Utilizing Curriculum Learning to Decrease Time to Threshold Performance of Deep Q Learning in a Simulated Highway

- Sean Hulseman
  CS138, Tufts, December 2024

The problem of continuing tasks in RL

Uncertainty in Sensor Data: Self-driving systems rely on sensor data that is often uncertain and prone to errors.

Complex Decision-Making:

- Example: A sensor detects an object signaling a potential collision.
- The AI agent must decide: **Stop or Swerve?**

Risky and Unexpected Outcomes:

- Actions taken in response to uncertain data can lead to unpredictable and potentially dangerous behavior.

Real-World Challenges:

- Reinforcement Learning (RL) must address the complexity of ongoing decision-making in dynamic and uncertain environments.

Problem statement

Primary Goal:

- Develop a custom Deep Q-Network (DQN) to train an RL agent for reliable, safe driving with minimized computational costs.

Key Challenges:

- Actions build on prior successful, non-terminal decisions, increasing training complexity.
- Exploration-Exploitation Tradeoff:
    - Fundamental in RL but leads to **exponential computational costs** in off-policy methods like DQNs.

Secondary Goal:

- Accelerate training using **Curriculum Learning**:
    - Simpler environments simulate aspects of the final goal.
    - A structured regimen automates progression to the goal environment.

Research Focus:

- Balance safety, reliability, and computational efficiency in RL for real-world tasks.

Transfer Learning in Reinforcement Learning

There are several metrics to evaluate value transfer (Taylor, 2009):

1. **Asymptotic Performance**: Performance after convergence in the target task.
2. **Initial Performance**: Starting performance in the target task.
3. **Total Reward**: Accumulated reward during training.
4. **Area Ratio**: Difference in learning curves (transfer vs. non-transfer).
5. **Time-to-Threshold**: Time to reach a predefined performance threshold.
    - **Focus of this Experiment**: Evaluating **Time-to-Threshold** for value transfer effectiveness.

Transfer Learning vs. Multitask Learning (Zhuang et al., 2020):

- **Multitask Learning**:
  - Trains on multiple related tasks simultaneously.
  - Leverages inter-task relevance.
  - Equal attention to all tasks.
- **Transfer Learning**:
  - Transfers knowledge from related domains to a target task.
  - Emphasis on the **target task** over the source task.

Curriculum Learning (CL)

Inspired by Human Learning

- Progression from simpler tasks to more complex ones.
- Improves **efficiency** and **robustness** of machine learning models.

Motivations for CL

1. **Guidance**:
   - Enhances training time and performance, especially for challenging tasks.
   - Structures learning into manageable stages.
   - Example: Self-driving vehicles, where direct training fails due to high dimensionality and sparse rewards.
2. **Denoising**:
   - Focuses on cleaner, more relevant data first.
   - Improves robustness and generalizability in noisy or heterogeneous datasets.
   - Example: Neural Machine Translation (NMT), prioritizing simpler translations early in training.

Deep Q-Network (DQN)

1. Experience Replay

- Stores experiences (state, action, reward, new state) in a replay memory of set size
- Benefits:
  - Reduces correlation between consecutive samples.
  - Stabilizes learning by breaking the sequence of states.

2. Stochastic Gradient Descent (SGD)

- Minimizes the difference between predicted and target Q-values.
- Enables approximation of optimal action-value functions in **high-dimensional state spaces**.
- **Key Achievement**:
  - Demonstrated human-level performance in Atari games directly from raw pixel inputs.
  - Highlighted RL and deep learning's potential for complex, high-dimensional tasks.

3. Target Q-Values

$$Q_{\text{target}}(s,a) = r + \gamma \max_{a'} Q_{\text{target}}(s',a')$$

- **Goal**: Estimate the action-value function Q(s,a)Q(s, a)Q(s,a), the expected return from taking action aaa in state sss.
- **Bellman Equation**: Computes the target Q-value for each transition

Highway Env - Customizations

State representation: Kinematics Features Vector (FxN) for each observation. Where F the the number of features an N is the number of cars (the first car in the feature is the ego car, RL agent.

