Generalized Reinforcement Learning for Gameplay

Amir H. Payberah
payberah@kth.se
2021-09-27

Scalable Learning Systems (SCALES)

- Francisco Pena, Postdoc
- Sara Karimi, Industrial PhD student (King)
- Sina Sheikholeslami, PhD student
- Kamal Hakimzadeh, PhD student
- Shirin Tahmasebi, PhD student
- Amirhossein Layegh, PhD student
- Tianze Wang, PhD student

SCALES Research Topics

- Distributed Deep Learning
- Reinforcement Learning
- Natural Language Processing
- Big Data Processing Systems

SCALES Research Topics

- Distributed Deep Learning
- Reinforcement Learning
- Natural Language Processing
- Big Data Processing Systems

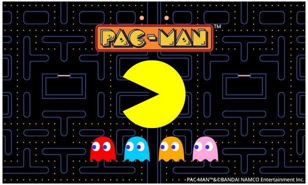
Let's Start

AI and Games

[https://medium.com/free-code-camp/an-introduction-to-reinforcement-learning-4339519de419]

From Scripted to Smart Games

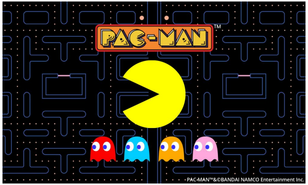
- Games and AI have a long history.
- Through the years, games became more intelligent and less scripted.



From Scripted to Smart Games

- Games and AI have a long history.
- Through the years, games became more intelligent and less scripted.

- 1980 - Pacman
- 1991 - Civilization
- 1998 - Starcraft: Brood War
- 2005 - World of Warcraft
- 2016 - AlphaGo



What Is The Challenge?

- Generalization: the ability of an agent that is trained on one environment to perform well in a new environment with different characteristics.

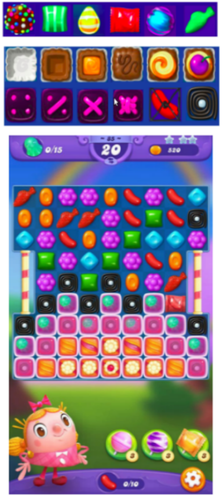
Problem Setting

Candy Crush Friends Saga (CCFS)



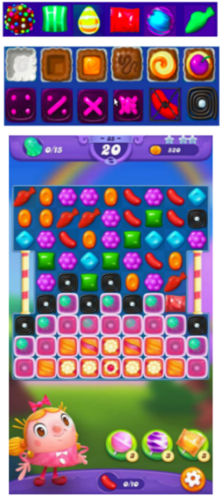
CCFS Features

- Match-3 game



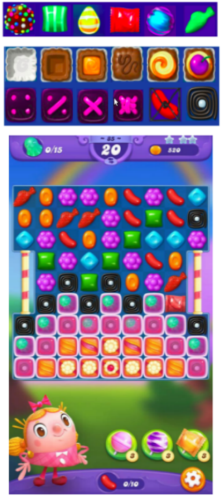
CCFS Features

- Match-3 game
- Stochastic transitions



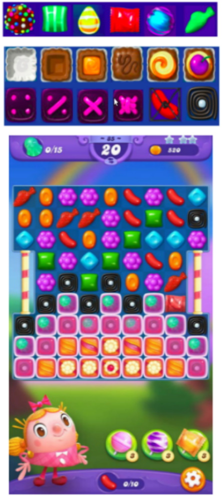
CCFS Features

- Match-3 game
- Stochastic transitions
- Various game elements, e.g., Candy and Blocker



CCFS Features

- Match-3 game
- Stochastic transitions
- Various game elements, e.g., Candy and Blocker
- Various game objectives, e.g., Spreading Jam



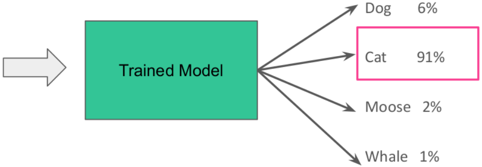
Supervised Learning for CCFS

Supervised Learning for CCFS (1/5)

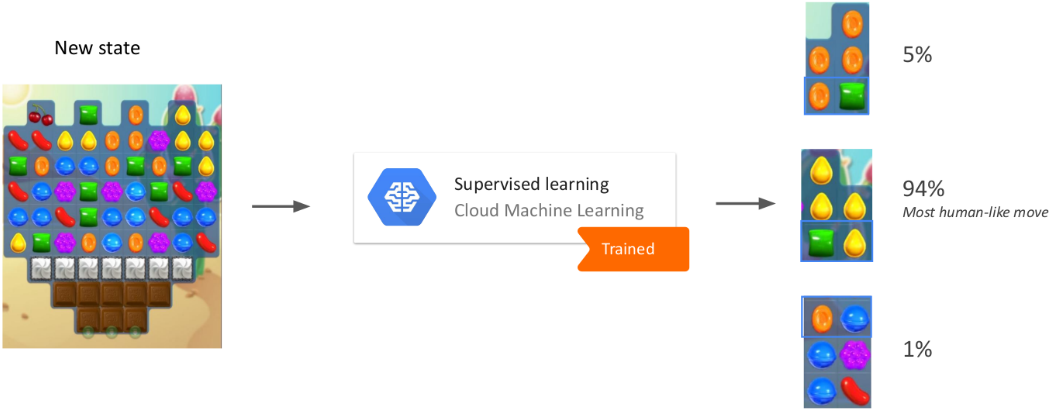
- Supervised learning: given lots of labelled observations, predict the label of an unseen observation.

