



دانشگاه اصفهان  
University of Isfahan

دانشکده مهندسی کامپیوتر

مبانی و کاربردهای هوش مصنوعی

پروژه دوم (Crawling robot)

استاد درس:

دکتر مرضیه حسینی

دستیاران درس:

کیارش گیلانیان

مهرزاد انصاری پور

زهرا مرتضوی

بهار 1404

## مقدمه

در دلِ دنیای وب، ربات کوچکی به نام «لینکی» زندگی می‌کرد. لینکی یک خزنده‌ی وب بود که هر روز در میان صفحات بی‌انتها پرسه می‌زد، لینک‌ها را دنبال می‌کرد و اطلاعات ارزشمند را جمع‌آوری می‌کرد. اما برخلاف دیگر ربات‌ها، لینکی یک هدف مشخص داشت: **یافتن بهترین کامپیوتر کوانتومی برای صاحبش.**

صاحب لینکی، کیارش بود که برای آموزش مدل جدید هوش مصنوعی‌اش دنبال کامپیوتر کوانتومی خاصی می‌گشت. او به لینکی مأموریت داده بود تا **در میان میلیون‌ها فروشگاه اینترنتی، بهترین انتخاب ممکن را پیدا کند؛** یعنی کامپیوتری با بالاترین امتیاز، قیمت مناسب، و ارسال سریع.

اکنون کیارش تصمیم گرفته که برنامه لینکی را از نو بازنویسی کند و از شما می‌خواهد که در تکمیل کردن آن کمک‌اش کنید.

## پیاده‌سازی

در این بخش قصد داریم با استفاده از مطالبی که درباره‌ی فرایند تصمیم مارکوف (MDP) یاد گرفته‌ایم، یک ربات خزنده بسازیم که در میان صفحات HTML جستجو کرده و بیشترین پاداش را جمع‌آوری کند.

فایل زیپ کد پایتونی که باید تکمیل کنید به همراه دو سری داده‌ی تست در اختیارتون قرار گرفته و شما باید تابع `value_iteration` را در کد پایتون تکمیل کنید. همچنین دقت کنید که برای تکمیل `value_iteration` نیازی به تغییر دیگر قسمت‌های کد و فایل‌های دیگر نیست.

پس از اینکه تابع `value_iteration` را پیاده‌سازی کردید، می‌تونید با اجرای کد روی مجموعه داده‌های `corpus0` و `corpus1` عملکرد آن را بسنجید و ببینید که درست کار می‌کنه یا نه.

## توضیح مفاهیم

در مجموعه داده `corpus0` ما 15 تا صفحه HTML داریم که هر یک دو حالت دارند.

1. صفحه‌ی پایانی هستند: یعنی پاداش مشخصی دارند (مثبت یا منفی) و ربات پس از رسیدن به آن صفحه، آن مقدار پاداش را دریافت کرده و جستجو خاتمه می‌یابد.
2. صفحات غیرپایانی هستند: یعنی می‌توان از آن به صفحات دیگری که به وسیله لینک متصل هستند، منتقل شد.

هدف ما یافتن سیاست بهینه است به طوری که بدانیم در هر یک از صفحات به وسیله‌ی کدام لینک مسیر را ادامه دهیم.

محیط ما غیرقطعی است یعنی به ازای هر لینکی که روی آن کلیک کنیم:

- 60 درصد به مقصد مورد نظر منتقل می‌شویم.
- 10 درصد در مکان فعلی می‌مانیم.
- 30 درصد به یکی از لینک‌های دیگر به طور یکنواخت منتقل می‌شویم.

\*\*\* اگر فقط یک لینک خروجی داشته باشیم به احتمال 90 درصد به مقصد هدف منتقل شده و به احتمال 10 درصد در صفحه فعلی می‌مانیم.

همچنین به ازای هر لینکی که روی آن کلیک کنیم و صفحه جدیدی باز شود، مقداری از حجم اینترنت ما کم می‌شه که پاداش آن را  $-0.05$  در نظر می‌گیریم.

ضریب تخفیف (discount factor) نیز 0.97 است یعنی به ازای پاداش‌هایی که دیرتر به آن‌ها برسیم از ارزش آن پاداش کم می‌شه.

اینکه چطوری فایل‌های HTML را هندل کنید و لینک‌های بین صفحات را پیدا کنید، از قبل پیاده‌سازی شده است و شما فقط باید تابع `value_iteration` را پیاده کنید.

\*\*\* برای اینکه بفهمید هر قسمت از کد دقیقاً چه کاری انجام می‌ده، اسلایدهای درس را مرور کنید.

## نیازمندی‌ها

برای تحویل پروژه حتماً حتماً داک بنویسید و خروجی‌های کدتون را به ازای مجموعه داده‌های تست نمایش دهید. همچنین توضیح بدید که هر بخش از کدتون چه منطقی را پیاده‌سازی می‌کنه.

## امتیازی

پیاده سازی روش `policy iteration` امتیازی محسوب می‌شه یعنی باید هر دو روش `value iteration` و `policy iteration` را پیاده‌سازی کنید تا نمره اضافی بگیرید.

هرگونه خلاقیت یا پیاده‌سازی الگوریتم‌های بیشتر نمره اضافی به همراه خواهد داشت.