Dynamics Curriculum Learning for Deep Reinforcement Learning Agents

Briton Park, Danqing Wang, and Xiaoer Hu

Abstract

Existing curriculum learning algorithms for RL provide an agent with a sequence of tasks characterized by different initial state distributions or goals. Here, we present another approach to design different tasks -- by varying state transition dynamics, ranging from “easy” environments in which the dynamics are simplified or tuned to make it easier to reach the goal, to “hard” environments that are closer to the real world. Our experiments on car racing, lunar lander and cart pole tasks show evidence of the effectiveness of the method. For the car racing task, we vary the friction of the road to represent changes in the transition dynamics of the environment and the width of the road to represent changes in the initial state distributions of the environment. There is evidence that curriculum learning leads to improvements when training is sparser for the Q-learning algorithm based on a convolutional neural network. For the lunar landing task, we designed dynamics curriculums by varying the main and side engine power. Within a certain extent, higher engine power will help learning, while extreme high engine power will make the lander overly sensitive and impede learning. Our dynamic curriculum learning agents where the agent learns with moderate high engine power first show evidence of faster learning than baseline. For the cart pole task, we varied the magnitude of the gravity and cart mass. Smaller gravity or smaller cart mass can both help the system to achieve high reward much faster and more stable. Our experiment results show that with a dynamic curriculum, the agents can learn better.

Introduction

Teachers typically employ curriculums in order to effectively relay complex concepts and ideas. For example, high school mathematics is usually taught following some specified order reflecting an increasing level of complexity, such as starting with algebra, then trigonometry, and finally calculus. Employing a curriculum is advantageous, because students are able to exploit previously learned concepts to better understand new ideas.

Our work applies a similar notion of curriculum learning to the training of deep reinforcement learning agents. Overall, we find some evidence for the utility of curriculum in facilitating learning across three reinforcement learning tasks.

Related Work

Curriculum learning has previously been employed in machine learning to facilitate model training convergence and improve the final model quality. The idea of training machines through curriculum learning was first introduced by Elman 1993 [1]. The idea was to start with easier subtasks and gradually increase their difficulty for learning simple language grammar. An interesting finding from the study was that without curriculum learning, the model could not learn at all. More recently, Weinshall 2018 applied the concept of curriculum learning for convolutional neural networks by sorting the training set based on the performance of a pre-trained network on a larger dataset and found improvements on both the convergence speed and final accuracy [2].

Previous works have also explored applying the concept of curriculum learning to train reinforcement learning agents. Providing agents with a sequence of tasks characterized by different state initializations or goals improved the speed of convergence and quality of the final solution [3,4]. Another way curriculum learning has been employed in this field is through the teacher-student curriculum learning (TSCL) framework [5]. In TSCL, the teacher, a policy for selecting tasks, guides the student's training process by selecting proper subtasks. This task selection helps the student learn tasks which facilitates the students learning process or at risk of being forgotten by the student.

Background

We explore two different approaches to curriculum learning. The first approach entails providing an agent with different initial state distributions, and the second approach entails presenting the agent with a sequence of environments characterized by different state transition dynamics, ranging from "easy" dynamics to "hard" dynamics. Our curriculum learning methods are evaluated across three reinforcement learning tasks: car racing, lunar landing, and cart pole.

Tasks

Car racing

For the first task, we went with the car racing task from the OpenAI gym (https://gym.openai.com/envs/CarRacing-v0/). The goal of the task is to train a car racing agent to race around a track quickly. The agent must learn to map to individual actions like braking, accelerating, and steering left or right from a particular state, which consists of 96x96 pixels. The rewards consist of -0.1 every frame and +1000/N for every track tile visited where N is the total number of tiles in the track. Each episode consists of one run across the track. The episodes terminate early if the reward is negative 10 times in a row.

The difficulty in the task is properly controlling the car on the track. It is easy for the car to spin out of control due to its speed and the friction of the road. Thus, the agent must learn to stay on the road to collect rewards and slow down before sharp turns to avoid spinning out of control, but at the same time, try to finish the track quickly.

Lunar lander

The second task is the continuous version of Lunar Lander from the OpenAI gym (https://gym.openai.com/envs/LunarLanderContinuous-v2/). The task aims to land the craft from the top of the screen to landing pad. Depending on the operation taken and position it landed, the craft will gain different rewards. Reward for moving from the top of the screen to the landing pad and zero speed is about 100 to 140 points. If the lander moves away from the landing pad, it loses reward. The episode finishes if the lander crashes or comes to rest, receiving an additional -100 or +100 points. Each leg with ground contact is +10 points.

