Artificial Intelligence Club Week 3 Slides

January 21th, 2026

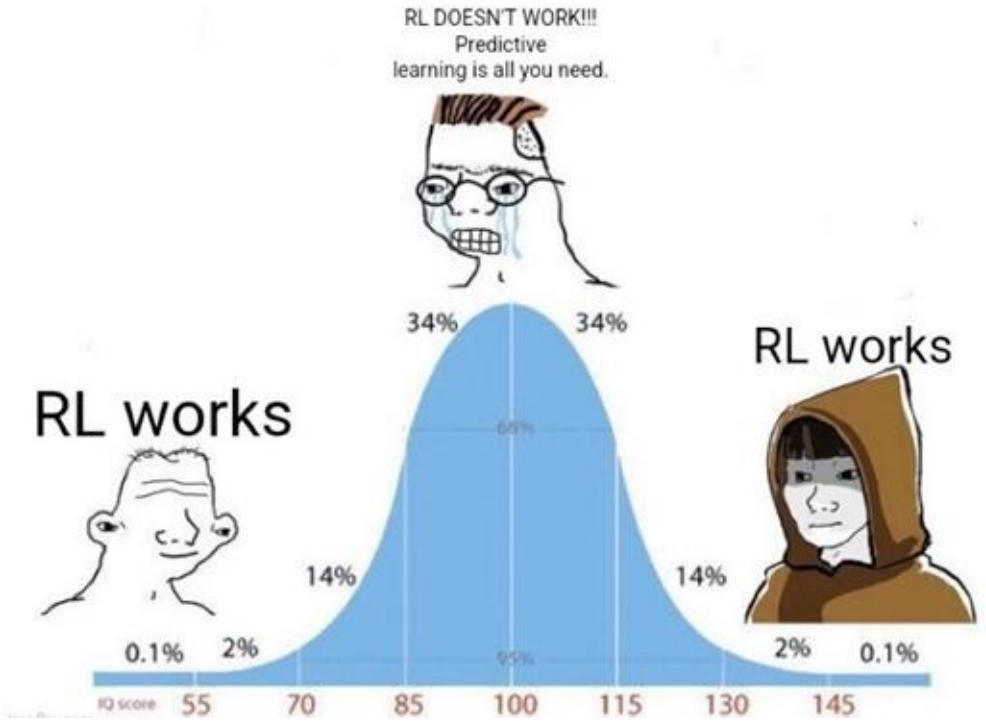
Please try to sit together near the front!

WEBSITE

INSTAGRAM

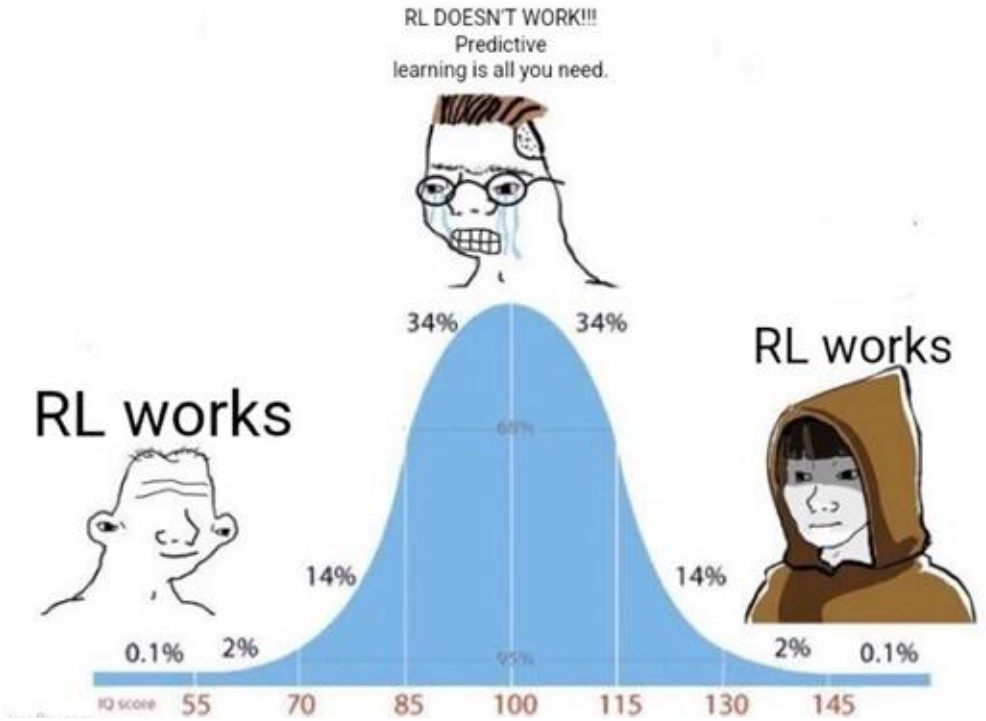
How Do We Learn in Real Life?

