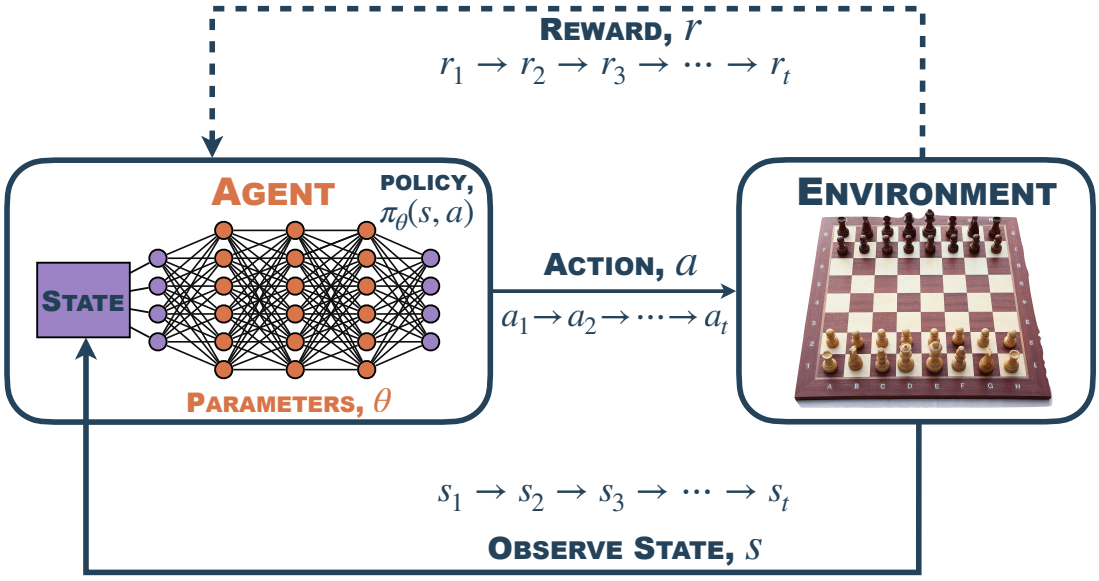

Figure 11.1: Schematic of reinforcement learning, where and agent senses its environmental state $\mathbf{s}$ and takes actions $\mathbf{a}$ according to a policy $\boldsymbol{\pi}$ that is optimized through learning to maximize future rewards $\mathbf{r}$. In this case, a deep neural network is used to represent the policy $\boldsymbol{\pi}$. This is known as a *deep policy network*.

11.1 Overview and Mathematical Formulation

Figure 11.1 provides a schematic overview of the reinforcement learning framework. An RL agent senses the state of its environment and learns to take appropriate actions to achieve optimal immediate or delayed rewards. Specifically, the RL agent arrives at a sequence of different states $\mathbf{s}_k \in \mathcal{S}$ by performing actions $\mathbf{a}_k \in \mathcal{A}$, with the selected actions leading to positive or negative rewards $\mathbf{r}_k$ used for learning. The sets $\mathcal{S}$ and $\mathcal{A}$ denote the sets of possible states and actions, respsectively. Importantly, the RL agent is capable of learning delayed rewards, which is critical for systems where the optimal solution involves a multi-step procedure. Rewards may be thought of as sporadic and time-delayed labels, leading to RL being considered a third major branch of machine learning, called *semi-supervised* learning, which complements the other two branches of supervised and unsupervised learning. One canonical example is learning a set of moves, or a long term strategy, to win a game of chess. As is the case with human learning, RL often begins with an unstructured *exploration*, where trial-and-error are used to learn the rules, followed by *exploitation*, where a strategy is chosen and optimized within the learned rules.

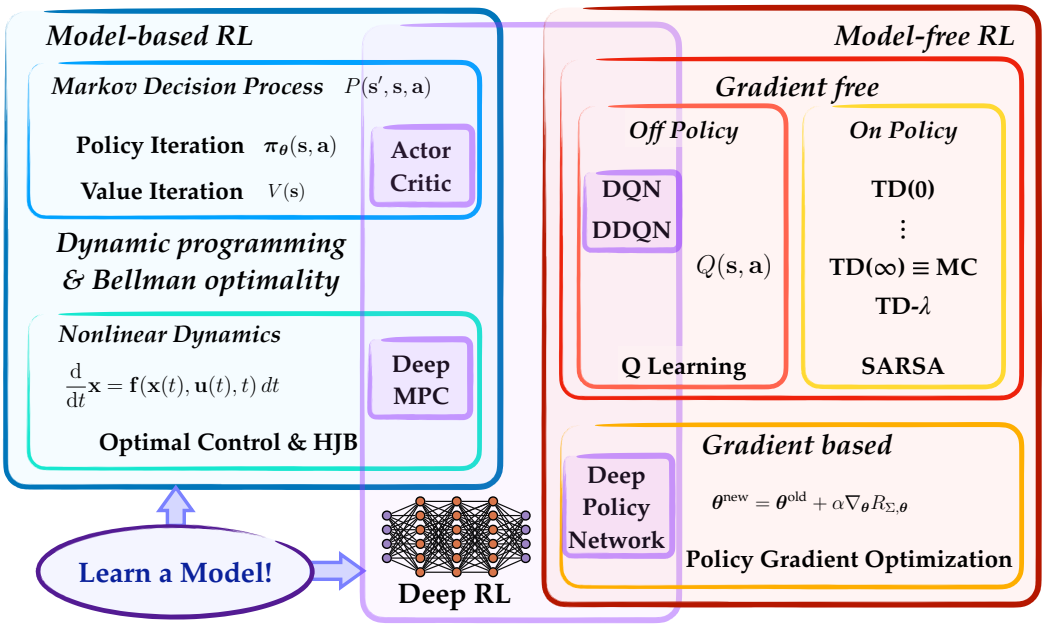

Figure 11.3: Rough categorization of reinforcement learning techniques. This organization is not comprehensive, and some of the lines are becoming blurred. The first major dichotomy is between model-based and model-free RL techniques. Next, within model-free RL, there is a dichotomy between gradient-based and gradient-free methods. Finally, within gradient-free methods, there is a dichotomy between on-policy and off-policy methods.

based RL versus *model-free RL*. When there is a known model for the environment, there are several strategies for learning either the optimal policy or value function through what is known as *policy iteration* or *value iteration*, which are forms of dynamic programming using the Bellman equation. When there is no model for the environment, alternative strategies, such as Q-learning, must be employed. The reinforcement learning optimization problem may be particularly challenging for high-dimensional systems with unknown, nonlinear, stochastic dynamics and sparse and delayed rewards. All of these techniques may be combined with function approximation techniques, such as neural networks, for approximating the policy π, the value function V, or the quality function Q (discussed in subsequent sections), making them more useful for high-dimensional systems. These model-based, model-free, and deep learning approaches will be discussed below. Note that this section only provides a glimpse of the many optimization approaches used to solve RL problems, as this is a vast and rapidly growing field.