Niche Particle Swarm Optimization for Neural Network Ensembles

Camiel Castillo**, Geoff Nitschke, Andries Engelbrecht

Department of Computer Science, University of Pretoria, South Africa
castillo.camiel@kpmg.nl, gnitschke@cs.up.co.za, engel@cs.up.co.za

Abstract. This research investigates a swarm intelligence based multi-objective optimization algorithm for optimizing the behavior of a group of *Artificial Neural Networks* (ANNs), where each ANN specializes to solving a specific part of a task, such that the group as a whole achieves an effective solution. *Niche Particle Swarm Optimization* (NichePSO) is a speciation technique that has proven effective at locating multiple solutions in complex multivariate tasks. This research evaluates the efficacy of the NichePSO method for training a group of ANNs that form a neural network ensemble (NNE) for the purpose of solving a set of multivariate tasks. NichePSO is compared with a gradient descent method for training a set of individual ANNs to solve different parts of a multivariate task, and then combining the outputs of each ANN into a single solution. To date, there has been little research that has compared the effectiveness of applying NichePSO versus more traditional supervised learning methods for the training of neural network ensembles.

1 Introduction

In nature, biological systems such as ant and termite colonies optimize solutions to their tasks via having a set of simple individuals specialize to solving different (and complementary) parts of the problem [2]. A goal of artificial life is to replicate the mechanisms that allow groups of behaviorally simple individuals to cooperate in order to optimize solutions to complex tasks [5]. Particle Swarm Optimization (PSO) has close ties to artificial life models such as that of Reynolds [13] and Heppner [9], which indicated that emergent group dynamics such as bird flocking behavior are based on local interactions. These studies were the foundation for the development of PSO with applications that include industrial process control [11] and multi-objective function optimization [1].

Most evolutionary and swarm intelligence techniques are designed to converge on a single solution in a search space, where the quality of the solution depends on a task dependent fitness function. These techniques implicitly assume that only a single solution exists in the search space, and therefore that the search space is univariate. When presented with a multivariate task domain, typical univariate techniques will either favor a single solution, or fail to converge due to the confusion introduced by multiple possible solutions [3]. Niching techniques attempt to overcome the deficiencies of univariate optimization techniques by explicitly assuming that multiple solutions exist in a search space.

** Current work address: KPMG N.V. Burgemeester Rijnderslaan 10-20. 1185 MC Amstelveen. The Netherlands.

This paper evaluates the efficacy of a PSO based niching method [3] compared with an established gradient descent method [14] for training Neural Network Ensembles (NNEs) [8] to solve a set of multivariate classification and regression tasks. To date, there has been little research that compares the effectiveness of using more traditional supervised learning techniques such as back propagation to train NNEs with more recent niche based (multi-population) swarm intelligence techniques such as that of Zhang *et al.* [15] and Brits *et al.* [3] to train NNEs. Results elucidate that NichePSO outperforms back propagation as a NNE training method for a majority of the multivariate classification and regression tasks. Traditionally, back propagation has been successfully applied as a supervised learning approach to train NNEs for solving such tasks [10]. Given this, the following research goal, hypotheses, and performance measure were formulated.

– **Research Goal:** To elucidate that the NichePSO algorithm [3] is able to outperform a back propagation algorithm [14] for training NNEs applied to solve a given set of multivariate classification and regression tasks.
– **Hypothesis 1:** For the given task set, back propagation is able to train a NNE such that the NNE outperforms each of its constituent ANNs.
– **Hypothesis 2:** For the given task set, NichePSO is able to train a NNE such that the NNE outperforms each of its constituent ANNs.
– **Hypothesis 3:** For the given task set, NichePSO is able to train a NNE such that it outperforms a back propagation trained NNE.
– **Performance Measure:** The portion of misclassified cases and the mean squared error, for classification and regression tasks, respectively.

