IPSJ Transactions on Mathematical Modeling and Its Applications Vol. 4 No. 4 81–91 (Nov. 2011)

Regular Paper

Artificial Neural Network Based on Simulated Evolution and Its Application to Estimation of Landslide

Mona Abo-El Dahb,†1 Yao Zhou,†1
Umair Farooq Siddiqi†1 and Yoichi Shiraishi†1

The conventional steepest descent method in the back propagation process of an artificial neural network (ANN) is replaced by Simulated Evolution algorithm. This is called SimE-ANN and is applied to the estimation of landslide. In the experimental results, the errors of displacement and resistance of the piles in SimE-ANN are 50.2% and 28.0% smaller than those of the conventional ANN in average over 10 sets of data, respectively. However, the experimental results also show the effects of overtraining of SimE-ANN and the appropriate selection of training data should be investigated as future work.

1. Introduction

An artificial neural network (ANN) is considered one of the most widely used tools in learning processes in various applications [1]. The back propagation (BP) algorithm is a well known training algorithm in the neural network applications. In BP process, the error between the output of an ANN and the training data is propagated backwards minimizing the error by tuning the parameters of the ANN. The steepest descent method is used as a minimization method. This algorithm tends to be trapped in a local minimum. Its operation is based on the gradient method and requires continuous derivative of the goal function, so it has low convergence speed during the learning process. As the result, it can easily get local optimum, but it may not find global optimum and may cause the oscillation effect. Due to random choices of the network parameters, the initial values at every time are totally different which can affect the prediction ability after training. These problems have attracted the researchers to improve the BP performances. The stochastic algorithms are used to overcome the defect of the BP algorithm, for example, Genetic Algorithms (GAs) [2] are used [3)–5)]. Genetic algorithm is a global search algorithm which derives from imitating evolution of the creature. It has hill-climbing ability which helps the algorithm not to be trapped by the local minimum. However, these approaches have some drawbacks. One of the insufficient points of the GA approach is that the CPU time needed in the learning process is large because all the ANN parameters are optimized at the same time. Moreover, GAs usually take over the configuration of solutions as a scheme of chromosome and the change of value is only done by the mutation operator. Therefore, the GA parameters cannot be chosen in a quantifiable method [2]. Building from the above results, the application of GAs is seemed not appropriate in the optimization of parameter values. This paper proposes the use of another stochastic algorithm which is Simulated Evolution (SimE) algorithm [2] instead of the conventional steepest descent method to overcome its drawback. This approach is based on the analysis of the fundamental states of how to use SimE algorithm to improve the ability of BP. SimE algorithm is based on the analogy with the principle of natural selection thought to be followed by various species in their biological environments. It is a powerful general iterative heuristic for solving combinatorial optimization problems. SimE algorithm has no crossover operator and it only has the mutation operator. This operator is applied to some of the genes in a chromosome with inferior goodness values. Therefore, SimE algorithm is simple and is expected to be very fast. The ANN with the back propagation using SimE algorithm, called SimE-ANN, is applied to the estimation of landslide [8)–10)] in geotechnical field in order to evaluate its performances by comparing with the conventional steepest descent method. The use of ANN in this study is to predict both displacement and resistance of piles, which are used to mitigate landslide hazards in terms of ground movement and instability of slopes. Conventionally, the performances of piles are estimated by the finite element method (FEM) with a large amount of CPU time. However, recently, the ANN approach is becoming a good candidate in this field. A set of experimental data is 56 and is used for both training and testing stages. The 38 data are used for training and the remaining 18 data are used for performance evaluations. Then, the comparison between the proposed method and the conventional BP algorithm is done for evaluating the proposed algorithm. The results show that

†1 Department of Production Science and Technology, Gunma University

the proposed SimE-ANN improves errors of displacement and resistance of the piles 50.2% and 28.0%, respectively.

In Section 2, a general ANN and especially the back propagation process of training are described. Section 3 shows the previous work based on GAs and their insufficiencies are described. Next, the SimE algorithm is described in Section 4. Section 5 explains the customization of SimE algorithm for SimE-ANN. In Section 6, the experimental results obtained when SimE-ANN is applied to estimate the landslide of geotechnical application are shown and the efficiency of the proposed SimE-ANN is proven. Finally, Section 7 concludes our work and shows the suggestions for future work.

2. Artificial Neural Network

In recent years, due to the pervasive development in computer software, an ANN is being introduced in many applications. As aforementioned, an ANN is constructed from the interconnected processing elements, which are called neurons (i.e., nodes) as shown in **Fig. 1**. In fact, a neural network has great applications since it has a simple structure as well as multiple adjustable parameters.

The operation of ANN depends on the connection between the neurons and the training algorithm. One of the most popular training algorithms is the BP algorithm, which has been proposed by Rumelart in 1986 [1].

2.1 Back Propagation Process

Figure 2 shows the mechanism of the learning process. The signal flows of this algorithm can be divided into two directions: forward-propagation of information and counter-propagation of error. During the process of forward-propagation, the input information is transmitted into the output layer through the input layer and the hidden layer. Neurons in every layer can affect the neural condition in the next layer. If the expected output results cannot be obtained in the output layer, the error alteration should be worked out. Then the process must add counter-propagation. The error signal is counter-transformed into the neurons in every layer along the primary connection in order to get the expected goal. The main ideology of BP algorithm can be concluded briefly as follows: adjust both weights and error through counter-propagation. It makes the input data approach the expected data as quickly as possible. The training process can be over when the sum of the squared errors is lower than the specified value in the output layer. In the end, the weights and the error should be saved.

