UT Austin Villa: High Dimensional Parameter Optimization for Kicking

Patrick MacAlpine and Peter Stone

Department of Computer Science, The University of Texas at Austin
{patmac,pstone}@cs.utexas.edu

This presentation discusses work by the UT Austin Villa team toward learning robot skills—specifically kicks—which are represented by high dimensional parameterized policies containing 100s or even 1000s of parameters. The goal of this work is to find ways to optimize as many parameters of a fixed duration kick motion policy as possible—and ideally all possible parameters—so as to perform optimization across the largest set of potential kick motion policies.

Currently the UT Austin Villa team specifies kicking motions through a periodic state machine with multiple key frames, where a key frame is a parameterized static pose of fixed joint positions. Figure 1 shows an example series of poses for a kicking motion. While some joint positions are specified by hand, a subset of values for joint positions are optimized using the CMA-ES algorithm and overlapping layered learning [1] methodologies.

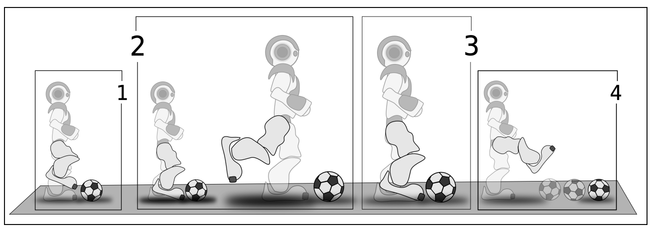

Figure 1. Example of a fixed series of poses that make up a kicking motion.

The UT Austin Villa team has noticed a couple trends when optimizing parameter values for kicks: policies with more parameters allow for longer kicks, and policies with more parameters allow kicks to be executed faster—an important consideration when opponents attempt to stop kicks from being executed. As adding more parameters to a policy increases the space of policies that can be represented, it is not surprising that policies with more parameters have allowed for kicks that can travel farther and be executed faster.

Ideally we might like to learn a joint position (parameter value) for each of the agents 22 joints at every simulation cycle (20 ms) so as to learn a policy over the entire range of possible poses. To optimize values for every joint position at every simulation cycle during a two second kicking motion (the maximum duration of our kicks) would require learning over 2000 parameter values, and unfortunately CMA-ES does not scale well to thousands of parameters. To test the scalability of CMA-ES we first optimized ≈260 parameters for a kick consisting of learned parameters for every joint over 12 simulation cycles. This kick takes less than 0.25 seconds to execute, travels close to 20 meters in distance, and provides a substantial increase in the team's performance. Next, we tried to optimize values for every joint position across 24 simulation cycles (over 500 parameters), but were unable to learn a good kick likely due to CMA-ES not scaling well to that many parameters.

An alternative to representing kicks as a series of fixed pose motions is to represent the policy of a kicking motion as a deep neural network, and then use deep learning to learn kicking motions. In ongoing work we are passing as input to a neural network the amount of time since a kicking motion has started, and then have an output for each of the robot's joints specifying the target joint angle position to move the joint to. Before training the neural network, we seed the network with the policy of our longest kick using supervised learning and backprop. Then, with the network initialized with a policy that mimics our longest kick's motion, we train the neural network with the Trust Region Policy Optimization (TRPO) algorithm. Preliminary results on using deep nets to represent a kicking motion—which enable learning of all joints angles at every simulation cycle—are promising.

The presentation will include videos of both a fast kick and initial efforts toward developing kicks using deep learning.

1. P. MacAlpine, M. Depinet, and P. Stone. UT Austin Villa 2014: RoboCup 3D simulation league champion via overlapping layered learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, January 2015.