	Sequential (as opposed to one-shot)	**Evaluative** (as opposed to supervised)	**Sampled** (as opposed to exhaustive)
Supervised learning	×	×	✓
Planning (Chapter 3)	✓	×	×
Bandits (Chapter 4)	×	✓	×
Tabular reinforcement learning (Chapters 5, 6, 7)	✓	✓	×
Deep reinforcement learning (Chapters 8, 9, 10, 11, 12)	✓	✓	✓

Sequential feedback

1. Reward can be delayed.
2. Hard to assign credit for rewards.

Immediate feedback

1. Supervised learning.
2. Rewards are assigned to the action just taken.

Evaluative feedback

1. goodness of feedback is relative.
2. " is it the best !!"

Supervised feedback

1. Classification problem.
2. No guessing; if the model makes a mistake, the correct answer is provided immediately.

Sampled feedback

1. Generalisation of gathered feedbacks.
2. Supervised learning

Exhaustive feedback

1. Has access to all possible samples.
2. Tabular reinforcement learning.

Function approximation RL.

- high dimensions of state and action space.
- continuous state and action space.
- Through this, we can use generalization.

$$Q(s, a; \theta_i)$$

Ideal objective.

$$L_i(\theta_i) = \mathbb{E}_{s,a}\left[\left(q_*(s,a) - Q(s,a;\theta_i)\right)^2\right]$$

Where Optimal action-value function is

$$q_*(s,a) = \max_\pi \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right], \forall s \in S, \forall a \in A(s)$$

On-policy and off-policy TD targets

on-policy

$$y_i^{\text{Sarsa}} = R_{t+1} + \gamma Q\left(S_{t+1}, A_{t+1}; \theta_i\right)$$

off-policy

$$y_i^{Q\text{-learning}} = R_{t+1} + \gamma \max_a Q\left(S_{t+1}, a; \theta_i\right)$$

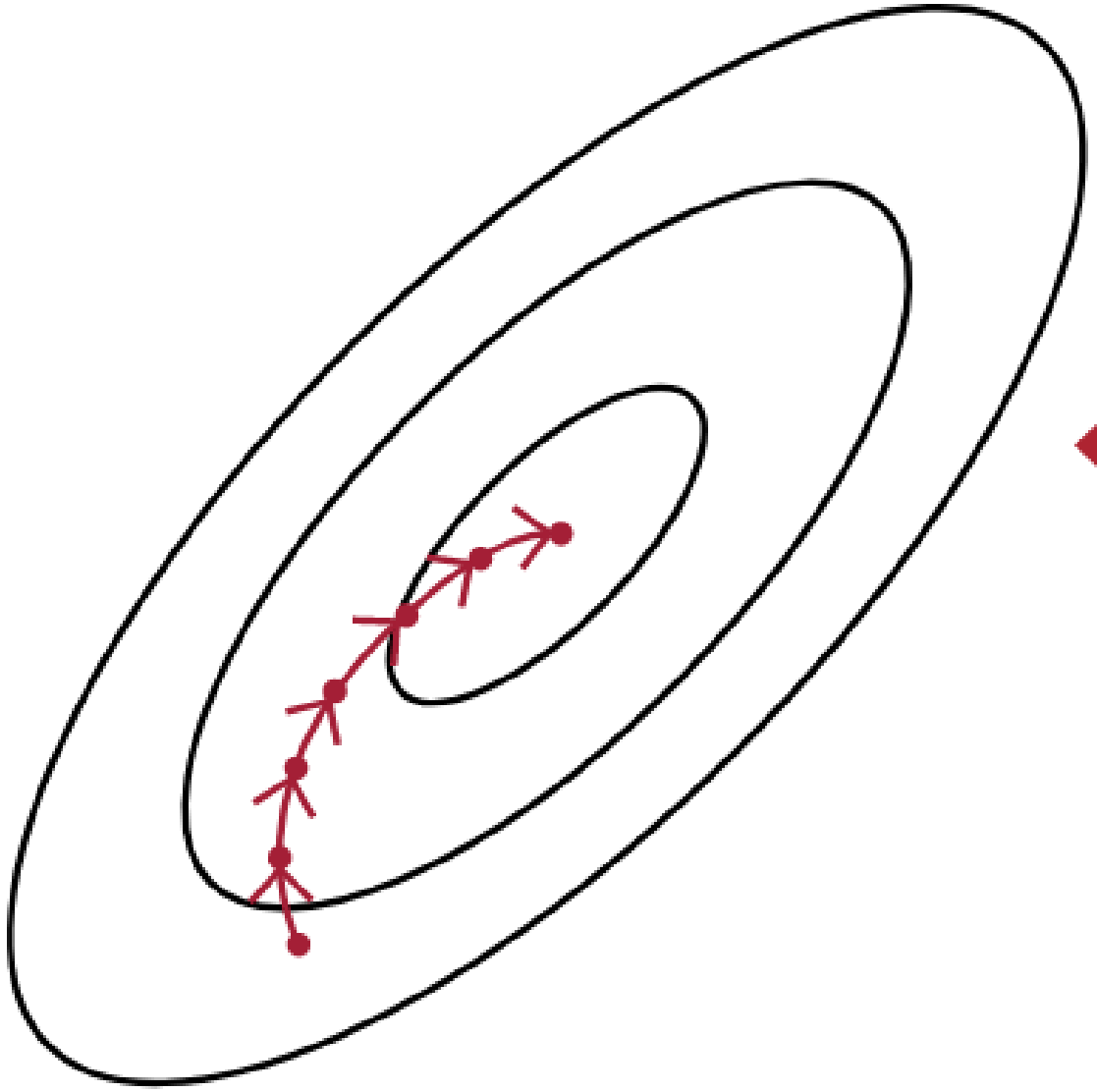
Q-learning target, an off-policy TD target

