Towards Solving Neural Networks with Optimization Trajectory Search

Lia T. Parsenadze
Kyushu University
leephars@gmail.com

Danilo Vasconcellos Vargas
Kyushu University
vargas@inf.kyushu-u.ac.jp

Toshiyuki Fujita
Kyushu University
tfujita@econ.kyushu-u.ac.jp

ABSTRACT

Modern gradient based optimization methods for deep neural networks demonstrate outstanding results on image classification tasks. However, methods that do not rely on gradient feedback fail to tackle deep network optimization. In the field of evolutionary computation, applying evolutionary algorithms directly to network weights remains to be an unresolved challenge. In this paper we examine a new framework for the evolution of deep nets. Based on the empirical analysis, we propose the use of linear sub-spaces of problems to search for promising optimization trajectories in parameter space, opposed to weight evolution. We show that linear sub-spaces of loss functions are sufficiently well-behaved to allow trajectory evaluation. Furthermore, we introduce fitness measure to show that it is possible to correctly categorize trajectories according to their distance from the optimal path. As such, this work introduces an alternative approach to evolutionary optimization of deep networks.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

Optimization trajectory search, search methodologies, neural network optimization

ACM Reference Format:
Lia T. Parsenadze, Danilo Vasconcellos Vargas, and Toshiyuki Fujita. 2019. Towards Solving Neural Networks with Optimization Trajectory Search. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3319619.3326796

1 INTRODUCTION

Deep neural networks are complex optimization problems often tackled with approximation of gradients. They are used to guide some random point in parameter space to an optimal position. Although such methods have shown impressive results, there are still no alternatives to this purely mathematical approach. Evolutionary methods are still unable to undertake direct neural network optimization, thus development of new methods is required.

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

ACM ISBN 978-1-4503-6748-6/19/07.
https://doi.org/10.1145/3319619.3326796

In our investigation we analyze a possibility of searching for linear sub-spaces with best optimization trajectories. In other words, we propose encoding individuals not as points, but as lines in parameter space. Based on our analysis of properties of linear sub-spaces, we hypothesize that 1-D encoding, opposed to 0-D encoding, may be a better way of performing a search in high-dimensional space.

We conclude this from the following experimental observations:

- **Loss is unimodal along random lines.** This has two main conclusions: (1) Optimal network configuration in a given linear sub-space is easy to determine. (2) Unimodality in random directions ensures that populations of sufficient size can be generated.
- **Consistency in sub-optimal space.** We confirm smoothness of loss surface as lines move away from the optima, which is required for evolvable setting.
- **Lines are classifiable.** Individuals encode multiple solutions, and as such a method to measure the overall line fitness is required. We propose such method and show that trajectories are classifiable according to their performance.

2 RELATED WORK

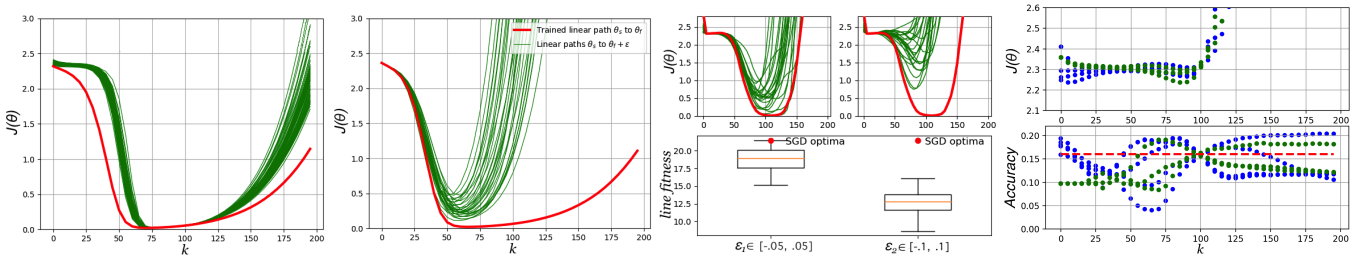
In the field of deep learning, genetic computation is often used for automatic development of network structures [6, 7]. Neuroevolution evolves only weights of the neural networks while its structure remains fixed [2, 3], but with increasing size of networks this approach faces significant challenges [4].

The possibility of evolving 1-D linear directions instead of a point in neural network parameter space has not yet been explored. In previous work we have seen the use of 1-D and 2-D linear interpolations for analysis of gradient path [1] and for visualization of loss surfaces [5]. We explore further and discuss the practical and algorithmic value of directional search.

3 EXPERIMENTS

In our experiments we examine high-dimensional space of neural network parameters and its linear sub-spaces. To do that, we use 1-D interpolations computed in a sub-space of two points θ_i and θ_j. Let $L_{ij}(\theta_i, \theta_j)$ denote a set of solutions such that $\forall\theta_k \in L_{ij}$, $\theta_k = (1 - \frac{k}{n})\theta_i + \frac{k}{n}\theta_j$ and $k = 0, 1, ...n$ and $|L_{ij}| = n$. As such, L_{ij} is a set containing solutions θ_i, θ_j and $n - 2$ discrete points on a line between them. Finally, the optimization function $J(\theta)$ is interpolated from the set, $J(\theta_k), \forall\theta_k \in L_{ij}$.