Supervised Learning for CCFS (1/5)

- Supervised learning: given lots of labelled observations, predict the label of an unseen observation.

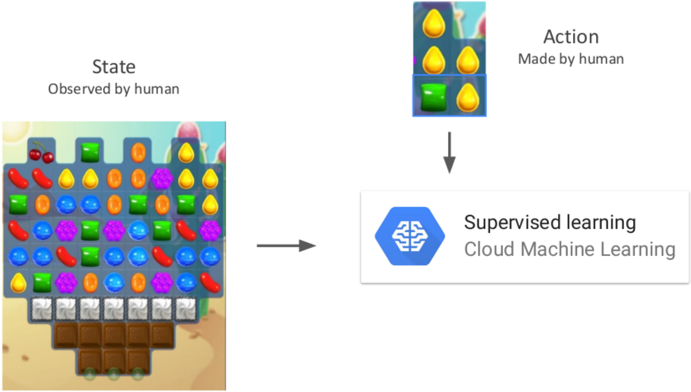


Supervised Learning for CCFS (2/5)



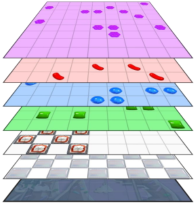
[Gudmundsson et al., Human-Like Playtesting with Deep Learning, IEEE CIG 2018.]

Supervised Learning for CCFS (3/5)

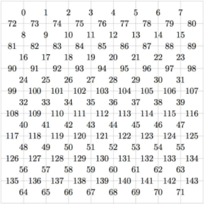


[Gudmundsson et al., Human-Like Playtesting with Deep Learning, IEEE CIG 2018.]

Supervised Learning for CCFS (4/5)



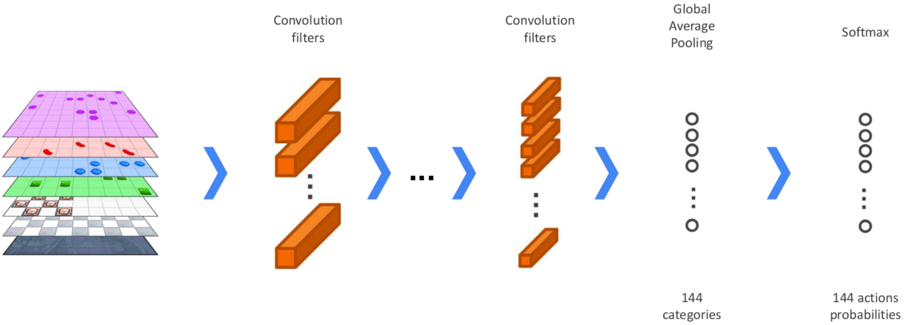
100+ binary feature layers



action space of size 144

[Gudmundsson et al., Human-Like Playtesting with Deep Learning, IEEE CIG 2018.]

Supervised Learning for CCFS (5/5)



[Gudmundsson et al., Human-Like Playtesting with Deep Learning, IEEE CIG 2018.]

Supervised Learning for CCFS - Challenges

- Requires large volume of players data.
- Generalization over wide variety of game content.

Supervised Learning for CCFS - Challenges

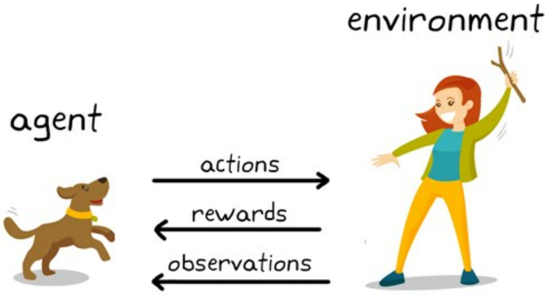
- Requires large volume of players data.
- Generalization over wide variety of game content.

- Possible solution: using Reinforcement Learning (RL) as a general framework that does not require player data.

Reinforcement Learning

Reinforcement Learning

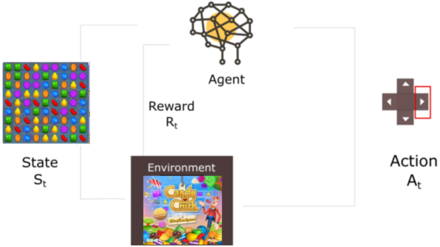
- RL: an agent learns from the environment by interacting with it and receiving rewards for performing actions.



[https://www.kdnuggets.com/2019/10/mathworks-reinforcement-learning.html]

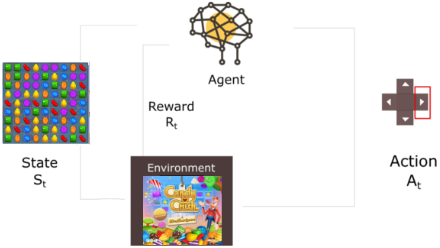
RL - Basic Concepts

- Environment: physical world in which the agent operates.



