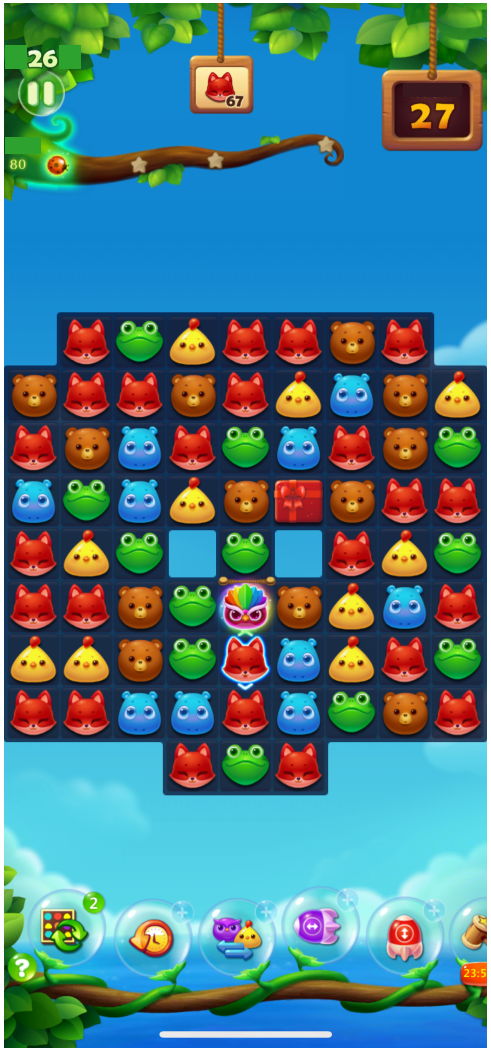
Happy Elements

MAKE THE WORLD HAPPY

Towards Deep Reinforcement Learning for Game Optimization

Rein Houthooft

Happy Elements



- Company focus: **casual** mobile games
- Main product: “**Anipop**” 
- Extremely popular (>**100M** users/month)
- Generates **TBs of data** each day
- **AI Lab** founded 2 years ago

Happy Elements

Workflow

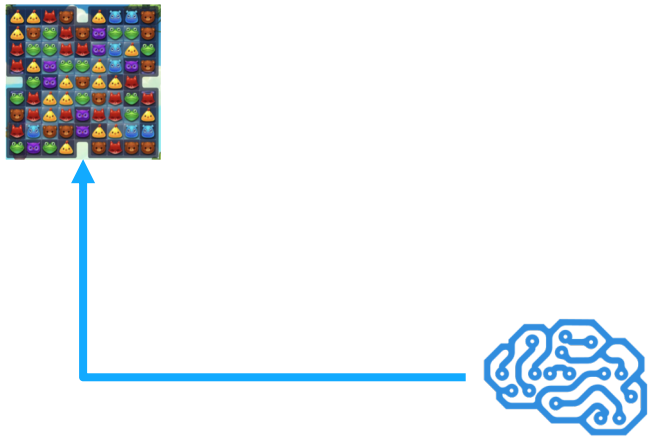
- Hypothesis: **Identify in-game parameters** to be altered dynamically
- Validation: Look for **correlation** between parameters and key metrics
- Prototype: Build **prototype**, verify and visualize offline
- Production: Deploy in **AB experiment**, analyze results and follow up