1. We try something



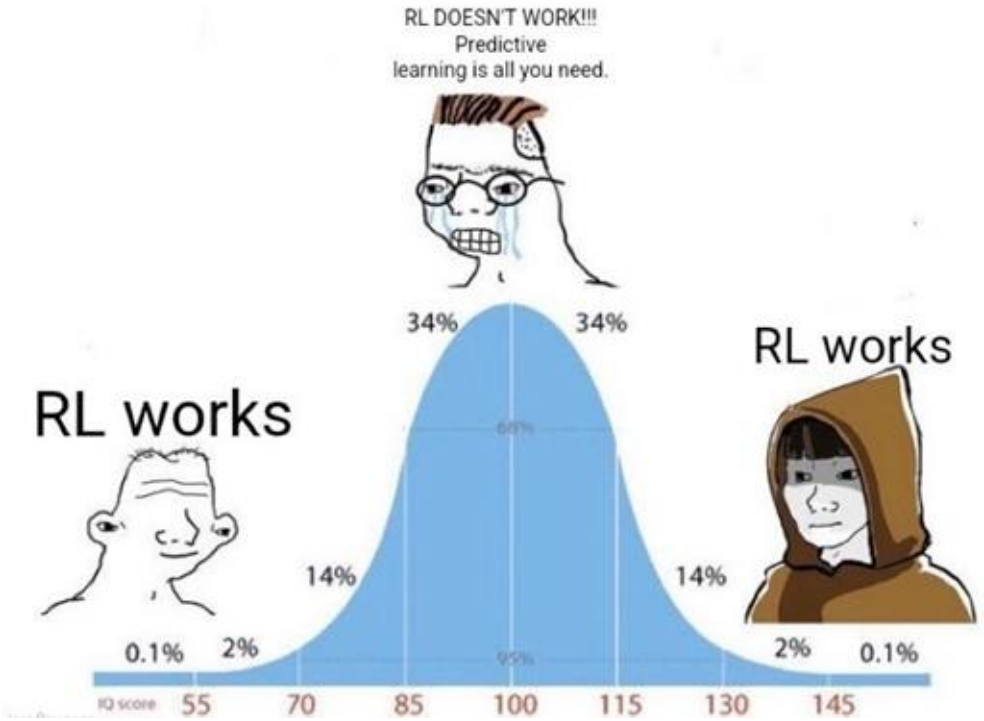
How Do We Learn in Real Life?

1. We try something
2. We see what happens



How Do We Learn in Real Life?

1. We try something
2. We see what happens
3. We adjust our behavior

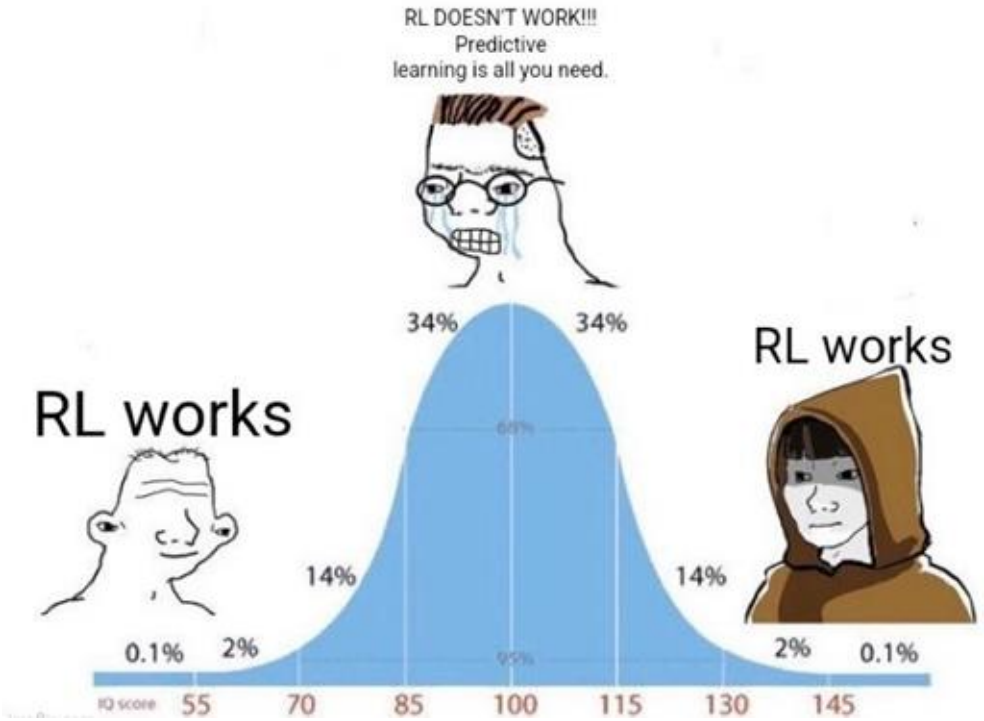


How Do We Learn in Real Life?