2 Methods for Training Neural Network Ensembles

This section describes comparative methods evaluated for solving a given set of classification and regression tasks. These methods are: *Gradient Descent trained Ensembles* (GDE) and *Niche Particle Swarm Optimized Ensembles* (NPSOE). GDE uses back propagation to train a NNE, and NPSOE uses the NichePSO algorithm to train a NNE in order to solve a given task set. Previous research has indicated that when a single network is not capable of correctly representing a given data set, the fusion of a set of networks, each of which is specialized to a part of the data set, can significantly improve performance [8]. The key idea behind the performance increase yielded by NNEs, is that each network in the ensemble specializes to solving a complementary part of the task. Collectively, these specializations result in a task performance that is superior to that of a single ANN applied to solve the same task. From a behavioral perspective, an input layer is processed by all constituent ANNs of a NNE, and a *fusing* scheme is then applied in order to combine the outputs of each ANN into one NNE output layer [8]. This research uses a uniformly weighted output scheme for the regression tasks, and a majority voting scheme for classification tasks [10].

When training a NNE using GDE or NPSOE (for either a classification or regression task), the input layer consisted of attributes from training data. This input layer was split over the constituent ANNs of the NNE. For measuring the performance of a trained NNE, the validation data was passed as the input

Table 1. GDE and NPSOE method parameter settings.

GDE and NPSOE Method Parameter Settings	
Number of hidden nodes (NPSOE / GDE)	8
Input / hidden node transfer function (NPSOE / GDE)	linear
Output node transfer function (NPSOE / GDE)	sigmoid
Learning rate (GDE)	0.001
Momentum (GDE)	0.01
Iterations (NPSOE / GDE)	50000
Number of particles (NPSOE)	30
Weight range (NPSOE / GDE)	[-1.0, +1.0]
Number of networks (NPSOE / GDE)	7
Number sub-swarms (NPSOE)	7
Initial ρ (NPSOE)	0.1
ρ increment value (NPSOE)	15
ρ decrement value (NPSOE)	5

layer to the NNE, and NNE performance compared to NNE performance using training data. An average task performance was calculated over multiple runs.

Both the GDE and NPSOE methods used a homogeneous ensemble, meaning each of the constituent ANNs was the same. The number of input neurons used by each ANN equalled the number of attributes that were being passed as the input layer for a given classification or regression task. The number of outputs always equalled one, which was the prediction or classification value. Hence, the value type of the input and output neurons depended on the value of the attributes being used by a given classification or regression task. For both methods, prior to training, the weights of each ANN were randomly initialized within the range [-1.0, 1.0]. Also, for both methods, the fusion of each of the outputs of each ANN was done according to a majority voting [8] or weighted average [12] scheme for classification and regression tasks respectively. Table 1 presents the parameter values used by the GDE and NPSOE methods. These values were derived in a set of exploratory experiments, and minor changes to these parameter values produced similar results for both GDE and NPSOE applied to all tasks.

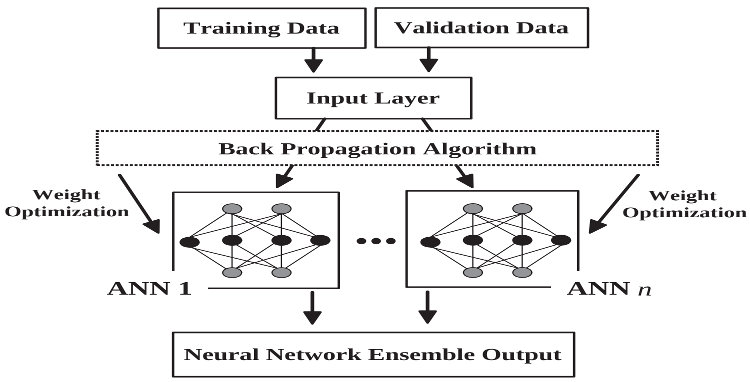

Fig. 1. Architecture of back propagation trained neural network ensemble.

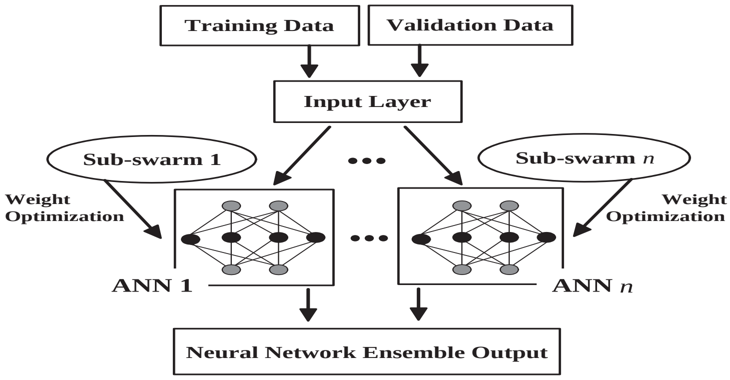

Fig. 2. Architecture of the NichePSO trained neural network ensemble.