Happy Elements

Happy Elements

Problem Formulation

Gameplay modification: Action sequences

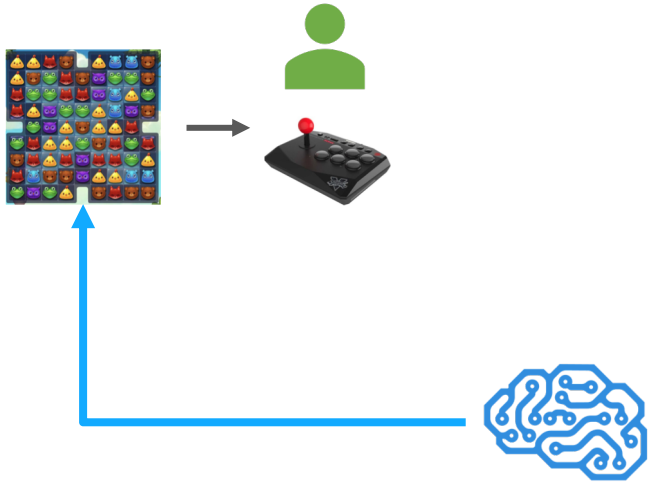


Objective: Rewards = player revenue/retention

Happy Elements

Problem Formulation

Gameplay modification: Action sequences

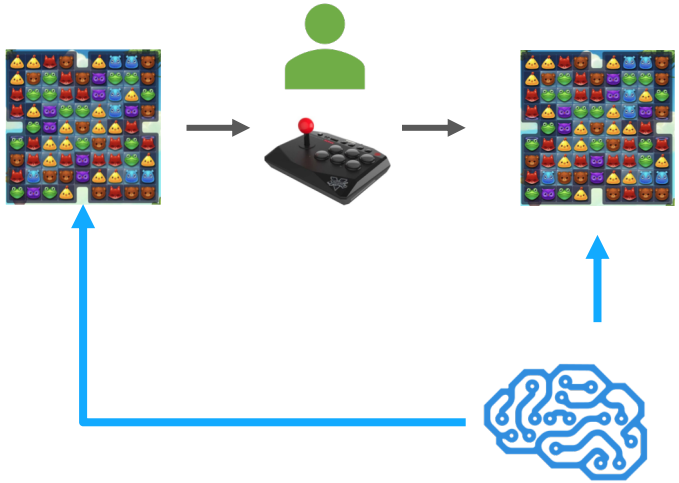


Objective: Rewards = player revenue/retention

Happy Elements

Problem Formulation

Gameplay modification: Action sequences

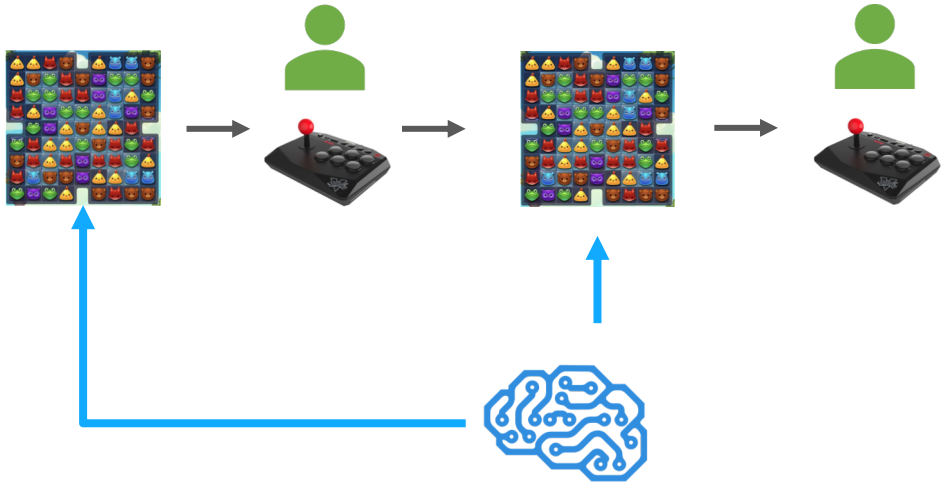


Objective: Rewards = player revenue/retention

Happy Elements

Problem Formulation

Gameplay modification: Action sequences

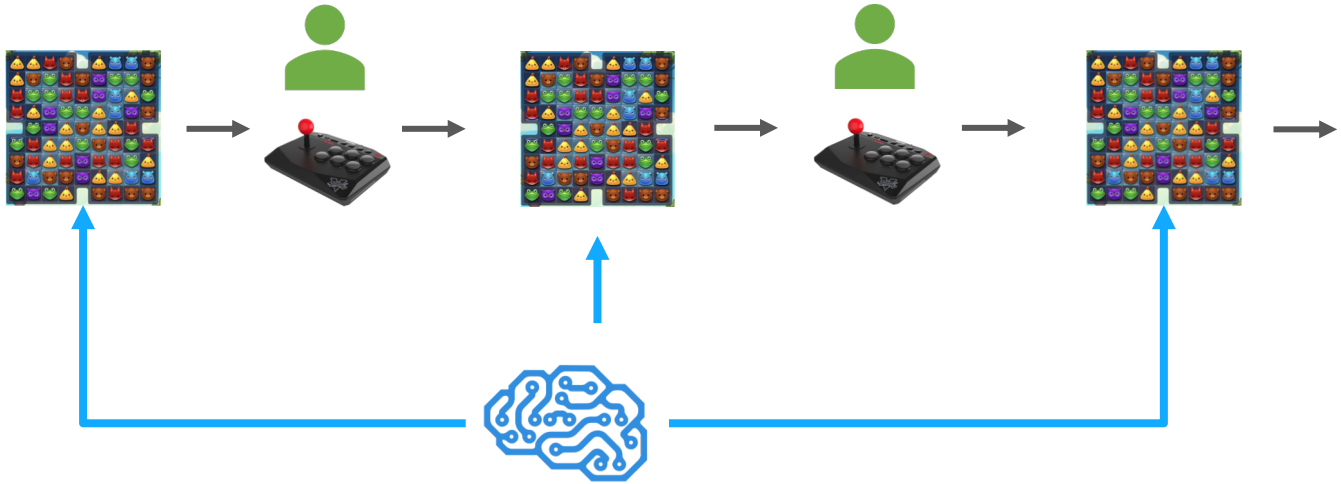


Objective: Rewards = player revenue/retention

Happy Elements

Problem Formulation

Gameplay modification: Action sequences



Objective: Rewards = player revenue/retention

Happy Elements

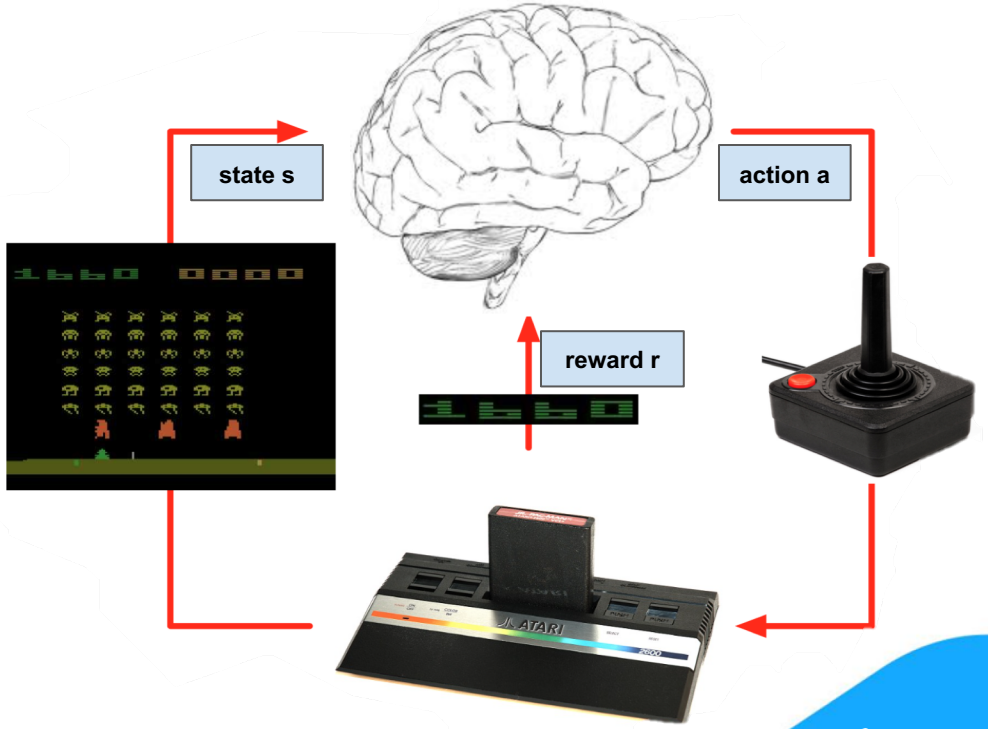
Deep Reinforcement Learning

- **Reinforcement learning** defines problem via high-level **objective**
- **Deep learning** is a **paradigm** for building flexible **solutions**
- **Deep reinforcement learning** integrates both above points

Reinforcement Learning

- What **action *a*** to take in **state *s*** to **optimize** the expected **reward E[*r*]**?

- For example, video game:
  - state s = screen
  - action a = controller
  - reward r = score

