

## GENOUD-BP: A novel training algorithm for artificial neural networks

Zhiyong YANG<sup>1, 2, a</sup>, Shiyuan ZHANG<sup>3, b</sup>, Taohong ZHANG<sup>1, 2, c, \*</sup>

<sup>1</sup>School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

<sup>2</sup>Beijing Key Laboratory of Knowledge Engineering for Materials Science

<sup>3</sup>School of Automation & Electrical Engineering, Beijing University of Posts & Telecommunications

<sup>a</sup>email:joshuaale@163.com, <sup>b</sup>email:angelazhang0526@163.com

<sup>c</sup>email:waterswordzth@163.com

\*Corresponding author

**Keywords:** neural networks, optimization, evolutionary algorithms, GENOUD, BP

**Abstract.** BP Algorithm, as a well-known training method for artificial neural network, has been widely used in all the main fields of science and engineering. However, owing to the overwhelming dependency on gradient of loss function, BP-ANN still suffers several drawbacks, for example, the training process is prone to stuck at local optima, cause early convergence while the whole process is lack of generalization performance. Aimed at improving the current BP training Algorithm, a new algorithm called GENOUD-BP is proposed in this paper by introducing the GENOUD algorithm which combines the global searching power of genetic algorithms and convergence speed of traditional gradient based optimization algorithms. Two UCI datasets are employed to carry out benchmark experiments, the result of which shows that the GENOUD-BP significantly outperforms traditional BP algorithms.

### Introduction

Back propagation (BP) Algorithm, as a well-known training method for artificial neural network, has been widely used in all the main fields of science and engineering [1]. However, considering its gradient based nature and its non-convex loss function, the traditional ANN based on BP algorithm may be prone to stuck at local minima and may suffer from early convergence. Aimed at better approximating the global solution, related works[2] has done to apply well-known evolutionary algorithms(EAs) such as Genetic Algorithm (GA) to optimize the weights in ANN, which is referred as evolutionary neural networks. Owing to the stochastic nature of EAs, such EA based ANNs still suffers from slow convergence speed at the neighbor of optimal solutions. In this paper, a novel training algorithm called GENOUD-BP, where the GENOUD algorithm combining the global searching power of genetic algorithms and convergence speed of traditional BP algorithm, is integrated into BP framework, is proposed in order to balance the performance and efficiency of BP algorithm.

### GENOUD Algorithm for optimization

GENOUD algorithm proposed by Sekhon et al [3] is a powerful optimization algorithm combines EA with BFGS, recently a R package called rgenoud [4] has also published in CRAN for analyzers to utilize GENOUD on R. An EA uses a collection of heuristic rules to modify a population of individual solutions in such a way that the best solution of each generation tends to be, on average, better than that of the previous generation. The EA in GENOUD is fundamentally a genetic algorithm (GA), where, in general, though the implementation may vary, the basic set of operators can be defined as reproduction, mutation, crossover and inversion. The Genetic algorithm in GENOUD uses nine operators that are listed in Table 1, adapted from [4]. The operators extend and modify a set of operators used in GENOCOP [4]. Among those operators in Table 1, the

Local-minimum crossover plays an important role in that it computes a convex combination of the best solution found so far by GA and the solution of BFGS, an efficient and classical quasi-Newton optimization Algorithm based on gradient, which could enhance the converge speed of GA in the neighbor of optimum.

Table.1. nine operators used by GENOUD

---

P1: Cloning. Copy  $X_i$  into the next generation  $X_{t+1}$

P2. **Uniform Mutation.** At random choose  $i \in N$ . Select a value  $x' \sim U(x_{i,min}, x_{i,max})$  Set  $X_i = x'$

P3. **Boundary Mutation.** At random choose  $i \in N$ . Set either  $X_i = x_{i,min}$  or  $X_i = x_{i,max}$  with probability 1/2 of using each value.

P4. **Non-uniform Mutation.** At random choose  $i \in N$ . Compute  $p = (1 - \frac{t}{T})^B u$ , where t is the current generation number, T is the maximum number of generations,  $B > 0$  is a tuning parameter and  $u \sim U(0,1)$  Set either  $X_i = (1-p)x_i + px_{i,min}$  or  $X_i = (1-p)x_i + px_{i,max}$ , with probability 1/2 of using each value.

P5 **Polytope Crossover.** Using  $m = \max(2, n)$  vectors x from the current population and m random numbers  $p_j \in (0,1)$  such that  $\sum_{j=1}^m p_j = 1$ . set  $X_i = \sum_{j=1}^m p_j x_j$

P6 **Simple Crossover.** Choose a single integer i from N. Using two parameter vectors x and y, set  $X_i = (1-p)y_i + px_i$  and  $X_i = (1-p)x_i + py_i$ , where  $p \in (0,1)$  is a fixed number.