2.2 Steepest Descent Method

In the ANN learning process, an algorithm is needed to minimize the sum of the squared errors by tuning the ANN parameters. Generally, the steepest descent method is used up to now. The BP with the steepest descent method is described below using the N-layered ANN shown in **Fig. 3** and the neuron shown in **Fig. 4**. The output of neuron shown in Fig. 4 is expressed by Eqs. (1) and (2).

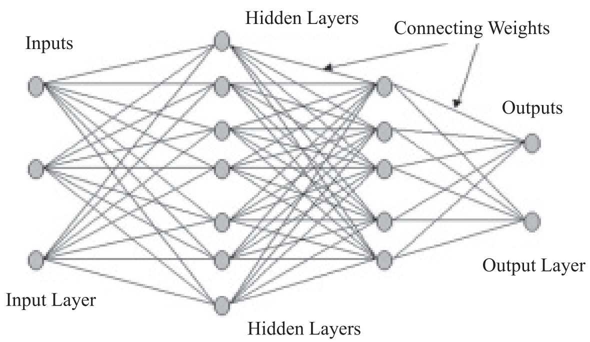


Fig. 1 ANN with two hidden layers.

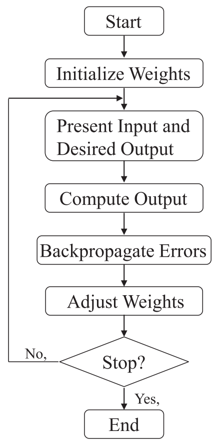


Fig. 2 Flow chart of ANN learning process.

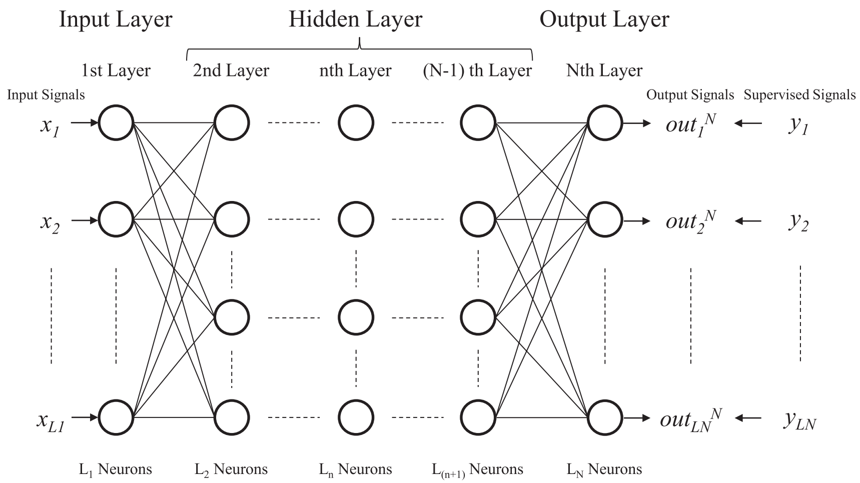


Fig. 3 N-layered ANN.

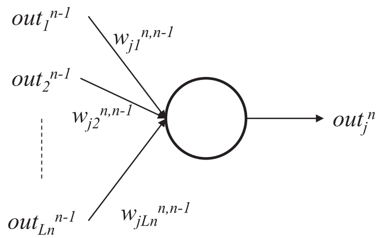


Fig. 4 Configuration of neuron.

$$net_j^n = \sum_{i=1}^{Ln} w_{ji}^{n,n-1} out_i^{n-1} \tag{1}$$

$$out_j^n = f(net_j^n) \tag{2}$$

where, net_j^n is the input value of j-th neuron in n-th layer and $w_{ji}^{n,n-1}$ is the weight assigned to the edge connecting i-th neuron in $(n-1)$-th layer and j-th neuron in n-th layer. The threshold function f is usually defined as a sigmoid activation function. The error E which shows the distance of the output of an ANN and the training data is defined by Eq. (3) as a mean squared error. This

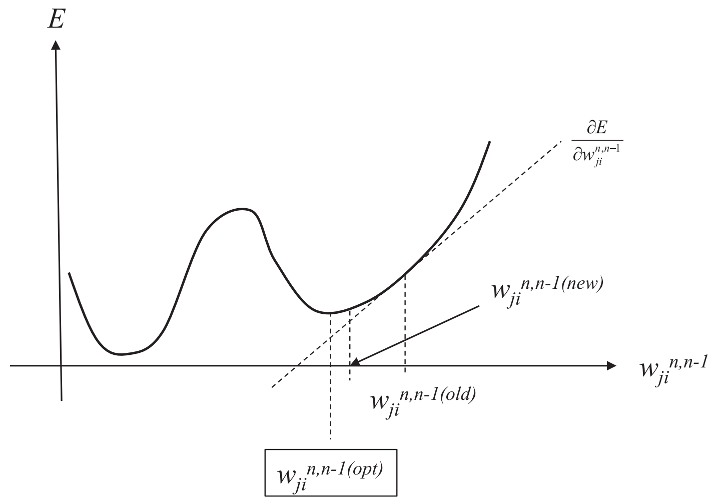


Fig. 5 Steepest descent method.

