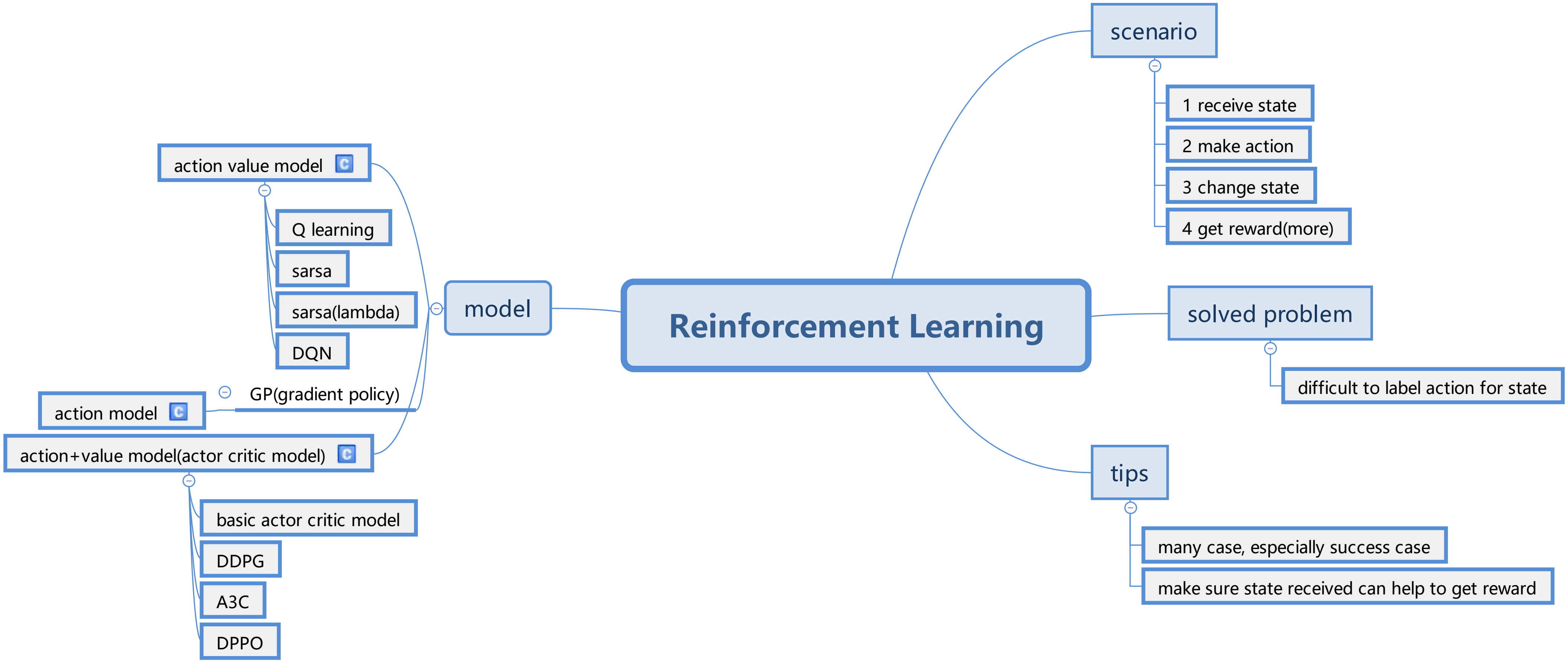
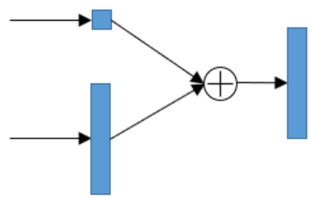
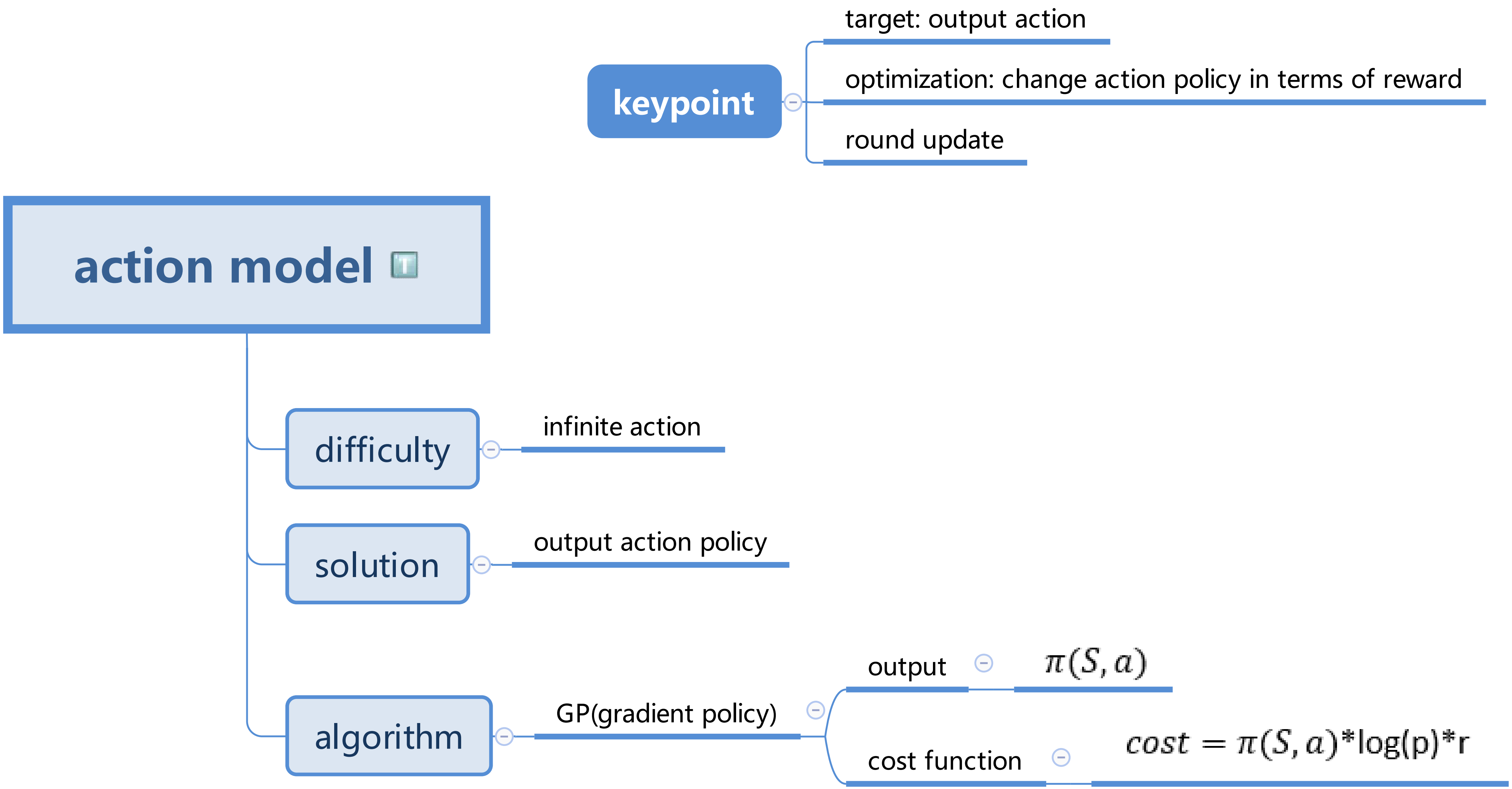