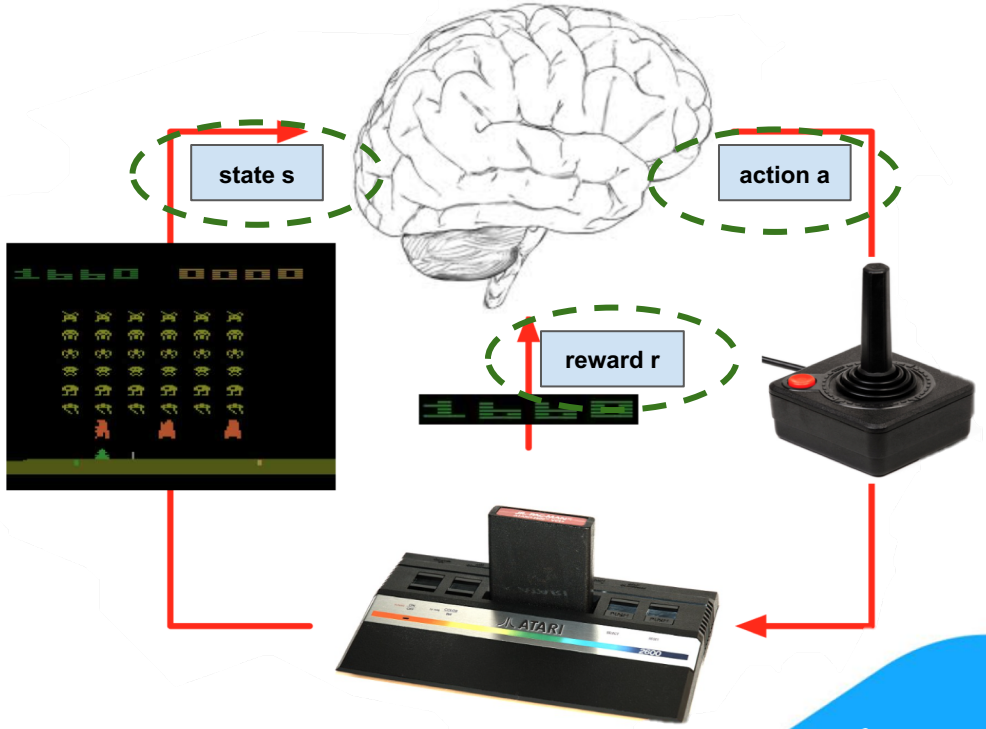


Happy Elements

Reinforcement Learning

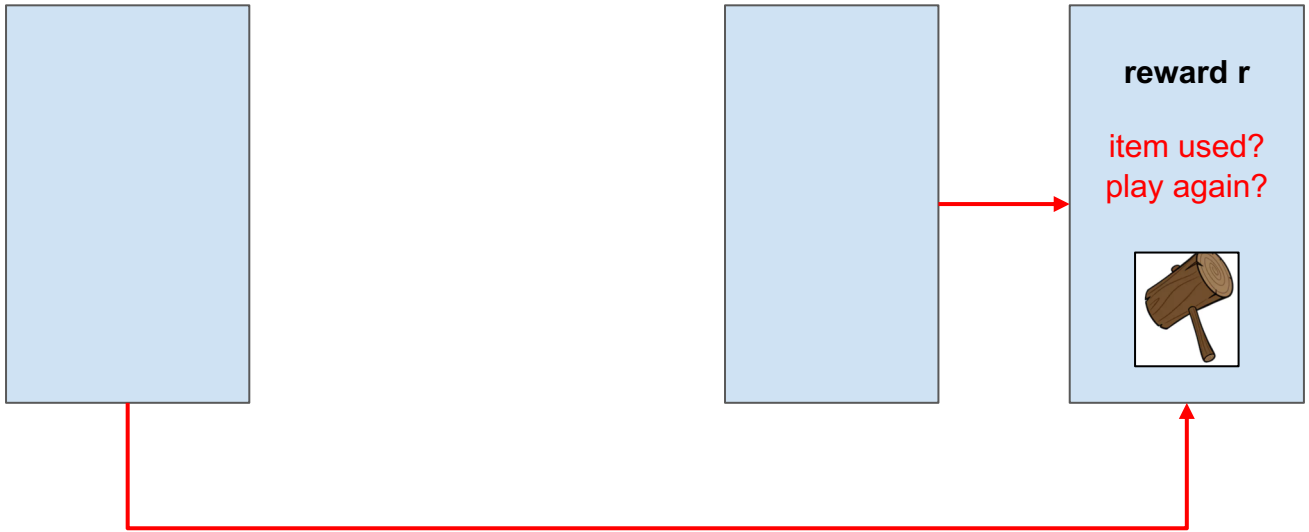
- What **action *a*** to take in **state *s*** to **optimize** the expected **reward E[*r*]**?

- For example, video game:
  - state s = screen
  - action a = controller
  - reward r = score