error E is a function of the output, out_i^N, which is determined by the weights $w_{ji}^{n,n-1}$, and it must be minimized during the learning process.

$$E = \frac{1}{2}\sum_{i=1}^{LN}(y_i - out_i^N)^2 \tag{3}$$

In minimizing the error E, the steepest descent method is applied. In this method, the weight $w_{ji}^{n,n-1}$ is modified according to Eq. (4).

$$w_{ji}^{n,n-1(new)} = w_{ji}^{n,n-1(old)} + \Delta w_{ji}^{n,n-1} \tag{4}$$

where $\Delta w_{ji}^{n,n-1}$ is the modification weight value.

By using **Fig. 5**, the gradient of tangent line of error function E is used to determine $\Delta w_{ji}^{n,n-1}$ as described in Eq. (5).

$$\Delta w_{ji}^{n,n-1} = -\eta \frac{\partial E}{\partial w_{ji}^{n,n-1}} \tag{5}$$

where η is the learning coefficient which controls the speed of convergence of the ANN.

Equation (5) is transformed to the next Eq. (6),

$$\Delta w_{ji}^{n,n-1} = -\eta \frac{\partial E}{\partial net_j^n} \frac{\partial net_j^n}{\partial w_{ji}^{n,n-1}} = -\eta \frac{\partial E}{\partial net_j^n} out_j^{n,n-1} = -\eta \cdot \delta_j^n \cdot out_i^{n,n-1}. \tag{6}$$

Here, the local gradient δ_j^n is defined by the next Eq. (7),

$$\delta_j^n = \frac{\partial E}{\partial net_j^n}. \tag{7}$$

The weight $w_{ji}^{n,n-1}$ can be determined by δ_j^n which is represented by the following Eqs. (8) and (9).

$$\delta_j^n = -(y_j - out_j^N) out_j^N (1 - out_j^N) \qquad (n = N) \tag{8}$$

$$\delta_j^n = \left\{ \sum_{k=1}^{L_{n+1}} \delta_k^{n+1} w_{kj}^{n+1,n} \right\} out_j^n (1 - out_j^n) \qquad (n < N) \tag{9}$$

Therefore, δ_j^n can be obtained by using the training data y_j $(j = 1, \ldots, L_N)$ when $n = N$ or by using δ_k^{n+1} when $n < N$ and this means such representation as "back propagation". If the steepest descent method is used for minimizing the error E, the global optimum set of weights can be obtained when the initial set of weights is near the optimum ones. However, in general, as shown in Fig. 5, when the initial set of weights is not in the appropriate location, the global optimum set of weights cannot be obtained.

3. Previous Work

In order to overcome the disadvantage of the steepest descent method as mentioned before, many researches [3)–5),11),12)] were done using stochastic algorithms such as Genetic Algorithm (GA) [3)–5)]. Montana and Davis constructed a chromosome as a sequence of connecting weights and thresholds of neurons when the network architecture of an ANN is given. Then, they applied the standard GA steps to minimize the error values [3)]. Rivero, et al. proposed the application of GAs to optimize the structure and the weights of an ANN at the same time [4)]. In their approach, a pair of genes represents the connectivity (0/1) and the weight of the connection, respectively. For example, a pair (1, 0.2) means that there is a connection with weight 0.2 between two specified neurons. Then, the standard steps of GAs optimize the structure and the weights of an ANN at the same time. Similarly, Feng, et al. tried to optimize the structure and the parameters of an ANN by applying GAs [5)]. They also represented the size parameters and the weights of an ANN in a chromosome and applied the standard type of GAs. The above mentioned previous work all compared the errors between the outputs of the ANN and the training data defining them as fitness values. However, the exact CPU times were not clearly stated although they evaluated the qualities of the obtained solutions. Only Feng, et al. stated that as a shortcoming of GA-BP, their method run a long time. Fundamentally, GAs need a large amount of time to get a near-optimal solution. Building on the previous work, many researches are needed to improve and overcome the challenges of the weight optimization and improving the performance of the network model. Therefore, this paper proposes the use of SimE algorithm instead of the steepest descent method to overcome the mentioned drawbacks.

4. Simulated Evolution Algorithm

In this section, SimE algorithm is described. SimE algorithm is invented by Kling and Banerjee in 1987 [6)]. It is a general search strategy for solving a variety of optimization problems. SimE algorithm is based on the analogy with the principle of natural selection thought to be followed by various species in their biological environments. While many chromosomes are treated in GAs, only one chromosome is changed to survive and bear the natural selection in SimE algorithm. It is a powerful general iterative heuristic for solving combinational optimization problems [6)]. Recently, SimE algorithm has proven to have a better performance than SA algorithm, though the specific problem is adopted for the performance evaluation as well as it is conceptually simple and elegant. In addition, it shows superiority in CPU time over GAs in some applications [7)]. SimE algorithm starts from the initial assignment and then, follows an evaluation based approach. It aims to reach the best assignments from one generation to the next. The general outline of SimE algorithm is described in **Figs. 6** and **7**.