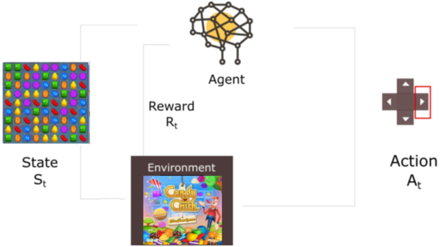
RL - Basic Concepts

- Environment: physical world in which the agent operates.
- State: current situation of the agent/environment.



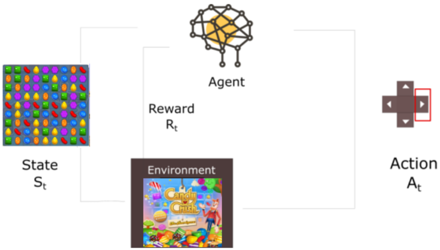
RL - Basic Concepts

- Environment: physical world in which the agent operates.
- State: current situation of the agent/environment.
- Policy: method to map agent's state to actions.



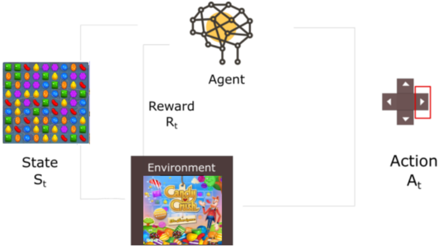
RL - Basic Concepts

- Environment: physical world in which the agent operates.
- State: current situation of the agent/environment.
- Policy: method to map agent's state to actions.
- Reward: feedback from the environment.



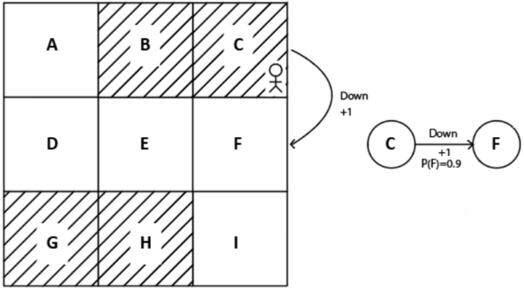
RL - Basic Concepts

- Environment: physical world in which the agent operates.
- State: current situation of the agent/environment.
- Policy: method to map agent's state to actions.
- Reward: feedback from the environment.
- Value: future reward that an agent would receive by taking an action in a particular state.



RL - Markov Decision Processes (1/2)

- Markov Decision Processes (MDP): modeling sequential decision making, where actions influence not just immediate rewards, but also subsequent states.



[P. Dayan et al., Reinforcement learning: The Good, The Bad and The Ugly, 2008.]

RL - Markov Decision Processes (2/2)

- Goal: maximizing the expected cumulative reward.

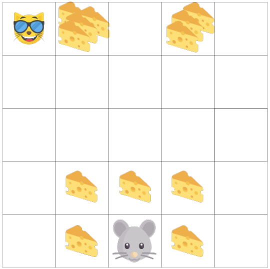
$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots$$

[https://medium.com/free-code-camp/an-introduction-to-reinforcement-learning-4339519de419]

RL - Markov Decision Processes (2/2)

- Goal: maximizing the expected cumulative reward.

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots$$



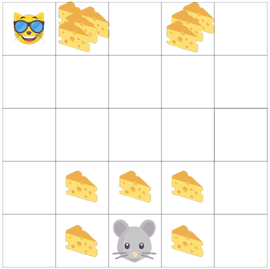
- But, the rewards that come sooner are more probable to happen (they are more predictable).

[https://medium.com/free-code-camp/an-introduction-to-reinforcement-learning-4339519de419]

RL - Markov Decision Processes (2/2)

- Goal: maximizing the expected cumulative reward.

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots$$



- But, the rewards that come sooner are more probable to happen (they are more predictable).

- Discounted cumulative expected.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots, \gamma \in [0, 1)$$

[https://medium.com/free-code-camp/an-introduction-to-reinforcement-learning-4339519de419]

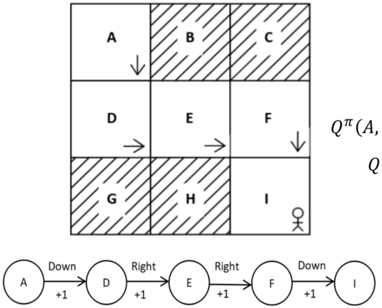
RL - Q-Value

- $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$

RL - Q-Value

- $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$
- Q-value: $Q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$

- $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$
- Q-value: $Q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$



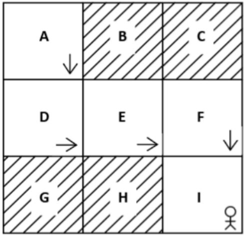
$$Q^\pi(A, \text{down}) = [R(\tau) | s_0 = A, a_0 = \text{down}]$$

$$Q(A, \text{down}) = 1 + 1 + 1 + 1 = 4$$

[P. Dayan et al., Reinforcement learning: The Good, The Bad and The Ugly, 2008.]

