Does Kinematic-Based Pretraining Improve Evolution of Quadrupedal Gaits?

Kevin J. Ayala Ahumada,[1] Jared M. Moore,[2] and Anthony J. Clark[1]

[1]Pomona College, Claremont, CA, USA
[2]Grand Valley State University, Allendale, MI, USA
anthony.clark@pomona.edu

Abstract

Neural networks are often chosen as controllers in evolutionary robotics. In all but a few cases, neural networks are evolved from scratch. In this study, we investigate the effect of pretraining neural networks using a biologically inspired walking gait. We first generate joint angles for a walking gait using an inverse kinematics model. We then train a conventional feed-forward neural network to reproduce these joint angles. The pretrained model is used to seed an initial population of neural networks, which are coevolved along with the morphology of a quadrupedal robot using Lexicase selection. Our initial results show that while pretraining does not necessarily lead to higher fitness at the end of evolution, it does lead to more consistent performance and more lifelike final behaviors. This exploration has left us with many questions about the importance and process of pretraining in evolutionary robotics, and we believe our results suggest the technique is worth further investigation.

Introduction and Related Work

Neural networks (NNs) are a natural choice for controllers in evolutionary robotics. We can evolve small NNs with few parameters to exhibit simple behaviors such as those demonstrated by Braitenberg vehicles (Braitenberg, 1986), but we can also evolve the architecture and connectivity of networks to produce more complex behaviors such as object grasping (Huang et al., 2014). Most evolutionary robotics studies using NNs do so because of their open-ended nature—giving more control over to evolution can often lead to unexpected and interesting behaviors (Lehman et al., 2020).

Unsurprisingly, NN controllers are almost always evolved from scratch, which encourages evolution of novel behaviors. However, there are some advantages to seeding evolution with a pretrained neural network (Moore and Clark, 2021). For example, pretraining can help evolution by reducing the number of generations required to reach a certain fitness level.

In this study, we investigate the effect of pretraining on evolution. Specifically, we simulate both the kinematics and dynamics of a quadrupedal robot (see Figure 1), pretrain NN controllers, and then continue optimizing NNs along with a quadruped's morphology using Lexicase selection (Helmuth et al., 2015).

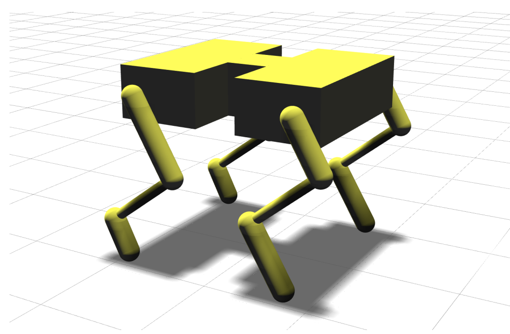

Figure 1: A four-legged quadrupedal animat with 24 degrees-of-freedom. The animat has four legs, each leg has three joints (a hip, knee, and ankle), and each joint is a two-degree-of-freedom universal joint. The initial morphology and gait were designed to match the walking pattern of a large breed canine (although the animat is wider to improve stability).

We pretrain NNs using supervised learning (i.e., minibatch stochastic gradient descent) and a dataset of joint angles generated by an inverse kinematics (IK) model. Our IK system, implemented using the cyclic coordinate descent (CCD) IK algorithm (Wang and Chen, 1991), generates joint angles specifically for a quadrupedal animat. Pretraining details are described in Section Kinematics and Pretraining and using the pretrained network as a seed in evolution is described in Section Gait Evolution. Code and data are available at `https://github.com/anthonyjclark/gaitpt`.

Our preliminary results indicate that pretraining leads to more consistent performance during evolution. These results also point to pretrained networks leading to more "lifelike" gaits, where the animat was more likely to take a higher step, alternate leg movement, and keep its body raised. We suggest that these smoother gaits are likely easier to transfer from simulation to reality.

Kinematics and Pretraining

NN inputs include 24 joint angle sensor measurements corresponding to the leg joints (see Figure 1), four touch sensors (one attached to each foot), and a pure sinusoid (which helps drive an oscillating pattern) for a total of 29 inputs. The NN outputs 24 new joint angles, which are forwarded to the animat's motor controller.

The kinematic gait pretraining process includes **(1)** designing a gait (e.g., walking, trotting), **(2)** stepping the gait forward in time while computing joint angles using inverse kinematics, and then **(3)** training a neural network to reproduce the joint angles.

Each step involves a number of design choices. In step (1), we chose a stable quadrupedal walking gait for generating joint angles (Datt and Fletcher, 2012). While computing joint angles in step (2), for the sake of simplicity, each leg follows a simple "robotic" path wherein the foot ramps up and ramps down using the same number of time steps—instead of a more lifelike path where the foot lifts and thrusts asymmetrically. The kinematic model starts standing on all four legs in a neutral position (as shown in Figure 1) and then initiates a cyclic walking gait that repeats for a specified number of cycles. We can also configure the number of repeated cycles, which provides a way to adjust the balance between the number of training examples corresponding to the initial transient phase (transitioning from standing on all fours to walking) and the steady-state gait.