$$y_i^{Q\text{-learning}} = R_{t+1} + \gamma \max_a Q\left(S_{t+1}, a; \theta_i\right)$$

$$L_i\left(\theta_i\right) = \mathbb{E}_{s,a,r,s'}\left[\left(r + \gamma \max_{a'} Q\left(s', a'; \theta_i\right) - Q\left(s, a; \theta_i\right)\right)^2\right]$$

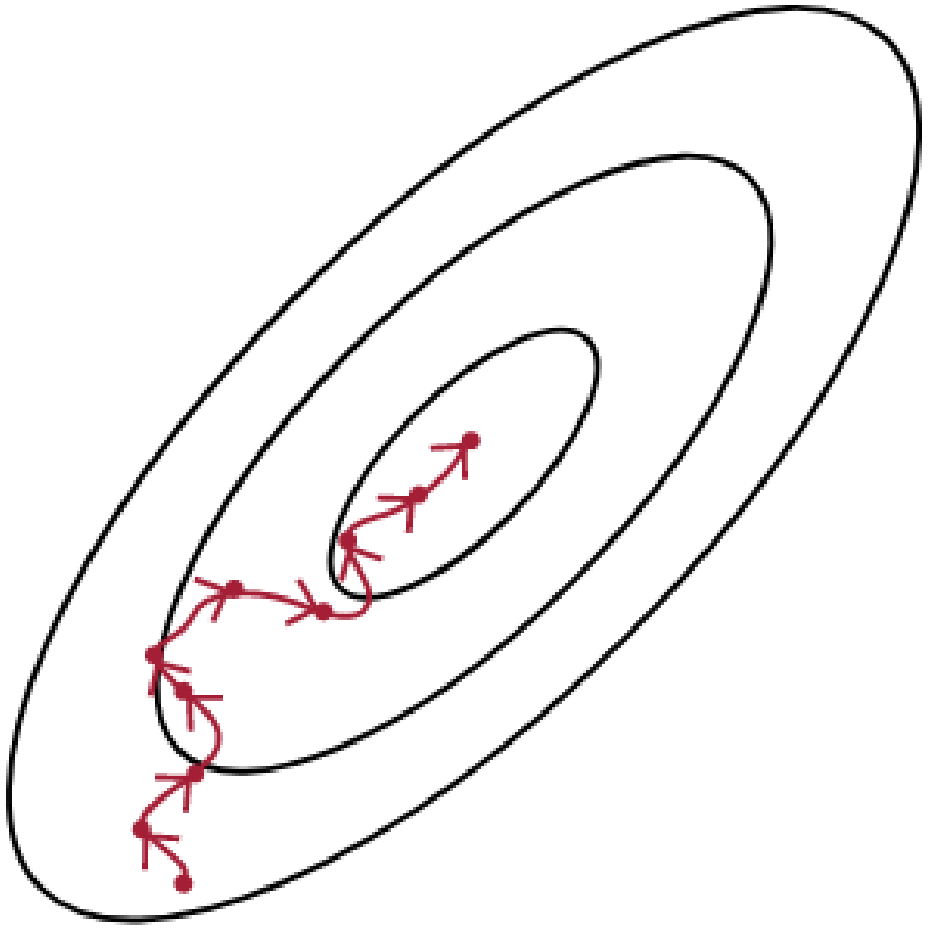
$$\nabla_{\theta_i} L_i\left(\theta_i\right) = \mathbb{E}_{s,a,r,s'}\left[\left(r + \gamma \max_{a'} Q\left(s', a'; \theta_i\right) - Q\left(s, a; \theta_i\right)\right) \nabla_{\theta_i} Q\left(s, a; \theta_i\right)\right]$$