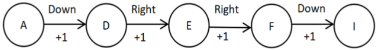
RL - Q-Value

- $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$
- Q-value: $Q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$
- Optimal Q-value?



$$Q^\pi(A, \text{down}) = [R(\tau) | s_0 = A, a_0 = \text{down}]$$

$$Q(A, \text{down}) = 1 + 1 + 1 + 1 = 4$$



[P. Dayan et al., Reinforcement learning: The Good, The Bad and The Ugly, 2008.]

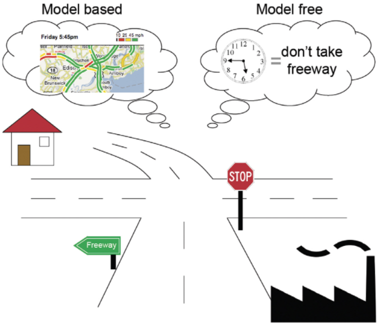
RL - Model-Based vs. Model-Free

- Model: mimics the behavior of the environment

RL - Model-Based vs. Model-Free

- Model: mimics the behavior of the environment

- Model-based: using an explicit representation of the model of the environment



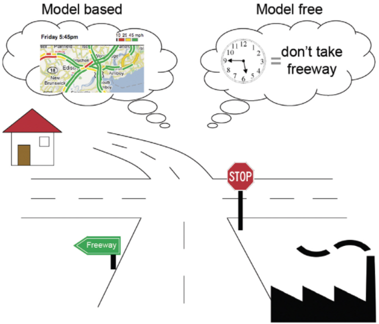
[P. Dayan et al., Reinforcement learning: The Good, The Bad and The Ugly, 2008.]

RL - Model-Based vs. Model-Free

- Model: mimics the behavior of the environment

- Model-based: using an explicit representation of the model of the environment

- Model-free: a representative of the model of the environment is not available or not practical



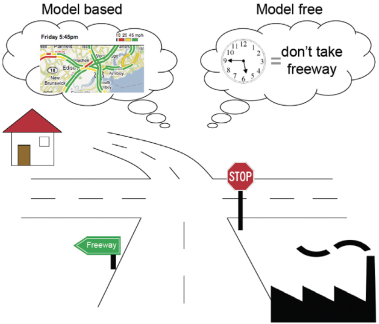
[P. Dayan et al., Reinforcement learning: The Good, The Bad and The Ugly, 2008.]

RL - Model-Based vs. Model-Free

- Model: mimics the behavior of the environment

- Model-based: using an explicit representation of the model of the environment
  - Dynamic Programming

- Model-free: a representative of the model of the environment is not available or not practical



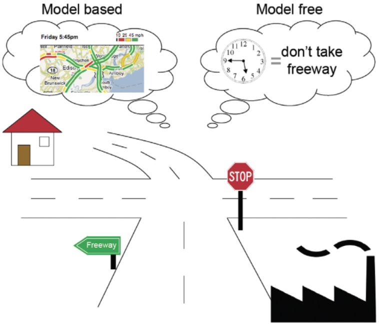
[P. Dayan et al., Reinforcement learning: The Good, The Bad and The Ugly, 2008.]

RL - Model-Based vs. Model-Free

- Model: mimics the behavior of the environment

- Model-based: using an explicit representation of the model of the environment
  - Dynamic Programming

- Model-free: a representative of the model of the environment is not available or not practical
  - Monte Carlo and Temporal Difference (e.g., Q-learning)



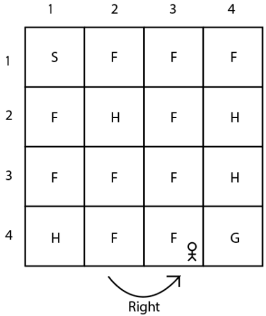
[P. Dayan et al., Reinforcement learning: The Good, The Bad and The Ugly, 2008.]

RL - Q-Learning

- A model-free approach to learn the value of an action, i.e., $Q_{\pi}(s, a)$.

RL - Q-Learning

- A model-free approach to learn the value of an action, i.e., $Q_\pi(s, a)$.



State	Action	Value
(1,1)	Up	0.5
⋮	⋮	⋮
(4,2)	Up	0.3
(4,2)	Down	0.5
(4,2)	Left	0.1
(4,2)	Right	0.8
⋮	⋮	⋮
(4,4)	Right	0.5

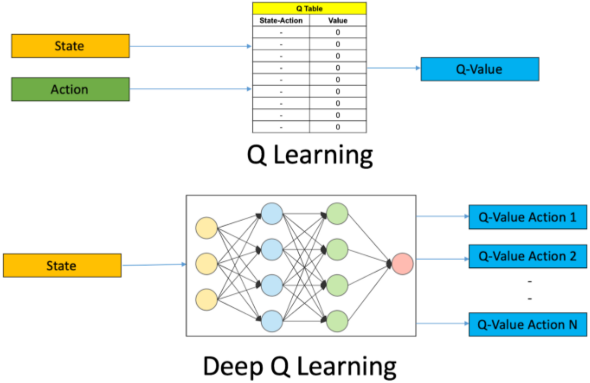
[S. Ravichandiran, Deep Reinforcement Learning with Python, Packt Publishing Ltd., 2018.]