P7 **Whole Non-uniform Mutation.** Do non-uniform mutation for all the elements of X.

P8 **Heuristic Crossover.** Choose  $p \sim U(0,1)$ . Using two parameter vectors, x and compute  $z = p(x-y) + x$ . If z satisfies all constraints, use it. Otherwise choose another p value and repeat. Set z equal to the better of x and y if a satisfactory mixed z is not found by a preset number of attempts. In this fashion produce two z vectors.

P9 **Local-minimum Crossover.** Choose  $p \sim U(0,1)$ . Starting with x, run BFGS optimization up to a preset number of iterations to produce  $x'$ . Compute  $z = px + (1-p)x'$  If z satisfies boundary constraints, use it. Otherwise shrink p by setting  $p = p/2$  and recompute z. If a satisfactory z is not found by a preset number of attempts, return x.

**Notation:**  $X = [X_1 \cdots X_n]$  is the vector of n parameters  $X_i$   $x_{i,min}$  is the lower bound and  $x_{i,max}$  is the upper bound on values for  $X_i$ .  $x_i$  is the current value of  $X_i$ , and x is the current value of X.  $N = \{1, 2, \dots, n, \dots\}$ .  $p \sim U(0,1)$  means that p is drawn from the uniform distribution on the [0,1] interval.

---

### GENOUD-BP: Embedding GENOUD in BP algorithm

The GENOUD-BP algorithm is proposed according to the following ideas. The training process of artificial neural networks could be regarded as the minimization of the overall loss function J with respect to the weights and biases. GENOUD Could be embedded into the training of ANN, by Setting  $X_i$  be all the weights and bias for a specific ANN model, and the fitness of a ANN (i.e.an individual in the population in GENOUD) be the loss function by (1) then using GENOUD algorithm and the best individual for the last iteration is the optimized parametric solution for the current ANN.

Table.2. definitions in GENOUD-BP

$J^{(i)}$	The loss function for the ith record
$x^{(i)}$	The general input of the ith record, with the first element being 1
$y^{(i)}$	The output of the ith record
$h(x^{(i)})$	The estimated output of the ith record
$W^{(L)}$	The weight and biases matrix between the (L -1)th layer and the Lth layer of ANN, with the bias being the first column of W
$J$	the overall loss function for a sample with size n
$f(\cdot)$	$\mathbb{R}^m \rightarrow \mathbb{R}^n$ The fitness function of a individual
$vec(\cdot)$	$\mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{mn \times 1}$ The vectorization operator for a matrix by rows

We also developed a R package for GENOUD-BP called GenoudBP based on the r package rgenoud, which has been published on GitHub [5]. All mathematical definitions used in GENOUD-BP are listed in Table 2. The Algorithm in detail is described in Algorithm 1. Note that both the weights and biases are merged to  $\vec{W}^{(L)}$ s

$$f(X_i) = f(\vec{W}) = J = \sum_{i=1}^n J^{(i)} + \lambda \|\vec{W}\|_2^2 \quad (1)$$

Where  $J^{(i)}$  could either be a square error for a regression problem (2) or cross-entropy for

classification (3).  $\vec{W} = vec \begin{pmatrix} W^{(1)} \\ \vdots \\ W^{(N)} \end{pmatrix}$ , in which all the Weight matrices are transferred into a vector for

computational convenience.

$$J^{(i)} = (y^{(i)} - h(x^{(i)}))^2 \quad (2)$$

$$J^{(i)} = y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)})) \quad (3)$$

## Numerical Experiments and Results

To test the performance of the proposed Algorithm, two UCI datasets for classification are employed to compare the performance of GENOUD-BP, GENOUD-BP-NBFGS (the same as GENOUD-BP except that BFGS is never used), BP-SCG (BP optimization with Scaled Conjugate Gradient), Quickprop (BP optimization with Quickprop[6]), with review of datasets and the numerical results presented as follows

### Algorithm.1. GENOUD-BP

**Input** : population size maximum iteration number maxiter

#### Initialization

for = i in 1:n

randomly initialize  $\vec{W}$

$X_i := \vec{W}$

end for

$best_0 = \min_i f(X_i)$  // get the best initial solution

#### Optimization

Sol := final solution get from GENOUD

#### Build the optimized ANN

Transform Sol to the matrix form of Ws to rebuild the optimized ANN

Two datasets are employed in this paper

a. The Pima Indians Diabetes Data Set (denoted pima in this paper):[7]

This is dataset for forecasting diabetes mellitus, with 768 records and 8 attributes covering the Demographic characteristics, physiologic information for adult females  
b.Blood Transfusion Service Center Data Set (denoted transfusion in this paper):[8]

