An NEAT/HyperNEAT approach for evolving deep neural network architectures

1 Description

Neuroevolutionary approaches are used to evolve the architecture and/or parameters of neural networks [1, 2, 3, 4]. Two examples of these approaches are NEAT [5] and HyperNEAT [6]. The first approach combines the search for appropriate network weights with the complexification of the network structure. The main idea of HyperNEAT is to use a Compositional Pattern Producing Network (CPPN) to represent the connectivity of a neural network layer or the connectivity between neural network layers. The network will receive as inputs two coordinates $(x_1, y_1), (x_2, y_2)$ in a Cartesian space and outputs a value (of the strength) between them. The grid where the neurons are located is called the substrate. The connective CPPNs are usually evolved using NEAT.

While NEAT and HyperNEAT have shown promising results in the evolution of networks that can work as agents (e.g., in games) [7, 8], their application to evolve the architecture of DNNs has been less explored [9, 10].

2 Objectives

The goal of the project is to apply NEAT and/or HyperNEAT to evolve the network architecture of DNNs for improving their performance on a given classification or regression task.

The student should: 1) Select a ML task and an DNN architecture (MLPs can be used). 2) Estimate the the quality of the network for the architecture/parameters manually set. 3) Use NEAT/HyperNEAT to search for an architecture and/or set of parameters that improve the performance of the original network. Different Python implementations of NEAT and HyperNEAT are available [1].

As in other projects, a report should describe the characteristics of the design, implementation, and results. A Jupyter notebook should include calls to the implemented function that illustrate the way it works.

3 Suggestions

- Read the relevant bibliography about NEAT and HyperNEAT.
- Start with one of the architectures and/or tasks implemented in the course labs.
- Create a tensorflow class that can serve to evaluate any network configuration evolved by HyperNEAT.
- Implementations can use any Python library that implements DNNs.

References

[1] C. Cotta, E. Alba, R. Sagarna, and P. Larrañaga. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, chapter Adjusting weights in artificial neu-

[1]See software packages mention in `http://eplex.cs.ucf.edu/hyperNEATpage/`

ral networks using evolutionary algorithms, pages 357–374. Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.

[2] Dario Floreano, Peter Dürr, and Claudio Mattiussi. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1):47–62, 2008.

[3] Gene Sher, Kyle Martin, and Damian Dechev. Preliminary results for neuroevolutionary optimization phase order generation for static compilation. In *Proceedings of the 11th Workshop on Optimizations for DSP and Embedded Systems*, pages 33–40. ACM, 2014.

[4] M. O'Halloran, S. Cawley, B. McGinley, R.C. Conceição, F. Morgan, E. Jones, and M. Glavin. Evolving spiking neural network topologies for breast cancer classification in a dielectrically heterogeneous breast. *Progress In Electromagnetics Research*, 25:153–162, 2011.

[5] Kenneth O Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.

[6] K.O. Stanley, D.B. D'Ambrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.

[7] Erin J Hastings, Ratan K Guha, and Kenneth O Stanley. Evolving content in the galactic arms race video game. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, pages 241–248. IEEE, 2009.

[8] Thomas G van den Berg and Shimon Whiteson. Critical factors in the performance of HyperNEAT. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 759–766. ACM, 2013.

[9] Sebastian Risi, Joel Lehman, and Kenneth O Stanley. Evolving the placement and density of neurons in the HyperNEAT substrate. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 563–570. ACM, 2010.

[10] Sebastian Risi and Kenneth O Stanley. Enhancing es-hyperneat to evolve more complex regular neural networks. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1539–1546. ACM, 2011.