Cart Pole

Cart Pole from the OpenAI gym (https://gym.openai.com/envs/CartPole-v1/) is our third task. In this task, a pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. This system is controlled by applying a force of left or right to the cart. The pendulum starts upright, and the goal is to prevent it from falling over. For every time step, if the pole remains upright, a reward of +1 will be provided. The episode ends when the pole is more than 15 degrees from vertical, or the cart moves more than 2.4 units from the center, or the episode length is more than 200.

Methods

Car racing

Architecture

For the car racing RL agent architecture, we went with a Q learning algorithm based on a convolutional neural network. The convolutional neural network consists of two sequences of convolution layers and max pooling layers followed by a dense layer and a final output softmax layer. Relu activation functions are used for the convolution and dense layers. The optimization function is the Adam algorithm and the loss is based on the mean squared error.

Curriculum learning

We came up with two ways to apply curriculum learning for this task. They consist of changing the dynamics of the environment and changing the initial condition of the environment.

The first way is changing the friction of the road, which is a change in the dynamics of the environment. Three friction modes were set for the task. The first is the original set up of the road, which is the most difficult version. The second is increasing the friction of the original road by 50% (medium version), and the third is doubling the friction of the road (easy version). Increasing the friction of the road makes the task easier, since the racing car is easier to control and less likely to spin out of control.

The second way is changing the width of the road. This is related to changing the initial condition of the environment. Three road widths were set for the task. The first is the original width of the road; the second is increasing the width of the road by 50% (medium version), and the last is doubling the width of the original road (easy). Increasing the width of the road makes the task easier, since it's easier to stay on the road and collect rewards if it is wider.

Lunar lander

Architecture

We utilized a policy gradient method for the RL agent based on a convolutional neural network that is of the same architecture as the previous task. We further added reward-to-go and discounting to reduce the variance.

Curriculum learning

To study the dynamics curriculum learning, we focused on the engine power (MAIN ENGINE POWER and SIDE ENGINE POWER) of the lander. To a certain extent, higher engine power gives the player a larger control over the agent and makes it easier to

land well. However, too large engine power would make the lander overly sensitive and result in difficulty in learning. Here, we designed a series of curriculum with varying engine power for better learning.

Cart Pole

Architecture

A policy gradient method for the RL agent based on a convolutional neural network that is of the same architecture as the previous two tasks is applied in the cart pole. Besides, reward-to-go and discounting are also added to reduce the variance.

Curriculum learning

For the cart pole environment, we focused on the magnitude of the gravity that the system has, as well as the mass of the cart. Smaller gravity gives the cart a better control over the agent and makes it easier to avoid falling over. Similarly, the mass of the cart is also critical. If the mass of the cart is smaller, the system is easier to control. In this work, a series of curriculum with varying the gravity magnitude and the mass of the cart is applied for better learning.

Experiments

Car racing

We evaluated the total rewards of 6 different RL agents. The details of each agent is described below. The agents are evaluated on 5 tasks: the unmodified car racing task, easy road width task, medium road width task, easy friction task, and medium friction task. The evaluation is averaged across 5 random runs on each task.

Table 1: Description of RL agents for car racing

Agent name	Description
Original_600	RL agent trained on 600 episodes of the unmodified car racing task
Original_300	RL agent trained on 300 episodes of the unmodified car racing task
Width_600	RL agent trained on 200 episodes of the easy road width version, 200 episodes on the medium road width version, and 200

	episodes on the unmodified car racing task
Width_300	RL agent trained on 100 episodes of the easy road width task, 100 episodes on the medium road width task, and 100 episodes on the unmodified car racing task
Friction_600	RL agent trained on 200 episodes of the easy road friction version, 200 episodes on the medium road friction version, and 200 episodes on the unmodified car racing task
Friction_300	RL agent trained on 100 episodes of the easy road friction version, 100 episodes on the medium road friction version, and 100 episodes on the unmodified car racing task

Lunar lander

We evaluated the total rewards of 7 different RL agents. The details of each agent is described below. The evaluation is averaged across 3 random runs.