In SimE, a solution is represented by a chromosome consisting of genes as in GAs. A set of chromosomes are handled as a population in GAs but only one chromosome is treated in SimE algorithm. Moreover, only a mutation operator is

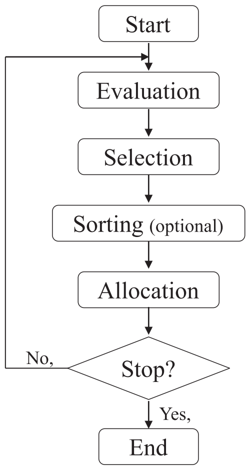


Fig. 6 Flow chart of SimE algorithm.

```
Algorithm Simulated_Evolution:
/* M: set of genes */;
/* B: selection weights */;
/* Stopping condition and weight of selection are automatically adjusted */;
Initialization Process;
Repeat
   /* Evaluation Process */;
     ForEach m ∈ M Do g_m = O_m/C_m EndForEach;

   /* Selection Process */;
     Ps=Pr=∅
     ForEach m ∈ M Do
         If Selection(m,B) Then Ps = Ps ∪ {m}
                Else Pr = Pr ∪ {m}
         EndIf;
     EndForEach;
     Sort(Ps);
   /* Allocation Process */
     ForEach m ∈ Ps Do Allocation(m) EndForEach;
Until Stopping Condition is Satisfied;
Return (Best Solution);
End Simulated_Evolution.
```

Fig. 7 SimE algorithm.

applied to a chromosome and the genes are perturbed over some generations. The execution of SimE algorithm consists of three main basic processes, Evaluation, Selection, and Allocation, in addition, Sorting which may be optionally applied in some applications. Initially, a chromosome is created randomly from all genes satisfying the problem constraints. Then, the three steps are executed in sequence until the chromosome's average goodness value satisfies a stopping condition [6]. In SimE algorithm, there are two conditions and the algorithm will terminate when one of them is met. The first condition is that the average of goodness values (defined later) reaches a maximum value or no noticeable improvement to the goodness is observed. The second condition is to run the algorithm for prefixed number of iterations. SimE algorithm is tailored to optimize the weights of ANN and each step of the algorithm is described in detail in Section 5.

5. SimE-ANN

This paper proposes the use of SimE algorithm instead of steepest descent method in the BP algorithm to optimize the ANN weights. As mentioned in Section 4, SimE algorithm consists of three main steps which are Evaluation, Selection, and Allocation.

5.1 Evaluation Process

The evaluation step consists of evaluating the goodness of each gene of the chromosome. In the ANN application, the chromosome is defined as a thread consisting of all the modified weight values $\Delta w_{ji}^{n,n-1}$ of ANN as described in Eq. (5). The values of genes must be optimized to minimize the error E. The goodness value is defined as follows.

$$g_m = \frac{O_m}{C_m} \tag{10}$$

where O_m is the optimum evaluation value of the weight and C_m is the current value of the weight. Usually, the optimum value O_m is not known and g_m should be calculated as a function of C_m without using O_m. Thereofre, in this problem, the goodness $g_{ji}^{n,n-1}$ corresponding to $\Delta w_{ji}^{n,n-1}$ ($i\ j = 1, 2, \ldots, LN$, $n = 1, 2, \ldots, N$) is defined as the negative normalized value by the maximum value among all absolute values of $\Delta w_{ji}^{n,n-1}$'s. The goodness values should be used to determine which genes must be selected (that is, natural selection) and selected

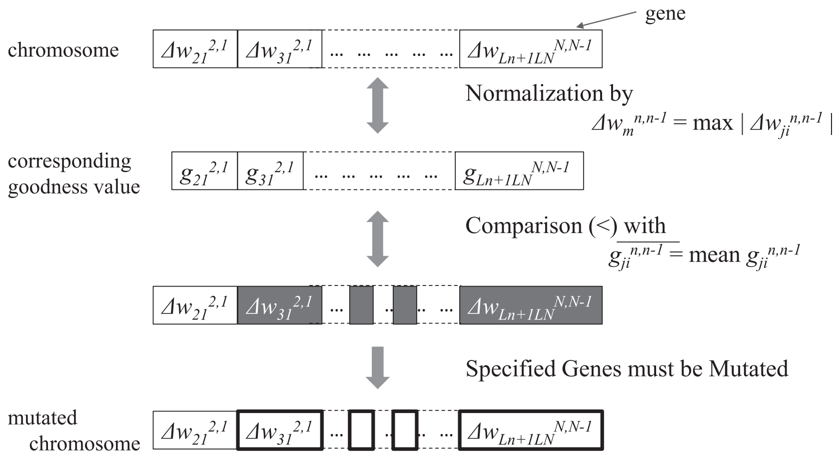


Fig. 8 Outline of SimE processes.

genes are mutated in the subsequent processes as described in **Fig. 8**.

5.2 Selection Process

The inputs of the process are the chromosome and the estimated goodness of each gene. The portions of the chromosome, the selection set P_s and the remaining set P_r, is generated based on the goodness value $g_{ji}^{n,n-1}$ and the selection bias B. The value of B is used for fine-tuning the selection of genes. Its value is recommended to be in $[-0.2, 0.2]$ and in many cases, "B $= 0$" would be a reasonable choice. The fundamental idea of the selection process is to select such genes with comparatively small goodness values (this means the current value of gene is no good). Therefore, the genes whose goodness values are smaller than the mean value of all goodness values are selected. For example, the selected genes are hatched in Fig. 8. Here, such genes as much more sensitive to the change of error E are selected.

