به نام خدا

دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

محمد کاظم هرندی

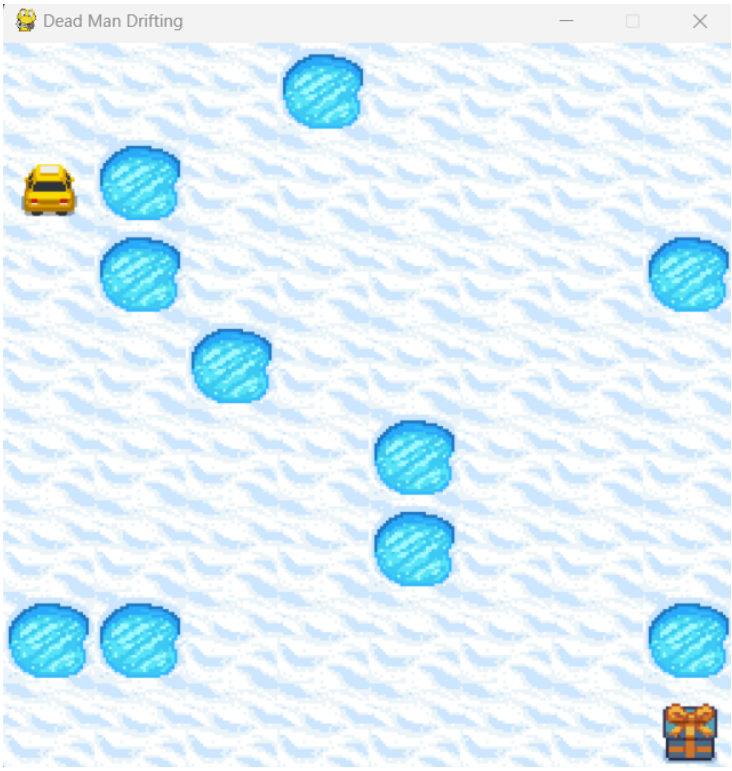
۴۰۰۳۶۲۳۰۳۹

مبانی و کاربردهای هوش مصنوعی

گزارشکار پروژه دوم

پاییز ۱۴۰۳

معرفی پروژه



در این پروژه، هدف ما پیاده‌سازی یک عامل هوشمند برای مسیردهی در محیط Frozen Lakeبا استفاده از الگوریتم تصمیم مارکوف بود. محیط Frozen Lake یک محیط شبکه‌ای ۸ خانه‌ای است که عامل (تاکسی یا الف) باید از خانه شروع (خانه ۰) به خانه هدف (خانه ۶۳) برسد. به دلیل لغزنده بودن زمین، احتمال دارد تاکسی به جهت‌های دیگر نیز برود. در این پروژه از کتابخانه‌های gymnasium و numpy استفاده شده است.

فرآیند تکرار سیاست

فرآیند تکرار سیاست یکی از الگوریتم‌های کلاسیک در حل مسائل تصمیم‌گیری مارکوف است. این فرآیند شامل دو مرحله اصلی است: ارزیابی سیاست و بهبود سیاست. در ادامه به توضیح کامل هر یک از این مراحل می‌پردازیم.