Table 2: Description of RL agents for lunar lander

Agent name	Description
Original_100	RL agent trained on 100 episodes of the unmodified lunar lander task, that is 13 for main, 0.6 for side
High_100	RL agent trained on 100 episodes of the lunar lander task with high engine power: 50 for main, 10 for side
2nd_High_100	RL agent trained on 100 episodes of the lunar lander task with 2nd highest engine power: 25 for main, 5 for side

3rd_High_100	RL agent trained on 100 episodes of the lunar lander task with 3rd highest engine power: 13 for main, 2.5 for side
Extreme_100	RL agent trained on 100 episodes of the lunar lander task with extremely high engine power: 500 for main, 50 for side
C1_100	RL agent trained on 20 episodes of the High_100 version, 20 episodes on the 2nd_High_100 version, 20 episodes on the 3rd_High_100 version and 40 episodes on the original task
C2_100	RL agent trained on 30 episodes of the High_100 version and 70 episodes on the original task

Cart Pole

We evaluated the total rewards of 8 different RL agents. The details of each agent is described below. The agents are evaluated on 7 tasks: the unmodified cart pole task, smaller gravity task, slightly smaller gravity task, larger gravity task, smaller cart mass task, and larger cart mass task. The evaluation is averaged across 3 random runs on each task.

Table 3: Description of RL agents for cart pole

Agent name	Description
Original_100	RL agent trained on 100 episodes of the unmodified cart pole task
GravityMoon_100	RL agent trained on 100 episodes of the cart pole task with smaller gravity applied (the gravity on the Moon = 1.6)
GravityMars_100	RL agent trained on 100 episodes of the cart pole task with smaller gravity applied (the gravity on the Mars = 3.7)
GravityJupiter_100	RL agent trained on 100 episodes of the

	cart pole task with larger gravity applied (the gravity on the Jupiter = 25)
MassCart0.9_100	RL agent trained on 100 episodes of the cart pole task with smaller cart mass (0.9)
MassCart1.5_100	RL agent trained on 100 episodes of the cart pole task with larger cart mass (1.5)
C1_100	RL agent trained on 50 episodes of the GravityMoon_100 version, and 50 episodes on the original task
C2_100	RL agent trained on 30 episodes of the MassCart0.9 version and 70 episodes on the original task

Results

Car racing

The results for each RL agent across the variations of the car racing tasks are detailed in the table below.

Table 4: Total rewards for each RL agent across car racing tasks, averaged across 5 runs

Experiment	Original task	Friction_easy	Friction_medium	Width_easy	Width_medium
Original_600	**866**	1008	1040	**1010**	**908**
Width_600	640	980	**1124**	577	389
Friction_600	784	**1022**	1036	44	89
Original_300	215	344	247	248	364
Width_300	269	388	298	147	270
Friction_300	**398**	**446**	**534**	**552**	**524**

Amongst RL agents trained on 600 episodes total, the Original_600 agent performs the best on the original task and across many of the variations of the task as well. From

these agent results, it does not seem like applying curriculum learning in this fashion led to better agents.

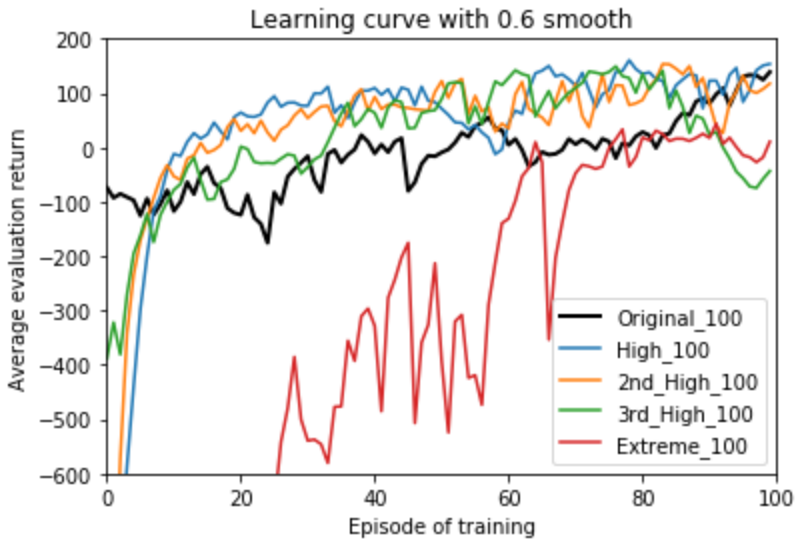
However, amongst RL agents trained on 300 episodes total, the Friction_300 agent performs the best across all the variations of the task, including the unmodified task. The difference in rewards compared to the Original_300 agent is substantial, more than 100 points in each task. Interestingly Width_300 performs better than Original_300 on the original task and the modified road friction tasks but worse on the modified width tasks by about 100 points.

