Welcoming the Era of Deep Neuroevolution

By Kenneth O. Stanley & Jeff Clune

December 18, 2017

On behalf of an Uber AI Labs team that also includes Joel Lehman, Jay Chen, Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, & Xingwen Zhang.

In the field of deep learning, deep neural networks (DNNs) with many layers and millions of connections are now trained routinely through stochastic gradient descent (SGD). Many assume that the ability of SGD to efficiently compute gradients is essential to this capability. However, we are releasing a suite of five papers that support the emerging realization that neuroevolution, where neural networks are optimized through evolutionary algorithms, is also an effective method to train deep neural networks for reinforcement learning (RL) problems. Uber has a multitude of areas where machine learning can improve its operations, and developing a broad range of powerful learning approaches that includes neuroevolution will help us achieve our mission of developing safer and more reliable transportation solutions.

Genetic algorithms as a competitive alternative for training deep neural networks

Using a new technique we invented to efficiently evolve DNNs, we were surprised to discover that an extremely simple genetic algorithm (GA) can train deep convolutional networks with over 4 million parameters to play Atari games from pixels, and on many games outperforms modern deep reinforcement learning (RL) algorithms (e.g. DQN and A3C) or evolution strategies (ES), while also being faster due to better parallelization. This result is surprising both because GAs, which are not gradient-based, were not expected to scale well to such large parameter spaces and also because matching or outperforming the state-of-the-art in RL using GAs was not thought to be possible. We

further show that modern GA enhancements that improve the power of GAs, such as novelty search, also work at DNN scales and can promote exploration to solve deceptive problems (those with challenging local optima) that stymie reward-maximizing algorithms such as Q-learning (DQN), policy gradients (A3C), ES, and the GA.

This GA policy scores 10,500 on Frostbite. DQN, AC3, and ES score less than 1,000 on this game.

The GA plays Asteroids well. It outperforms DQN and ES on average, but not A3C.

Safe mutations through gradient computations

In a separate paper, we show how gradients can be combined with neuroevolution to improve the ability to evolve recurrent and very deep neural networks, enabling the evolution of DNNs with over one hundred layers, a level far beyond what was previously shown possible through neuroevolution. We do so by computing the gradient of network outputs with respect to the weights (i.e. not the gradient of error as in conventional deep learning), enabling the calibration of random mutations to treat the most sensitive parameters more delicately than the least, thereby solving a major problem with random mutation in large networks.

Both animations show a batch of mutations of a single network that could solve the maze (with start at lower left and goal at upper left). Normal mutations mostly lose the ability to reach the end while safe mutations largely preserve it while still yielding diversity, illustrating the significant advantage of mutating safely.

How ES relates to SGD

Our papers complement an already-emerging realization, which was first noted by a team from OpenAI, that the evolution strategy variety of neuroevolution can optimize deep neural networks competitively on Deep RL tasks. However, to date, the broader implications of this result have remained a subject of some speculation. Setting the stage for further innovation with ES, we provide deeper insight into its relationship to SGD through a comprehensive study that examines how close the ES gradient approximation actually comes to the optimal gradient for each mini-batch computed by SGD on MNIST, and also

how close this approximation must be to perform well. We show that ES can achieve 99 percent accuracy on MNIST if enough computation is provided to improve its gradient approximation, hinting at why ES will increasingly be a serious contender in Deep RL, where no method has privileged access to perfect gradient information, as parallel computation increases.

ES is not just traditional finite differences

Adding further understanding, a companion study confirms empirically that ES (with a large enough perturbation size parameter) acts differently than SGD would, because it optimizes for the expected reward of a *population* of policies described by a probability distribution (a *cloud* in the search space), whereas SGD optimizes reward for a single policy (a *point* in the search space). This change causes ES to visit different areas of the search space, for better or for worse (both cases are illustrated). An additional consequence of optimizing over a population of parameter perturbations is that ES acquires robustness properties not attained through SGD. Highlighting that ES optimizes over a population of parameters also emphasizes an intriguing link between ES and Bayesian methods.

Random perturbations of the weights of a walker learned by TRPO lead to significantly less stable gaits than random perturbations of a walker of equivalent quality evolved by ES. The original learned walkers are at the center of each nine-frame composite.

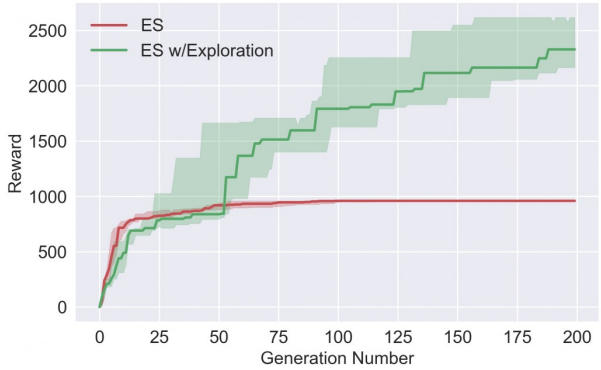
Traditional finite differences (gradient descent) cannot cross a narrow gap of low fitness while ES easily crosses it to find higher fitness on the other side.

