

## گزارش پروژه ی سوم هوش مصنوعی

### کنترل هوشمند چراغ راهنمایی با استفاده از Deep Q-Learning

ارزو صفری

مهسا ارجمندفر

## مقدمه

هدف این پروژه، طراحی و ارزیابی یک سیستم کنترل هوشمند چراغ راهنمایی با استفاده از الگوریتم یادگیری تقویتی (Deep Q-Network (DQN برای بهینه‌سازی جریان ترافیک در یک تقاطع چهارراه است. این سیستم با هدف کاهش زمان انتظار وسایل نقلیه و تعداد توقف‌ها در یک محیط شبیه‌سازی شده با SUMO پیاده‌سازی شده است. در این گزارش، رویکرد مسئله، الگوریتم‌های استفاده‌شده، تنظیمات آزمایش، نتایج به‌دست‌آمده، و پیشنهادهایی برای بهبودهای آینده ارائه شده‌اند.

## رویکرد مسئله و الگوریتم‌های پیاده‌سازی شده

### رویکرد مسئله

هدف اصلی، بهینه‌سازی زمان‌بندی فازهای چراغ راهنمایی در یک تقاطع چهارراه است. این تقاطع شامل چهار جهت (شمال، جنوب، شرق، غرب) با سه خط در هر جهت است. الگوریتم DQN برای یادگیری سیاست بهینه تغییر فازها (شمال-جنوب سبز یا شرق-غرب سبز) استفاده شده است. فازهای زرد (۳ ثانیه) برای ایمنی ترافیک در نظر گرفته شده‌اند. این سیستم با دو روش پایه مقایسه شده است:

- زمان ثابت (Fixed-Time): تغییر فاز هر ۲۰۰ مرحله با فازهای سبز (۳۰ ثانیه) و زرد (۵ ثانیه).

- تصادفی (Random): انتخاب تصادفی فازها (۰ یا ۲) با حداقل مدت‌زمان فاز ۱۵۰ مرحله.

## الگوریتم‌های پیاده‌سازی شده

الگوریتم DQN با استفاده از مفاهیم زیر پیاده‌سازی شده است:

- Experience Replay: ذخیره تجربیات (حالت، عمل، پاداش، حالت بعدی، پایان) در حافظه و نمونه‌گیری تصادفی برای یادگیری پایدار.

- Target Network: استفاده از شبکه هدف برای به‌روزرسانی مقادیر Q و کاهش نوسانات.

- استراتژی  $\epsilon$ -greedy: تعادل بین اکتشاف (انتخاب تصادفی با احتمال  $\epsilon$ ) و بهره‌برداری (انتخاب عمل با بالاترین Q-value).

پیدامسازی در Python با استفاده از کتابخانه‌های TensorFlow، NumPy، و TraCI انجام شده است. فایل‌های اصلی پروژه عبارتند از:

- dqn\_agent.py: پیدامسازی عامل DQN با شبکه عصبی (دو لایه مخفی با ۱۲۸ نورون و فعال‌سازی ReLU).

- rl\_environment.py: محیط شبیه‌سازی SUMO با تعریف فضای حالت (طول صف‌های نرمال‌شده، فاز، زمان فاز)، فضای عمل (۰: شمال-جنوب سبز، ۲: شرق-غرب سبز)، و پاداش.

- main.py: آموزش مدل DQN با ۲۰۰ اپیزود و ذخیره مدل‌ها در checkpoints.

- test\_model.py: تست مدل آموزش‌دیده با  $\epsilon=0$ .

- evaluation(fixed).py: ارزیابی روش زمان ثابت.

- evaluation(random).py: ارزیابی روش تصادفی.

- evaluation(dqn).py: ارزیابی روش DQN.

- plot.py: تولید نمودارهای مقایسه‌ای و رفتار سیستم.

تابع پاداش بر اساس تغییرات زمان انتظار (delta\_wait) و تعداد وسایل نقلیه متوقف‌شده تعریف شده است، با جریمه برای تغییرات زودهنگام فاز (کمتر از ۱۰ ثانیه).

