Neuroevolution for Complex Domains

Degree programme: BSc in Computer Science | Specialisation: Computer Perception and Virtual Reality
Thesis advisor: Prof. Dr. Jürgen Eckerle
Expert: Han van der Kleij (SBB AG)

Using a genetic algorithm to generate neural networks allows to not only configure weights, but also evolve its topology. Such an algorithm was developed with focus on exploration to tackle a common issue in reinforcement learning tasks: getting trapped in local optima.

Neuroevolution using genetic algorithms

In neuroevolution, evolutionary algorithms are used to generate artificial neural networks. This method can be applied in a broad range of domains. In this work, the evolutionary algorithm used is a genetic algorithm. This searching algorithm has a valuable property: it is truly gradient-free, and thus, the risk of becoming trapped in local optima, a common issue in reinforcement learning tasks, is lower.

Topology and weight evolving neural networks

While for most methods, the topology of the neural network is chosen manually, here, this is not the case. A genetic algorithm is, with the right encoding for neural networks, able to modify the topology using mutation and crossover. Starting with a minimal network topology (**1**) not only allows beneficial crossover by carefully marking newly added nodes and connections for gene alignment, but also (**2**) encourages finding minimal solutions in terms of topology. Fig. 1 shows the best neural network from the first (top left) and last (bottom left) generation. The evolved neural network solved the mountain-car problem reliably using 5 hidden neurons and recurrent connections.

Exploration through speciation

In genetic algorithms, exploration is often implemented by rewarding diversity. This way, an individual can perform worse, but being different in weights and topology, it may be preferred by selection over a better performing, more common one. Here, speciation is used to enhance diversity. Individuals are clustered into different species, based on their similarity. Within each species, individuals share their fitness among them, hindering a species to take over the whole population.

Results and conclusion

In the mountain-car environment, one of the selected problems in this work, the reward (based on the number of frames used to solve the problem, at most 200) is delayed until the end of an episode. Thus, a huge amount of exploration is needed to generate a configuration obtaining at least some reward. Only after that, fine-tuning is possible. Fig. 1 presents the performance of the algorithm compared to that of other algorithms.

More generally, the method used in this work could solve a broad range of reinforcement learning tasks where other algorithms either did not get a solution in time or obtained a smaller reward. While having minimal or small solutions in terms of topology is usually not a goal, it is unquestionably a valuable property. Variations on handling diversity were implemented and showed that exploration is a main key to success.

Leandro Patricio Kai Lerena
leandro.lerena@gmail.com

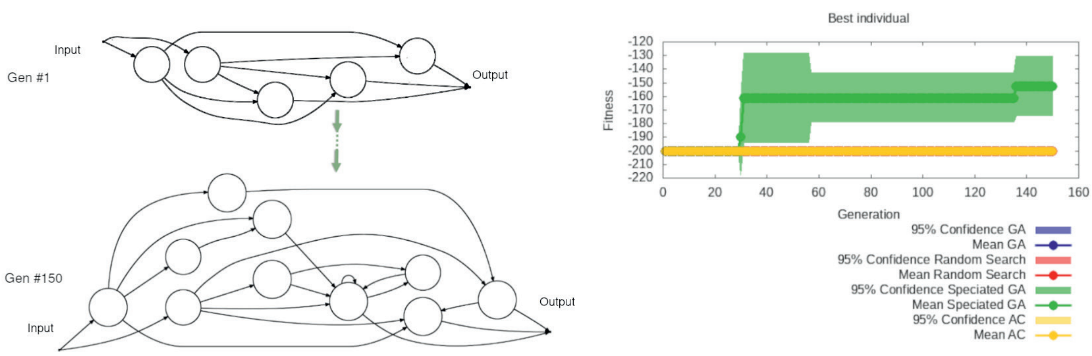

Fig. 1: Evolution of a neural network in 150 generations and its performance (green) compared to other algorithms, which did not solve the problem in the given time.