Batch gradient descent



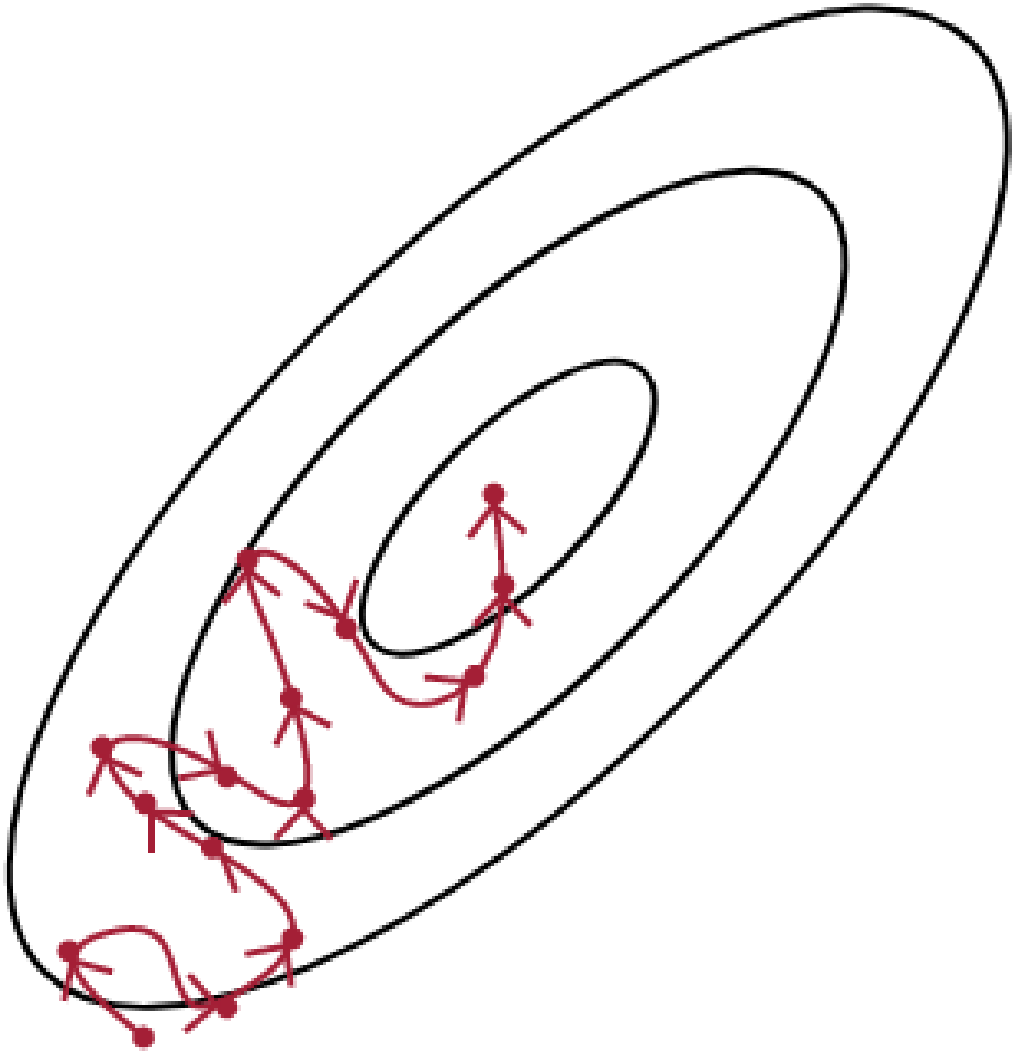
(i) Batch gradient descent goes smoothly toward the target because it uses the entire dataset at once, so lower variance is expected.

Mini-batch gradient descent



(1) In mini-batch gradient descent we use a uniformly sampled mini-batch. This results in noisier updates, but also faster processing of the data.

Stochastic gradient descent



(1) With stochastic gradient descent, in every iteration we step through only one sample. This makes it a noisy algorithm. It wouldn't be surprising to see several steps taking us further away from the target, and later back toward the target.