# On the Potential of Evolution Strategies for Neural Network Weight Optimization


Hao Wang
Leiden University
Leiden, The Netherlands
h.wang@liacs.leidenuniv.nl

Thomas Bäck
Leiden University
Leiden, The Netherlands
t.h.w.baeck@liacs.leidenuniv.nl

Aske Plaat
Leiden University
Leiden, The Netherlands
a.plaat@liacs.leidenuniv.nl

Michael Emmerich
Leiden University
Leiden, The Netherlands
m.t.m.emmerich@liacs.leidenuniv.nl

Mike Preuss
Leiden University
Leiden, The Netherlands
m.preuss@liacs.leidenuniv.nl



## ABSTRACT

Artificial neural networks typically use backpropagation methods for the optimization of weights. In this paper, we aim at investigating the potential of applying the so-called evolutionary strategies (ESs) on the weight optimization task. Three commonly used ESs are tested on a multilayer feedforward network, trained on the well-known MNIST data set. The performance is compared to the Adam algorithm, in which the result shows that although the $(1+1)$-ES exhibits a higher convergence rate in the early stage of the training, it quickly gets stagnated and thus Adam still outperforms ESs at the final stage of the training.



## CCS CONCEPTS

• **Computing methodologies → Randomized search**; **Machine learning**;

## KEYWORDS

Evolution strategy, backpropagation, neural network training

**ACM Reference Format:**
Hao Wang, Thomas Bäck, Aske Plaat, Michael Emmerich, and Mike Preuss. 2019. On the Potential of Evolution Strategies for Neural Network Weight Optimization. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322014


## 1 INTRODUCTION

In the past, Evolutionary Algorithms were already suggested as a means to optimize the weights and the structure of multilayer neural networks [3]. However, this had little effect on the practical use of a neural network and especially deep learning methods. Modern libraries such as e.g., `keras` do not even contain evolutionary optimization algorithms. Recently, first results on optimizing the weights of deep convolutional networks using Evolution Strategies [4] or Genetic Algorithms [5] also showed promising results and even classical evolution strategies could show competitive performance on challenging benchmarks, such as the Atari-game challenge [7]. This paper is a first step towards filling this gap of knowledge and investigating whether evolution strategies can be competitive to or even better than state-of-the-art weight optimization algorithms such as Adam. Towards this goal, we select three variants of evolution strategies, namely the $(1+1)$-ES with $1/5$ success rule, the $(\mu \overset{+}{,} \lambda)$-MSC-ES, and the $(\mu, \lambda)$-sep-CMA-ES, which are tested on the well-known MNIST data set.


Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.



*GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic*



© 2019 Association for Computing Machinery.



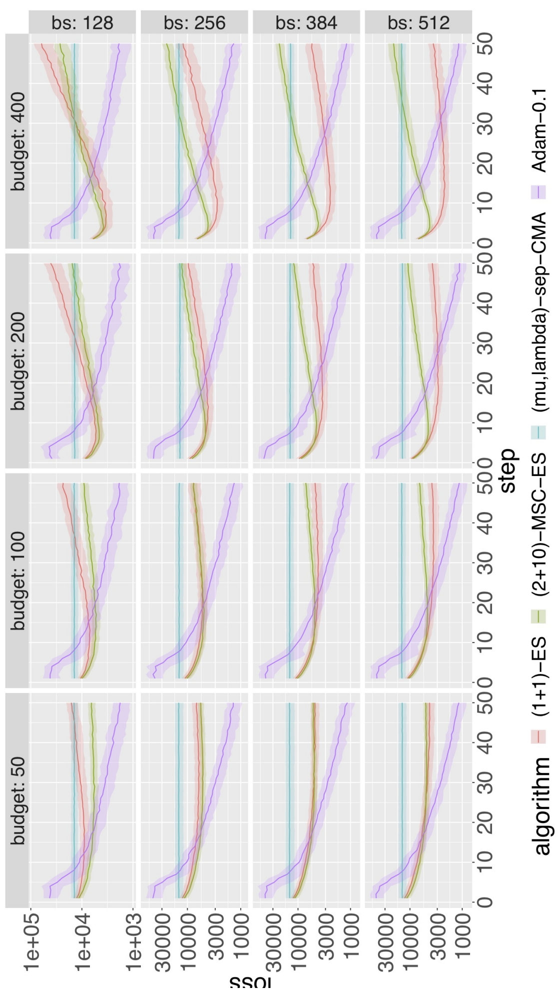
ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322014