1. We try something
2. We see what happens
3. We adjust our behavior

Examples:

- Learning to ride a bike
- Playing a new video game
- Studying for an exam



Reinforcement Learning

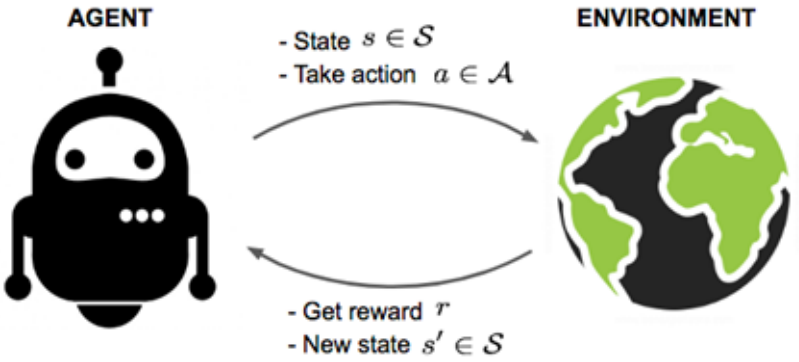
Reinforcement Learning is when an ***agent*** learns what to do by interacting with an ***environment*** and receiving ***rewards*** or ***penalties***.

NO labeled data

Learning comes from experience

Reinforcement Learning Loop

1. Agent observes the state
2. Agent takes an action
3. Environment respond
4. Agent receives a reward/penalty
5. Repeat



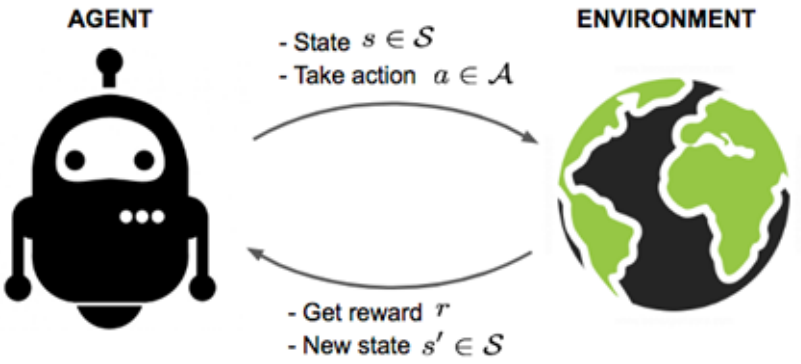
Where do you see the feedback here?

Reinforcement Learning Example

- Agent → Dog
- Environment → House / Outside
- Action → Sit, bark, jump
- Reward → Treat or praise
- Goal → Learn good behavior

Rewards ***guide*** behavior

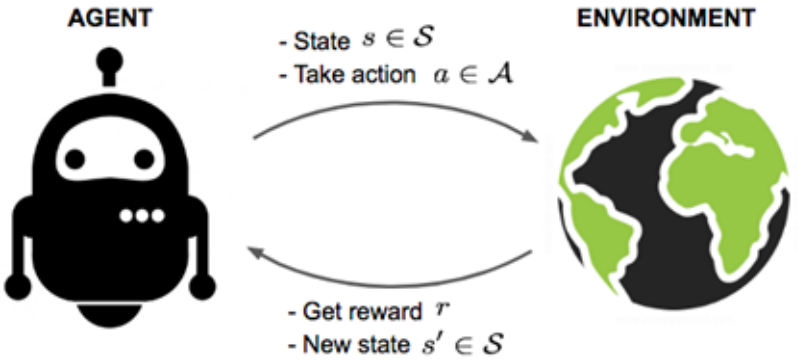
No one tells the dog the “correct formula”



RL IRL

- Video game AI
- Robot navigation
- Recommendation systems
- Self-driving decisions

RL is about decisions over time, not single predictions.