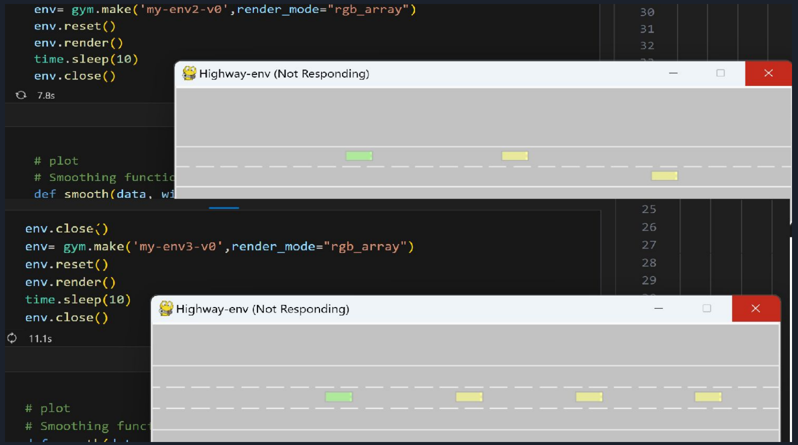
Actions: Discrete Meta Actions

Rewards: Standard pre built rewards from Gymnasium’s Highway Env. Velocity in the target zone is rewarded.

Duration: Standard was 40s. My curriculum envs had 20, 30 seconds for their durations

Lane Count: Standard was 4 lanes. My curriculum envs had 2, then 3 lanes.

Simulation Frequency: set to 5 instead of 15 to decrease the number of computations per second

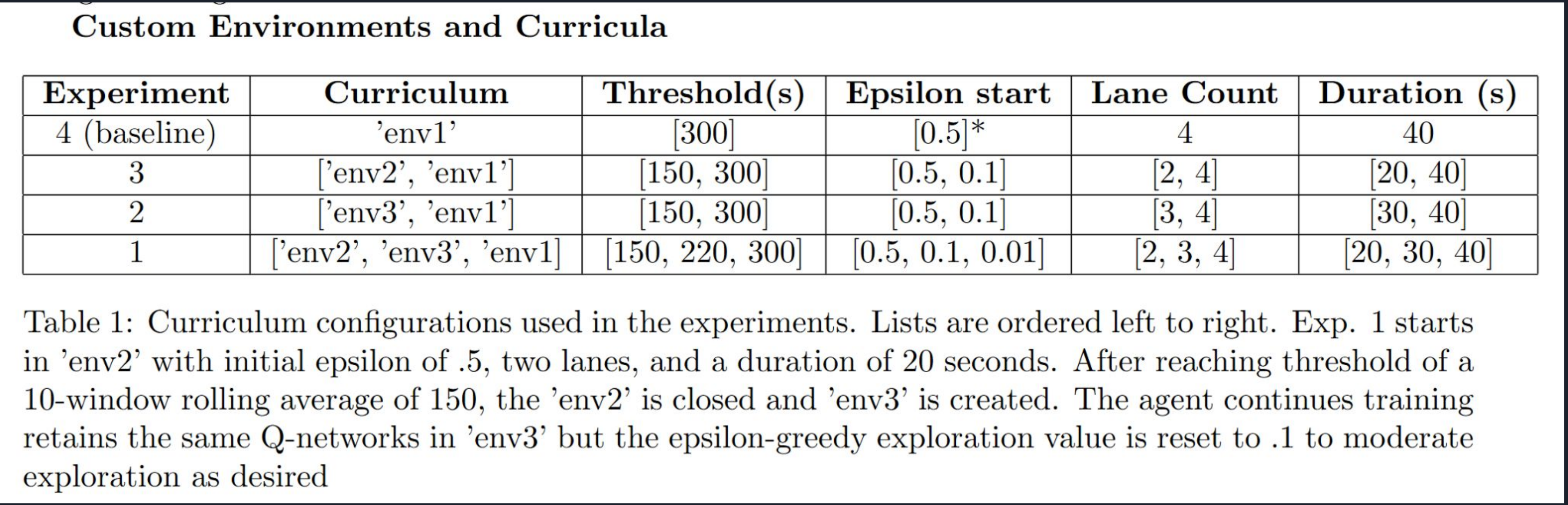


```
env= gym.make('my-env2-v0',render_mode="rgb_array")
env.reset()
env.render()
time.sleep(10)
env.close()
7.8s

# plot
# Smoothing functio
def smooth(data, wi
```

```
env.close()
env= gym.make('my-env3-v0',render_mode="rgb_array")
env.reset()
env.render()
time.sleep(10)
env.close()
11.1s

# plot
# Smoothing funct
```

