Case Study - Hybrid System

Introduction

A Hybrid system is an intelligent system which is framed by combining atleast two intelligent technologies like Fuzzy Logic, Neural networks, Genetic algorithm, reinforcement Learning, etc. The combination of different techniques in one computational model make these systems possess an extended range of capabilities. These systems are capable of reasoning and learning in an uncertain and imprecise environment. These systems can provide human-like expertise like domain knowledge, adaptation in noisy environment etc.

What is reinforcement learning?

Reinforcement learning is the training of machine learning models to make a sequence of decisions. The agent learns to achieve a goal in an uncertain, potentially complex environment. In reinforcement learning, an artificial intelligence faces a game-like situation. The computer employs trial and error to come up with a solution to the problem. To get the machine to do what the programmer wants, the artificial intelligence gets either rewards or penalties for the actions it performs. Its goal is to maximize the total reward.
Although the designer sets the reward policy–that is, the rules of the game–he gives the model no hints or suggestions for how to solve the game. It’s up to the model to figure out how to perform the task to maximize the reward, starting from totally random trials and finishing with sophisticated tactics and superhuman skills. By leveraging the power of search and many trials, reinforcement learning is currently the most effective way to hint machine’s creativity. In contrast to human beings, artificial intelligence can gather experience from thousands of parallel gameplays if a reinforcement learning algorithm is run on a sufficiently powerful computer infrastructure.

Examples of reinforcement learning.

Applications of reinforcement learning were in the past limited by weak computer infrastructure. However, as Gerard Tesauro’s backgamon AI superplayer developed in 1990’s shows, progress did happen. That early progress is now rapidly changing with powerful new computational technologies opening the way to completely new inspiring applications.
Training the models that control autonomous cars is an excellent example of a potential application of reinforcement learning. In an ideal situation, the

computer should get no instructions on driving the car. The programmer would avoid hard-wiring anything connected with the task and allow the machine to learn from its own errors. In a perfect situation, the only hard-wired element would be the reward function.

For example, in usual circumstances we would require an autonomous vehicle to put safety first, minimize ride time, reduce pollution, offer passengers comfort and obey the rules of law. With an autonomous race car, on the other hand, we would emphasize speed much more than the driver's comfort. The programmer cannot predict everything that could happen on the road. Instead of building lengthy "if-then" instructions, the programmer prepares the reinforcement learning agent to be capable of learning from the system of rewards and penalties. The agent (another name for reinforcement learning algorithms performing the task) gets rewards for reaching specific goals.

Another example: deepsense.ai took part in the "Learning to run" project, which aimed to train a virtual runner from scratch. The runner is an advanced and precise musculoskeletal model designed by the Stanford Neuromuscular Biomechanics Laboratory. Learning the agent how to run is a first step in building a new generation of prosthetic legs, ones that automatically recognize people's walking patterns and tweak themselves to make moving easier and more effective. While it is possible and has been done in Stanford's labs, hard-wiring all the commands and predicting all possible patterns of walking requires a lot of work from highly skilled programmers.

Challenges with reinforcement learning

The main challenge in reinforcement learning lays in preparing the simulation environment, which is highly dependant on the task to be performed. When the model has to go superhuman in Chess, Go or Atari games, preparing the simulation environment is relatively simple. When it comes to building a model capable of driving an autonomous car, building a realistic simulator is crucial before letting the car ride on the street. The model has to figure out how to brake or avoid a collision in a safe environment, where sacrificing even a thousand cars comes at a minimal cost.

Scaling and tweaking the neural network controlling the agent is another challenge. There is no way to communicate with the network other than through the system of rewards and penalties.This in particular may lead to catastrophic forgetting, where acquiring new knowledge causes some of the old to be erased from the network (to read up on this issue, see this paper, published during the International Conference on Machine Learning).

A "jumper" jumping like a kangaroo instead of doing the thing that was expected of it-walking-is a great example, and is also one that can be found in our recent blog post.