5.3 Allocation Process

The allocation process is very important to optimize the solution and it has a great impact on the solution quality. Here, the mutation operator is applied to the selected genes in P_s. After the mutations, the new chromosome consisting of P_s and the remaining genes P_r is generated as shown in Fig. 8. As in Eq. (6), the value of gene $\Delta w_{ji}^{n,n-1}$ can be changed by δ_j^n. Therefore, the value of δ_j^n of the selected gene should be controlled. This value can be freely changed as a real number and a range limiter for the new values of δ_j^n's is created in order to avoid the divergence. This is realized by that the new value of δ_j^n must be randomly changed as a real number in $[\delta_j^n - r\sigma, \delta_j^n + r\sigma]$, where σ is the standard deviation of all δ_j^n's and r ($r = 1$, 2 or 3) is a multiplier for σ. Here, we assume that δ_j^n's ($i = 1, 2, \ldots, LN,\ n = 1, 2, \ldots, N$) would obey the normal distribution. The idea is that if the standard deviation is large, the new value of δ_j^n should be largely changed and the value r should be determined experimentally. Here, the mutation operator determines the new value of δ_j^n stochastically. Then, by using the new values of selected δ_j^n's, the new chromosome can be obtained by using Eq. (6). To test the validity of SimE algorithm in the ANN weight optimization, an actual problem, the estimation of the landslide is used.

6. Application of SimE-ANN to Estimation of Landslide

This section describes the application of the proposed SimE-ANN to the estimation problem of a landslide [8)–10)] to evaluate its performances. A landslide is one of the most dangerous natural hazards which result many problems for human and damages for earthwork and structure projects. The induced seismic force during landslide results in significant damage to the ground profile, especially in mountain areas due to the sharp slopes in these areas. There are many ground movements and instability for slopes in these areas occurred during landslide shocks. The use of piles to mitigate landslide hazards in mountain areas is considered one of the common methods for landslide mitigations. In fact, there are many parameters affecting the design of such mitigation system as the intensity of seismic waves, type of soil, type of materials used in piles, the degree of slopes and the predicted pile displacement as well as the resistance of piles during a landslide [8)–10)]. Both displacement and resistance of piles are considered to be the main factors which are responsible for landslide by reducing ground movement and failure. Therefore, many numerical and empirical methods were developed to determine both two factors to obtain the safe design for piles [8)–10)]. It is a well-known numerical analysis method using 3-D FEM (3-dimensional Fi-

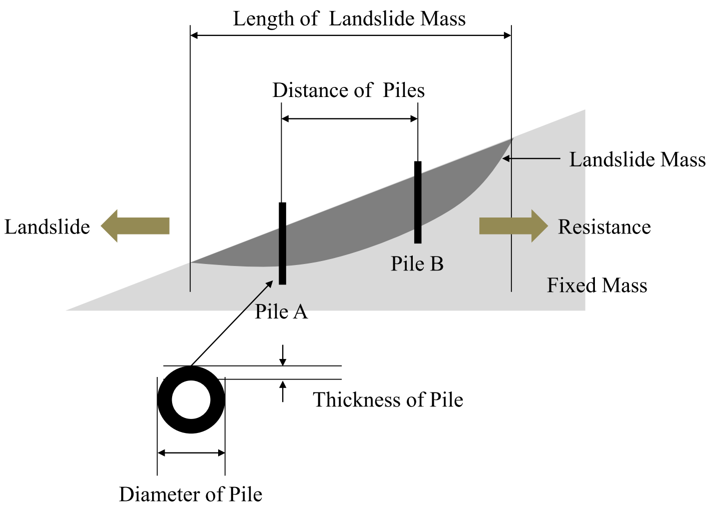


Fig. 9 Model of landslide.

nite Element Method), which is very complicated and it takes more time and high cost. On the other hand, results obtained from empirical methods cannot provide more accurate and reliable prediction for ground displacements. Subsequently, most researches have nowadays resorted to the use of computer software and computer aided tools to develop a simple technique for solving such complicated computations in the area of geotechnical engineering [8]. Wakai, et al. obtained a good result by using ANN; however, this ANN uses the conventional back propagation algorithm. In this study, the actual data obtained by 3-D FEM [8] are used for evaluating the performances of proposed SimE-ANN compared with the conventional ANN.

6.1 Problem Definition

The model of a landslide [8] is shown in **Fig. 9**. The input and the output data of the landslide estimation problem are shown in **Table 1**. The landslide estimation problem is to generate a pair of data, that is, the displacement of pile heads and the maximum resistance of piles. The dimensional parameters in Table 1 are shown in Fig. 9.

Table 1 Input and output data of landslide.