Reinforcement Learning
scenario
1 receive state
2 make action
3 change state
4 get reward(more)
solved problem
difficult to label action for state
tips
many case, especially success case
make sure state received can help to get reward
model
action value model
Q learning
sarsa
sarsa(lambda)
DQN
GP(gradient policy)
action model
action+value model(actor critic model)
basic actor critic model
DDPG
A3C
DPPO

action value model

- keypoint
  - 1 mapping between action value and state&action
  - 2 choose action with biggest value
- Q learning
  - mapping form
    - table
  - optimized function
    - $Q(s,a) \coloneqq Q(s,a) + \alpha(R + \gamma \max(Q(S',)) - Q(s,a))$
  - action attempt
- sarsa
  - optimized function
    - $Q(s,a) \coloneqq Q(s,a) + \alpha(R + \gamma Q(S',a') - Q(s,a))$
- sarsa(lambda)
  - optimized function
    - $$Q(s,a) \coloneqq Q(s,a) + \alpha E(\lambda)(R - Q(s,a))$$
      $$E(\lambda) = \begin{cases} 1, if\ S,a \\ \lambda \dot{E}(\lambda), if\ \sim(S,a) \end{cases}$$
- DQN
  - mapping form
    - deep neural network
  - difficulty
    - 1 infinite state
    - 2data dependency
  - solution
    - 1 neural network
    - 2 two sets of parameters
    - 2 memory
  - cost function
    - $cost = (R + \gamma \max(Q(S',)) - Q(s,a))^2$
  - algorithm
    - double DQN
      - difficulty
        - overestimate
      - solution
        - cost function
          - $cost = (R + \gamma Q(S',a',\dot{\theta}) - Q(s,a,\theta))^2$
    - prioritized experience replay DQN
      - difficulty
        - convergence
    - dueling DQN
      - difficulty
        - convergent velocity
      - solution
        - output




action model
keypoint
target: output action
optimization: change action policy in terms of reward
round update
difficulty
infinite action
solution
output action policy
algorithm
GP(gradient policy)
output
$\pi(S,a)$
cost function
$cost = \pi(S,a)$*log(p)*r

action+value model(actor critic model)

- difficulty
  - efficiency: round update
  - convergency
- solution
  - add critic model
  - reduce data dependency
- algorithm
  - basic actor critic model
    - cost function
      - ccost = $(\mathrm{R}+\gamma c(S')-c(S))^2$
      - $acost = td * \pi(S,a)$*log(p), td = $(\mathrm{R}+c(S',a')-c(S,a))^2$
  - DDPG
    - output
      - action
    - cost function
      - ccost = $(\mathrm{R}+\gamma c(S',a(S',\tau'),\theta')-c(S,a,\theta))^2$
      - $acost = -c(S,a(S,\tau),\theta')$
  - A3C
    - data collection
      - parallel environment
  - DPPO
    - data collection
      - parallel environment
    - cost function
      - ccost = $(\mathrm{R}+\gamma c(S',\theta')-c(S,\theta))^2$
      - $acost = -td * \min(\frac{\pi(S,a,\tau)}{\pi(S,a,\tau')}, clip\left(\frac{\pi(S,a,\tau)}{\pi(S,a,\tau')}, 1-\varepsilon, 1+\varepsilon\right))$
        td=$(\mathrm{R}+\gamma c(S',\theta')-c(S,\theta'))^2$