There is some evidence of the utility of curriculum learning applied in this way, because friction_300 performs the best out of the agents trained on 300 episodes. Most of the performance gains come from agents trained only on 300 episodes total compared to those trained on 600 total. From the experiments done, one potential takeaway is that increasing the training procedure negates the benefits of curriculum learning to an extent. However, more experiments with RL agents trained on varying numbers of episodes need to be done to better understand how much benefit curriculum learning can give in different circumstances.

Lunar lander

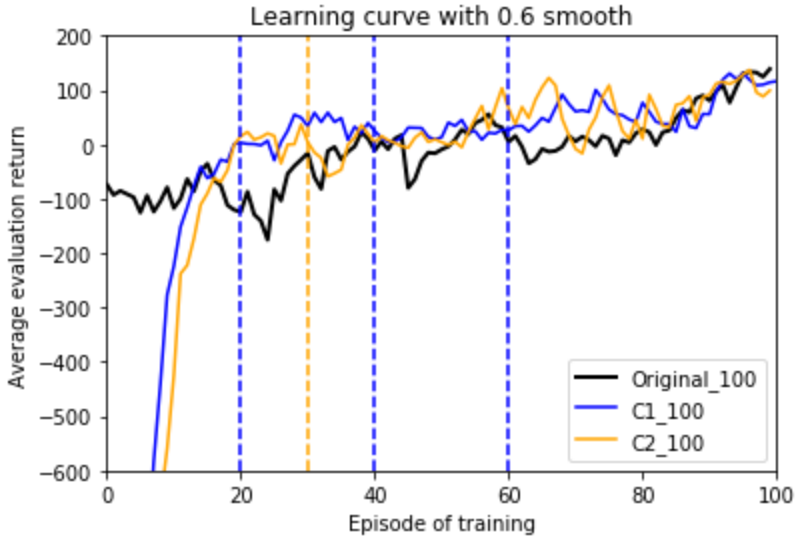
The training curves for RL agents are shown below.

Figure 1: Learning curve for Original_100, High_100, 2nd_High_100, 3rd_High_100 and Extreme_100.



The learning curve shows that within a certain range, higher main and side engine power will help learning, while the extreme power will impede learning. This validates our choice of different engine power as tuning parameters for our curriculum design.

Figure 2: Learning curve for Original_100, C1_100, C2_100. The blue and orange dashed lines mark the change of curriculums for C1_100 and C2_100, respectively.

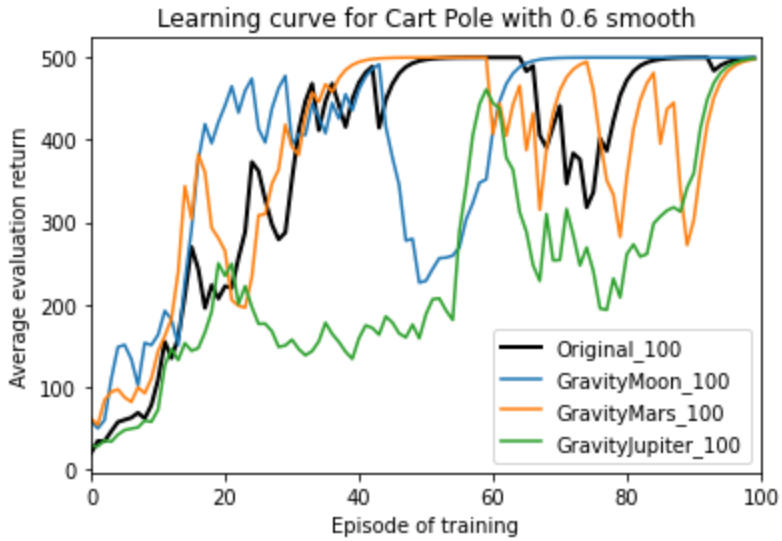


For the agents with curriculum C1 and C2, there's some evidence that with curriculum, the agents learn faster. Further tuning of the curriculum parameters can be studied to further improve the agent's behaviour.

