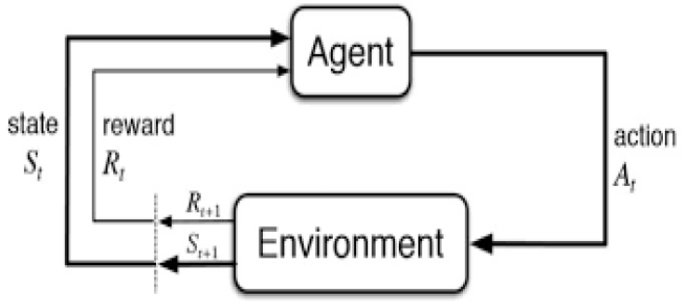
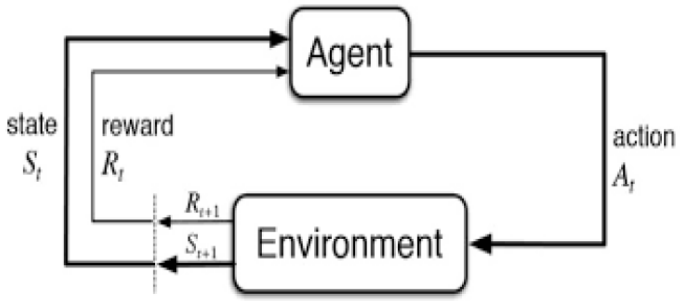
# RL model




* Identify,
  (i) Environment
  (ii) Agent
  (iii) States
  (iv) Actions
  (v) Reward

Example: RL based Trading Bot (stock market)




* Environment - Stock Market
* Agent - RL Algorithm (Q-Learning)
* States - # stocks in the hand
  Features: stock price
  money in the hand
  (Assuming trading is done for only one stock)
* Actions - Sell, buy, Hold
  Labels
  Sell all the stocks own
  buy stocks for all the money own

Assumptions
(i) single trade for a day
(ii) if sell, all the stocks will be sold
(iii) If buy, buy stocks for all the money own
(iv) 200$ given when money in the hand becomes 0

How Agent works.

Day 1: State: 1000, 5$, 200$
(# stocks, stock price, money in the hand.)
Agent → Action: sell
(random action since no previous experience)

Day 2: State: 0, 5-6$,

How Agent works.

(# stocks, stock price, money in the hand.)

$t = t$ Day 1: State: 1000, 5$, 200$

Agent → Action (sell) (random action since no previous experience)

$(R)_t = (5200 + 0 \times 5.6) - (200 + 5 \times 1000) = 0$

$t = t+1$ Day 2: State: 0, 5.6$, 5200$

Agent → Action (buy)

