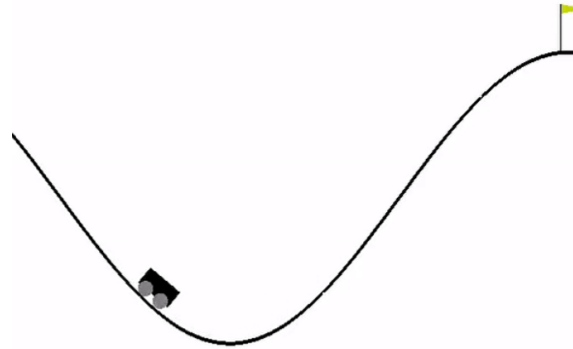


Exercise 9

Cross-Entropy Method (CEM)



This exercise will implement the Cross-Entropy Method (CEM) on the MountainCar environment. To give you some air to breathe, this will only be a mini-exercise for you to see, how powerful CEM can actually be. We will solve the environment by iteratively sampling the neural-network weights of our policy from a Gaussian distribution with fixed σ and fitting our Gaussian towards the mean of our "elite" weights of the current population.

Based on the skeleton we provide to you, you should be able to complete this exercise with around five lines of code.

Programming Tasks: Inside `cem.py`, implement the `cem` method, which requires you to fill in three core steps of the method:

1. Initialize the current Gaussian mean vector by sampling it from a standard normal distribution (e.g., using `numpy.random.randn`).
2. Sample a population of weights from our current distribution and evaluate them inside the MountainCar environment. You can evaluate them using `Agent.evaluate`.
3. Update the distribution towards the elite population.

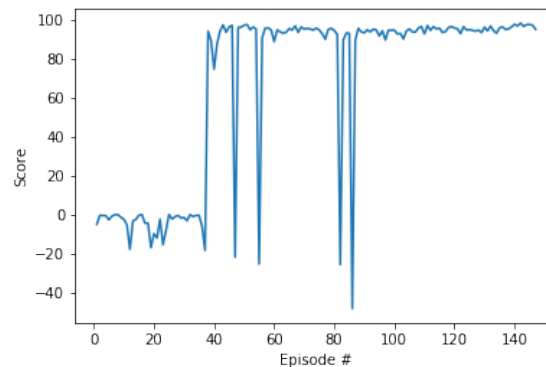


Figure 1: A typical learning curve for CEM on the MountainCar environment.