Cart Pole

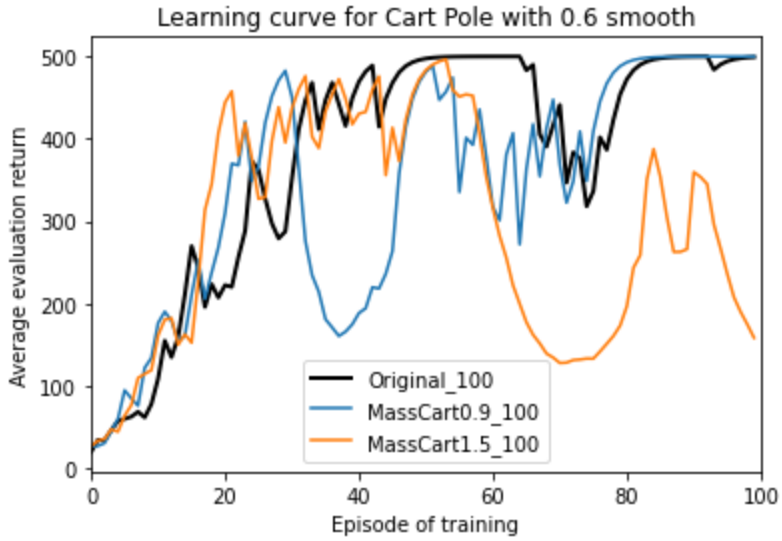
The training curves for RL agents are shown below.

Figure 3: Learning curve for Original_100, GravityMoon_100, GravityMars_100, and GravityJupiter_100.



The learning curve shows that smaller gravity can improve the learning curve to achieve the high value much faster, and also more stable, while the larger gravity will impede the system to learn. This validates our choice of different gravity as tuning parameters for our curriculum design.

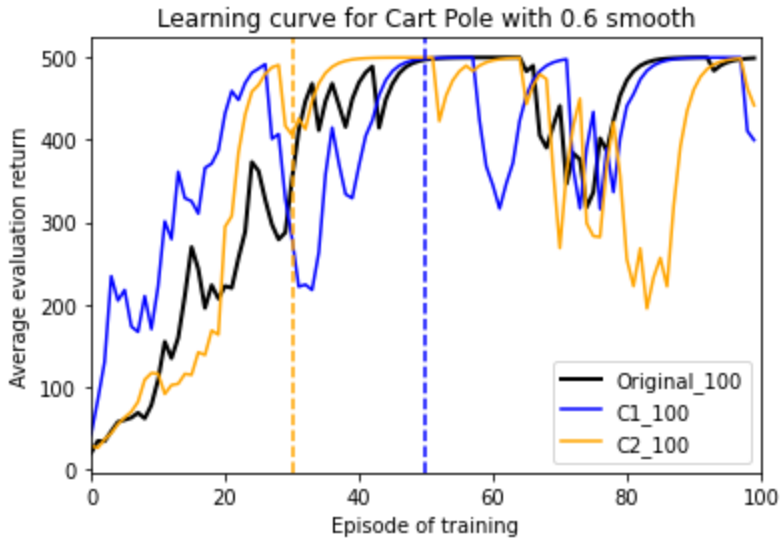
Figure 4: Learning curve for Original_100, MassCart0.9_100, and MassCart1.5_100.



The learning curve shows that smaller mass can improve the learning curve to achieve high rewards much faster, and are also more stable. With a larger cart mass, the

system cannot maintain stability with more episodes. This validates our choice of different cart mass as tuning parameters for our curriculum design.

Figure 5: Learning curve for Original_100, C1_100, C2_100. The blue and orange dashed lines mark the change of curriculums for C1_100 and C2_100, respectively.



For the agents with curriculum C1 and C2, we can clearly see with a dynamic curriculum, the agents can learn faster. Future work can be done to better tune the curriculum parameters to further improve the agent's performance.

Future Work

One natural follow-up to our work is to further tune the curriculum learning parameters such as how many episodes we learn for each curriculum and the difficulty level of the curriculums. Furthermore, it will be interesting to integrate with other curriculum methods such as varying initial conditions and rewards to design more comprehensive curriculums. Automatic curriculum design for these comprehensive curriculums is also a promising direction to go.

Conclusion

In this work, we presented a novel curriculum design for dynamics curriculum learning. We found that it performed modestly better than the baseline on environments such as Car racing, Lunar lander and Cart pole. Varying state dynamic transition can be an effective means for curriculum design.

References

1. Jeffrey L. Elman. “Learning and development in neural networks: The importance of starting small.” Cognition 48.1 (1993): 71-99.
2. Daphna Weinshall, Gad Cohen, and Dan Amir. “Curriculum learning by transfer learning: Theory and experiments with deep networks.” ICML 2018.
3. Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In International Conference on Machine Learning (ICML), 2009.
4. C. Florensa, D. Held, M. Wulfmeier, and P. Abbeel. Reverse curriculum generation for reinforcement learning. arXiv preprint arXiv:1707.05300, 2017.
5. Tambet Matiisen, et al. “Teacher-student curriculum learning.” IEEE Trans. on neural networks and learning systems (2017).