RL - Deep Q-Network (DQN)

- When the number of states and actions becomes very large.

RL - Deep Q-Network (DQN)

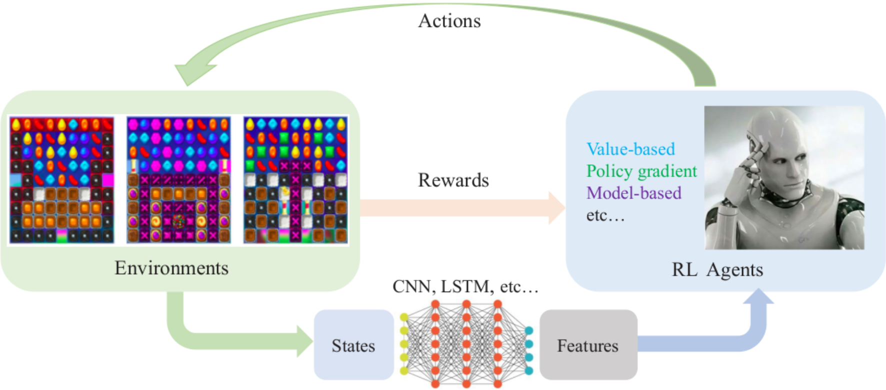
- When the number of states and actions becomes very large.



[https://medium.com/@novacek_48158/connect-x-with-dqn-and-pbt-be11915dd860]

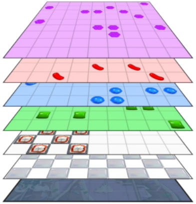
RL for CCFS

RL for CCFS



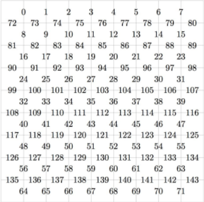
[Shao et al., A Survey of Deep Reinforcement Learning in Video Games, arXive 2019.]

State Space and Action Space



100+ binary
feature layers

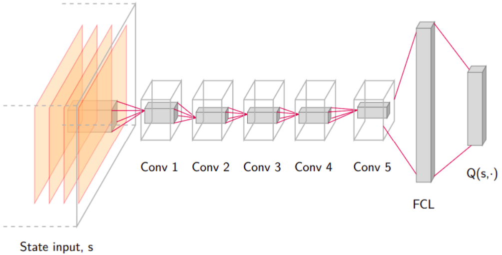
action space of
size 144



[Gudmundsson et al., Human-Like Playtesting with Deep Learning, IEEE CIG 2018.]

Network Architecture

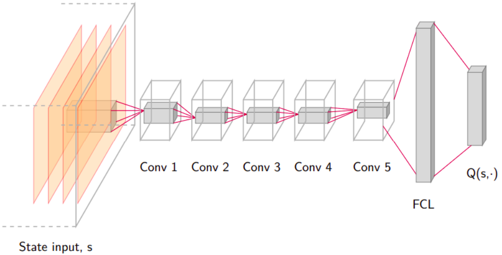
- DQN algorithm.



[A. Karnsund, DQN Tackling the Game of Candy Crush Friends Saga, KTH/King 2019.]

Network Architecture

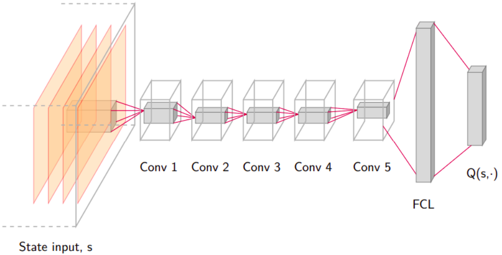
- DQN algorithm.
- CNN with five convolutional layers and two fully-connected layers.



[A. Karnsund, DQN Tackling the Game of Candy Crush Friends Saga, KTH/King 2019.]

Network Architecture

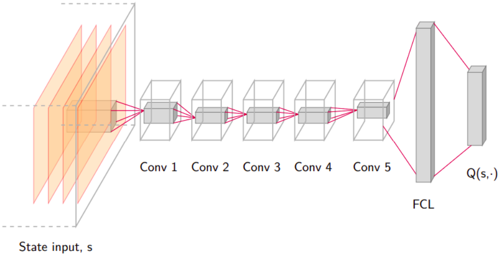
- DQN algorithm.
- CNN with five convolutional layers and two fully-connected layers.
- One Q-value for each action.



[A. Karnsund, DQN Tackling the Game of Candy Crush Friends Saga, KTH/King 2019.]

Network Architecture

- DQN algorithm.
- CNN with five convolutional layers and two fully-connected layers.
- One Q-value for each action.



- What about the policy and reward function?

[A. Karnsund, DQN Tackling the Game of Candy Crush Friends Saga, KTH/King 2019.]

Extrinsic and Intrinsic Motivation (1/2)

- Extrinsic motivation: do something because of some external reward.

Extrinsic and Intrinsic Motivation (1/2)

- Extrinsic motivation: do something because of some external reward.
- E.g., Progressive Jam (PJ): rewards an agent for making a move that spreads at least one more tile with Jam.