To demonstrate the RFMTC marketing model (a modified version of RFM), this dataset adopted the donor database of Blood Transfusion Service Center in Hsin-Chu City in Taiwan for demonstrating RFMTC (a modified version of RFM),with 5 attributes and 748 instances

The parameter settings for all four algorithms are summarized as follows: For data separation, the ratio of the size of training set and that of the test set is 0.85/0.15 .The number of hidden layer nodes for The Pima Indians Diabetes Data Set is set to 10 and for transfusion is set to 8, other parameters are listed in Table 3 where maxiter denotes the maximum number of iteration

Table.3.parameter settings

BP-SCG(denoted SCG hence after)	maxiter =1000
Quickprop	the same as the former
GENOUD-BP(denoted GBP hence after)	maxiter=100,populationsize=10, $\lambda = 0.2$ , execute BFGS for P9 since the last half of the iterations
GENOUD-BP-NBFGS(denoted GBPN hence after):	maxiter=100,population size=10, $\lambda = 0.2$ never executing BFGS for P9

The performance of each of the four aforementioned algorithms on the two datasets are evaluated 20 times respectively and the performance of the test sets are summarized in Table 4. To better measure both the sensitivity and specificity of the classification results, the well-known receiver operating characteristic curve (ROC) is introduced and the AUC [9] for ROC is employed as the metric for algorithm performance. As listed in Table 4, it is concluded that for both datasets, the performance of GBP and GBPN are obviously better than the other two algorithms due to the introduction of Genetic Algorithm and the standard deviations(std) for the former group are smaller than the that of latter group which also shows a better robustness, furthermore ,with the fact that the mean performance of GBP is slightly better than that of GBPN,it is concluded that the utilization of BFGS could help GENOUD-BP algorithm to converge faster to the global optima.

Table.4. numerical results for all the algorithms

datasets	Item	SCG	Quickprop	GBPN	GBP
pima	min	0.651866	0.6857848	0.7365215	<b>0.7526652</b>
	max	0.855145	0.8500882	0.8733686	<b>0.9051146</b>
	mean	0.755481	0.7741496	0.8166844	<b>0.8228659</b>
	std	0.052068	0.0407715	<b>0.0366407</b>	0.0367042
transfusion	min	0.609195	0.61229	0.652962	<b>0.6777188</b>
	max	0.832041	0.824053	<b>0.8391473</b>	0.833118
	mean	0.716822	0.7584623	0.7754181	<b>0.7882547</b>
	std	0.05455	0.0508087	0.0478426	<b>0.0448634</b>

## Conclusion

In this paper we have proposed a novel algorithm GENOUD-BP that combines the Genetic Algorithm with gradient-based optimization algorithms. Based on data from two real world classification problems, it has been shown that the proposed GENOUD-BP algorithm obtains better prediction precision and robustness than other three aforementioned algorithms: SCG, Quickprop and GENOUD-BP-NBFGS.

## Acknowledgement

This paper is sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry and supported by National Key Technology R&D Program in 12th Five-year Plan of China (No. 2013BAI13B06).

## Reference

- [1] Yingjie Xu, Tao You, Chenglie Du. An integrated micromechanical model and BP neural network for predicting elastic modulus of 3-D multi-phase and multi-layer braided composite[J]. *Composite Structures*, 122, 2015.308-315
- [2] Yanxi Zhang, Xiangdong Gao, Seiji Katayama, Weld appearance prediction with BP neural network improved by genetic algorithm during disk laser welding[J]. *Journal of Manufacturing Systems*, 34, 2015, pp 53-59
- [3] Sekhon, Jasjeet Singh, Walter R, Mebane, Jr. Genetic Optimization Using Derivatives: Theory and Application to Nonlinear Models [J]. *Political Analysis*, 7, 1998 pp 187-210
- [4] Walter Mebane, Jr, Jasjeet S. Sekhon. 2011. Genetic Optimization Using Derivatives: The rgenoud package for R. *Journal of Statistical Software*[J], 42(11): 1-26.
- [5] <https://github.com/joshuaas/GenoudBP>
- [6] Thomas Udelhoven, Brigitta Brigitta Schütt. Capability of feed-forward neural networks for a chemical evaluation of sediments with diffuse reflectance spectroscopy[J] *Chemometrics and Intelligent Laboratory Systems*, 51(1), 2000, pp 9-22
- [7] <http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes>
- [8] <http://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center>
- [9] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, Markus Müller pROC : an open-source package for R and S+ to analyze and compare ROC curves[J]. *BMC Bioinformatics*, 2011, 12, p.77