## 2 CANDIDATE EVOLUTION STRATEGIES

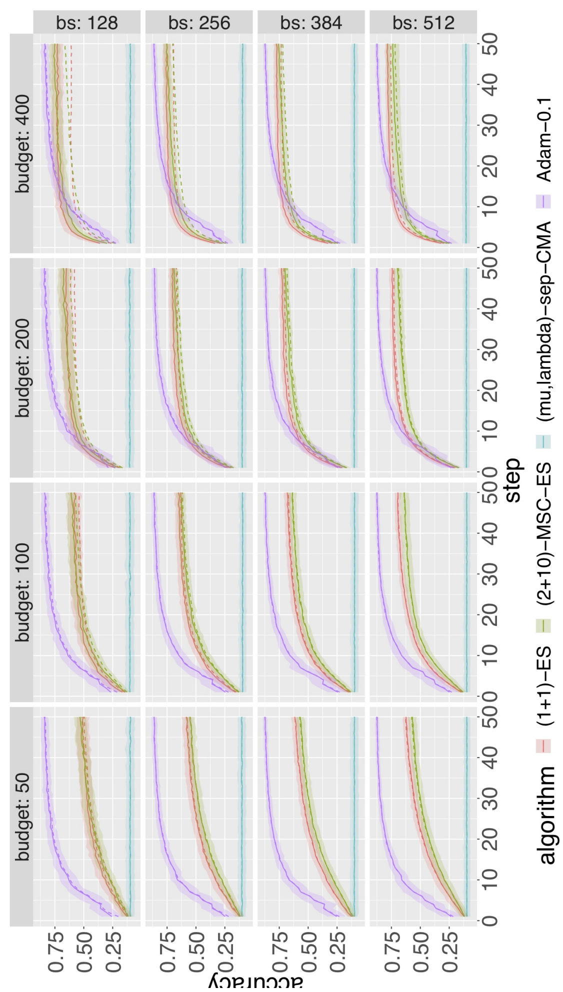
This paper focuses on evolution strategies as a subclass of evolutionary algorithms that is standard for continuous optimization, within which three commonly used of evolution strategies are considered for the neural network training: 1) The $(1+1)$-**ES with** $1/5$ **success rule** [1] is the simplest ES algorithm that employs an isotropic multivariate normal distribution to generate the mutation. The global step-size is controlled by the well-known $1/5$ success rule. 2) The $(\mu \overset{+}{,} \lambda)$-**self-adaptive step-size control (MSC) ES** [1] is a population-based ES algorithm, in which $\lambda$ candidate individuals are generated using the isotropic multivariate normal distribution. The global step-size is controlled in a self-adaptive manner. 3) The $(\mu, \lambda)$-**sep-CMA-ES** [6] (also called separable-CMA-ES) is the simplified variant of the well-known CMA-ES [2].

## 3 EXPERIMENT AND CONCLUSION

In the experiments reported here, a neural network is constructed to handle the well-known MNIST data set. The network structure is specified as follows: 1) **Input**: flattened MNIST image of size $784 = 28 \times 28$. 2) **First layer**: 200 linear units with biases. 3) **Second layer**: 100 linear units with biases. 4) **Third layer**: 50 softmax units with biases. 5) **Output layer**: 10 softmax units with biases. 6) **Loss function**: the so-called cross entropy is chosen as the objective function for ESs. Note that, there are **182660** parameters in this larger neural network. In order to determine the proper batch size and evaluation budget for the ES, several different batch sizes ($\{128, 256, 384, 512\}$) and evaluation budgets ($\{50, 100, 200, 400\}$) are tested, yielding 16 combinations of those. The experiment results are shown in Fig. 1. It is obvious that the Adam method performs




(a) Loss function




(b) Accuracy

**Figure 1: Evolution strategies are compared to the Adam method. The results are averaged over 100 runs where the shaded bands show the standard deviation.**

better in terms of loss function as well as accuracy. Taking a closer look at the loss values in Fig. 1, the convergence rate of ESs is faster than that of Adam in the first 10 steps. Then the loss function value appears to stagnate and even increase in some case (e.g., budget 400 and batch size 128). This phenomenon might be due to the fact that applying a relatively larger evaluation budget (e.g., 400) to an elitist ES algorithm can result in a potential overfitting to the current batch of data. In general, we arrive at the following conclusions: 1) A large evaluation budget leads to fast initial convergence of the loss function. The converged loss function weights, however, would quickly overfit the subsequent training batches. 2) A relatively large batch size should be taken together in combination with ES algorithms, to alleviate the overfitting issue as mentioned above.

## REFERENCES

[1] Thomas Bäck, Christophe Foussette, and Peter Krause. 2013. *Contemporary Evolution Strategies* (1 ed.). Springer-Verlag Berlin Heidelberg.

[2] Nikolaus Hansen. 2006. *The CMA Evolution Strategy: A Comparing Review.* Springer Berlin Heidelberg, Berlin, Heidelberg, 75–102. https://doi.org/10.1007/3-540-32494-1_4

[3] Michael Hüsken, Yaochu Jin, and Bernhard Sendhoff. 2005. Structure optimization of neural networks for evolutionary design optimization. *Soft Computing* 9, 1 (2005), 21–28.

[4] Ilya Loshchilov and Frank Hutter. 2016. CMA-ES for Hyperparameter Optimization of Deep Neural Networks. *CoRR* abs/1604.07269 (2016). http://arxiv.org/abs/1604.07269

[5] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. 2017. Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning. *arXiv e-prints*, Article arXiv:1712.06567 (Dec. 2017), arXiv:1712.06567 pages. arXiv:cs.NE/1712.06567

[6] Raymond Ros and Nikolaus Hansen. 2008. A simple modification in CMA-ES achieving linear time and space complexity. In *International Conference on Parallel Problem Solving from Nature*. Springer, 296–305.

[7] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. *arXiv e-prints*, Article arXiv:1703.03864 (March 2017), arXiv:1703.03864 pages. arXiv:stat.ML/1703.03864