Reinforcement learning (cont)

- Deep Q-Networks (DQN) (still value-Based)
  - what it dose: using Neural Networks to approximate the Q function (insted of a table which breaks down in big / continous enviroments)
  - key inovation (2015, Deepmind): Apply deep learning + Q-functions
  - Ex: instead of storing all possible board states in memory on a table (action vs states 2D Grid) which is impossible the DQN learns "Features" of the game to predict Q values
- So Q-learning = tabular (small problems), DQN = Q learning + deep NN (big / complex problems)

- Policy Gradients (poicly Based)
  - What it dose: Directly learn the policy $\pi(a|s)$ (Probability of taking each action in a given state)
  - use gradient asent to adjust policy parameters to maximize expected award
  - Example: in a continous action (like steering angle for car) its easier to directly learn "how much to turn" insted of assigning Q-values to infintely many actions

- Actor-Critic Method (combines both)
  - What it dose: actor = learns policy (decides which action to take
    critic = learns the value function (estimates how good the action was)
  - they work together: actor suggests action → critic judges it and gives Feedback (good or bad) → Actor updates policy based on this feedback
  - Why usefull: more stable that PG, good for continous/complex action spaces
  - Ex: A3C, PPO

Scanned with CamScanner