Extrinsic and Intrinsic Motivation (1/2)

- Extrinsic motivation: do something because of some external reward.
- E.g., Progressive Jam (PJ): rewards an agent for making a move that spreads at least one more tile with Jam.

#(tiles covered in jam) / #(tiles on the board)

$$R(\boxed{\mathbf{s_t}, a_t}) = \begin{cases} \boxed{\frac{J}{B}}, & \Delta j > 0 \\ 0, & \Delta j = 0 \end{cases}$$

state and action at step t

Extrinsic and Intrinsic Motivation (1/2)

- Extrinsic motivation: do something because of some external reward.
- E.g., Progressive Jam (PJ): rewards an agent for making a move that spreads at least one more tile with Jam.
- Limition: not generalized and only focused on external rewards.

$$R(\boxed{\mathbf{s_t}, a_t}) = \begin{cases} \boxed{\frac{J}{B}}, & \Delta j > 0 \\ 0, & \Delta j = 0 \end{cases}$$

#(tiles covered in jam)
#(tiles on the board)

state and action
at step t

Extrinsic and Intrinsic Motivation (2/2)

- Intrinsic motivation: do something because it is inherently enjoyable.

Extrinsic and Intrinsic Motivation (2/2)

- Intrinsic motivation: do something because it is inherently enjoyable.
- E.g., Learning basic skills (helps human players to achieve the level objective faster).

Extrinsic and Intrinsic Motivation (2/2)

- Intrinsic motivation: do something because it is inherently enjoyable.
- E.g., Learning basic skills (helps human players to achieve the level objective faster).
- An agent rewards itself for completing sub-goals that can be different from the goal of the environment.

Question?

- Using intrinsic rewards, can an agent learn a set of basic skills to achieve extrinsic rewards?

Question?

- Using intrinsic rewards, can an agent learn a set of basic skills to achieve extrinsic rewards?
- Can an agent employ this set of skills to win new levels? (generalization)

Basic Skills and Generalization

- Find a good reward function to learn basic skills.

Basic Skills and Generalization

- Find a good reward function to learn basic skills.
- Creating a Special candy is a basic skill.

Basic Skills and Generalization

- Find a good reward function to learn basic skills.
- Creating a Special candy is a basic skill.

  - Six skills, one for each Special Candy: $\mathtt{X} = \{\mathtt{x}_1, \mathtt{x}_2, \cdots, \mathtt{x}_6\}$

Basic Skills and Generalization

- Find a good reward function to learn basic skills.
- Creating a Special candy is a basic skill.

  - Six skills, one for each Special Candy: $\mathtt{X} = \{\mathtt{x}_1, \mathtt{x}_2, \cdots, \mathtt{x}_6\}$
- Train an RL agent to use these skills more frequently.

Basic Skills and Generalization

- Find a good reward function to learn basic skills.
- Creating a Special candy is a basic skill.

  - Six skills, one for each Special Candy: $\mathtt{X} = \{\mathtt{x}_1, \mathtt{x}_2, \cdots, \mathtt{x}_6\}$
- Train an RL agent to use these skills more frequently.
- Problem: some special candies are easier to create than others, thus they will be created more often.

Rarity of Events (RoE)

- Use the frequency of occurrence of each skill as a weight.

Rarity of Events (RoE)

- Use the frequency of occurrence of each skill as a weight.
  - Skills that are used less are rewarded more.

Rarity of Events (RoE)

- Use the frequency of occurrence of each skill as a weight.
  - Skills that are used less are rewarded more.

$$R(\underbrace{\boxed{\mathbf{s}_t, a_t}}_{\text{state and action at step t}}) = \sum_{x \in X} \overbrace{\boxed{c_t^{(x)}}}^{\#(\text{candies of type } x \text{ created at time t})} \times \left[\frac{1}{\max\left(\underbrace{\boxed{\tau}}_{\text{hyperparameter}}, \underbrace{\boxed{\mu_t^{(x)}}}_{\text{mean frequency of creation of candy } x}\right)} \right]$$

[Niels et al., Automated curriculum learning by rewarding temporally rare events, IEEE CIG, 2018.]

Rarity of Events (RoE)

- Use the frequency of occurrence of each skill as a weight.
  - Skills that are used less are rewarded more.
- Issue 1: Rewards might be much higher than 1 (If $\mu_t^{(x)} < 1$), thus gradient updates become unstable.

#(candies of type *x* created at time t)

$$R(\boxed{\mathbf{s}_t, a_t}) = \sum_{x \in X} \boxed{c_t^{(x)}} \times \left[\frac{1}{\max\left(\boxed{\tau}, \boxed{\mu_t^{(x)}}\right)} \right]$$

state and action at step t

hyperparameter

mean frequency of creation of candy *x*

[Niels et al., Automated curriculum learning by rewarding temporally rare events, IEEE CIG, 2018.]