ES stalls as a path of high fitness narrows, while traditional finite differences (gradient descent) traverses the same path with no problem, illustrating along with the previous video the distinction and trade-offs between the two different approaches.

Improving exploration in ES

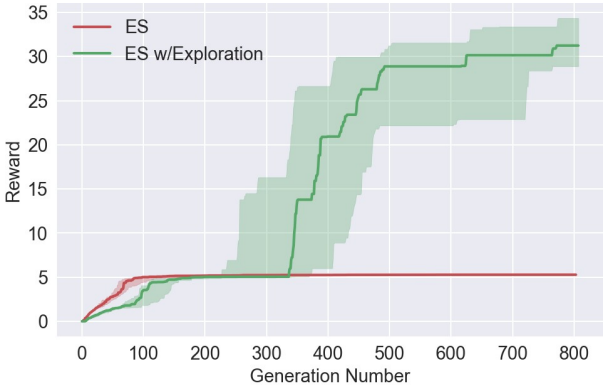
An exciting consequence of deep neuroevolution is that the collection of tools previously developed for neuroevolution now become candidates

for enhancing the training of deep neural networks. We explore this opportunity by introducing new algorithms that combine the optimization power and scalability of ES with methods unique to neuroevolution that promote exploration in RL domains via a *populations* of agents incentivized to act differently from one another. Such population-based exploration is different from the single-agent tradition in RL, including recent work on exploration in deep RL. Our experiments reveal that adding this new style of exploration improves the performance of ES in many domains that require exploration to avoid deceptive local optima, including some Atari games and a humanoid locomotion task in the Mujoco simulator.



ES ES w/ Exploration Reward During Training

With our hyperparameters, ES converges quickly to a local optimum of not coming up for oxygen because doing so temporarily foregoes earning reward. With exploration, however, the agent learns to come up for oxygen and thus accrue much higher rewards in the future. Note that Salimans et al. 2017 do not report ES, with their hyperparameters, encountering this particular local optima, but the point that ES without exploration can get stuck indefinitely on some local optima (and that exploration helps it get unstuck) is a general one, as our paper shows.



ES ES w/ Exploration Reward During Training

The agent is tasked with running as far forward as it can. ES never learns to escape the deceptive trap. With a pressure to explore,

however, one of the agents learns to navigate around the trap.

Conclusions

For neuroevolution researchers interested in moving towards deep networks there are several important considerations: first, these kinds of experiments require more computation than in the past; for the experiments in these new papers, we often used hundreds or even thousands of simultaneous CPUs per run. However, the hunger for more CPUs or GPUs should not be viewed as a liability; in the long run, the simplicity of scaling evolution to massively parallel computing centers means that neuroevolution is perhaps best poised to take advantage of the world that is coming.

The new results are so different from what was previously observed in lower-dimensional neuroevolution that they effectively overturn years of intuitions, in particular on the implications of search in high dimensions. As has been discovered in deep learning, above some threshold of complexity, it appears that search actually becomes easier in high dimensions in the sense that it is less susceptible to local optima. While the field of deep learning is familiar with this way of thinking, its implications are only beginning to be digested in neuroevolution.

The reemergence of neuroevolution is yet another example that old algorithms combined with modern amounts of computing can work surprisingly well. The viability of neuroevolution is interesting because the many techniques that have been developed in the neuroevolution community immediately become available at DNN-scale, each offering different tools to solve challenging problems. Moreover, as our papers show, neuroevolution searches differently than SGD, and thus offers an interesting alternative approach to have in the machine learning toolbox. We wonder whether deep neuroevolution will experience a renaissance just as deep learning has. If so, 2017 may mark the beginning of the era, and we are excited to see what will unfold in the years to come!

Here are the five papers we are releasing today, along with a summary of their key findings:

- Deep Neuroevolution: Genetic Algorithms are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning
- Evolves DNNs with a simple, traditional, population-based genetic algorithm that performs well on hard deep RL problems. On Atari, the GA performs as well as evolution strategies and deep reinforcement learning algorithms based on Q-learning (DQN) and policy gradients (A3C).
- The "Deep GA" successfully evolves networks with over four million free parameters, the largest neural networks ever evolved with a traditional evolutionary algorithm.
- Suggests intriguingly that in some cases following the gradient is not the best choice for optimizing performance.
- Combines DNNs with Novelty Search, an exploration algorithm designed for tasks with deceptive or sparse reward functions, to solve a deceptive, high-dimensional problem on which reward-maximizing algorithms (e.g. GA and ES) fail.
- Shows the Deep GA parallelizes better than, and is thus faster than, ES, A3C, and DQN, and enables a state-of-the-art compact encoding technique that can represent million-parameter DNNs in thousands of bytes.
- Includes results for random search on Atari. Surprisingly, on some games random search substantially outperforms DQN, A3C, and ES, although it never outperforms the GA.