## تنظیمات آزمایش و محیط شبیه‌سازی

### محیط شبیه‌سازی

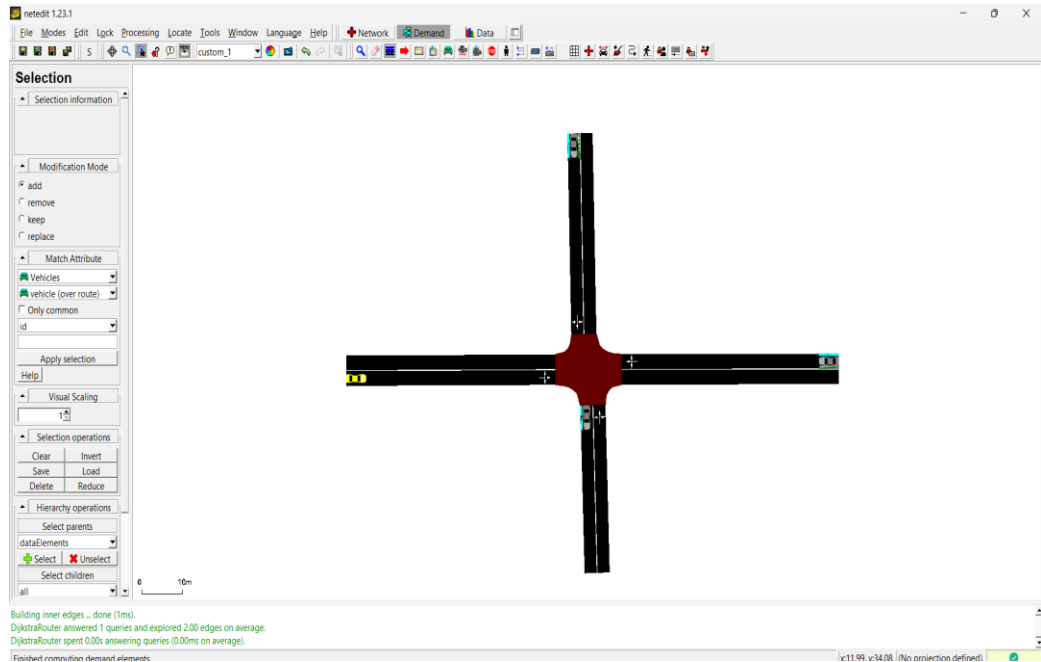
محیط شبیه‌سازی با استفاده از SUMO و فایل پیکربندی cross.sumocfg طراحی شده است.

مشخصات محیط:

- تقاطع: یک تقاطع چهارراهه با چهار جهت (شمال، جنوب، شرق، غرب) و سه خط در هر جهت.

- جریان ترافیک: تعریف‌شده در route.rou.xml با یک وسیله نقلیه در هر جهت در هر ۵ ثانیه.

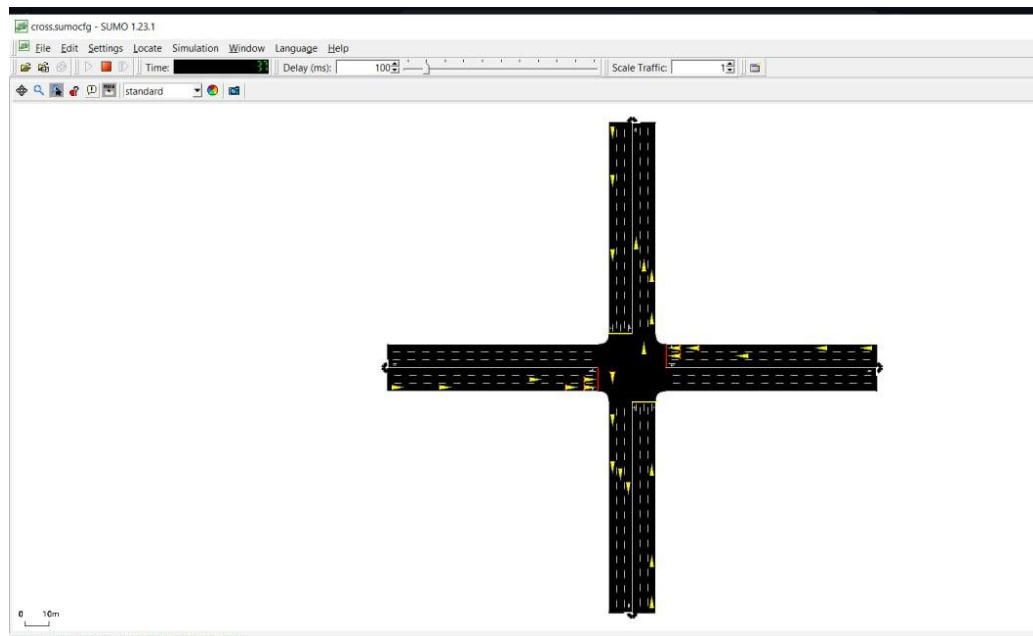
- فازها: دو فاز اصلی (۰: شمال-جنوب سبز، ۲: شرق-غرب سبز) با فازهای زرد (۱ و ۳) دیرای ایمنی.