Finally, in step (3) we can configure the training hyperparameters (e.g., learning rate, batch size, number of epochs, data augmentations) and the NN architecture (e.g., number of hidden layers, number of neurons per layer, activation functions). We found that the most important hyperparameter for improving the transfer from pretraining to simulation is the noise added to the training data. Joint angles are augmented each epoch by adding a random value sampled from a normal distribution (with a configurable magnitude) and touch sensors are randomly flipped. Adding noise to the training dataset helps the NN *adapt* from the pure kinematic training process to the physically constrained (e.g., motor torque limits) and imperfect simulation environment (Tobin et al., 2017). NNs are trained using PyTorch.

Gait Evolution

The Lexicase selection algorithm is seeded with the pretrained NNs and the morphology used during the inverse kinematics pretraining stage. Lexicase selection is a many-objective optimization algorithm (Helmuth et al., 2015); in this study, we maximize distance traveled, minimize power consumption, and optimize behaviors for stability and fewer leg direction switches. These objectives were chosen to encourage fast and stable gaits. Evolution will search for the best NN parameters (updating the pretrained values) and morphologies (e.g., body and leg segment dimensions, resting angles, maximum joint forces).

Evolutionary experiments are run for 2000 generations with 120 individuals. We evaluate individuals by simulating them for 10 seconds using a physics engine.

Results and Discussion

Here we compare two treatments, evolution with and without pretraining. Both treatments were replicated 10 times with different random seeds. In Figure 2, we plot the distance traveled by the best individual in each generation for each replicate run.

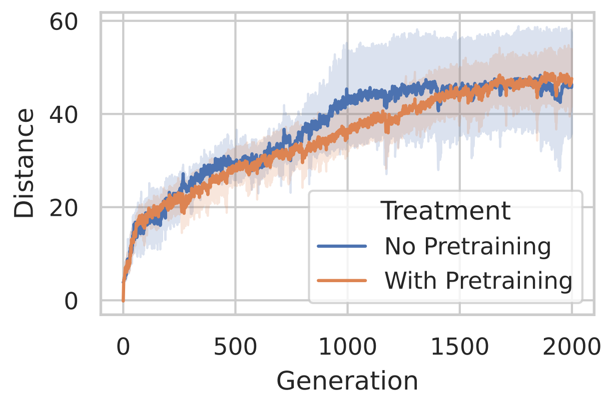

Figure 2: Distanced traveled by the best individual in each generation. Shaded regions indicate the 95% confidence interval of the mean for the 10 replicate runs of each treatment.

Treatments result in similar distances traveled, but the pretraining treatment included the overall farthest traveling individual, a higher minimum distanced traveled by the best individual in each replicate, and a higher mean distanced traveled by the best individuals from each replicate. The pretraining treatment also had a smaller standard deviation in the best individuals, indicating more consistent performance. This consistency can be seen in Figure 2 where the pretraining treatment shows a more continual increase in distance traveled over time, and a tighter confidence interval. These relative performance gains were also seen when changing the pretraining process (e.g., changing the amount of noise or the number of gait cycles in the training dataset). Visualized gaits also appear smoother for the pretraining treatment, though we do not include a metric for this quality.

Future Directions. We are continuing this work and exploring the following directions: different gaits (i.e., amble, trot, canter, gallop), more biologically consistent gaits with asymmetric lift and thrust, incorporating pretraining in the evolution loop, exploring different neural network architectures (e.g., recurrent and convolutional networks), and adding NN outputs for controlling motion of the spine.

Acknowledgements

Thanks to Pomona College and Grand Valley State University for supporting this research.

References

Braitenberg, V. (1986). *Vehicles: experiments in synthetic psychology*. Bradford book psychology. MIT Press, 9. print edition.

Datt, V. L. and Fletcher, T. F. (2012). Gaits: Gait footfall patterns.

Helmuth, T., Spector, L., and Matheson, J. (2015). Solving uncompromising problems with lexicase selection. *IEEE Transactions on Evolutionary Computation*, 19(5):630–643.

Huang, P.-C., Lehman, J., Mok, A. K., Miikkulainen, R., and Sentis, L. (2014). Grasping novel objects with a dexterous robotic hand through neuroevolution. In *2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, pages 1–8. IEEE.

Lehman, J., Clune, J., Misevic, D., Adami, C., Altenberg, L., Beaulieu, J., Bentley, P. J., Bernard, S., Beslon, G., Bryson, D. M., Cheney, N., Chrabaszcz, P., Cully, A., Doncieux, S., Dyer, F. C., Ellefsen, K. O., Feldt, R., Fischer, S., Forrest, S., Fŕenoy, A., Gagńe, C., Le Goff, L., Grabowski, L. M., Hodjat, B., Hutter, F., Keller, L., Knibbe, C., Krcah, P., Lenski, R. E., Lipson, H., MacCurdy, R., Maestre, C., Miikkulainen, R., Mitri, S., Moriarty, D. E., Mouret, J.-B., Nguyen, A., Ofria, C., Parizeau, M., Parsons, D., Pennock, R. T., Punch, W. F., Ray, T. S., Schoenauer, M., Schulte, E., Sims, K., Stanley, K. O., Taddei, F., Tarapore, D., Thibault, S., Watson, R., Weimer, W., and Yosinski, J. (2020). The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *Artificial Life*, 26(2):274–306.

Moore, J. M. and Clark, A. J. (2021). Supervision and evolution: Pretraining neural networks for quadrupedal locomotion. In *The 2021 Conference on Artificial Life*. MIT Press.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE.

Wang, L.-C. and Chen, C. (1991). A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):489–499.