Input Data			Output Data	
E1	Landslide Mass	Young's Modulus	S	Displacement of Pile Heads
Φ		Internal Friction Angle	R	Maximum Resistance of Piles
c		Adhesion		
l		Length		
E2	Fixed Mass	Young's Modulus		
B	Piles	Distance		
T		Thickness		
D		Diameter		

6.2 Application of Conventional ANN and SimE-ANN

To solve the landslide estimation problem, the proposed SimE-ANN is applied and its performances are evaluated against the conventional ANN. First of all, the network structure and the parameters of an ANN are determined based on the results of Wakai, et al. [8]. The numbers of neurons in the input and the output layers are 8 and 2, and one hidden layer is constructed consisting of 10 neurons. The same topology of network is used for the evaluation of SimE-ANN for the exact comparison of SimE and the steepest descent method. The conventional ANN and SimE-ANN were implemented by C++ language. The number of data generated by FEM is 58 - 38 of them are used as training data for training ANN's and the remaining 18 are used for testing the ANN's. The stopping condition of the learning process is either of them, the number of trainings reaches 1,000 or the mean squared error is less than 0.0001. In the steepest descent method, the learning coefficient is set as $\eta = 0.1$ through several experiments. Here, in the SimE algorithm, the length of chromosome is 100 and the number of generations is set to 100 by repeating some experiments. Moreover, the learning coefficient η is 0.001. The threshold values of each neuron in both of the conventional ANN and SimE-ANN are set to the random numbers in $[-1, 1]$. The value r in the allocation process (described in Section 5.3) is set to 1 by doing some experiments. In both ANNs, the initial values of weights are randomly determined. In the learning process of the ANN, the learning algorithm is executed in sequence until it satisfies the stopping conditions.

Table 2 Average errors of 1,000 times of trainings over four cases.

ANN	Mean	Variance
Conventional ANN	1.74999000	0.22668060
SimE-ANN	0.00307171	0.00000678
Reduction (%)	99.825	99.998

6.3 Performance Comparison of Steepest Descent and SimE Algorithms

The objective of the steepest descent and SimE algorithms is the error minimization during the training process, where the error is the sum of the root of mean squared errors (RMSE) of displacement and the RMSE of resistance. These errors are related to the stopping condition of the training process. The training processes of the conventional ANN (with the steepest descent algorithm) and SimE-ANN are repeated 1,000 times over 38 training data.

As shown in **Table 2**, the errors of the training processes of SimE-ANN are reduced 99.8% and 99.9%, respectively and this shows the superiority of the SimE algorithm to the steepest descent algorithm in the error minimization. This fact may lead to the speedup of the learning of SimE-ANN.

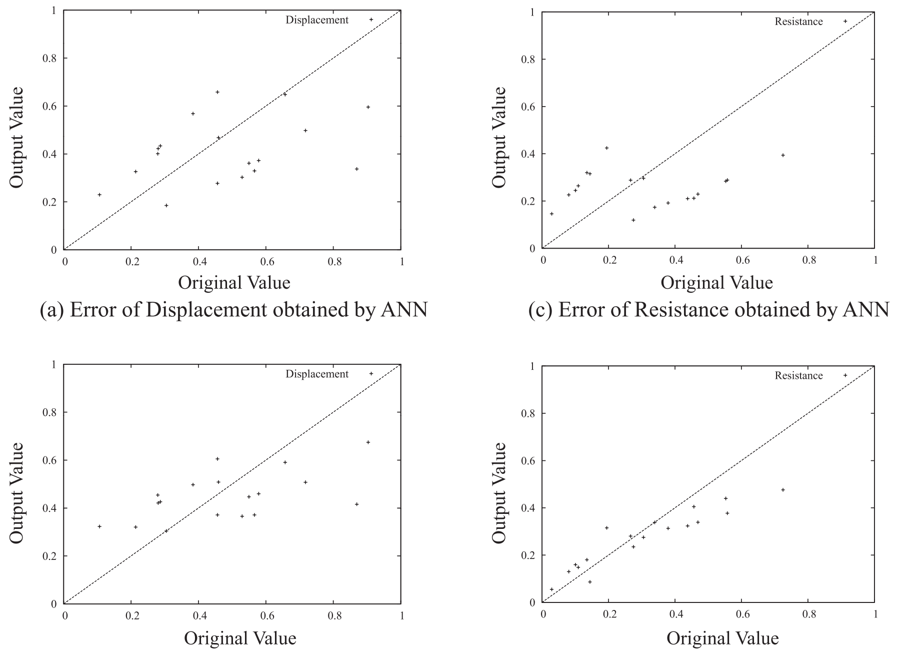
6.4 Learning Abilities of Conventional ANN and SimE-ANN

In order to evaluate the learning abilities of the conventional ANN and SimE-ANN, 10 combinations of 38 training data and 18 test data are randomly generated and used. Based on the stopping condition stated in Section 6.2, the conventional ANN and SimE-ANN are given trainings. The comparisons of these ANN's for 10 sets are shown in **Table 3**.

As is clear from Table 3, the errors of displacement and resistance of SimE-ANN is 50.8% and 28.0% of the conventional ANN, respectively. The number of trainings is 37% reduced. However, not all of the outputs of SimE-ANN are better than those of the conventional ANN, such as in the cases of displacements of No.1 and No.8. This may be because SimE-ANN has an effect of overtraining and as a result, the performances of outputs for the set of test data depend on the choice of the training data. That is, if the choice of the training data is adequate, the parameters can be tuned to obtain the good results for the test data. The comparisons of errors for No.10 case are graphically shown in **Fig. 10**.