2.1 GDE Method: Back Propagation trained Ensemble

Figure 1 illustrates the architecture of the method for training a NNE with back propagation. First, the training data is given to the input layer of the NNE. Each ANN is then trained by a back propagation algorithm [14]. The validation data is then passed to the NNE input layer, the output is compared with that produced by the training data, and the weights of the NNE are adapted accordingly.

2.2 NPSOE Method: NichePSO trained ensemble

NichePSO [4] is a niche-based PSO method that dynamically creates sub-swarms that converge upon multiple newly discovered optima in the search space. The function of the initial main swarm is thus to continually explore the search space. Figure 2 illustrates the architecture used for training a NNE with NichePSO (NPSOE). For the experiments presented in section 3, NPSOE derived one sub-swarm in order to optimize the weights of each constituent ANN. Each sub-swarm particle represents the weight vector of a given ANN. Each sub-swarm uses the GCPSO algorithm [6] as the particle velocity update strategy. GCPSO was selected since it has been demonstrated to work well with low particle numbers, and is more appropriate for exploitation than exploration. In order to train each ANN the training data is passed to each ANN as the input layer, and the output is compared to that produced when the validation data is passed. The error is used as a fitness value by the NichePSO algorithm. The weights of each ANN were set according to each sub-swarm's best particle. For a complete description of the NichePSO algorithm refer to Brits *et al.* [4].

3 Experiments

The performance of the GDE and NPSOE methods were both evaluated on five regression, and five classification tasks. The performance of each method was measured as the number of misclassified cases for classification tasks, and mean squared error for regression tasks. For the performance evaluation of each method, the validation set, and not the training set, was used. For each method applied to each task, 20 simulation runs were executed, and results presented are averages of these 20 runs. Method parameters for all classification and regression tasks are as presented in table 1. With the exception of the *random pattern classification* task, and the *Friedman#1* synthetic data set used for the first regression task, the data sets used for all classification and regression tasks were taken from the *UCI Machine Learning Repository*[1].

3.1 Classification Tasks

- **Classification Task 1: Random pattern classification:** The classification of random patterns task investigated by Hansen and Salamon [8] is used. There were 1000 training and 400 validation patterns. Each pattern was a real valued input vector with 20 attributes. Given this training set, the task was to correctly classify each validation pattern to one of the five classes.
- **Classification Task 2: Ozone Level Detection:** This data set uses 2536 instances. Each instance contains 73 attributes. The task is to correctly classify an ozone reading given a set of environment related attributes.
- **Classification Task 3: Abalone:** This data set consists of 4177 data patterns, each with eight real valued attributes. The task is to correctly classify abalones as belonging to a particular age range, given a set of attributes.
- **Classification Task 4: Wine:** The wine data set contains 178 types of wine, each with 13 real valued attributes. The attributes represent physical characteristics of the wines. All wines belong to one of three classes. The task is to correctly classify each wine to the correct class.
- **Classification task 5: Glass:** The glass database consists of 214 patterns each representing a piece of glass. Each pattern contains ten real valued attributes. The task is to classify each piece of glass as crime scene processed, or not, for a given set of attributes.

3.2 Regression Tasks

- **Regression Task 1: Friedman #1:** The Friedman#1 synthetic data set [7] corresponds to training vectors with five input and one output variable. The data set was created by randomly generating real valued input vectors with attributes in the range [0.0, 1.0], and computing a corresponding output [7]. A set of 1200 patterns was split into a training set baring 1000 instances, and the remaining 200 patterns were assigned to the validation set.
- **Regression Task 2: MPG Auto:** This data set consists of 398 instances, having eight real valued attributes. The task is to determine the fuel consumption of cars with given attributes.
- **Regression Task 3: Computer Hardware:** This data set contains 209 instances, where each instance has nine integer attributes. The task is to predict the relationship between hardware and performance given a set of computer hardware attributes.

