Walk To Run: Teaching An Intelligent Agent To Play QWOP

Written by Sonny Smith, Matthew Oros, Michael Terekhov
275 Eastland Road
Berea, Ohio 44017

Abstract

We have decided to recreate the game QWOP and create an AI that can sufficiently play the game. In QWOP, a player attempts to control a track runner's thighs and legs using Q and W and calves using O and P to make them run without falling over. Our goal is to create an AI that can play this game and improve at it to eventually be able to travel 100m. We intend to use a combination of approaches that includes a genetic algorithm, a neural network, and reinforcement learning to train the AI to improve its gameplay. Our plan is to develop the AI using milestones such as crawling, standing, walking, etc. We intend to further expand this plan with more specific algorithms, milestones, etc.

The core issue that this program intends to solve is creating a biomechanical model of locomotion. This issue is bioinspired because most land-borne animals must learn to balance and walk on their own. This problem is explored in many fields including engineering, medicine, and robotics. Our goal is to successfully develop a program that can learn to maneuver the character without falling over.

Our approach to the problem is to develop a neural network and use a genetic algorithm and reinforcement learning to train it.

The neural network will be given a set of inputs, which involves the positions of all the limbs in the environment. The goal is to create a deep enough neural network with enough layers so we can train it by merging the best performing networks.

To optimize the weights of the neural network we will use a genetic algorithm. The approach that we will use is population-based. In such an approach, we generate multiple agents with various neural networks, which have randomly assigned weights and biases for the first generation and as they evolve the weights will change based on the successful performances of the networks.

To further improve the agent's performance we will use reinforcement learning. For this portion of the development, we will develop a utility function that would indicate how well the agent performs to make further updates to the neural network's weights based on the rewards received by the agent during run-time. The simplest version of the utility function that we have in mind is to give rewards if the figure moves forward and penalize it for falling or moving backward to update the weights in real-time. We will potentially use a replay buffer to store the outputs of the utility function (agent's experiences) for further improvements in the stability of the learning process.

To determine the effectiveness of the solution, objective quantitative measurements will be taken such as the the distance the figure has traversed. To determine effectiveness of the training solution, there would have to be evidence of an improvement in overall performance in movement from the first generation to multiple later generations. Even if the original goal is not met, any improvements indicates effectiveness and thus some success.

We believe that we will be successful because we have planned out a system of a neural network and genetic algorithm/reinforcement learning to help accomplish our goal. Our neural network has been coded to fully perceive the character's body and movements relative to the environment. This will allow us to determine if the character is balancing and measure its movement. We already have a functional draft of a neural network that can be used and upon which we can expand. We believe that our [genetic algorithm/reinforcement learning] approach will allow our program to learn the best approach to balancing and walking. Due to the genetic algorithm, we would be able to assign starting weights that would have some meaning due to trial and error. Due to reinforcement learning the weights and the actions would be polished (further improved). We believe that such a combination will give us better results compared to the ones that would be given if we were only to use one of the mentioned above algorithms.

If the original goal of having the figure traverse 100 meters is too ambitious for the time constraint given or complexity of the training solution, then smaller goals can be assigned such crawling, balance, or jumping. Even if only these smaller goals are achieved then it would still be evidence of the effectiveness of our learning solution. This would further increase the likelihood of producing a meaningful result even if the original goal is not achieved.