Table 3 Performances of ANN's.

	SimE-ANN			Conventional ANN		
No.	Displacement	Resistance	# of Trainings	Displacement	Resistance	# of Trainings
1	0.119378113	0.058263025	1000	0.077502278	0.062431909	1000
2	0.011996932	0.014316911	231	0.115124478	0.06218942	1000
3	0.071388331	0.034047879	1000	0.109280469	0.065601986	1000
4	0.06757277	0.018134157	830	0.077444002	0.118705515	1000
5	0.01883213	0.00724077	1000	0.109155979	0.081434685	1000
6	0.019155201	0.015697981	397	0.081138558	0.087446479	1000
7	0.060557039	0.009754596	32	0.084746761	0.057921378	1000
8	0.077622755	0.002238956	1000	0.05484456	0.054658022	1000
9	0.020467046	0.023617703	663	0.104664079	0.067877571	1000
10	0.007347548	0.021880045	103	0.119378672	0.074405727	1000
Mean	0.047431787	0.020519202	625.6	0.093327984	0.073267269	1000
Ratio (%)	50.82268433	28.00596039	62.56	100	100	100



(a) Error of Displacement obtained by ANN
(c) Error of Resistance obtained by ANN
(b) Error of Displacement obtained by SimE-ANN
(d) Error of Resistance obtained by SimE-ANN

Fig. 10 Comparison of errors for test data.

Table 4 Errors of ANN's for same data set.

	SimE-ANN			Conventional ANN		
No.	Displacement	Resistance	# of Trainings	Displacement	Resistance	# of Trainings
1	0.194952175	0.14760843	17	0.189103496	0.191121486	1000
2	0.211090888	0.223425453	1000	0.154037083	0.106022557	1000
3	0.210265773	0.176608139	12	0.156308218	0.106063184	1000
4	0.211574083	0.11316096	22	0.155676697	0.106704357	1000
5	0.249162043	0.155182915	41	0.157104677	0.108313727	1000
6	0.129887039	0.123510407	1000	0.154222252	0.106540265	1000
7	0.210305209	0.1579564	13	0.154975771	0.108587656	1000
8	0.176428235	0.259514059	1000	0.156238081	0.107514946	1000
9	0.162734691	0.20136116	1000	0.153466205	0.107765323	1000
10	0.196887066	0.270937208	1000	0.155567379	0.107972424	1000
Mean	0.19532872	0.182926513	510.5	0.158669986	0.115660592	1000
Variance	0.000956258	0.002666954	-	0.0001040699	0.0006334541	-

6.5 Stochastic Nature of SimE-ANN as Drawback

SimE algorithm has a stochastic nature. This means that the obtained results are stochastically variable. This can be a drawback of these kinds of algorithms. To check the variance of the obtained results, an experiment to apply both of the ANN algorithms to the same set of training data and test data 10 times is executed. The results are shown in **Table 4**. As shown in this table, the variances of errors for test data in SimE-ANN are 9.3 times and 4.2 times larger than those in the conventional ANN. However, these variances of errors are 0.45% and 1.45% of the respective mean value and they are not likely to give a serious problem. However, SimE-ANN must be applied to the same set of training data and test data several times and the variances of errors should be measured before the actual application.

6.6 Estimated Comparison with Improved BP's

The original BP with the steepest descent method [1] is used for comparison with SimE-ANN in the experiment due to the need of comparison with the conventional research in the landslide estimation [8)–10)]. However, many improvements have already been made with the original BP [11),12)]. The detailed comparison of SimE-ANN with such improved BPs must be done and this is future work in this research. For the BP with the introduction of entropy term [12)], its convergence ratio is evaluated with that of the original BP, but the actual processing times are not compared. In the L-BP (BP using the Lyapunov coefficient) [11)], the acceleration of the learning process is compared. The examples used in Ref. 11) are an Exor problem and a simple pattern recognition problem. In fact, L-BP reduced the number of trainings 96% in Exor problem and 98% in a simple pattern recognition problem, respectively. However, the convergence of learning in L-BP is not stable and the authors conclude that the stability of L-BP is not thought better than that of original BP. The sizes and the characteristics of these examples are different from those of landslide estimation problem and so, the exact comparison is inadequate. However, the convergence of SimE-ANN is guaranteed by the asymptotic optimality of SimE algorithm [2)]. Morevoer, as shown in Table 3, SimE-ANN reduced the number of trainings 37.4%. Therefore, it can be concluded that the speedup of SimE-ANN is 1/2.6 of L-BP but it is superior to L-BP in the convergence of learning.

7. Conclusions

A new artificial neural network based on a stochastic algorithm, called SimE-ANN, is proposed. The implemented stochastic algorithm is called Simulated Evolution and this algorithm replaces the conventional steepest descent method in minimizing the error between the output of ANN and the training data. The performances of SimE-ANN are evaluated against the conventional ANN for the landslide estimation problem. After trainings for 38 data from 56 data, SimE-ANN reduced the error of displacement of piles and the error of resistance of piles, 50.8% and 28.0%, respectively in average, for the 18 test data. However, in some cases, the errors of the test data in SimE-ANN are no better than those of the conventional ANN because of the estimation that SimE-ANN has a strong effect of overtraining. Therefore, the appropriate selection of the set of training data for SimE-ANN should be investigated as future work. Moreover, SimE-ANN should be applied to various kinds of learning problems.