$R = (928 \times 4.7 + 200) - (5200) = -638.4$

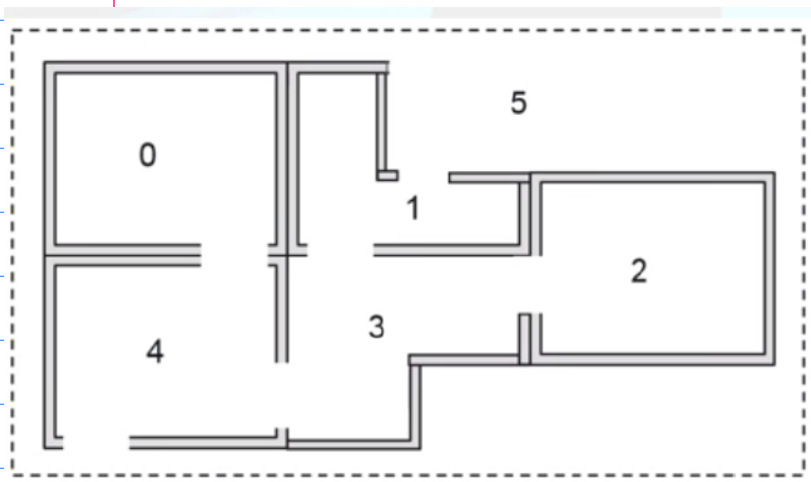
Day 3: State 928, 4.7$, 200$

$(\text{Reward})_t = (\text{Total Asset})_{t+1} - (\text{Total Asset})_t$

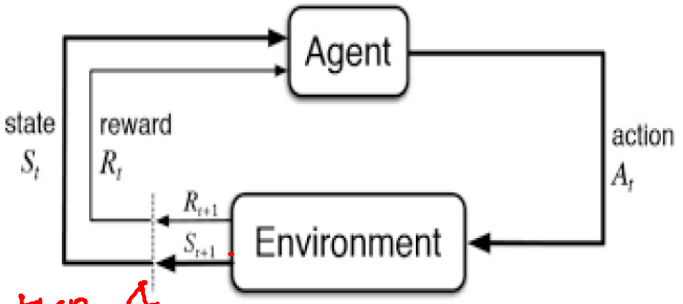
Total Asset = Money in the hand + (# stocks · stock price)

R matrix → RL Algorithm (Q-learning) → Q matrix

# A Simple Problem




Environment : Buildings
Agent : RL Agent (a learning)
states : Room #
Action : movement
0, 1, 2, 3, 4, 5




t=1 State 2 → action 3

t=2 State 3 $R_{2,3} = 0$ → action 4

($R_{3,4}$)

t=3 state 4 $R_{3,4} = 0$ → action 0

t=4 state 0 $R_{4,0} = 0$ → action 4

t=5 state 4 $R_{0,4} = 0$ → action 5

state 5 $R_{4,5} = 100$ (goal ↑)

t=6

R matrix

| state / actions | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | | | | | 0 | |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | 0 | | | |
| 4 | 0 | | | 0 | | |
| 5 | | | | | 100 | |

after several episodes,

t=1

Reward matrix 0

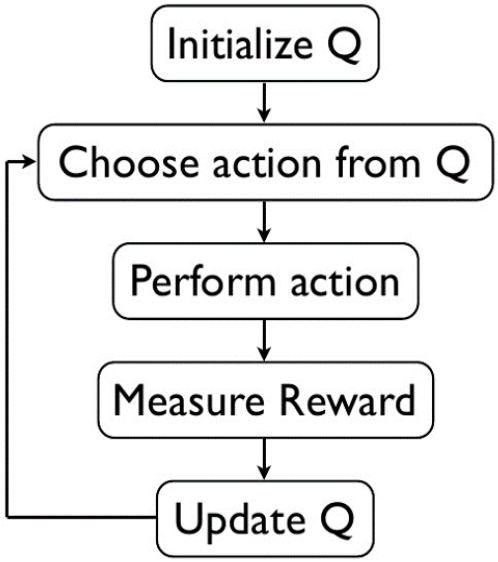
| State \ Action | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | -1 | -1 | -1 | -1 | 0 | -1 |
| 1 | -1 | -1 | -1 | 0 | -1 | 100 |
| 2 | -1 | -1 | -1 | 0 | -1 | -1 |
| 3 | -1 | 0 | 0 | -1 | 0 | -1 |
| 4 | 0 | -1 | -1 | 0 | -1 | 100 |
| 5 | -1 | 0 | -1 | -1 | 0 | 100 |

The -1's in the table represent null values

(-1) → impossible actions

Q matrix

using Q Learning Algorithm




Q Learning Algorithm

$$Q(state, action) \leftarrow (1-\alpha)Q(state, action) + \alpha\left(reward + \gamma \max_{a} Q(next\ state, all\ actions)\right)$$

Action

| State | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | -1 | -1 | -1 | -1 | 0 | -1 |
| 1 | -1 | -1 | -1 | 0 | -1 | 100 |
| 2 | -1 | -1 | -1 | 0 | -1 | -1 |
| 3 | -1 | 0 | 0 | -1 | 0 | -1 |
| 4 | 0 | -1 | -1 | 0 | -1 | 100 |
| 5 | -1 | 0 | -1 | -1 | 0 | 100 |

R =

The -1's in the table represent null values

Action

| States | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0. | 0. | 0. | 0. | 80. | 0. |
| 1 | 0. | 0. | 0. | 64. | 0. | 100. |
| 2 | 0. | 0. | 0. | 64. | 0. | 0. |
| 3 | 0. | 80. | 51.2 | 0. | 80. | 0. |
| 4 | 64. | 0. | 0. | 64. | 0. | 100. |
| 5 | 0. | 80. | 0. | 0. | 80. | 100. |

R Matrix

Q matrix