Research seminar

A new optimizer using Particle swarm theory

iCIS | Data Science
Radboud University

Objective

In this paper **LBEST** is introduced which is a small modification of the method described in the original Particle Swarm Optimization(PSO) paper 'Particle swarm optimization'[1].

They examine the LBEST method with a NN training example. They also tell a bit about their original algorithm.

[1] Kennedy, James, and Russell Eberhart. 'Particle swarm optimization.' Neural Networks, 1995. Proceedings., IEEE International Conference on. Vol. 4. IEEE, 1995.

Context

Published

- Published October 1995 in 'Proceedings of the sixth international symposium on micro machine and human science'.
- Their original paper, which introduced PSO, was published in November 1995 in 'The proceedings IEEE International Conference on neural networks'.

Micro machine and human science

The conference proceedings where this paper was published is about micro machines(or other technologies) that are used to improve or understand human's life.

Development of PSO

Authors

- **James Kennedy** (Social psychologist)
- **Russell C. Eberhart** (Electrical Engineer)

Social behavior

- They started doing their research as a way modeling social behavior.
- The authors note that one of the reasons for LBEST is that it was 'effective' in simulating social behavior.

Method

In the original(global best) PSO you have a population consisting of particles. Each particle moves with a certain velocity v_i trough the search space. Each iteration the particle updates it's position x_i and v_i based on three variables.

$$^{2}v_i = w * v_i + c_1 * r() * (p - x_i) + c_2 * r() * (g - x_i)$$

$$x_i = x_i + v_i$$

Where w, c_1 and c_2 are parameters, r() is random number generator, p denotes the best known position of the particle and g denotes the best known position of all particles.

[2] The w parameters was actually introduced by *Shi, Y., & Eberhart, R. (1998, May). A modified particle swarm optimizer. In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on (pp. 69-73). IEEE.*

Method

LBEST

In LBEST you simply replace the g parameter by a parameters l which denotes the best position known in the local area. In this case the particle compares to either it's 2 nearest neighbors or it's 6 nearest neighbors. This, according to the authors, makes the algorithm less vulnerably to local optima more iterations might be required.

$$v_i = w * v_i + c_1 * r() * (p - x_i) + c_2 * r() * (l_i - x_i)$$

Often the v_i value is limited to a maximum.

Evidence

They use LBEST to find the weights for a 2x3x1 Neural Network modeling a XOR network. The algorithm stops when the MSE of the network is < 0.02. The median of the number of iterations is shown here for different $c_1 = c_2$ values.

VMAX	ACC_CONST 2.0	1.0	0.5
2.0	31.5(2)	38.5(1)	27(1)
4.0	36(1)	26	25
6.0	26.5	29	20

Neighbors = 2

They note that the number of iterations never exceeded 2000.

Evidence

VMAX	ACC_CONST 2.0	1.0	0.5
2.0	38.5	47	37.5
4.0	28.5	33	53.5
6.0	29.5	40.5	39.5

Neighbors = 6

They **do not** note that the number of iterations never exceeded 2000.

Claims

- The neighbors $= 2$ example is less vulnerable to local-optima but takes more iterations to complete.
- Also they note that groups of different particles tend to exist in different areas of the search space
- The neighbors $= 6$ example is more vulnerable to local-optima problems, but still less than GBEST.

Impact

- About 9400 citations to date ('Particle swarm optimization' has 38.000).
- A lot of variants on the original Local best approach have been proposed. Most change the number of neighbors over time.
- One typical application is where the goals is to find the multiple local-optima instead of one global optimum. 'Multimodal optimization' [3]
- In general PSO is used for all kinds of problems Neural Network design, Multi-objective problems and planning problems etc.
- They often occur in hybrid forms

[3] Li, X. (2004, June). Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. In Genetic and Evolutionary Computation–GECCO 2004 (pp. 105-116). Springer Berlin Heidelberg.