00:26 -00:09 00:35

Copy and paste this HTML code into your webpage to embed.

*space*play / pause

*q*unload | stop

*f*ullscreen

shift + ←→slower / faster

←→seek

. seek to previous

Surprisingly, a DNN found through random search plays Frostbite well, outperforming DQN, A3C, and ES, but not the GA.

- Safe Mutations for Deep and Recurrent Neural Networks through Output Gradients
- *Safe mutations through gradients* (SM-G) greatly improves the efficacy of mutation in large, deep, and recurrent networks by measuring the sensitivity of the network to changes in particular connection weights.
- Computes gradients of *outputs* with respect to weights instead of gradients of error or loss as in conventional deep learning, allowing random, but safe, exploratory steps.
- Both types of safe mutation require no additional trials or rollouts in the domain.
- The result: deep networks (over 100 layers) and large recurrent networks now evolving effectively only through variants of SM-G.
- On the Relationship Between the OpenAI Evolution Strategy and Stochastic Gradient Descent
- Explores the relationship between ES and SGD by comparing the approximate gradient computed by ES with the exact gradient computed by SGD in MNIST under different conditions.
- Develops fast proxies that predict ES expected performance with different population sizes.
- Introduces and demonstrates different ways to speed up and improve the performance of ES.
- *Limited perturbation ES* significantly speeds up execution on parallel infrastructure.
- *No-mini-batch ES* improves the gradient estimate by replacing the mini-batch convention designed for SGD with a different approach customized for ES: A random subset of the whole training batch is

assigned to each member of the ES population within each iteration of the algorithm. This ES-specific approach provides better accuracy for ES with equivalent computation, and the learning curve is much smoother than even for SGD.

- No-mini-batch ES reached 99 percent accuracy in a test run, the best reported performance of an evolutionary approach in this supervised learning task.
- Overall helps to show why ES would be able to compete in RL, where gradient information obtained through trials in the domain is less informative with respect to the performance objective than in supervised learning.
- ES Is More Than Just a Traditional Finite Difference Approximator
- Highlights an important distinction between ES and traditional finite differences methods, which is that ES optimizes for an optimal distribution of solutions (as opposed to one optimal solution).
- One interesting consequence: Solutions found by ES tend to be robust to parameter perturbation. For example, we show that Humanoid Walking solutions from ES are significantly more robust to parameter perturbation than similar solutions found by GAs and by TRPO.
- Another important consequence: ES is expected to solve some problems where conventional methods would become trapped, and vice versa. Simple examples illustrate these different dynamics between ES and conventional gradient-following.
- Improving Exploration in Evolution Strategies for Deep Reinforcement Learning via a Population of Novelty-Seeking Agents
- Adds the ability to encourage deep exploration in ES.
- Shows that algorithms that have been invented to promote exploration in small-scale evolved neural networks via populations of exploring agents — specifically novelty search (NS) and quality diversity (QD) algorithms — can be hybridized with ES to improve its performance on sparse or deceptive deep RL tasks, while retaining scalability.
- Confirms that the resultant new algorithms, NS-ES and a version of QD-ES called NSR-ES, avoid local optima encountered by ES to achieve higher performance on tasks ranging from simulated robots learning to

walk around a deceptive trap to the high-dimensional pixel-to-action task of playing Atari games.

- Adds this new family of population-based exploration algorithms to the deep RL toolbox.

To be notified of future Uber AI Labs blog posts, please sign up for our mailing list, or you can subscribe to the Uber AI Labs YouTube channel. If you are interested in joining Uber AI Labs, please apply at Uber.ai.

Header Image Credit: Eric Frank

- Categories: Uber Data /
- Tags: AI, AI Labs, Already-Emerging Realization, artificial intelligence, Atari, Deep Neuroevolution, DNNS, ES, Evolution, Finite Difference, Jeff Clune, Kenneth Stanley, Machine Learning, ML, MNIST, Mujoco, Neuroevolution, Novelty Search, OpenAI, Peter Dayan, Policy Gradients, Q-Learning, Reinforcement Learning, SGD, SM-R, Stochastic Gradient Descent, TRPO, Uber, Uber AI Labs, Uber Engineering