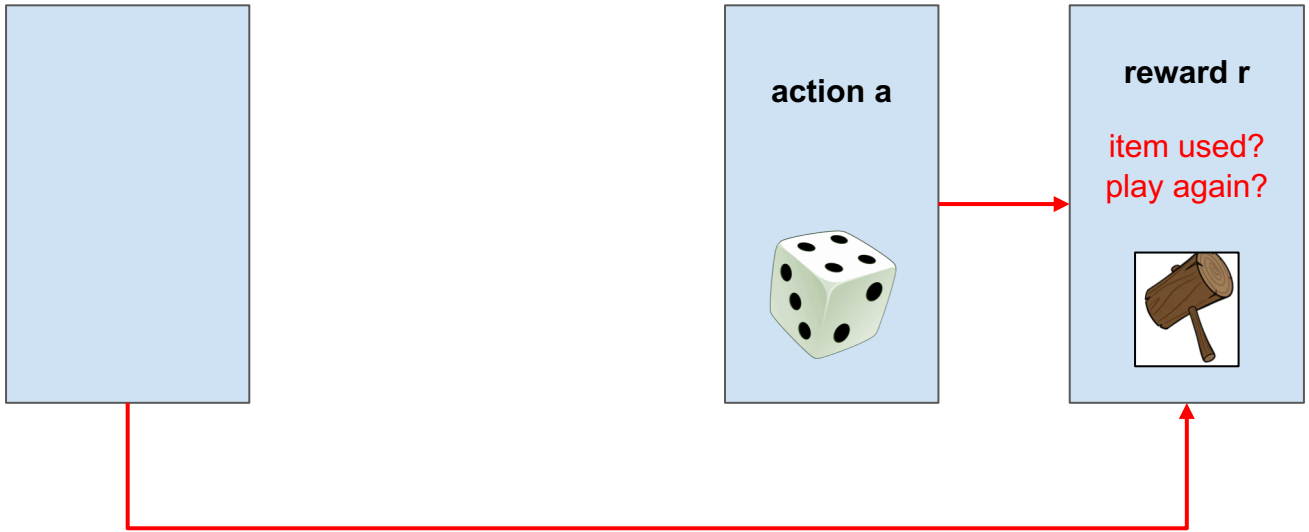
Happy Elements

Reinforcement Learning

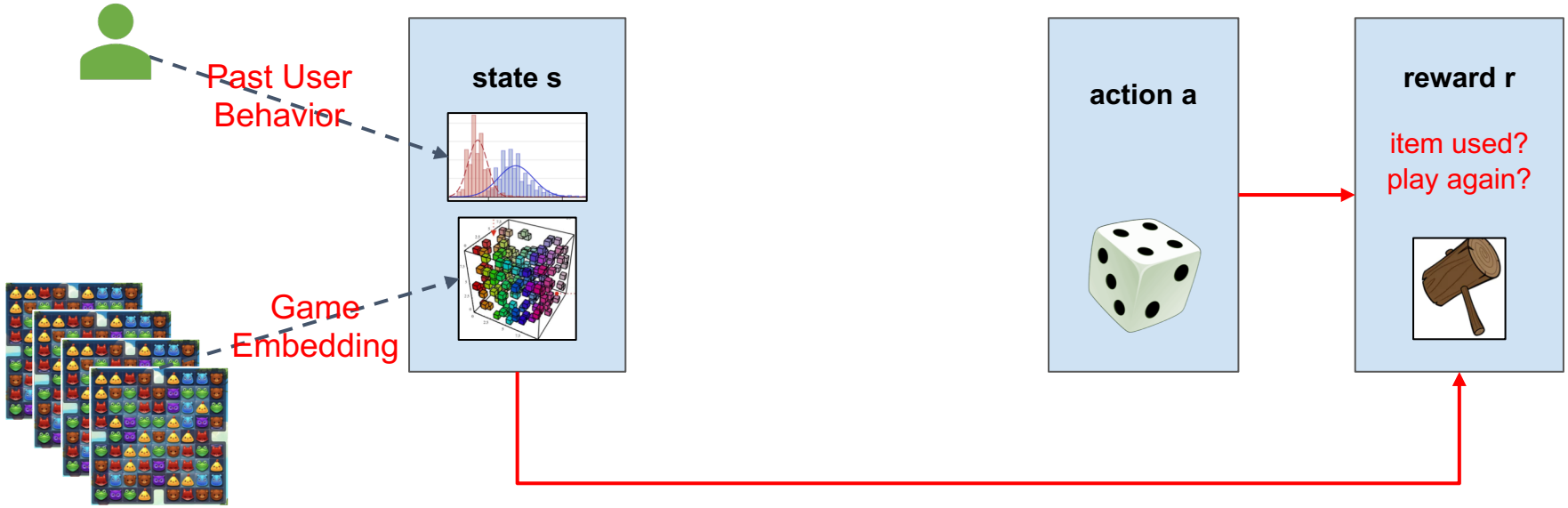


Happy Elements

Reinforcement Learning

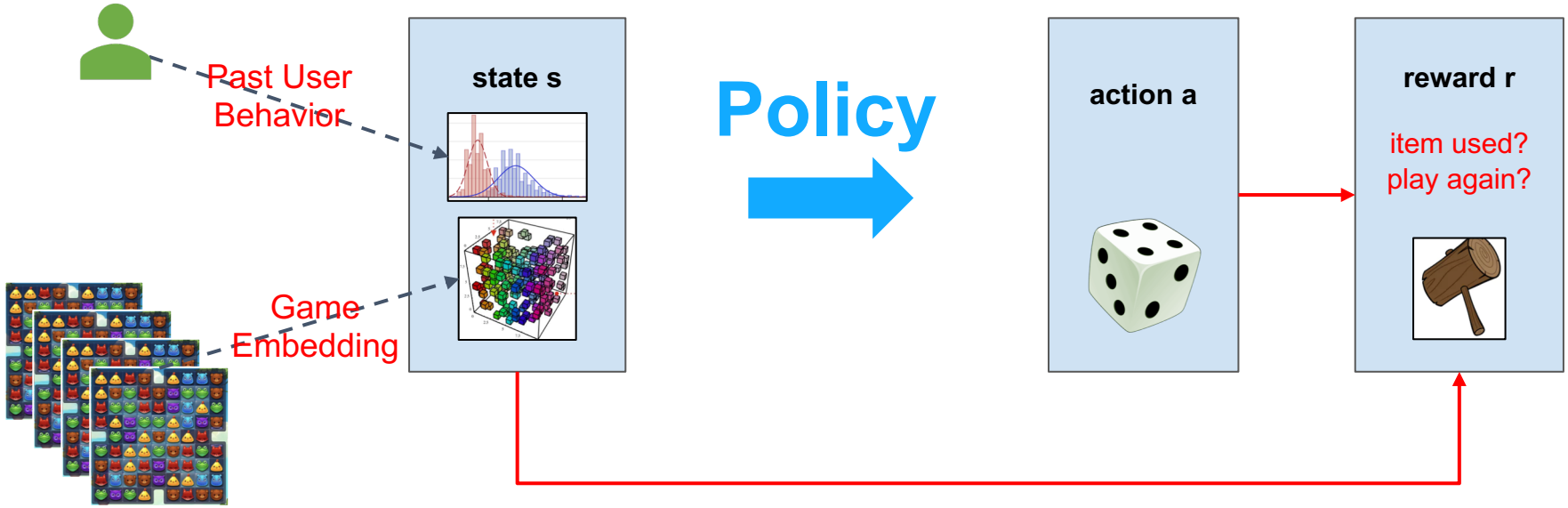


Reinforcement Learning



Happy Elements

Reinforcement Learning



Happy Elements

Deep Learning

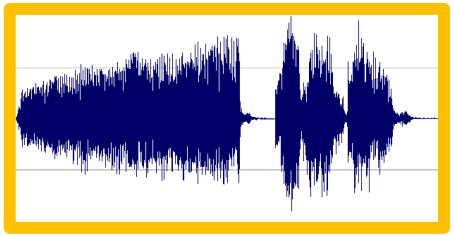
- Not just a ”fad”: **Paradigm shift** rather than new technique
- Ability to **optimize any sort** of target using **any type** of data flow
- Extremely **flexible** in fusing and integrating **heterogeneous data**

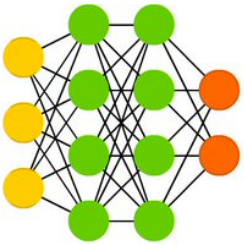
Happy Elements

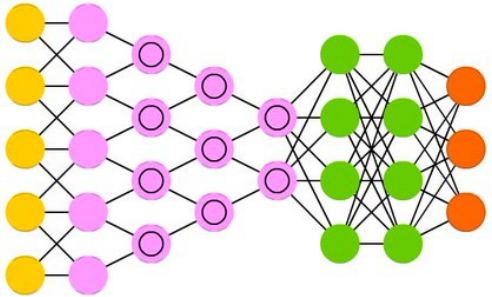
Deep Learning

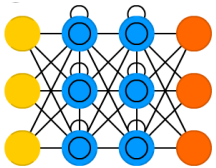
	A	B	C	D	E
1	Country	Salesperson	Order Date	OrderID	Order Amount
2	UK	Buxton	3/01/2018	10954	755
3	UK	Buxton	8/01/2018	10922	495
4	UK	Buxton	4/01/2018	10899	510
5	UK	Buxton	21/01/2018	10874	582
6	UK	Buxton	11/01/2018	10872	761
7	UK	Buxton	16/01/2018	10870	466
8	UK	Buxton	14/01/2018	10869	946
9	UK	Buxton	7/02/2018	10866	935
10					
11					