[1] http://archive.ics.uci.edu/ml/datasets.html.

- **Regression Task 4: Servo:** The servo data set consists of 167 instances, where each instance has two continuous and two discrete attributes. The task is to predict a a servo-mechanism rise time (the time required for the mechanism to respond to a change in position) given a set of attributes.
- **Regression Task 5: Wisconsin breast cancer:** The Wisconsin breast cancer data set comprises 198 instances each with 34 real valued attributes. The task is to predict cancer (benign or malignant) given a set of attributes.

4 Results and Discussion

Table 2 presents the results of an independent t-test (0.95 confidence value) applied in order to test for a statistically significant difference between the average task performance results of GDE and NPOSOE applied to each of the classification and regression tasks. That is, the NPSOE method out-performs GDE on three out of the five classification tasks and four out of the five regression tasks.

Hypothesis 1: T-tests applied to misclassification and mean square error results of the NNE and each of the constituent ANNs trained by back propagation indicates the following significant differences. For the classification tasks, it is only for the *random pattern* classification task, that back propagation trained NNEs are able to out-perform each of their constituent ANNs (table 2). For the regression tasks, the *Friedman #1*, *Wisconsin Breast Cancer*, and *MPG Auto* trained NNEs out-perform each of their constituent ANNs (table 2).

Hypothesis 2: T-tests applied to misclassification and mean square error results of the NNE and constituent ANNs trained by NichePSO indicates the following significant differences. For the classification tasks, NichePSO trained NNEs out-perform their constituent ANNs for the *random pattern*, *abalone*, *wine*, and *Ozone Level Detection* tasks (table 2). NichePSO trained NNEs out-perform each of their constituent ANNs for all regression tasks (table 2).

Hypothesis 3: T-tests applied to misclassification and mean square error results of the back propagation (GDE method) and the NichePSO (NPSOE method) trained NNEs indicates the following significant differences. For the classification tasks, NPSOE out-performs GDE for the *random pattern*, *glass* and *Ozone Level Detection* tasks (table 2). However, for the *abalone*, and *wine* tasks, both methods yield comparable results. NPSOE out-performs GDE for all of the regression tasks, except the *Wisconsin Breast Cancer* task. A statistical comparison of mean squared indicated comparable results for this task (table 2).

These results indicate that NichePSO (hypothesis 2) comparative to back propagation (hypothesis 1) is an appropriate method for training NNEs. That is, NichePSO trained NNEs outperform each of the constituent ANNs trained by NichePSO for 90% of the tasks. Where as, back propagation trained NNEs outperform each of the constituent ANNs trained by back propagation for only 40% of the tasks. Regarding hypothesis 3, results indicate that the NichePSO trained NNE (NPSOE), comparative to a back propagation trained NNE (GDE) is appropriate for solving the given set of multivariate classification and regression tasks. That is, NPSOE outperforms GDE with a statistically significant

Table 2. Overview of acceptance or rejection of hypotheses for the results of GDE and NPSOE when applied to each of the classification and regression tasks.

Classification Tasks	Hypothesis 1	Hypothesis 2	Hypothesis 3
	Accept/Reject	**Accept/Reject**	**Accept/Reject**
Random Pattern Classification	Accepted	Accepted	Accepted
Glass	Rejected	Rejected	Accepted
Abalone	Rejected	Accepted	Rejected
Wine	Rejected	Accepted	Rejected
Ozone Level Detection	Rejected	Accepted	Accepted
Regression Tasks	**Hypothesis 1**	**Hypothesis 2**	**Hypothesis 3**
	Accept/Reject	**Accept/Reject**	**Accept/Reject**
Friedman #1	Accepted	Accepted	Accepted
Wisconsin Breast Cancer	Accepted	Accepted	Rejected
MPG Auto	Accepted	Accepted	Accepted
Computer Hardware	Rejected	Accepted	Accepted
Servo	Rejected	Accepted	Accepted

difference for 70% of the tasks. Table 3 presents the task performance results for GDE and NPSOE applied to the classification and regression tasks.