Test environments

Curricula Assessed



Custom Environments and Curricula

Experiment	Curriculum	Threshold(s)	Epsilon start	Lane Count	Duration (s)
4 (baseline)	’env1’	[300]	[0.5]*	4	40
3	[’env2’, ’env1’]	[150, 300]	[0.5, 0.1]	[2, 4]	[20, 40]
2	[’env3’, ’env1’]	[150, 300]	[0.5, 0.1]	[3, 4]	[30, 40]
1	[’env2’, ’env3’, ’env1]	[150, 220, 300]	[0.5, 0.1, 0.01]	[2, 3, 4]	[20, 30, 40]

Table 1: Curriculum configurations used in the experiments. Lists are ordered left to right. Exp. 1 starts in ’env2’ with initial epsilon of .5, two lanes, and a duration of 20 seconds. After reaching threshold of a 10-window rolling average of 150, the ’env2’ is closed and ’env3’ is created. The agent continues training retains the same Q-networks in ’env3’ but the epsilon-greedy exploration value is reset to .1 to moderate exploration as desired

Results - A picture worth 1,000 words

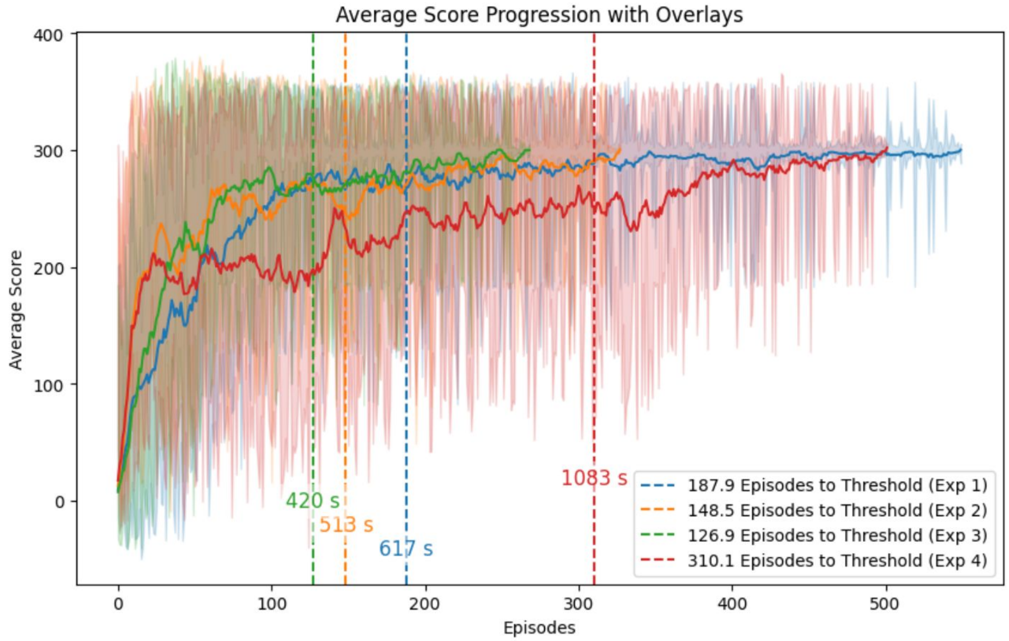
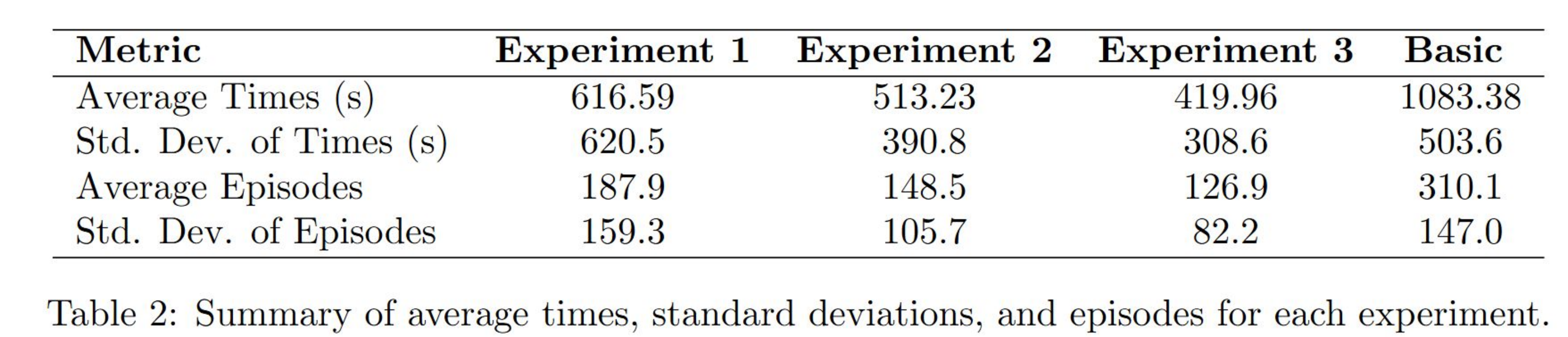