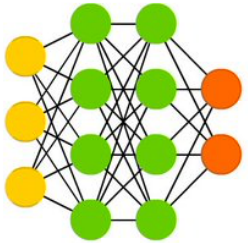
Buxton | Maxwell | Jarvis | Everton | Other Stuff



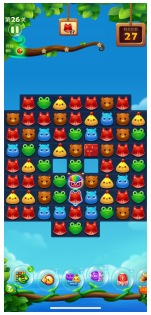


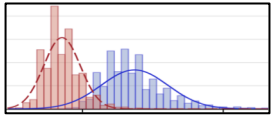


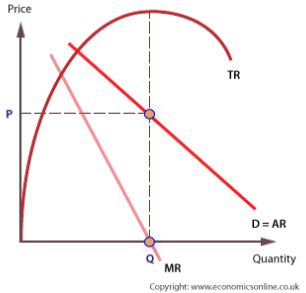


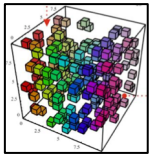


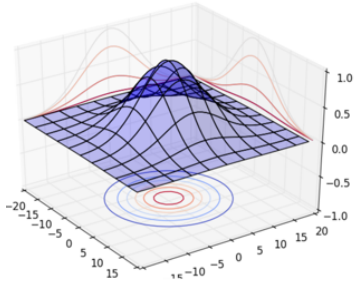
Happy Elements



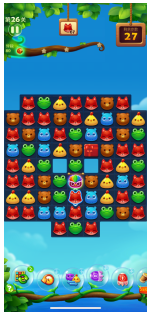


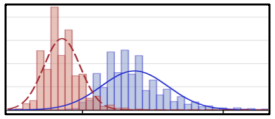


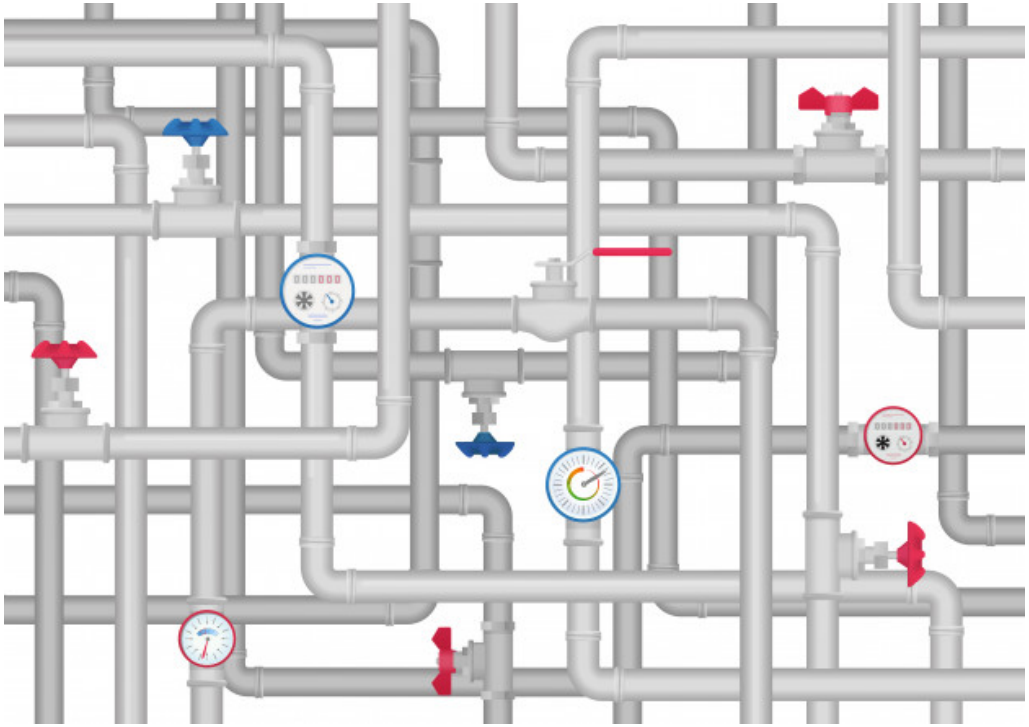


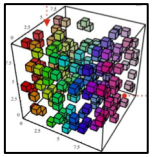


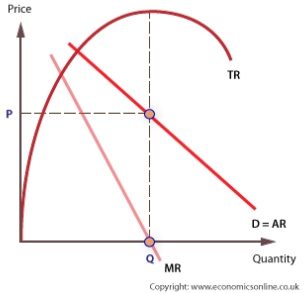
Happy Elements

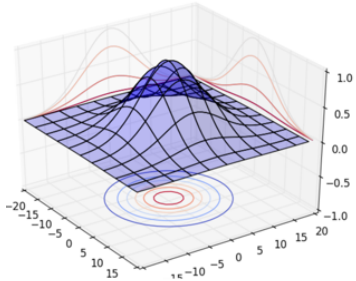