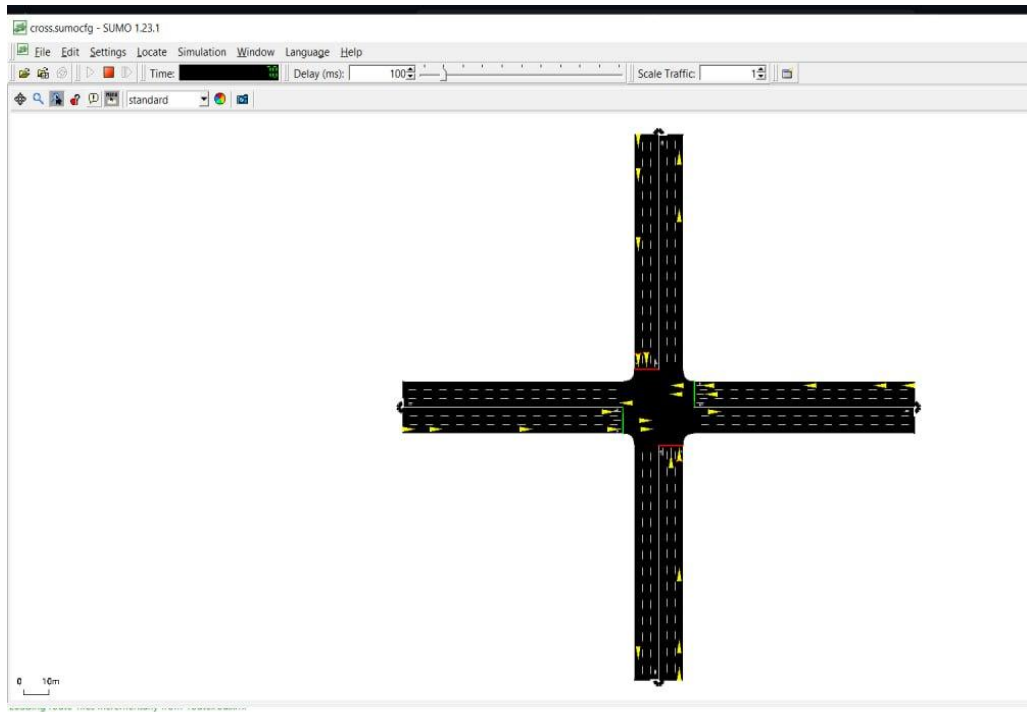
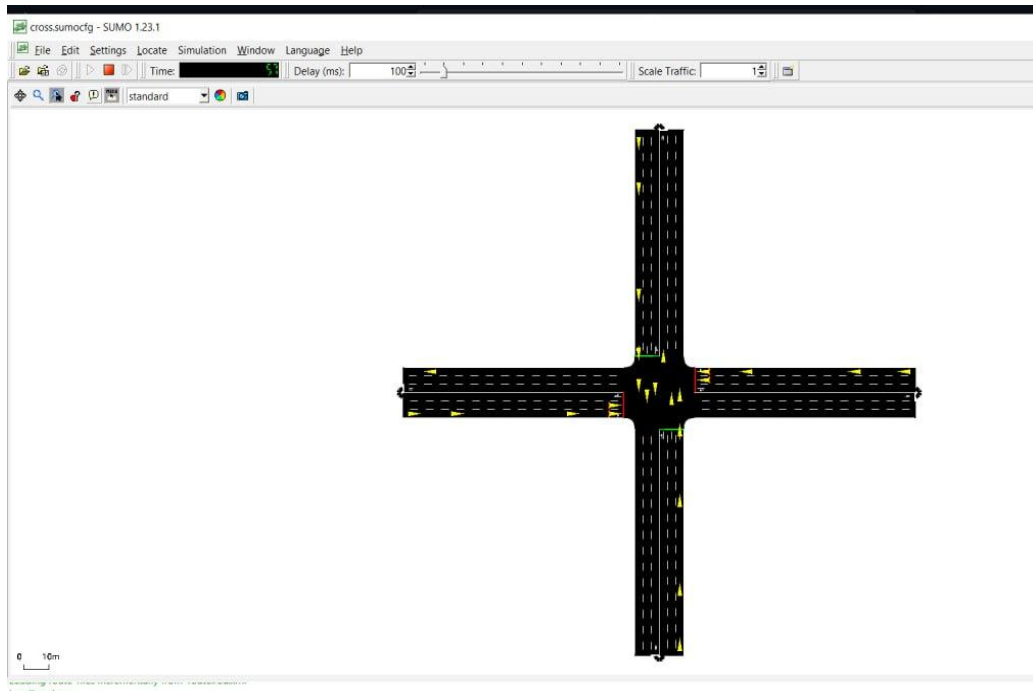


