Seminar in Deep Reinforcement Learning

The Path to Continual Learning

Curriculum Learning

Ramon Witschi, ETH Computer Science MSc, 19.05.2020
What is a Curriculum?
Curriculum over Training Data!

Curriculum Learning (ICML, 2009) – Bengio et al.
1. Start with simple examples
2. Gradually add more difficult ones
3. Arrive at target training distribution

Curriculum Learning (ICML, 2009) – Bengio et al.
1. Start with simple examples
2. Gradually add more difficult ones
3. Arrive at target training distribution

Curriculum Learning (ICML, 2009) – Bengio et al.
1. Start with simple examples
2. Gradually add more difficult ones
3. Arrive at target training distribution

Curriculum Learning (ICML, 2009) – Bengio et al.
Empirical Results

Faster Training & sometimes higher Test Scores

Curriculum Learning (ICML, 2009) – Bengio et al.
Faster Training *proven* on Linear Regression (Convex Optimization) 😊
Curriculum Learning meets Reinforcement Learning

Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play

Sukhbaatar et al.

Reverse Curriculum Generation for Reinforcement Learning

Florensa et al.

Mix & Match – Agent Curricula for Reinforcement Learning

Czarnecki et al.
Curriculum Learning meets Reinforcement Learning

- Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play
  - Sukhbaatar et al.

- Reverse Curriculum Generation for Reinforcement Learning
  - Florensa et al.

- Mix & Match – Agent Curricula for Reinforcement Learning
  - Czarnecki et al.
Model-Free Reinforcement Learning

Sample Inefficient 😞
Jointly learn Environment and optimize for Reward
“Unsupervised” Exploration!
Framework

Self Play Episode (no supervision -- internal reward only)

Alice’s turn

Bob’s turn

Target Task Episode (supervision from external reward)

Bob applied to target task
Internal Reward Structure

\[ R_A = \max(0, t_B - t_A) \]

Bob fast
or
too slow

\[ \rightarrow \quad 0 \quad \frown \]
Automatically creates a Curriculum over Exploration Tasks!
Internal Reward Structure

\[ R_A = \max(0, t_B - t_A) \]

if job is fast or too slow

⇒ 0

sad face
The graph shows the average probability on $s_t > 0$ over learning iterations. Different line styles and colors represent different methods:

- **Red**: Uniform Sampling (baseline)
- **Brown**: Oracle (rejection sampling)
- **Green**: Brownian from Good Starts
- **Blue**: Brownian from All Starts
- **Orange**: Asymmetric Self Play Start Generation

As learning iterations increase from 0 to 300, the probability increases for all methods, with the Asymmetric Self Play Start Generation showing the highest probability.
Curriculum Learning meets Reinforcement Learning

Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play
- Sukhbaatar et al.

Reverse Curriculum Generation for Reinforcement Learning
- Florensa et al.

Mix & Match – Agent Curricula for Reinforcement Learning
- Czarnecki et al.
Goal-Oriented Target Tasks
Goal-Oriented Target Tasks

Binary Reward Signal 😞
Goal-Oriented Target Tasks

Binary Reward Signal 😞

+ Model-Free Reinforcement Learning 😞
Goal-Oriented Target Tasks

Binary Reward Signal 😞

+ Model-Free Reinforcement Learning 😞

= (☹️ℓ⊙☹️) ‾‾‾‾‾‾‾
How do We Train the Agent?
Random Sampling of Starting States?
Add Regularization Term?
What’s the Trick?
Easy to Win, if you Start at the Goal!
Reverse Curriculum

1. Start almost there
2. Start increasingly further away
3. Profit from work already done
Reverse Curriculum

1. Start almost there

2. Start increasingly further away

3. Profit from work already done
Reverse Curriculum

1. Start almost there
2. Start increasingly further away
3. Profit from work already done
Automatically creates a Curriculum over Start States!
1. States Close to $s^g$ may be good Start States 💡