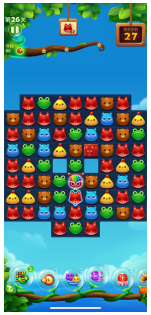


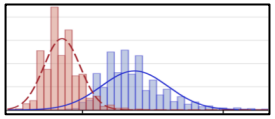


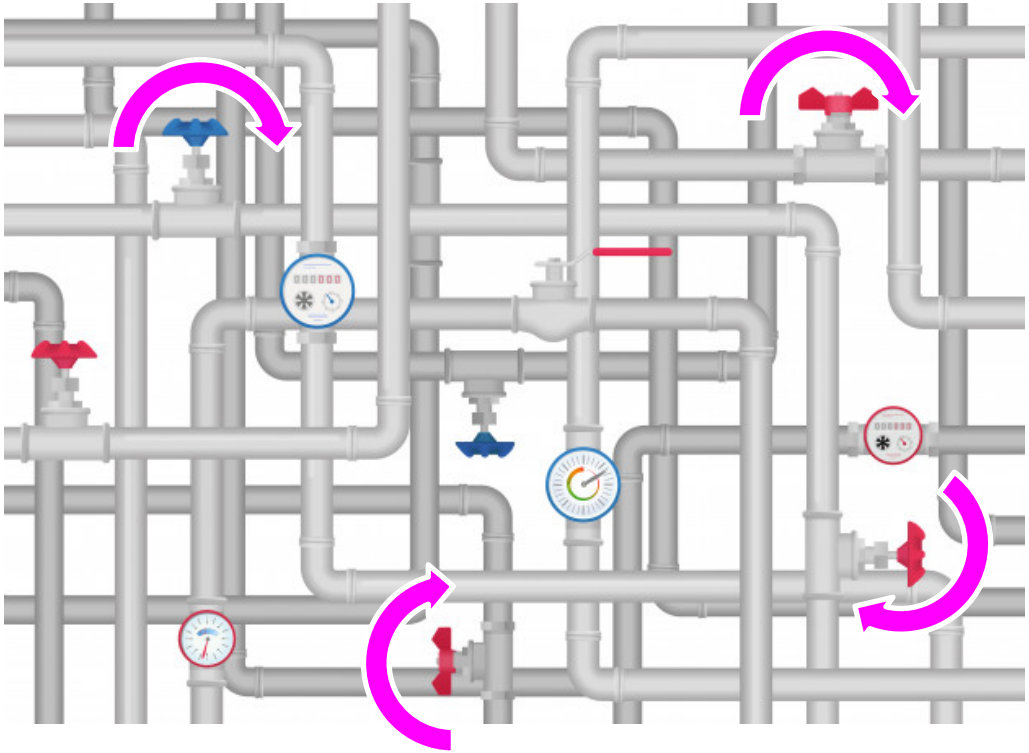


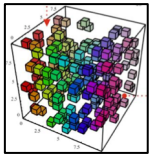


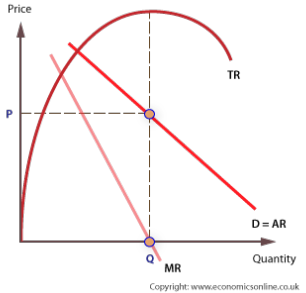
Happy Elements

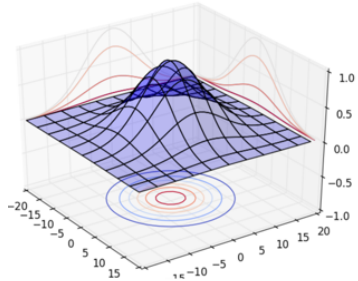






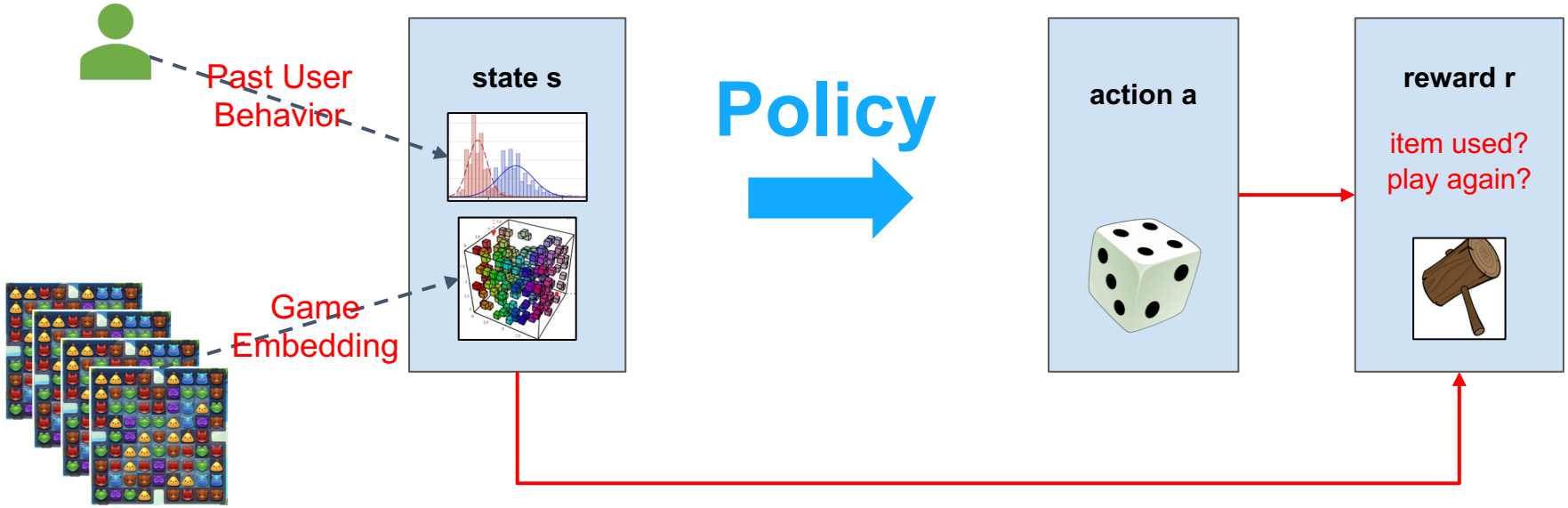






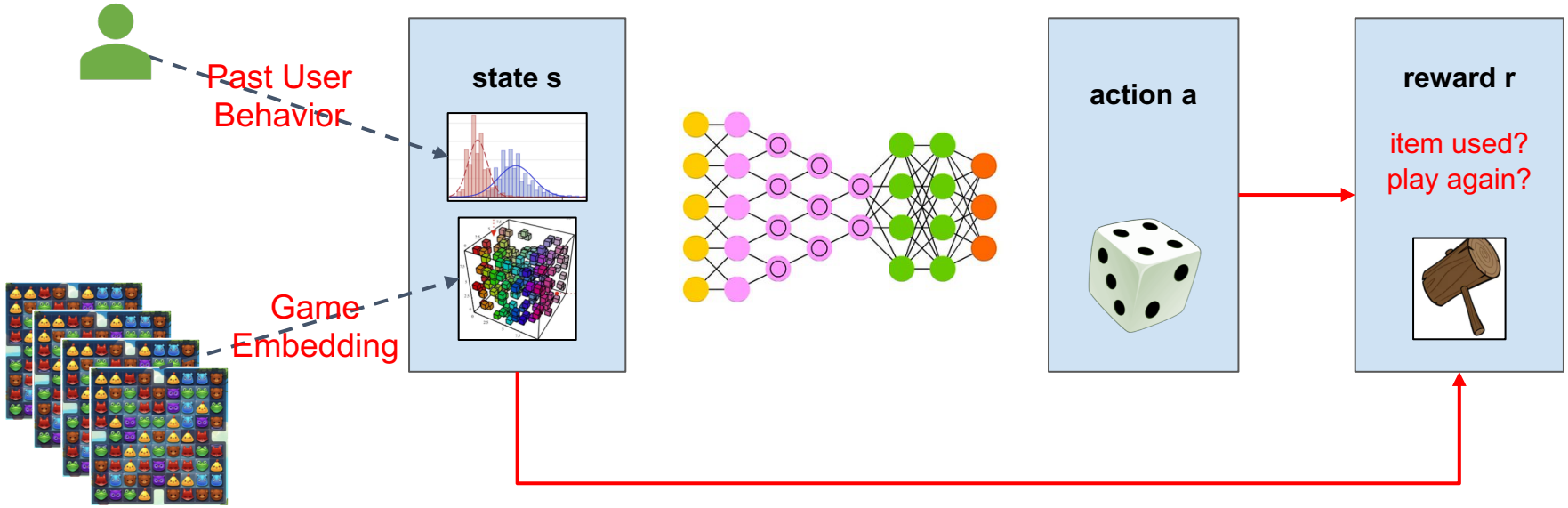
Happy Elements

Reinforcement Learning



Happy Elements

Deep Reinforcement Learning



Happy Elements

In Practice: Key Problems

- ML good at optimizing short-term targets:

  How do short-term targets relate to long-term objectives?