`netconvert --node-files=node.xml --edge-files=edge.xml -o crossroad.net.xml`

این دستور با استفاده از فایل‌های تعریف گره‌ها (node.xml) و لبه‌ها (edge.xml)، فایل شبکه را به‌طور خودکار ایجاد می‌کند. با این حال، برای کنترل دقیق‌تر ترافیک و اعمال تغییرات پویا (مانند تنظیم فازهای چراغ راهنمایی یا جریان وسایل نقلیه، از فایل‌های XML مانند rou.xml استفاده شد که انعطاف‌پذیری بیشتری در مدیریت سناریوهای ترافیکی فراهم می‌کند.

به عنوان مثال، فایل rou.xml امکان تعریف تعداد، نوع و مسیر حرکت وسایل نقلیه را با جزئیات بیشتری نسبت به تنظیمات پیش‌فرض NETEDIT می‌دهد، که برای بهینه‌سازی سیستم با الگوریتم DQN ضروری بود.





شبیه‌سازی تست سیستم کنترل هوشمند چراغ راهنمایی است که با استفاده از فایل `test_model.py` آخرین چک‌پوینت مدل آموزش‌دیده (مانند `checkpoints/dqn_ep12.keras`) اجرا شده است. مدل‌های اولیه که با چک‌پوینت‌های ابتدایی تست شدند، توانایی کنترل ترافیک را نداشتند و صف‌ها انباشت می‌شد. اما با افزایش اپیزودها، مدل به سیاست بهینه رسید و آموخت که به‌صورت دینامیک، با توجه به طول

صف‌ها و زمان انتظار، فازها را تغییر دهد. این پیشرفت در تصویر با جریان روان ترافیک قابل مشاهده است.

## تنظیمات DQN

پارامترهای DQN از فایل config.yaml استخراج شده‌اند:

learning\_rate: 0.0001 -

gamma: 0.95 -

epsilon\_start: 1.0 -

epsilon\_min: 0.01 -

epsilon\_decay: 0.995 -

batch\_size: 128 -

memory\_capacity: 50000 -

target\_update\_freq: 100 -

## تنظیمات آزمایش

- تعداد اپیزودها: ۲۰۰ اپیزود، هر کدام با حداکثر ۵۰۰ مرحله.

- روش‌های مقایسه:

- زمان ثابت: تغییر فاز هر ۲۰۰ مرحله با فازهای سبز (۳۰ ثانیه) و زرد (۵ ثانیه).

- تصادفی: انتخاب تصادفی فازها (۰ یا ۲) با حداقل مدت زمان فاز ۱۵۰ مرحله.