Rarity of Events (RoE)

- Use the frequency of occurrence of each skill as a weight.
    - Skills that are used less are rewarded more.
- Issue 1: Rewards might be much higher than 1 (If $\mu_t^{(x)} < 1$), thus gradient updates become unstable.
- Issue 2: Requires an hyperparameter to prevent cold start problem.

$$R(\underbrace{\boxed{\mathbf{s}_t, a_t}}_{\text{state and action at step t}}) = \sum_{x \in X} \overbrace{\boxed{c_t^{(x)}}}^{\text{\#(candies of type } x \text{ created at time t)}} \times \left[\frac{1}{\max\left(\underbrace{\boxed{\tau}}_{\text{hyperparameter}}, \underbrace{\boxed{\mu_t^{(x)}}}_{\text{mean frequency of creation of candy } x}\right)} \right]$$

[Niels et al., Automated curriculum learning by rewarding temporally rare events, IEEE CIG, 2018.]

Balanced Rarity of Events (BRoE)

- Same concept as RoE
  - Use the frequency as a weight.
  - The more a skill is used, the less it is rewarded.

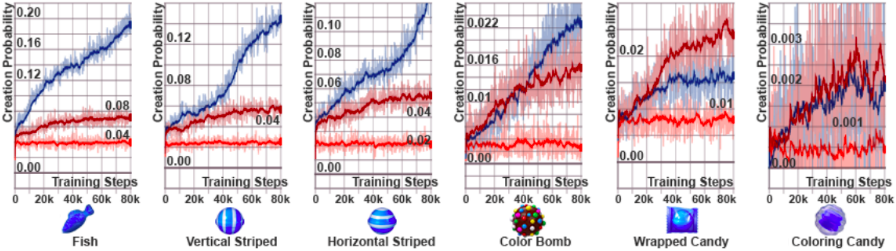
Balanced Rarity of Events (BRoE)

- Same concept as RoE
  - Use the frequency as a weight.
  - The more a skill is used, the less it is rewarded.
- Take into account the proportion of occurrence of a skill with respect to all the other skills.

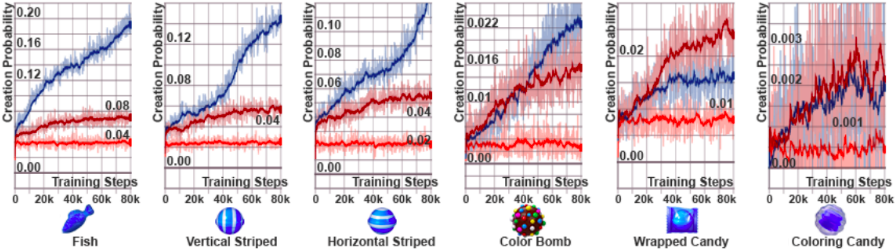
$$R(\underbrace{\boxed{\mathbf{s}_t, a_t}}_{\text{state and action at step t}}) = \sum_{x \in X} \overbrace{\boxed{c_t^{(x)}}}^{\text{\#(candies of type } x \text{ created at time t)}} \times \left(1 - \frac{\overbrace{\boxed{\mu_t^{(x)}}}^{\text{mean frequency of creation of candy } x}}{\sum_{x' \in X} \underbrace{\boxed{\mu_t^{(x')}}}_{\text{mean frequency of creation of candy } x'}}\right)$$

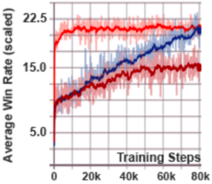
[F. Lorenzo et al., Use All Your Skills, Not Only The Most Popular Ones, IEEE CoG, 2020.]

Basic Skill Results



Basic Skill Results





Average win rate.

[F. Lorenzo et al., Use All Your Skills, Not Only The Most Popular Ones, IEEE CoG, 2020.]

More Basic Skills

- Special Candies Skills

- Blockers skills

Special Candies Skills

- Creating Special Candies (BRoE): reward given when a Special Candy is created.

Special Candies Skills

- Creating Special Candies (BRoE): reward given when a Special Candy is created.
- Using Special Candies (CU): reward given when a Special Candy is used.

Special Candies Skills

- Creating Special Candies (BRoE): reward given when a Special Candy is created.
- Using Special Candies (CU): reward given when a Special Candy is used.

Reward	Win Rate		Creation (%)	
	Train	Test	Train	Test
Random	4.03	1.77	1.93	2.10
PJ	**7.56**	3.20	1.71	1.90
CU	6.54	3.44	1.70	1.90
BRoE	7.54	**4.03**	**9.06**	**6.71**

[F. Lorenzo et al., Generalized Reinforcement Learning for Gameplay, AAAI RLG, 2021.]

Blockers Skills

- Damaging Blockers (DB): rewards for each layer removed from a Blocker.

Blockers Skills

- Damaging Blockers (DB): rewards for each layer removed from a Blocker.
- Progressive Tiles (PT): rewards for removing a Blocker completely from a tile.

Blockers Skills

- Damaging Blockers (DB): rewards for each layer removed from a Blocker.
- Progressive Tiles (PT): rewards for removing a Blocker completely from a tile.

