1. Reinforcement learning
   1. Map situations to actions to maximize a numerical reward signal
   2. Characterized by
      1. Trial and error search – agent is not told which actions to take, but instead must balance between excitation and exploration
      2. Delayed reward signal – actions affect not only the immediate award, but all subsequent states, and in doing so affect all subsequent rewards
      3. Different thought paradigm – defining a learning problem, instead of characterizing learning methods
   3. NOT supervised learning – agent must explore on its own
   4. Four main elements
      1. Policy – mapping form perceived states of the environment to actions to be taken within those states. Analagous to psychology’s stimulus-response rules. May be stochastic
      2. Reward Function – defines the goal. Maps each perceived state (or state action pair) of the environment to a signal number—the reward. An agent’s sole objective is to maximize the total reward it receives in the long run. Is unalterable by the agent. Is short sighted
      3. Value Function – specifies what’s good in the long run. Value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state. Takes into account the states that are likely to follow and the rewards available from those states. Even though values are second to rewards, we try to optimize value, because doing so leads to optimized long term reward. Value estimation is difficult – rewards are given to us directly by environment, values must be estimated and re- estimated over time
      4. Model of Environment (Optional) – mimics behavior of the environment. Given a state and an action, model might predict the resultant state and the next reward. Models are the basis of planning. Model based reinforcement learning is a relatively new development, and very closely related to dynamic programming
      5. V(s) = V(s) + α[V(s`) –V(s)]
         1. Agent makes choice randomly instead of following policy – exploration. s denotes state before greedy move, s` denotes state after move. V(s) is updated value of state s after greedy move. α is learning rate
         2. If α is reduced appropriately over time, for a fixed environment, method will converge on true representation of value functions.
      6. Advantage over evolutionary methods
         1. Evolutionary methods – if one policy leads to a win, all moves in policy are given credit. What happens during the game is ignored
         2. Value function methods – allow each individual state to be evaluated