- DQN: مدل آموزش‌دیده با  $\epsilon=0$  در checkpoints/dqn\_ep12.keras.

- تعداد ران‌ها: ۲ ران برای هر روش جهت محاسبه معیارها.

## نتایج به دست آمده

### معیارهای ارزیابی

چهار معیار برای مقایسه روش‌ها محاسبه شده‌اند:

- میانگین زمان انتظار: میانگین زمان انتظار وسایل نقلیه در تقاطع (ثانیه).
- تأخیر کل: مجموع زمان انتظار تمام وسایل نقلیه در تمام مراحل (ثانیه).
- طول صف‌ها: میانگین تعداد وسایل نقلیه در صف در چهار جهت.
- تعداد توقف‌ها: تعداد کل وسایل نقلیه متوقف شده در تمام مراحل.

### جدول مقایسه

جدول زیر میانگین و انحراف استاندارد معیارها را برای دو ران نشان می‌دهد:

روش	میانگین زمان انتظار (ثانیه)	تأخیر کل (ثانیه)	طول صف‌ها (خودرو)	تعداد توقف‌ها
زمان ثابت	$0,00 \pm 35,28$	$0,00 \pm 17642,36$	$0,00 \pm 8,82$	$0 \pm 17643$
تصادفی	$23,60 \pm 49,28$	$11797,94 \pm 24640,58$	$1,76 \pm 10,03$	$3518 \pm 20055$
DQN	$0.51 \pm 0.00$	$253.40 \pm 0.00$	$0.91 \pm 0.00$	$1821 \pm 0$

### تحلیل نتایج

- DQN: عملکرد بسیار بهتری نسبت به روش‌های پایه نشان داد:

- کاهش ۹۸,۶٪ در میانگین زمان انتظار (از ۳۵,۲۸ به ۰,۵۱ ثانیه).

- کاهش ۹۸,۶٪ در تأخیر کل (از ۱۷۶۴۲,۳۶ به ۲۵۳,۴۰ ثانیه).

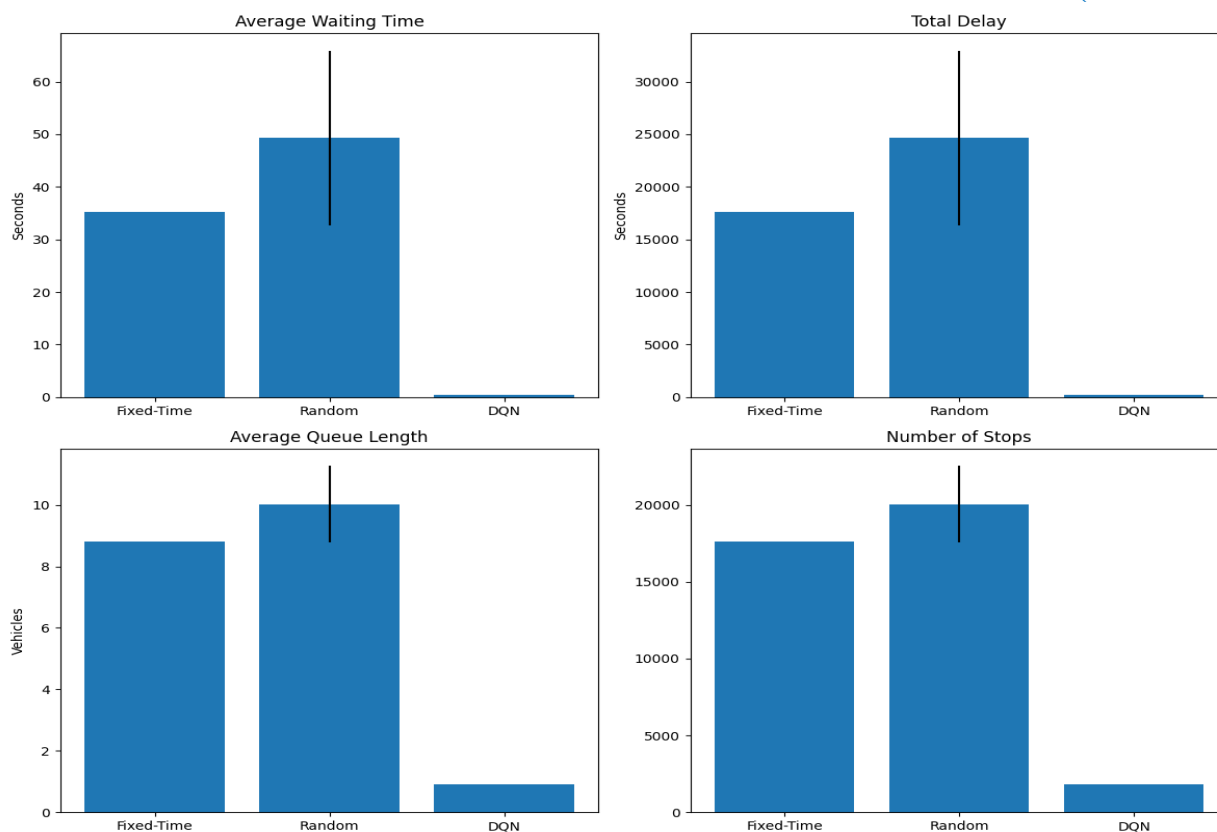
- کاهش ۸۹,۷٪ در طول صف‌ها (از ۸,۸۲ به ۰,۹۱ خودرو).

