

In general it is thought that count based exploration strategies can not work in problems with high dimensional state spaces, since the majority of states will only occur once in total and therefore their probability of occurrence is near zero. In this thesis however, I want to describe a method that allows a robot to efficiently explore its configuration space with the help of a neural network, specifically an autoencoder, and the CMA-ES algorithm. The counting of the state-action pairs (current joint position and velocity plus the desired velocity) is done by the autoencoder that is trained after every episode with all to this point acquired state-action pairs. Each one of those pairs that is forwarded into the autoencoder is compressed to a much lower dimension which now allows their actual counting. The exploration itself is done by the CMA-ES algorithm. In each iteration it proposes several different actions (desired joint velocity) and forwards them as state-action pairs into the autoencoder. For every single pair the compressed representation is retrieved and a density estimation is performed on it. The CMA-ES algorithm will now choose the action that yields the lowest probability. This is repeated until the global optimum has been found and thus the robot will occupy a state that does not occur as often as others states in its direct surrounding. The method allows for a thorough exploration of the robot's configuration space.

Gemeinhin wird angenommen, dass Erkundungsstrategien, die auf dem Zählen der einzelnen Zustände beruhen, in höher-dimensionalen Zustandsräumen nicht funktionieren können, da jeder einzelne Zustand nur insgesamt einmal vorkommt und dessen Auftretenswahrscheinlichkeit demnach nahe null ist. In dieser Arbeit möchte ich jedoch eine Methode vorstellen, die es einem Roboter mit der Hilfe von einem Neuralen Netzwerk, spezifischer einem Autoencoder, und dem CMA-ES-Algorithmus erlaubt, seinen Konfigurationsraum möglichst effizient zu erkunden. Das Zählen der State-Action-Paare (aktuelle Gelenkpositionen und -geschwindigkeiten sowie die gewünschte Gelenkgeschwindigkeit) wird vom Autoencoder vorgenommen, welcher nach jeder Episode mit all den bisher erlangten State-Action-Paaren trainiert wird. Jedes einzelne dieser Paare, das an den Autoencoder weitergeleitet wird, wird in eine viel niedrigere Dimension komprimiert, was das eigentliche Zählen möglich macht. Das Erkunden an sich wird vom CMA-ES-Algorithmus übernommen. In jeder Iteration schlägt dieser mehrere verschiedene Aktionen (gewünschte Gelenkgeschwindigkeiten) vor und leitet sie an den Autoencoder weiter. Für jedes einzelne Paar wird die komprimierte Repräsentation abgerufen und auf ihr eine sogenannte Dichteevaluierung durchgeführt. Der CMA-ES-Algorithmus wird nun die Aktion auswählen, die die geringste Wahrscheinlichkeit hervorgebracht hat. Dies wird solange wiederholt, bis das globale Optimum gefunden wurde und der Roboter aus diesem Grund einen Zustand einnimmt, der nicht so oft auftritt, wie andere Zustände in seiner direkten Nachbarschaft. Diese Methode erlaubt eine gründliche Erkundung des Konfigurationsraumes des Roboters.