 DOI: 10.3217/978-3-85125-663-5-41

The Quest for the Golden Activation Function*

Mina Basirat[1], Alexandra Jammer[2], and Peter M. Roth[1]

I. PROBLEM STATEMENT AND MOTIVATION

Deep Neural Networks have been shown to be beneficial for a variety of tasks, in particular allowing for end-to-end learning and reducing the requirement for manual design decisions. However, still many parameters have to be chosen in advance, also raising the need to optimize them. Moreover, since increasingly more complex and deeper networks are of interest, strategies are required to make neural network training efficient and stable. While initialization and normalization techniques are well studied, a relevant and important factor is often neglected: the selection of a proper activation function (AF). In [1], we tackled this problem and learned task-specific activation functions. For that purpose, we take two main observations into account. First, the positive and negative parts of activation functions have a different influence on information propagation. Second, the search space is very huge and hard to explore. Thus, motivated by evolution theory (e.g., [3], [4]) we introduced an approach to evolving piece-wise activation functions building on the ideas of Genetic Programming (e.g., [2]).

II. OVERVIEW OF THE APPROACH

The evolution typically starts from a population consisting of randomly selected candidate solutions, called *individuals*. These are described by a set of properties (*genes*: functions in our case), which can be altered by three breeding operations: (a) *Selection*, (b) *Crossover*, and (c) *Mutation*. In addition, we introduced two new operators, especially representing our problem: *Inheritance Crossover* and *Hybrid Crossover*. The first additionally allows for combining different positive and negative parts, whereas the second one introduces the possibility to combine functions using mathematical operators. Then, in an iterative process, where we refer to one iteration as a *generation*, each individual is evaluated and based on their fitness, we select a set of parent solutions for breeding. Subsequently, we apply breeding operators on pairs of individuals to generate new pairs of offsprings. This process is repeated until a pre-defined number of generations or a pre-defined optimality criterion is met. To this end, we are able to evolve even more complex activation functions. This is in particular remarkable as only very basic candidate solutions are provided (in contrast to, e.g., `Swish`). Moreover, our approach is adapting very well to different kinds of problems, also yielding different activation functions for different tasks.

*We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

[1]Graz University of Technology, {mina.basirat, pmroth}@icg.tugraz.at
[2]University of Graz, alexandra.jammer@uni-graz.at

III. ILLUSTRATIVE EXPERIMENTAL RESULTS

To demonstrate the benefits of our approach, we run experiments on two different classification benchmarks of different complexity, namely CIFAR-10 and CIFAR-100, and compared it to existing approaches. Illustrative results for ResNet-20 are shown in Table I (In addition, we carried out experiments for ResNet-56 and VGG-16.).

TABLE I: Results for CIFAR-10 using ResNet-20.

Activation Function	Accuracy
Ours (best)	79.24%
`Swish`	78.51%
`ELU`	73.00%
`ReLU`	71.98%
`SeLU`	65.79%
Random Search (best)	76.03%

In addition, Figs. 1 and 2 show the best-performing activation functions for CIFAR-10 and CIFAR-100, respectively. It cannot only be seen that for the different tasks different activation functions have been evolved, but also that the shapes of the top-performing functions are similar. For more details, we would like to refer to [1].

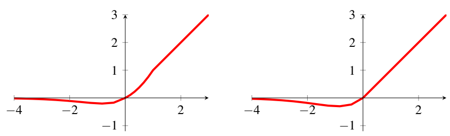

Fig. 1: Top 2 evolved AFs for CIFAR-10.

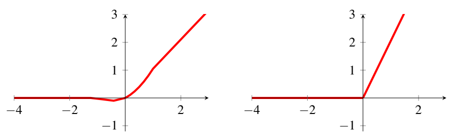

Fig. 2: Top 2 evolved AFs for CIFAR-100.

REFERENCES

[1] M. Basirat and P. M. Roth. Learning task-specific activation functions using genetic programming. In *Proc. Int'l Conf. on Computer Vision Theory and Applications*, 2019.

[2] K. A. De Jong. *Evolutionary Computation: A Unified Approach*. MIT Press, 2006.

[3] D. J. Futuyma and M. Kirkpatrick. *Evolution*. Oxford University Press, 4th edition, 2017.

[4] J. B. Losos, D. A. Baum, D. J. Futuyma, H. E. Hoekstra, R. E. Lenski, A. J. Moore, C. L. Peichel, D. Schluter, and M. J. Whitlock, editors. *The Princeton Guide to Evolution*. Princeton University Press, 2014.