Towards Efficient Multi-Agent Learning Systems

Kailash Gogineni, Peng Wei, Tian Lan, Guru Venkataramani
Outline

Multi-Agent RL : Background
Workload Characterization
Neighbor Sampling Strategy
Conclusion & Future work
Outline

Multi-Agent RL: Background

Workload Characterization

Neighbor Sampling Strategy

Conclusion & Future work
Multi-Agent RL

Source: Nowe, Vrancx & De Hauwere 2012
Outline

- Multi-Agent RL: Background
- Workload Characterization
- Neighbor Sampling Strategy
- Conclusion & Future work
Workload breakdown
Mini-batch sampling

Experience Replay Buffer

Experience Replay Buffer

Experience Replay Buffer

Experience Replay Buffer

Layout of Random sampling, batch size = 5
Mini-batch sampling

Experience Replay Buffer

Layout of Random sampling, batch size = 5

Random memory access patterns

Iter-1

Iter-2

Iter-3

Iter-N
More details in the paper

• Target Q calculation
  • State and action spaces grow exponentially

• Back-propagation phases of Actor & Critic networks
  • Sequential updates of Actor-Critic networks

• Mini-batch size limitations in MARL for real-world systems
  • Communication overhead
Methodology

• MARL Workloads
  – MADDPG
  – MASAC (more details in the paper)

• Multi-Agent Particle Environment
  – Competitive task (Predator-Prey) - $N$ predators work cooperatively to block the way of $M$ fast paced prey agents. The prey agents are environment controlled and they try to avoid the collision with predators.

• Hyper-parameters
  – The workloads are trained for 60K episodes, max episode length is 25, size of replay buffer is 1M, mini-batch size is 1024, entropy coefficient for MASAC is 0.05.

• Platform (experimental testbed)
  – Ampere Architecture NVIDIA-RTX 3090
MADDPG: Training time from 3 to 48 agents

**Entire workload**

**Update all trainers**

MASAC algorithm also exhibits the same pattern!
Outline

Multi-Agent RL: Background

Workload Characterization

Neighbor Sampling Strategy

Conclusion & Future work
Goal: Eliminate random memory access patterns

Can we retrieve neighboring transitions?

Experience Replay Buffer

Layout of Random sampling

Iter-1

Iter-2

Iter-3

......

Iter-N
Effect of neighbor sampling

Experience Replay Buffer

Iter-1

Random Sampling + Neighbor transitions

Iter-1

Random Sampling

Iter-1

Layout of Neighbor sampling strategy
Effect of neighbor sampling

Layout of Neighbor sampling strategy, micro-batch size = 2

Out of bounds?

Iter-1

Iter-2

Iter-3

......

Iter-N

Layout of Neighbor sampling strategy, micro-batch size = 2
Results

Training time performance improvement

<table>
<thead>
<tr>
<th>Number of agents</th>
<th>Mini-batch sampling phase (%)</th>
<th>Total training time (%</th>
<th>Average of mean rewards</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>26.66</td>
<td>26.68</td>
<td>27.39</td>
</tr>
<tr>
<td></td>
<td>5.6</td>
<td>7.8</td>
<td>10.2</td>
</tr>
</tbody>
</table>

Average of mean rewards

<table>
<thead>
<tr>
<th>Number of agents</th>
<th>MADDPG Average reward</th>
<th>Neighbor_sampling_optimized-MADDPG Average reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>21.04</td>
<td>103.96</td>
</tr>
<tr>
<td>6</td>
<td>20.05</td>
<td>105.94</td>
</tr>
<tr>
<td>12</td>
<td>870.39</td>
<td>872.49</td>
</tr>
</tbody>
</table>
Outline

Multi-Agent RL : Background

Workload Characterization

Neighbor Sampling Strategy

Conclusion & Future work
Conclusion & Future work

• We understand and characterize several MARL algorithms to identify key bottlenecks from a systems perspective

• We implemented a simple heuristic, neighbor sampling strategy to address the mini-batch sampling phase

• For future work, we will investigate:
  • Various efficient sampling strategies and design a hardware-friendly architecture to efficiently fetch the transitions in large-scale MARL
  • Use algorithmic optimizations into systems to reduce the observation-action space
Thank You!

Q & A
References

• CMU AI and [http://ai.berkeley.edu](http://ai.berkeley.edu)

• Google DeepMind – Multi-Agent Systems & AI