Reward	**Win rate**		**Clearing (%)**	
	Train	Test	Train	Test
Random	0.21	0.05	54.81	63.63
PJ	1.14	0.35	65.08	71.18
PT	**2.76**	**0.50**	**78.92**	**73.29**
DB	1.95	0.34	75.82	71.35

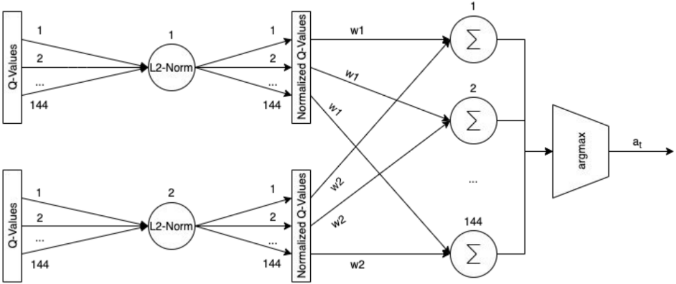
[F. Lorenzo et al., Generalized Reinforcement Learning for Gameplay, AAAI RLG, 2021.]

Let's Use All The Skills (1/3)

- Hybrid model: Average Baging (AB)
- The basic skills are pre-trained.

Let's Use All The Skills (1/3)

- Hybrid model: Average Baging (AB)
- The basic skills are pre-trained.



[F. Lorenzo et al., Generalized Reinforcement Learning for Gameplay, AAAI RLG, 2021.]

Let's Use All The Skills (2/3)

Level	Combination	Win Rate	
		L2	None
82	PJ+PT+BRoE	7.02	7.40
	PJ+PT+DB+BRoE	7.33	**8.08**
62	PJ+BRoE	16.44	16.25
	PJ+BRoE+CU	**17.37**	15.83
136	PJ+PT+BRoE	3.84	4.12
	PJ+PT+DB+BRoE	3.51	**4.41**
147	PJ+PT+BRoE	2.39	2.65
	PJ+PT+DB+BRoE	2.45	**3.01**
163	PJ+PT+BRoE	0.1	0.12
	PJ+PT+DB+BRoE	0.09	**0.14**

Let's Use All The Skills (2/3)

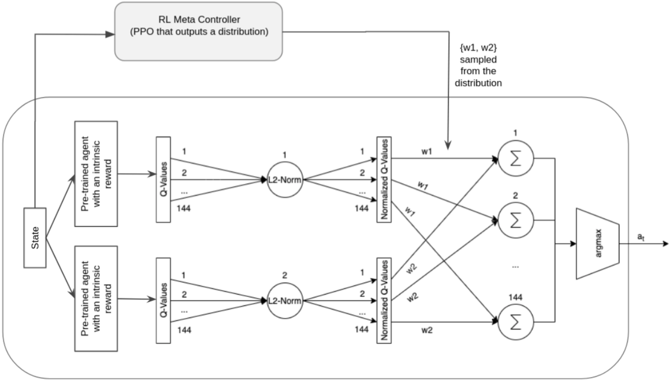
Level	Combination	Win Rate	
		L2	None
82	PJ+PT+BRoE	7.02	7.40
	PJ+PT+DB+BRoE	7.33	**8.08**
62	PJ+BRoE	16.44	16.25
	PJ+BRoE+CU	**17.37**	15.83
136	PJ+PT+BRoE	3.84	4.12
	PJ+PT+DB+BRoE	3.51	**4.41**
147	PJ+PT+BRoE	2.39	2.65
	PJ+PT+DB+BRoE	2.45	**3.01**
163	PJ+PT+BRoE	0.1	0.12
	PJ+PT+DB+BRoE	0.09	**0.14**

Level	Humans	AB	PJ	Random
82	**9.60**	8.08	1.33	0.13
62	**21.92**	17.37	10.21	5.22
136	3.10	**4.41**	0.81	0.08
147	**6.90**	3.01	0.55	0.08
163	**1.03**	0.14	0.01	0

[F. Lorenzo et al., Generalized Reinforcement Learning for Gameplay, AAAI RLG, 2021.]

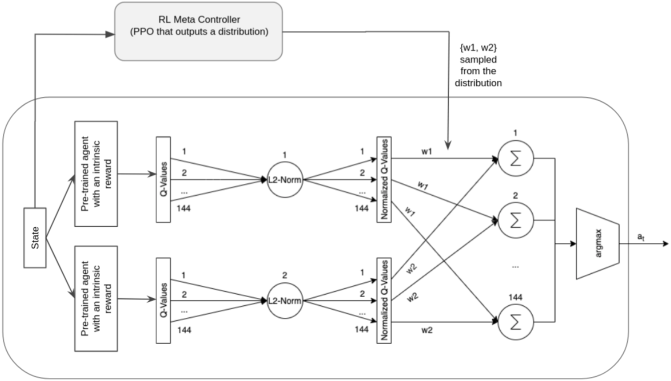
Let's Use All The Skills (3/3)

- Hybrid model controller.



Let's Use All The Skills (3/3)

- Hybrid model controller.
- Under development.



Summary and Future Work

Summary

- AI and Games
- Candy Crush Friends Saga (CCFS)
- Supervised learning for CCFS
- RL for CCFS
- Extrinsic vs. intrinsic rewards
- Hybrid model

Future Work

- Beyond CCFS
- Scalability

Questions?

Acknowledgements

Some content and images are derived from Sahar Asadi, Sara Karimi, and Francesco Lorenzo's slides from King