3255	0.0984	0.2591	0.0908
복합 지반변수	Rendon-Herrero (1983년)	0.6973	0.0826	0.7626	0.0719
	Azzouz 등 (1976년)	0.7691	0.0615	0.8253	0.0515
	Al-Khafaji 와 andersland (1992년)	0.7505	0.0599	0.6361	0.0717
인공 신경망	LM 알고리즘	0.9376	0.0311	0.8089	0.0485
	BP 알고리즘	0.9374	0.0311	0.8060	0.0481

RMSE는 Al-Khafaji 와 Anderland (1992등)의 식이 본 연구에서는 가장 좋은 것으로 나타났지만, 예측에서는 Azzouz 등 (1976년)의 제안 식이 가장 좋은 것으로 나타났다. LM 알고리즘과 비교했을 때는 추정식들보다 인공신경망의 결과가 더 좋음을 알 수 있다.

6. 결 론

국내 지반에 대한 상당한 양의 지반상수와 압축지수에 대하여 데이터 베이스를 수행하여 인공신경망을 이용하였다. 제안된 경험식과 기존의 예측식들과의 상호비교를 통하여 신뢰도를 검토하고 국내 지반에 대한 적합성을 검토한 결과, 다음과 같은 결론을 얻었다.

1. 압축지수 추정에 대한 최적신경망은 LM 알고리즘은 입력층 노드수 12, 은닉층 노드수 24, 출력층 노드수 1, λ 0.01이다. LM 알고리즘에서 예측의 상관계수는 0.8089, RMSE는 0.0485이다.
2. 단일 지반변수를 사용한 추정식은 Nishida (1956년)의 식이 학습과 예측자료에서 상관계수가 가장 높은 것으로 나타났지만, RMSE는 Terzaghi 와 Peck (1967년)의 식이 가장 좋은 값을 나타내는 것을 알 수 있다. 복합 지반변수를 사용한 추정식은 Azzouz 등 (1976년)의 식이 학습과 예측자료 모두에서 상관계수가 가장 높은 것으로 나타났다. RMSE는 Al-Khafaji 와 Anderland (1992등)의 식이 본 연구에서는 가장 좋은 것으로 나타났지만, 예측에서는 Azzouz 등 (1976년)의 제안 식이 가장 좋은 것으로 나타났다. LM 알고리즘과 비교했을 때는 추정식들보다 인공신경망의 결과가 더 좋음을 알 수 있다.
3. 제안된 경험식을 사용한다면 보다 합리적인 연약지반의 침하를 산정할 수 있어 경제적인 설계가 될 것으로 판단된다.

참고문헌

1. 송만순 (1988), “우리나라 해성점토의 토질계수의 상관관계”, 석사 학위논문, 한양대학교.
2. 윤길립, 김병탁 (2003), “국내 해양지반에 적합한 압축지수 예측식 제안”, 대한 토목학회지, **대한토목학회**, 제23권, 제3C호, pp. 169-176.
3. Al-Khafaji, A.W.N. and Andersland, O.B. (1992), "Equations for compression index approximation", *J. Geotech. Engrg.*, ASCE, 118(1), pp.148-153.
4. Azzouz, A.S., Krizek, R.J., and Corotis, R.B. (1976), "Regression analysis of soil compressibility", *Soil and Foundation*, JSSMFE, Tokyo, Japan, 16(2), pp.19-29.
5. Braja M. Das, (2007), *Principles of Geotechnical Engineering*, 6th Ed., Thomson Learning Asia Pte Ltd.
6. Cozzolino, V.M. (1961), "Statistical forecasting of compression index", *Proc., 5th Int. Conf. on Soil Mech. and Found. Engrg.*, Paris, France, Vol.1, pp.51-53.
7. Ghaboussi, J. and Sidarta, D. E., (1994), "Neural Network based Modeling in Geomechanics", *Computer Methods and Advanced in Geomechanics*, Sirlwardane & Zaman(ed.), Balkema, Rotterdam, pp.153-164.
8. Hecht-Nielsen, R. (1990), *Neurocomputing*, Addison-Wesly, Reading, Mass.
9. Herrero, O.R. (1983), "universal compression index equation; Closure", *J. Geotech. Engrg.*, ASCE, 109(5), pp.755-761.
10. Koppula, S.D. (1981), "Statistical estimation of compression index", *Geotech. Testing J.*, 4(2), pp.68-73.
11. Mayne, P.W. (1980), "Cam-clay predictions of undrained strength", *J. Geotech. Engrg. Div.*, ASCE, 106(11), pp.1219-1242.
12. Nishida, Y. (1956), "A brief note on compression index of soil", *J. Soil Mech. and Found. Engrg.*, ASCE, 82(3), pp.1-14.
13. Peumadu, D., Zhao R., Yao, C., and Ellis, G W, (1995), "Stress-strain modeling of sands using artificial neural network", *J. Geotech. Engrg.*, ASCE, 121(5), pp.429-435.
14. Rendon-Herrero, O. (1983), "Universal compression index", *J. Geotech. Engrg. Div.*, ASCE, 109(10), pp. 1349.
15. Rumelhart, D., Hinton, G., and Williams, R. (1986). "Learning-representations by back-propagating errors." *Nature.*, Vol. 323, pp.533-536.
16. Schmertmann, J.H.(1953), "Undisturbed Consolidation Behavior of Clay", *Transactions*, ASCE, Vol. 120, 1201.