Figure 2: Plot of average scores by episode for each run. The average time to threshold is used to label the thresholds for each experiment, which may not align with the episode count. The episode number on the X-axis provides a more detailed view of the agent's learning progress. However, time is a valuable resource, and curriculum learning seems to reduce the computational cost of training for continuous tasks.

Results table

Experiments 1,2,3 are different curricula and were each found to achieve the same performance threshold in less time than the basic DQN as seen in the table:



Metric	Experiment 1	Experiment 2	Experiment 3	Basic
Average Times (s)	616.59	513.23	419.96	1083.38
Std. Dev. of Times (s)	620.5	390.8	308.6	503.6
Average Episodes	187.9	148.5	126.9	310.1
Std. Dev. of Episodes	159.3	105.7	82.2	147.0

Table 2: Summary of average times, standard deviations, and episodes for each experiment.

Conclusions

Key Findings

- Curriculum Learning demonstrated significant benefits for complex tasks like highway driving:
  - **Faster Convergence**: Improved training efficiency in reinforcement learning.

Limitations

- Limited experimental rigor:
  - High testing costs and a small number of runs for each curriculum.
  - Uniform hyperparameter tuning with minimal fine-tuning during the process.
- Influence of epsilon resetting:
  - Potential impact on the observed benefits of curriculum learning.

Future Directions

- Explore **Curriculum Mapping**:
  - Investigate more refined and adaptive curricula.
  - Aim for greater computational savings with minimal trade-offs in generalizability.
- Conduct experiments with larger datasets and more diverse environments to validate findings.

Sources cited

1. Edouard Leurent. An environment for autonomous driving decision-making. https://github.com/eleurent/highway-env, 2018.
2. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.
3. Bartow Sutton. Reinforcement Learning. Westchester Publishing Services, 2020.
4. Matthew E. Taylor, Peter Stone, and Yaxin Liu. Transfer learning via inter-task mappings for temporal difference learning. Journal of Machine Learning Research, 8:2125–2167, 2007. Submitted 11/06; Revised 4/07; Published 9/07.
5. Xin Wang, Lichao Liu, Jing Yu, Li Zhang, and Yunchao Yang. A survey on curriculum learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(9):4555–4576, 2022.
6. Zhiyuan Xu, Wei Zhang, Tianyu Yang, Jure Xu, and Yifan Yu. Knowledge transfer in multi-task deep reinforcement learning for continuous control. arXiv preprint arXiv:2010.07494, 2020.
7. Fuzhen Zhuang, Zhiqiang Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Heng Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1):43–76, 2021