Introduction to Reinforcement Learning
Overview

- Supervised-, Unsupervised and RL
- Introduction to RL
- Elements of RL
Usual Process of Neural-Networks

- **Input**
- **Hidden Layer**
- **Compare Calculate Error**
- **Network Decisions**
- **Ground Truth**
Other possibility unsupervised learning
Introduction Reinforcement Learning (RL)

- No labels are given
- Reward function
Reinforcement Learning scenario
Policy of RL

- Determine behaviour of agent at given state
- Can be any function

$$\pi(a, s) = Pr(a_t = a | s_t = s) \quad a = \text{Action, } s = \text{State}$$
Reward signal of RL

- Reward function has to be modeled in some way
- At every state a reward is given
- A reward can be any Real Number \( r \in \mathbb{R} \)
- At which states do we get which rewards?
Value function of RL

Real function we want to optimize
Expected reward for each state

\[ T = \text{horizon (episode length)} \]
\[ \gamma \in [0, 1] = \text{decreasing factor} \]
\[ r_t = \text{reward at time } t \]
\[ \pi = \text{Policy} \]
\[ V_\pi(s) = \text{State-value for state } s \]

\[ V_\pi(s) = \mathbb{E}_\pi[R | s_0 = s] = \mathbb{E}_\pi \left[ \sum_{t=0}^{T} \gamma^t r_t | s_0 = s \right] \]
Model of RL

- Should mimic environment
- Prediction and planning
- Model-based and model-free methods
- Model-based is closer to human learning
Difficulties

- Exploration-Exploitation dilemma
- Credit Assignment Problem
- Reward Modeling
Multi-armed Bandits
Overview

- k-Armed Bandits problem
- Action-Values
- K-Armed Bandits Algorithms
- Nonstationary Problems
- Gradient Bandit Algorithms
Two-Armed Bandit

Reward Function = ?

<table>
<thead>
<tr>
<th>0</th>
<th>2</th>
<th>10</th>
<th>0</th>
<th>10</th>
<th>5</th>
</tr>
</thead>
</table>

| 10 | 0 | 5 | 0 | 3 |
K-armed Bandit Problem

- K-different possible actions
- Receive reward after each action
- Reward from \textit{stationary} probability distribution
- Goal: maximize the total reward after some time/actions
Action-value

\[ A_t = \text{Action selected at time } t \]
\[ R_t = \text{Reward received at time } t \]
\[ q_\star(a) = \mathbb{E}[R_t | A_t = a] \]
\[ Q_t(a) \approx q_\star(a) \]

Simple computation for \( Q_t \):

\[ Q_t(a) = \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\# \text{ of times } a \text{ taken prior to } t} \]

Simple selection rule for \( A_t \):

\[ A_t = \arg \max_a Q_t(a) \]
$\epsilon$-Greedy

With probability $1 - \epsilon$ exploit

With probability $\epsilon$ explore all actions
10-armed Testbed

Testbed evaluation of $\varepsilon$-Greedy

Nonstationary Problem

Reward function changes over time
More weights on recent rewards
Simple method, just take constant step-size

\[ Q_{n+1} = Q_n + \alpha [R_n - Q_n] \quad \alpha \in (0, 1] \]
Constant Step-Size

\[ Q_{n+1} = Q_n + \alpha [R_n - Q_n] \]
\[ = \alpha R_n + (1 - \alpha) Q_n \]
\[ = \alpha R_n + (1 - \alpha) [\alpha R_{n-1} + (1 - \alpha) Q_{n-1}] \]
\[ = \alpha R_n + (1 - \alpha) \alpha R_{n-1} + (1 - \alpha)^2 Q_{n-1} \]
\[ = \alpha R_n + (1 - \alpha) \alpha R_{n-1} + (1 - \alpha)^2 \alpha R_{n-2} + \ldots \]
\[ \ldots + (1 - \alpha)^{n-1} \alpha R_1 + (1 - \alpha)^n Q_1 \]
\[ = (1 - \alpha)^n Q_1 + \sum_{i=1}^{n} \alpha (1 - \alpha)^{n-i} R_i \]
Gradient Bandit Algorithms

Numerical Preference for action a denoted as $H_t(a)$

$$Pr(A_t = a) = \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}} = \pi_t(a)$$

$\overline{R_t} =$ average of all rewards up to time $t$

$$H_{t+1}(A_t) = H_t(A_t) + \alpha (R_t - \overline{R_t})(1 - \pi_t(A_t))$$

$$H_{t+1}(a) = H_t(a) - \alpha (R_t - \overline{R_t})\pi_t(a), \forall a \neq A_t$$