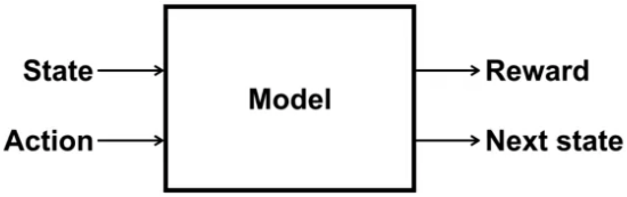
What is a model?

In this course, models store knowledge about the transition and reward dynamics



Models are used for planning; the process of using a model to improve a policy. One way to plan a model is to use the Stimulated Experience and perform Value function updates. By improving the value estimates we can make more informed decisions.

Types of models

1. Sample models:
    - Produces the actual outcome drawn from some underlying probabilities.
    - Can be computationally inexpensive due to random outcomes can be produced according to a set of rules.
2. Distribution models:
    - Specifies the likelihood of every outcome
    - Contains more information but can be difficult to specify and can become very large.

Random sample one-step tabular Q-learning

Assumed that:

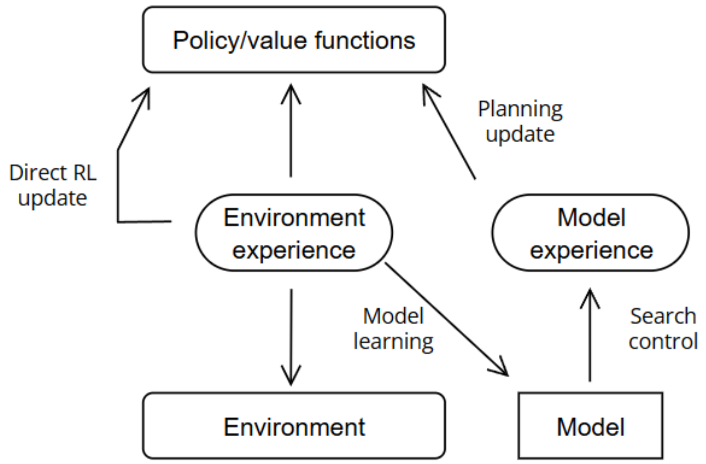
- We had a sample model of the transition dynamics
- We had the strategy for sampling relevant state action pairs

Procedure:

1. The algorithm chooses a state action pair at random from the set of all states and actions. It then queues the sample model with this state action pair to produce a sample of the next state and reward.
2. The algorithm performs a Q-learning update on this model transition

3. The algorithm improves the policy by greedifying with respect to the updated action values

The Dyna Architecture



Tabular Dyna-Q Algorithm

```
Initialize $Q(s, a)$ and $Model(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$

Loop forever:

    (a) $S \leftarrow$current (nonterminal) state
    (b) $A \leftarrow \epsilon - greedy(S, Q)$
    (c) Take action $A$; observe resultant reward $R$ and state $S'$
    (d) $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma max_a Q(S', a) - Q(S, A)]$
    (e) $Model(S, A) \leftarrow R, S'$ (assuming deterministic environment)
    (f) Loop repeat $n$ times:
            $S \leftarrow$ random previously observed state
            $A \leftarrow$ random action previously taken in $S$
            $R, S \leftarrow Model(S, A)$
            $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma max_a Q(S', a) - Q(S, A)]$
```

What if the model is inaccurate

The model can be inaccurate when the transitions they store are different from the transitions that happen in the environment.

- Incomplete model: the agent hasn't tried most of the actions in almost all of the states in the beginning of learning. The transitions associated with trying those actions in those states are simply missing from the model.
- Changing environment: taking actions in a state could result in a different next state and reward than what the agent observed before the change.

Planning with an inaccurate model improves the policy or value function with respect to the model, and not the environment

Dyna-Q can plan with an incomplete model by only sampling state-action pairs that have been previously visited

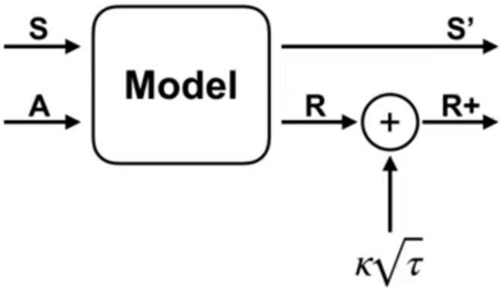
In-depth with changing environments

To encourage the agent to revisit its state periodically, we can add a bonus to the reward used in planning.

$$\text{New reward} = r + \kappa\sqrt{\tau}$$

r: actual reward
κ: a small constant to control the influence on the planning update
τ: time steps since transition was last tried

Adding this reward bonus to Dyna-Q's planning updates results in the Dyna-Q+ algorithm


S
A
Model
S'
R
+
R+
$\kappa\sqrt{\tau}$