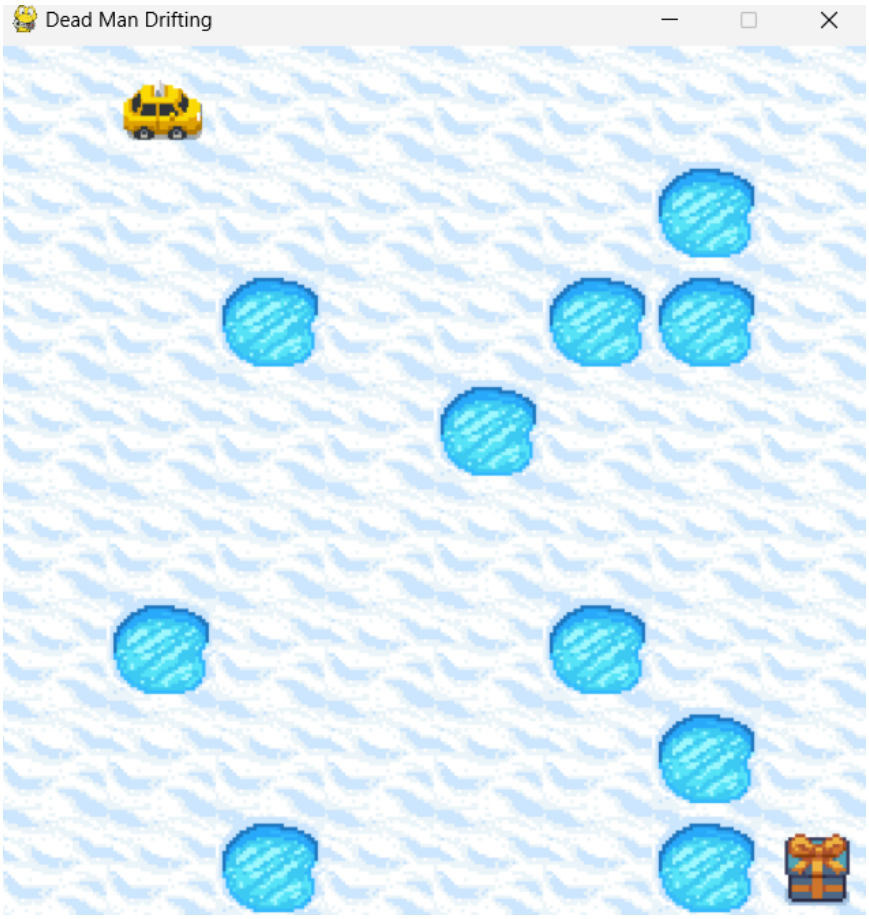
```python
v = np.zeros(num_state)  # Initialize the current value function to zeros
for i in range(1000):  # Limit the number of iterations for policy iteration

    # Policy Evaluation step
    v_old = -1 * np.ones(num_state)  # Reset old value function to ensure the loop condition is met

    while (np.abs(v - v_old)).max() > 1e-3:  # Continue until value function convergence
        v_old = v.copy()  # Update the old value function

        for s in range(num_state):  # For each state in the environment

            vs = 0  # Temporary variable for storing the value of state s

            # Calculate the expected value of the current policy
            for prob, st, r, done in env.P[s][policy[s]]:
                vs += prob * (r + gamma * v[st])  # Bellman equation for policy evaluation
            v[s] = vs  # Update the value of state s

        print(v)  # Print the current value function for debugging/observation

    # Policy Improvement step
    for s in range(num_state):  # For each state in the environment

        q = np.zeros(num_action)  # Initialize a temporary Q-value array for all actions

        # For each action, calculate its Q-value
        for a in range(num_action):

            for prob, st, r, done in env.P[s][a]:
                q[a] += prob * (r + gamma * v[st])  # Bellman equation for Q-value

        policy[s] = np.argmax(q)  # Update the policy to choose the action with the highest Q-value
```

ارزیابی سیاست

هدف از ارزیابی سیاست، محاسبه مقداردهی به حالات (v) برای یک سیاست مشخص است. این کار با استفاده از معادله بلمن انجام می‌شود. معادله بلمن رابطه‌ای بین مقداردهی به یک حالت و مقداردهی به حالات بعدی آن برقرار می‌کند. در این مرحله، ما به دنبال یافتن مقداردهی پایدار (یا نزدیک به پایدار) برای هر حالت هستیم.

بهبود سیاست

در مرحله بهبود سیاست، هدف ما بهبود سیاست موجود با استفاده از مقداردهی به حالات به‌دست‌آمده در مرحله ارزیابی سیاست است. در این مرحله، برای هر حالت بهترین عمل را که بیشترین مقدار-Q Valueرا فراهم می‌کند، انتخاب می‌کنیم .



اجرای سیاست بهینه

پس از یافتن سیاست بهینه، محیط با استفاده از این سیاست اجرا شده و تاکسی از حالت اولیه تا حالت نهایی حرکت می‌کند.

گسترش پروژه

تغییر گرافیک بازی

برای بهبود تجربه کاربری و جذاب‌تر کردن بازی، می‌توان از کتابخانه Pygame برای تغییر گرافیک بازی استفاده کرد. با تغییر تصاویر و افزودن انیمیشن‌های جذاب، بازی برای کاربران لذت‌بخش‌تر خواهد شد.

پیاده‌سازی در گیت‌هاب

با ایجاد مخزن در گیت‌هاب برای پروژه، می‌توان کدها و مستندات پروژه را به اشتراک گذاشت. این کار امکان همکاری با دیگران و دریافت بازخورد را فراهم می‌کند. همچنین به مستندسازی پروژه کمک می‌کند و باعث می‌شود که پروژه به صورت منظم‌تر و مستندتر باشد.

بهبود الگوریتم

علاوه بر الگوریتم تکرار سیاست، می‌توان الگوریتم‌های پیشرفته‌تری مانند یادگیری تقویتی (Reinforcement Learning) را برای بهبود عملکرد عامل هوشمند پیاده‌سازی کرد. این الگوریتم‌ها می‌توانند با استفاده از تجربه‌های گذشته، استراتژی‌های بهتری برای رسیدن به هدف پیدا کنند. از جمله الگوریتم‌های پیشرفته می‌توان به Q-Learning و SARSA اشاره کرد.

آزمایش و ارزیابی

با انجام آزمایش‌های مختلف و ارزیابی عملکرد عامل در شرایط مختلف، می‌توان نقاط قوت و ضعف الگوریتم را شناسایی کرد و بهبودهای لازم را انجام داد. این آزمایش‌ها می‌توانند شامل تغییرات در پاداش‌ها و احتمالات انتقال باشند. با استفاده از ابزارهای بصری‌سازی مانند نمودارها، می‌توان عملکرد عامل را در شرایط مختلف به صورت گرافیکی نمایش داد.