In the way described above, we define optimal trajectory. The set of starting network parameters θ_s is randomly initialized, and trained with gradient descent methods until optimization function converges at θ_f. Gradient trajectory $L_{sf}(\theta_s, \theta_f)$ is taken as optimal (fig. 1.a, red line). Next, we examine random trajectories leading to minima θ_f. To do that we initialize 50 random parameter vectors



(a) Directions random-to-optima **(b)** $\varepsilon \in [-.25, .25]$ **(c) Line fitness evaluation for** ε_1, ε_2 **(d) Linear sub-spaces of DE solutions**

Figure 1: Main experiments results

$\theta^{(l)}_{rand}, l = 1, ..., 50$, and interpolate while θ_f remains fixed, yielding $L^{(l)}_{rf}(\theta^{(l)}_{rand}, \theta_f), l = 1, ..., 50$ (fig. 1.a, green lines). First of all, this shows that surface of high-dimensional objective function is smooth, further contributing to the opinion that neural networks do not enter end escape local minimas during optimization. We see no evidence that there are saddle points on random directions leading to algorithmic optima. Secondly, we confirm that there are many unimodal sub-spaces leading to optima, not just gradient trajectory, implying that generating populations of individuals encoded as directions is possible.

Figure 1.b shows results of other series of experiments where we explore directions that are drifting apart from θ_f. To diverge from optima, we add an error ϵ to our random paths while fixing θ_s, thus computing $L^{(l)}_{s\epsilon}(\theta_s, \theta_f + \epsilon^{(l)}), l = 1, ..., 50$. This provides further argument that the surface is smooth, as linear sub-spaces demonstrate consistent pattern even when trajectories are away from optima. Furthermore, we see that unimodality is retained even with intensifying levels of noise, meaning optimal configuration within lines are easy to find. As such, it can be concluded that lines of different performance levels have analogous properties.

Naturally, we are interested in classifying trajectories according to their performance. We propose following trajectory fitness evaluation: $F = \sum_0^n acc(\theta_k), \forall \theta_k \in L_{ij}(\theta_i, \theta_j)$, where $acc(\theta_k)$ is accuracy of neural network with θ_k solution. To validate an ability of this metric to classify trajectories, we examine lines that are ϵ_1 and ϵ_2 distances away from optima. $\epsilon_1 \in [-.05, .05]$ is a smaller noise, meaning trajectories containing ϵ_1 are closer to optimal line (fig.1.c, upper-left), than the ones containing $\epsilon_2 \in [-.1, .1]$, which represents larger noise (fig.1.c, upper-right). Computing line fitness as described above, we see that two groups of trajectories are successfully classified. Box-plot on figure 1.c shows that the fitness measure of individual lines correspond to their distance from optimal trajectory, further demonstrating the consistency of the approach.

In our final experiment, we implement preliminary tests to support the proposed encoding. The test is conducted as follows: we apply direct differential evolution (DE) to a small CNN to obtain a population of solutions trapped in local minimas with similar levels of performance. Similar to the previous test, we examine linear sub-spaces of DE solutions. Figure 1.d shows the result. Upper figure shows loss interpolations, and the lower image - accuracy measurements. Red line indicates performance threshold for DE solutions. Examining linear sub-spaces of these solutions reveals some points that are better performing than others, as we see many above threshold. This preliminary test shows that linear sub-spaces contain better performing configurations than the ones initially available.

4 NETWORKS

All test were run on three neural networks and 2 data sets. Their configurations are as follows: (1) *MNIST*, MLP: 5 dense layers, dropout 0.2; (2) *MNIST*, CNN-1: 3 convolutional layers, maxpolling and 2 dense layers, no dropout, figure 1.a,b. (3) *CIFAR-10*, CNN-2: 5 convolutional layers with maxpolling and 4 dense layers, dropout 0.3, figure 1.c.

5 RESULTS SUMMARY

Based on experiments, we make the following conclusions:

- Landscapes of examined networks are obstacle-free, as there was no evidence of surface irregularity.
- Based on unimodality of linear sub-spaces, we propose trajectory encoding to be a discrete set of linearly aligned solutions.
- Encoded Individuals have consistent fitness mapping.
- Linear sub-spaces can contain directional information.

REFERENCES

[1] Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. 2014. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544* (2014).
[2] Nikolaus Hansen. 2006. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation*. Springer, 75–102.
[3] Matthew Hausknecht, Joel Lehman, Risto Miikkulainen, and Peter Stone. 2014. A neuroevolution approach to general atari game playing. *IEEE Transactions on Computational Intelligence and AI in Games* 6, 4 (2014), 355–366.
[4] Jan Koutník, Giuseppe Cuccu, Jürgen Schmidhuber, and Faustino Gomez. 2013. Evolving large-scale neural networks for vision-based reinforcement learning. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM, 1061–1068.
[5] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*. 6391–6401.
[6] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc Le, and Alex Kurakin. 2017. Large-scale evolution of image classifiers. *arXiv preprint arXiv:1703.01041* (2017).
[7] Masanori Suganuma, Shinichi Shirakawa, and Tomoharu Nagao. 2017. A genetic programming approach to designing convolutional neural network architectures. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 497–504.