- ML good at optimizing on fixed dataset:

  What when the data regime is highly non-stationary?

Optimization Horizon

- Often it is easy to define **short-term targets**:

  Did the user play another in-game level?
  Did the user make an in-app purchase?

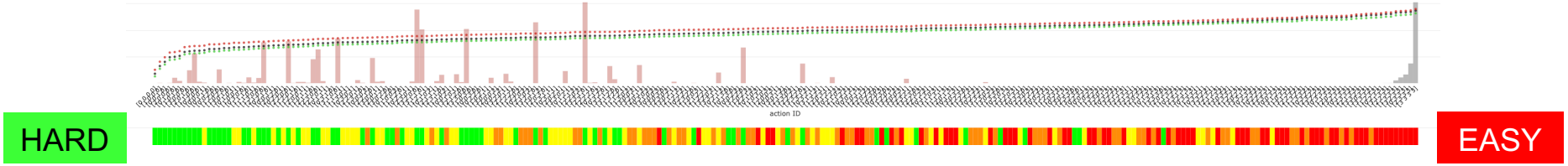
- But how does this lead to **long-term objectives**?

  User engagement over the next year
  Life-time value of player

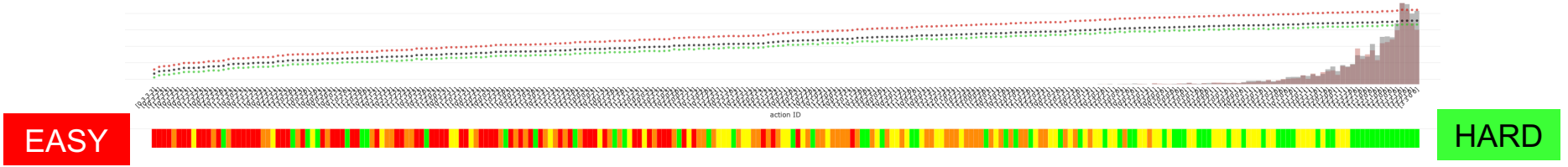
Happy Elements

Optimization Horizon

Retention



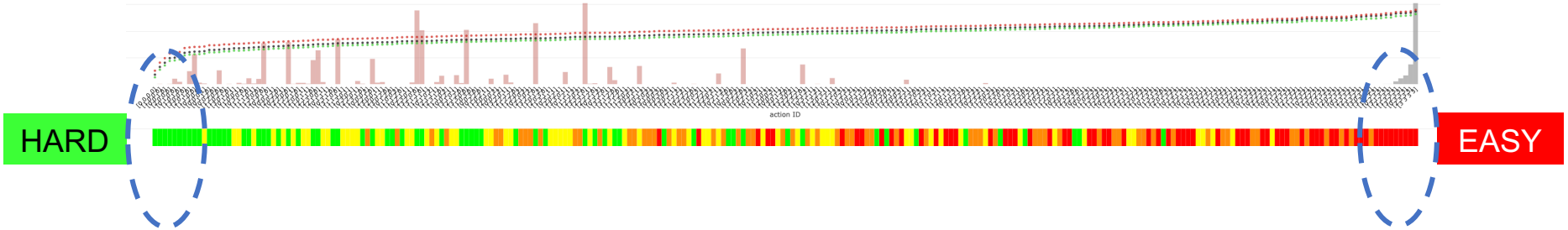
Revenue



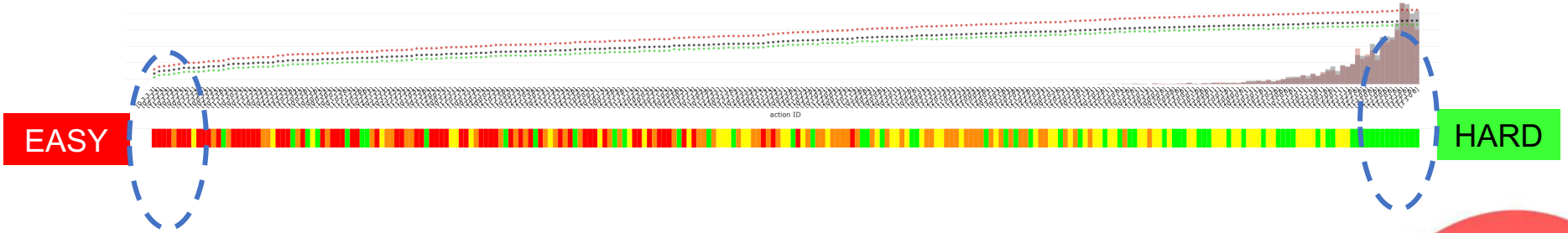
Happy Elements

Optimization Horizon

Retention



Revenue



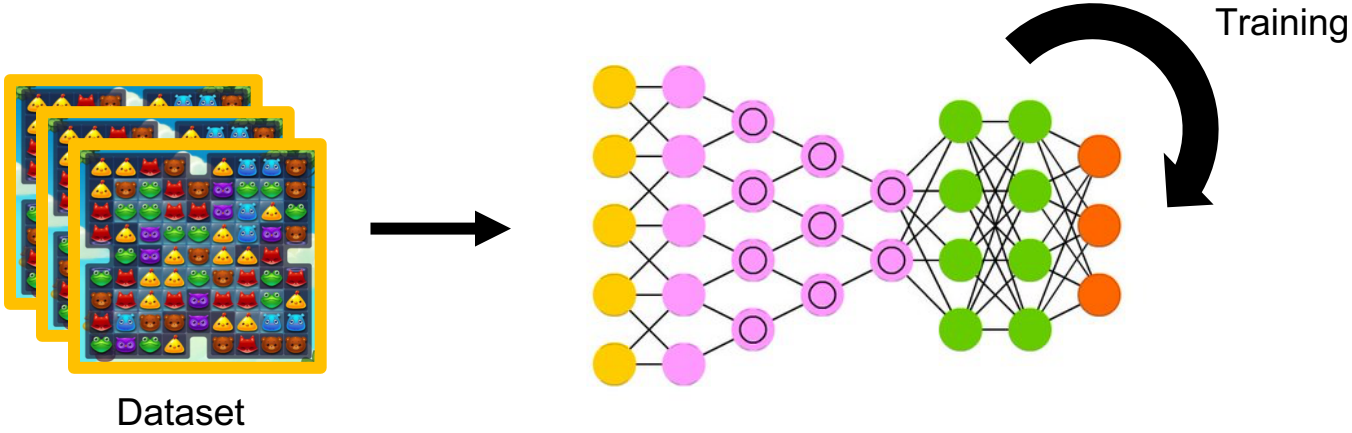
Happy Elements

Optimization Horizon

- This can be solved through **RL** by **formulating the right objective**
- Objective is a **sum of individual short-term targets** over a time horizon
- However problem remains in **how to accurate model** this objective