2. Random Walk in State-Space 😞

3. Brownian Motion in Action-Space 😊
States of Intermediate Difficulty (SoIDs)

1. States Close to $s^g$ may be good Start States💡
2. Random Walk in State-Space 😞
3. Brownian Motion in Action-Space ☺
States of Intermediate Difficulty (SoIDs)

1. States Close to $s^g$ may be good Start States 🧡
2. Random Walk in State-Space 😞
3. Brownian Motion in Action-Space 😊
$S^g$: goal states we want to reach from everywhere.
$s^g$: one goal state is provided
Iteration 1:
- Run Brownian motion
- Obtain trajectories from collected starts to train policy
- Label and filter starts based on training trajectories
Iteration 1:
- Run Brownian motion
- Obtain trajectories from collected starts to train policy
- Label and filter starts based on training trajectories
Iteration 1:

- Run Brownian motion
- Obtain trajectories from collected starts to train policy
- Label and filter starts based on training trajectories
Iteration 1:
- Run Brownian motion
- Obtain trajectories from collected starts to train policy
- Label and filter starts based on training trajectories
Iteration 2:
- Run Brownian motion
- Obtain trajectories from collected starts to train policy
- Label and filter starts based on training trajectories
Iteration 2:
- Run Brownian motion
- Obtain trajectories from collected starts to train policy
- Label and filter starts based on training trajectories
Iteration 2:
- Run Brownian motion
- Obtain trajectories from collected starts to train policy
- Label and filter starts based on training trajectories
Iteration 2:
- Run Brownian motion
- Obtain trajectories from collected starts to train policy
- Label and filter starts based on training trajectories
Iteration 3:
- Run Brownian motion
- Obtain trajectories from collected starts to train policy
- Label and filter starts based on training trajectories

\[ S_g \]
Iteration 4:
- Run Brownian motion
- Obtain trajectories from collected starts to train policy
- Label and filter starts based on training trajectories

\[ S_g \]

- Low Reward
- High Reward
- SoID state
Iteration 5:
- Run Brownian motion
- Obtain trajectories from collected starts to train policy
- Label and filter starts based on training trajectories
(d) Key insertion task
Curriculum Learning meets Reinforcement Learning

Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play
- Sukhbaatar et al.

Reverse Curriculum Generation for Reinforcement Learning
- Florensa et al.

Mix & Match – Agent Curricula for Reinforcement Learning
- Czarnecki et al.
Rethinking the Notion of Curriculum
Curriculum not Automatic!
What’s the Difficulty of an Agent?
Agents are Neural Networks!

*for all practical purposes*
Architectural Components
or
Performable Actions
or
Jointly-Learnable Tasks
and
Training Iterations
# Architectural Components

or

# Performable Actions

or

# Jointly-Learnable Tasks

and

# Training Iterations
# Architectural Components
or
# Performable Actions
or
# Jointly-Learnable Tasks
and
# Training Iterations
# Architectural Components or # Performable Actions or # Jointly-Learnable Tasks and # Training Iterations
Difficulty ✓
Scheduler: Tune Mixture Parameter $\alpha$
Scheduler: Tune Mixture Parameter $\alpha$

Could use hand crafted scheduler 😞
Scheduler: Tune Mixture Parameter $\alpha$

Could use hand crafted scheduler 😞

Could use naive hyperparameter tuning 😞
Scheduler: Tune Mixture Parameter $\alpha$

Could use hand crafted scheduler 😞

Could use naive hyperparameter tuning 😞

Population Based Training 😊
Population Based Training
Population Based Training

1. Tuning several mixture agents in parallel
Population Based Training

1. Tuning several mixture agents in parallel
2. Agent A periodically communicates with some B
Population Based Training

1. Tuning several mixture agents in parallel
2. Agent A periodically communicates with some B
3. Badly performing: Copy weights and hyperparameters ($\alpha$)
Explore Search Space with badly performing Agents
Curriculum Learning Is Here to Stay! 🎈
Yes, Mr. Frodo. It’s over now.
Yes, Mr. Frodo. It’s over now.

Take Care!