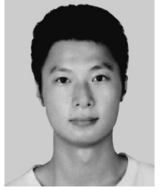
Acknowledgments The authors firstly would like to appreciate to the reviewers for suggesting the application of the proposed SimE-ANN to variable set of training data in order to investigate its performances. Secondly, the authors would also thank Prof. Akihiko Wakai for giving us the landslide data for evaluating the proposed SimE-ANN.

References

1) Anderson, J.A.: *An Introduction to neural network*, Cambridge, Mass., MIT Press (1995).
2) Sait S.M. and Youssef, H.: *Iterative Computer Algorithms with Applications in Engineering*, IEEE Computer Society Press (1999).
3) Montana, D. and Davis, L.: Training feedforward neural networks using genetic algorithms, *Proc. 11th International Joint Conference on Artificial Intelligence*, pp.762–767, Morgan Kaufmann (1989).
4) Rivero, D., Dorado, J., Fernández-Blanco, E. and Pazos, A.: A Genetic Algorithm for ANN Design, Training and Simplification, *IWANN 2009, Part I, LNCS*, Vol.5517, Cabestany, J., et al. (Eds.), pp.391–398 (2009).
5) Feng, W., Zhu, W. and Zhou, Y.: The Application of Genetic Algorithm and Neural Network in Construction Cost Estimate, *Proc. Third International Symposium on Electronic Commerce and Security Workshops* (*ISECS '10*), Guangzhou, P.R. China, 29–31, pp.151–155 (July 2010).
6) Kling, H.M. and Banerjee, P.: Concurrent ESP: A placement algorithm for execution on distributed processors, *Proc. IEEE International Conference on Computer-Aided Design*, pp.354–357 (1987).
7) Siddiqi, U.F., Shiraishi, Y., El-Dahb, M.A. and Sait, S.M.: Embedded Systems Synthesis Using Simulated Evolution (SimE) with ECU-Specific Optimization, *Proc. International Conference on Computer Mathematics and Natural Computing, ICCMNC 2011*, Penang, Malaysia (Feb. 2011).
8) Wakai, A., Ugai, K., Cai, F. and Tanaka, K.: Simple computer-aided design system for rational evaluation of maximum resistance of landslide prevention piles, *Journal of the Japan Landslide Society*, Vol.42, No.5, pp.18–28 (2006).
9) Wakai, A., Gose, S. and Ugai, K.: 3-D elasto-plastic finite element analyses of pile foundations subjected to lateral landing, *Soils and Foundations*, Vol.39, No.1, pp.97–111 (1999).
10) Wakai, A., Ugai, K. and Cai, F.: Finite element analysis of slopes reinforced with steel piles, *Proc. International Conferences on Slope Engineering*, Vol.2, pp.822–827 (2003).
11) Hashimoto, M., Inoue, H. and Narihisa, K.: Stability and Acceleration in the Back Propagation Learning Algorithm, *The Bulletin of the Okayama University of Science*, Vol.35A, pp.143–152 (1999) (in Japanese).
12) Akiyama, Y.: Improvement of the Back-propagation Learning Rule by Introducing an Entropy Term, *IPSJ Trans. Mathematical Modeling and its Applications*, Vol.40, No.SIG9(TOM2), pp.81–90 (1999) (in Japanese).

(Received April 18, 2011)
(Revised June 6, 2011)
(Revised(2) July 4, 2011)
(Accepted July 11, 2011)

Mona Abo El Dahb received her B.E. and M.S. degrees from Electrical and Electronic Department, Faculty of Engineering, El-Minia University, El Minia, Egypt. In 2011, she received Ph.D. degree from Gunma University. From 1998 to 2002, she was a researcher assistant, at El-Minia University, Egypt. She was an assistant lecturer at Helwan University, Egypt, from 2002 to 2007. She is currently a researcher in Department of Production Science and Technology, Gunma University. Her research includes embedded system design and neural network.



Yao Zhou received his B.E. degree from Beijing University of Technology in China in 2008 and M.E. degree from Gunma University in 2011. He is now with USTC E-Business Technology Co., Ltd. in China. He was engaged in the research on an automatic routing algorithm and the application of stochastic algorithms.

Umair Farooq Siddiqi received his B.E. degree from NED University of Engineering and Technology, Karachi, Pakistan in 2002 and M.S. degree from King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia in 2007. From 2007 to 2010, he was a research assistant in Department of Computer Engineering in KFUPM. Currently, he is a Ph.D. course student at Gunma University. His research includes the application of stochastic algorithms, hardware design and combinatorial optimization algorithms.

Yoichi Shiraishi received his B.S., M.S. and Ph.D. degrees in 1979, 1981 and 1993, respectively, all from Tokyo Institute of Technology. He has been engaged in the research and development of ECAD/EDA algorithms at Hitachi's Central ResearchLaboratory from 1981 to 1993. Now, he is with the Department of Production Science and Technology, Gunma University. His research includes combinatorial optimization algorithms, stochastic algorithms, embedded system design, signal processing, chaos theory and hardware design. He is a member of IEICE, JIEP, JFS, JSIAM, SICE, IEEE and ACM.