Happy Elements

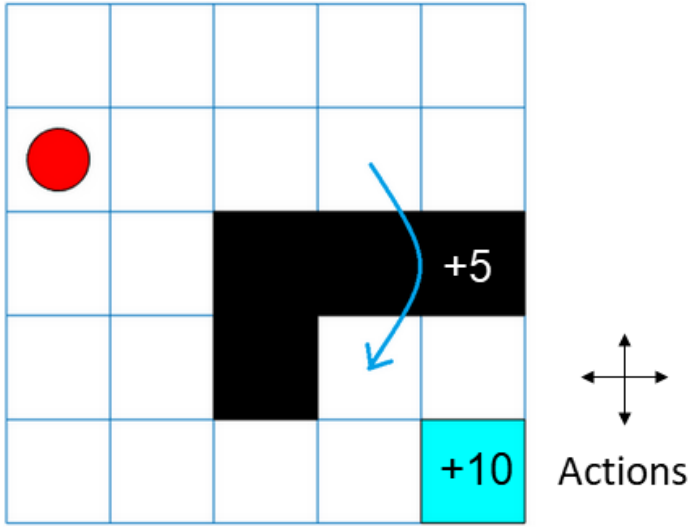
Exploration – Exploitation Duality



- Traditionally, ML works on a **fixed dataset**
- Practical RL in **constant motion**: model generates future dataset

Exploration – Exploitation Duality

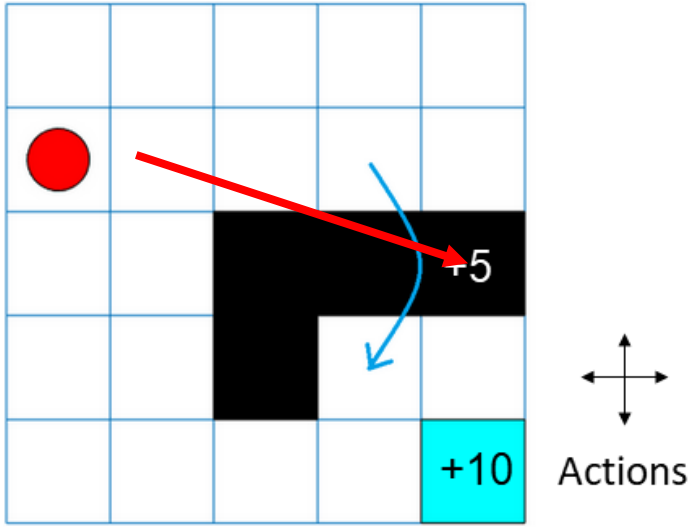
- Rewards might be **sparse**: learn from long-term signal
- + **dynamic interaction** with players: inherently **nonstationary** data regime
- Core problem: trading off **exploration vs exploitation**



Happy Elements

Exploration – Exploitation Duality

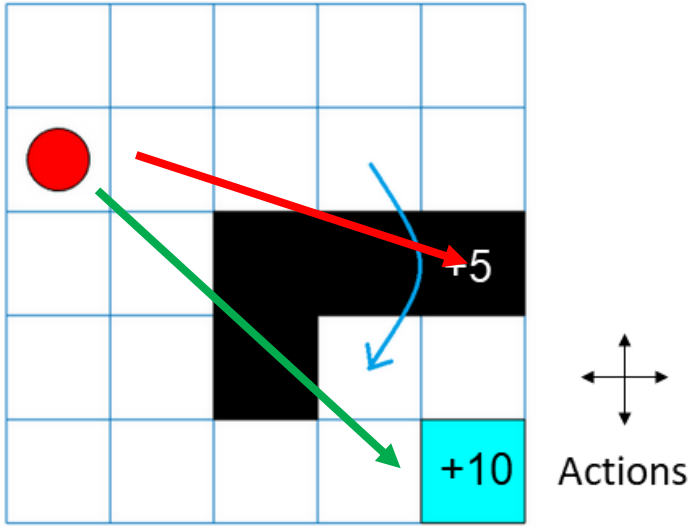
- Rewards might be **sparse**: learn from long-term signal
- + **dynamic interaction** with players: inherently **nonstationary** data regime
- Core problem: trading off **exploration vs exploitation**



Happy Elements

Exploration – Exploitation Duality

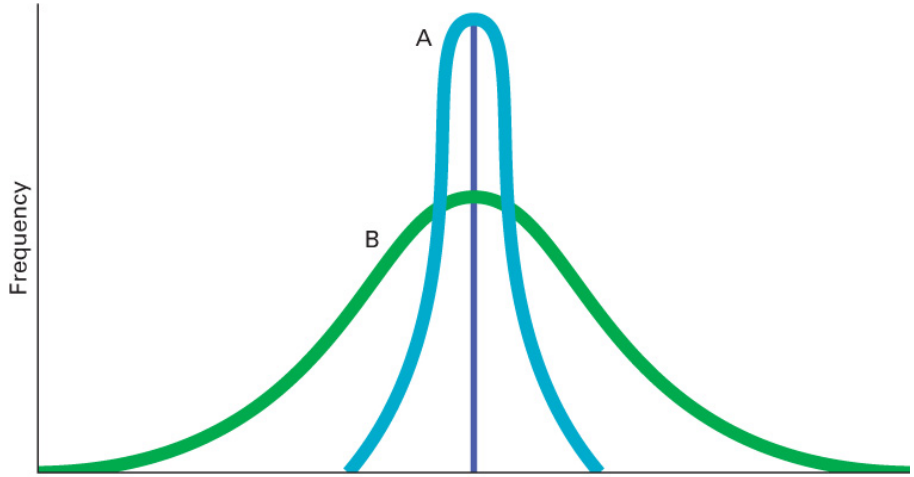
- Rewards might be **sparse**: learn from long-term signal
- + **dynamic interaction** with players: inherently **nonstationary** data regime
- Core problem: trading off **exploration vs exploitation**



Happy Elements

Exploration – Exploitation Duality

- Approximate Bayesian approach to capture **model uncertainty**
- Solving exploitation (A) vs. exploration (B) problem.



Happy Elements

Conclusions

Happy Elements

- Deployment of ML can **significantly improve revenue** and **engagement**
- **Nonstationary** data presents **difficult optimization** problem
- **Relationship** between **short**-term and **long**-term **metrics** hard to identify

Happy Elements

THANKS

http://en.happyelements.com/ai

References

- Neural network drawings: Fjodor van Veen
- Grid world diagram: MathWorks documentation