Table 3. Average portion of misclassified cases (classification) and mean squared error (regression), and standard deviation in parentheses for GDE and NPSOE.

Classification Tasks	GDE	NPSOE
Random Pattern Classification	0.33 (0.022)	0.291 (0.02)
Glass	0.378 (0.157)	0.291 (0.074)
Abalone	0.374 (0.01)	0.432 (0.016)
Wine	0.646 (0.068)	0.652 (0.047)
Ozone Level Detection	0.648 (0.015)	0.567 (0.02)
Regression Tasks	**GDE**	**NPSOE**
Friedman #1	2.169 (0.055)	0.249 (0.128)
Wisconsin Breast Cancer	29.217 (5.644)	51.413 (16.256)
MPG Auto	63.059 (0.018)	41.745 (16.938)
Computer Hardware	60.578 (6.299)	54.625 (6.853)
Servo	1.211 (0.228)	1.019 (0.006)

5 Conclusions

This research was an initial step for establishing NichePSO as being an appropriate algorithm for training neural network ensembles to solve complex multivariate tasks that require different networks to specialize to solve complementary parts of the task. This paper presented a set of multivariate classification and regression tasks. Such tasks have typically been solved via applying gradient descent algorithms to train neural networks or neural network ensembles. Results indicated that a neural network ensemble trained with NichePSO was able to exploit the multivariate nature of these tasks, which in turn lead to a significantly lower classification and prediction error rate compared to the back propagation trained ensemble. Given that NichePSO trained neural network ensembles have been successful at solving more the classification and regression tasks presented in this paper, the approach has potential applications to complex artificial life oriented tasks. For example, automating the design of a group of agent neural controllers such that the controllers develop specialized and complementary behaviors and a collective group behavior is produced that solves a given task.

References

1. C. Blum and X. Li. Swarm intelligence in optimization. In *Swarm Intelligence - Introduction and Applications*, pages 43–85. Springer, 2008.
2. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, Oxford, England, 1998.
3. R. Brits, A. Engelbrecht, and F. van den Bergh. A niching particle swarm optimizer. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning 2002 (SEAL 2002)*, pages 692–696. MIT Press, Singapore, 2002.
4. R. Brits, A. Engelbrecht, and F. van den Bergh. Locating multiple optima using particle swarm optimization. *Applied Mathematics and Computation*, 189(1):1859–1883, 2007.
5. P. Darwen and X. Yao. Every niching method has its niche: fitness sharing and implicit sharing compared. In *Proceedings of Parallel Problem Solving from Nature (PPSN) IV*, pages 398–407. Springer-Verlag, 1996.
6. A. Engelbrecht and F. van den Bergh. A new locally convergent particle swarm optimizer. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, page 96 101. IEEE Press, 2002.
7. J. Friedman, E. Grosse, and W. Stuetzle. Multidimensional additive spline approximation. *Journal of Scientific and Statistical Computing*, 4(1):291 301, 1983.
8. L. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.
9. F. Heppner and U. Grenander. A stochastic nonlinear model for coordinated bird flocks. In S. Krasner, editor, *The Ubiquity of Chaos.* AAAS Publications, Washington, D.C. USA, 1-10 1990.
10. A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, pages 231–238. MIT Press, Cambridge, USA, 1995.
11. L. Li, J. Zhou, X. Yu, and X. Li. Constrained power plants unit loading optimization using particle swarm optimization algorithm. *WSEAS Transactions on Information Science and Applications*, 4(2):296–302.
12. P. Munro and B. Parmanto. Competition among networks improves committee performance. In M. Mozer and J. Petsche, editors, *Advances in Neural Information Processing Systems*, pages 592–598. MIT Press, Cambridge, United Kingdom, 1997.
13. C. Reynolds. Flocks, herds and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–36, 1987.
14. D. Rumelhart, G. Hinton, and R. Williams. *Learning internal representations by error propagation: Parallel Distributed Processing, Volume 1: Foundations.* MIT Press, Cambridge, USA, 1986.
15. J. Zhang, J. Zhang, and K. Li. A sequential niching technique for particle swarm optimization. In *Advances in Intelligent Computing*, pages 390–399. Springer, Berlin, Germany, 2005.