EqPropNEAT
Extending NEAT with equlibrium propagation

Cs. Karikó, M. Kovács, Zs. Tornai

***Abstract*—Artificial neural networks have been employed successfully to solve a great variety of tasks over the past decade. In most applications, network architecture is usually decided with a trial-and-error process, relying on empirical experience. The neuoroevolution of augmenting topologies algorithm (NEAT) [1] proposes a solution to finding well fitting network architectures to a given task, however for learning weights, it only uses a simple genetic mutation rule.**

This paper proposes an extension of NEAT with an additional learning step employing equilibrium propagation [2] for learning edge weights. We claim that the resulting novel technique needs less iterations for convergence, as weights are adjusted by a method which was specifically invented for such task. Our experimental results show a significant reduction in training times on common neural network benchmark tasks, while achieving similar prediction quality.

***Index Terms*—neuroevolution, neat, equilibrium propagation.**

I. Introduction

THE process of designing neural network models always involves deciding on a particular network architecture. In order to avoid both overfitting as well as subpar prediction quality, the right amount of network complexity has to be found for a particular learning task. Over the years, a great amount of experience has accumulated regarding this decision, however these rules are more about empirical know-how, than proven theorems. For our experiment we took well tested and proven algorithms NEAT and Equlibrium Propagation(EP) [2] and tried to design a neural network which best represents a biological solution to a given task. For this reason EP was chosen rather than Backpropagation(BP) [3] to further fine tune the network in a way which would be plausible in a biological network. Due to the similarities between BP and EP we hope to capture all the positive aspects of BP while still maintaining the properties of a biological network. Due to the nature of EP our networks learns slower than a comparable network would utilizing BP, but as BP is considered "biologically implausible" we accept this shortcoming. In chapter **N** we will discuss the performance compared to BP and we will explain why the difference in real-world applications would not be significant.

II. Background

As mentioned in the previous chapter, the NEAT algorithm uses a simple genetic mutation rule for evolving edge weights. With a sufficiently large population this method can evolve correct connection weights, however this process usually requires lot of generations. A number of extensions have already been proposed for improving weight search by incorporating BP to the original algorithm [4]. Such attempts include Learning-NEAT [5], DeepNeat and CoDeepNeat [6]. A common property of these techniques is that s limited number of training epochs are run during evaluation for adjusting edge weights and determining network performance.

Equilibrium propagation proposes a new approach to training neural networks. Unlike conventional artificial neural network models, equilibrium propagation models the network as a continuos-time dynamic system. Training and evaluation is accomplished with a numerical simulation of changing energy states converging to an equilibrium. After the simulation has arrived at a fixed point, the results can be read from the state of output nodes. Training is achieved by "nudging" the states of the output nodes towards the correct prediction. This change than propagates throughout the network and weight updates can be computed from the two states of a given node during the evaluation and the training phase.

III. EqPropNEAT

Our method substitutes BP with equilibrium propagation. During the mutation stage of each generation, instead of assigning random weights to some connections, we employ equilibrium propagation iterations in order to adjust edge weights.

As the network's complexity increases with more and more generations, the search space for weights increases rapidly. In order to get the most out of each network architecture proposed by the NEAT algorithm, we suggest adjusting the number of equilibrium propagation steps run between generations according to network complexity. We have tested a number of different formulae for determining the number of necessary simulation iterations:

$$N_1 = V \tag{1}$$

$$N_2 = V + E \tag{2}$$

$$N_3 = V \cdot E \tag{3}$$

Where V and E denote the number of nodes and edges respectively.

TODO: fake figure about network complexity and needed iterations

References

[1] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.

[2] B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," *Frontiers in computational neuroscience*, vol. 11, p. 24, 2017.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.
[4] M. Y. Ibrahim, R. Sridhar, T. Geetha, and S. Deepika, "Advances in neuroevolution through augmenting topologies – a case study," in *2019 11th International Conference on Advanced Computing (ICoAC)*, 2019, pp. 111–116.
[5] L. Chen and D. Alahakoon, "Neuroevolution of augmenting topologies with learning for data classification," in *2006 International Conference on Information and Automation*, 2006, pp. 367–371.
[6] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy *et al.*, "Evolving deep neural networks," in *Artificial intelligence in the age of neural networks and brain computing*. Elsevier, 2019, pp. 293–312.