Impact on neural networks

- In general i don't see it being used very often
- I could find at least one example of an hybrid form that uses local best based method (Al-Kazemi & Mohan, 2002)
- But there are two papers that clearly suggests that PSO (or PSO alike) algorithms work better in terms of computing power required (Mirjalili, Hashim, & Sardroudi, 2012), (Gudise & Venayagamoorthy, 2003)
- There is one interesting paper that states that PSO works pretty well to find globally good solutions and uses an hybrid approach with back propagation to improve the details. (Zhang, Zhang, Lok, & Lyu, 2007)

Discussion

Questions

- I haven't heard about these algorithms being used in current neural networks. Have these algorithms eventually been used in NN applications?
- Is this only theoretical research or are there still application that use a algorithm that resembles this one?
- In what cases would you use this algorithm instead of the original algorithm?
- How does the author's argument from performance withstand more recent innovations/discoveries?
- Was any further research conducted on the method (LBEST) introduced in this paper and if yes, the obtained results were better than the GBEST version of the algorithm?
- Since the paper has been cited over 6500 times and appears to be the ground model for future PSO work, do you feel the information is still up to date and should be referenced?

Questions

- Are LBEST and GBEST, as presented in the paper, still used today?
- The paper is very short and contains quite some ideas that were already presented in the previous PSO paper. Do you think that it is too short or are there enough contributions? On what parts would you have liked the authors to go a bit more in depth, if at all?
- What is your opinion about section four? Does it belong in here or in the paper at all?
- Is there an importance of the ration between the number of particles used and the number of neighbors? Can we have a given method to compute that ration, or is it maybe a 'trail and error' method?
- Isn't LBEST the same as running GBEST on neighboring fields?
- Are you convinced of the claims made in this paper?

LBEST theory

Isn't LBEST just the same as starting GBEST multiple times with a smaller number of particles?

- It would be if every particle in a neighbor-group of particle has that same neighbor-group for every iteration.
- The authors suggest that this is almost the case for neighbor = 2. 'It appears that the invulnerability of this version to local optima might result from the fact that a number of "groups" of particles spontaneously separate and explore different regions'
- With smaller group sizes separate groups might be more likely to exist
- More likely when the local-optima are further apart, in the other case GBEST would have a higher chance of finding the global optimum.
- Variants exist that actually try to influence the behavior of groups.

‘Particle swarm optimization’ Evidence

They show some interesting early and basic results to support their method.

Neural networks

- Training a XOR model on a 2x3x1 Neural Network. With an $e < 0.05$ criterion, in an average of 30.7 iterations using 20 agents.
- Another NN on the iris dataset. ‘An average of 284 epochs’
- Seems to be less likely to overfit training data, based on differences in training and test set accuracy.

Comments

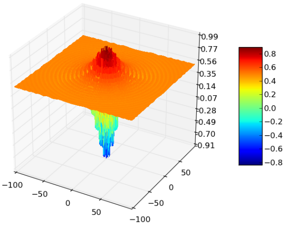
- They don’t describe how they measure against the back propagation trained networks. Epochs? Computing time? Accuracy measures?
- Unclear exactly how they applied the back propagation method(learning rate? error function).
- With what exact implementation of the PSO did they test the XOR NN.

‘Particle swarm optimization’ Schaffer f6 function

They also compare their PSO method to a ‘elementary‘ genetic algorithm(GA) using the shaffer f6 benchmark. They note that the PSO:

- ‘Approximates’ the number of evaluations of GA to reach certain performance levels.
- For every run found the global optimum.

[4]



Shaffer f6 function

[4] Obtained from http://www.cs.unm.edu/~neal.holts/dga/benchmarkFunction/schafferf6.html/

References

Al-Kazemi, B., & Mohan, C. K. (2002). Training feedforward neural networks using multi-phase particle swarm optimization. In *Neural information processing, 2002. iconip'02. proceedings of the 9th international conference on* (Vol. 5, pp. 2615–2619).

Gudise, V. G., & Venayagamoorthy, G. K. (2003). Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. In *Swarm intelligence symposium, 2003. sis'03. proceedings of the 2003 ieee* (pp. 110–117).

Mirjalili, S., Hashim, S. Z. M., & Sardroudi, H. M. (2012). Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation*, *218*(22), 11125–11137.

Zhang, J.-R., Zhang, J., Lok, T.-M., & Lyu, M. R. (2007). A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Applied Mathematics and Computation*, *185*(2), 1026–1037.