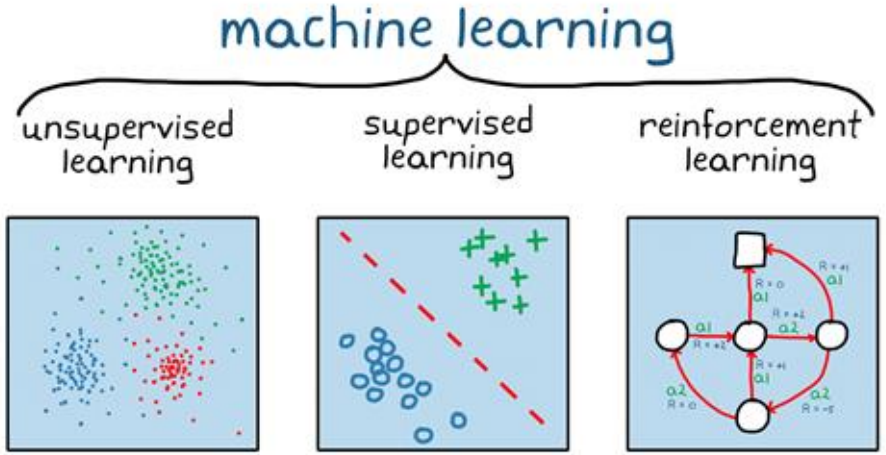


What makes RL Different?

- Supervised Learning → given correct answers
- Unsupervised Learning → find patterns
- Reinforcement Learning → learn from consequences

In RL, mistakes are expected

Exploration is part of learning



What makes RL hard

Multiple Things to Balance:

1. Long Term vs Short Term Rewards
2. Exploration vs Exploitation
3. Simple vs Complex Environment
4. Trial and Error

Q Learning

Q-learning is a method to learn which actions are best.

Learn a score for each (state, action) that estimates “how good” it is.

- State = situation (where you are)
- Action = move you can take
- Q(s,a) = predicted “future reward” if you take action a in state s

Q Learning

Q-learning is a method to learn which actions are best.

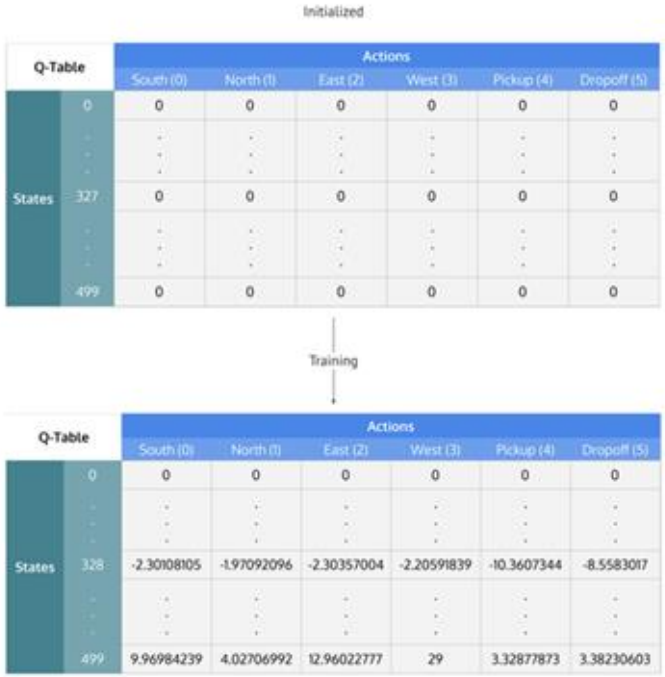
Learn a score for each (state, action) that estimates “how good” it is.

- State = situation (where you are)
- Action = move you can take
- Q(s,a) = predicted “future reward” if you take action a in state s

Q-Table: the agent's memory

- Rows = states
- Columns = actions
- Entries = Q-values

Every time the agent takes an action, it updates one Q-value.



The Math

New Q = Old Q + learning rate × (better estimate – old)

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$

Alpha (α) learning rate:

- High: learns fast, can be unstable

-Low: learns slow, more stable

Gamma (γ) discount factor:

- High: cares about long-term rewards

- Low: cares mostly about immediate rewards

What should the agent do?

How does the agent choose actions while learning?

- Exploit: pick highest Q-value action
- Explore: try random actions

Common strategy: epsilon-greedy

- With probability ε: random action
- Otherwise: best known action

When to use Q Learning

Strengths:

- Simple and effective in small/discrete worlds
- Learns without a model of the environment
- Eventually learns near-optimal behavior (with enough exploration)

Limitations:

- Q-table gets huge when states are large (images, continuous spaces)
- Needs lots of experience
- Sensitive to reward design (reward hacking)

LEVEL UP

- Q-learning: store a table
- DQN: use a neural network to approximate Q(s,a)

DQN replaces the table with a function approximator!

Feedback?

Survey