- کاهش ۸۹,۷٪ در تعداد توقف‌ها (از ۱۷۶۴۳ به ۱۸۲۱).

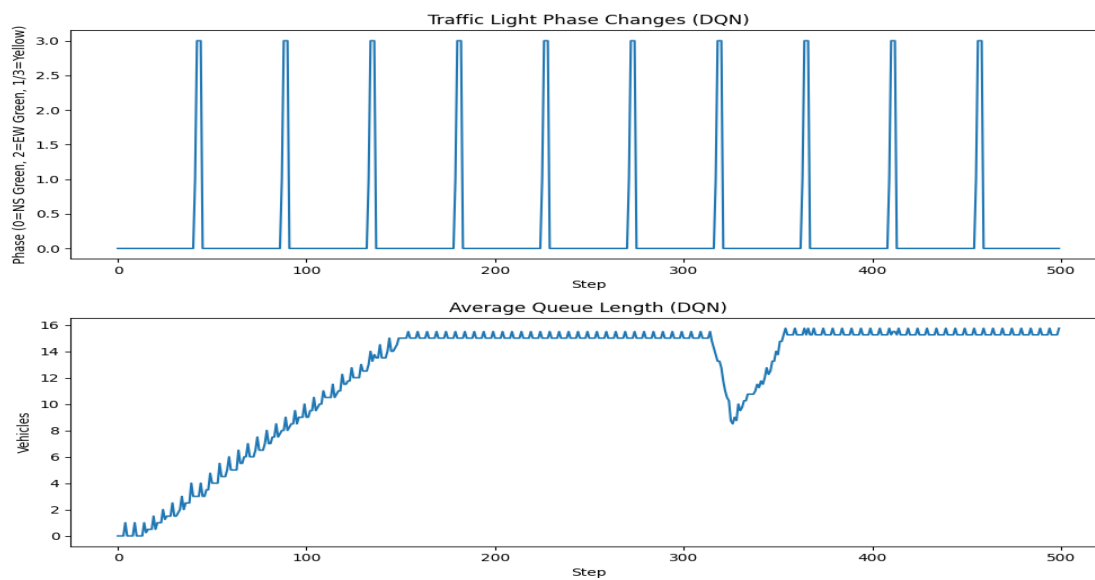
- زمان ثابت: عملکردی پایدار اما غیربهبوده داشت، زیرا به تغییرات پویای ترافیک پاسخ نمی‌دهد.
- تصادفی: بدترین عملکرد را داشت، با انحراف استاندارد بالا به دلیل انتخاب‌های غیرهوشمند فازها.

## نمودارها

مقایسه معیارها: نمودارهای میله‌ای مقایسه معیارها (میانگین زمان انتظار، تأخیر کل، طول صف‌ها، تعداد توقف‌ها) را برای سه روش نشان می‌دهد.

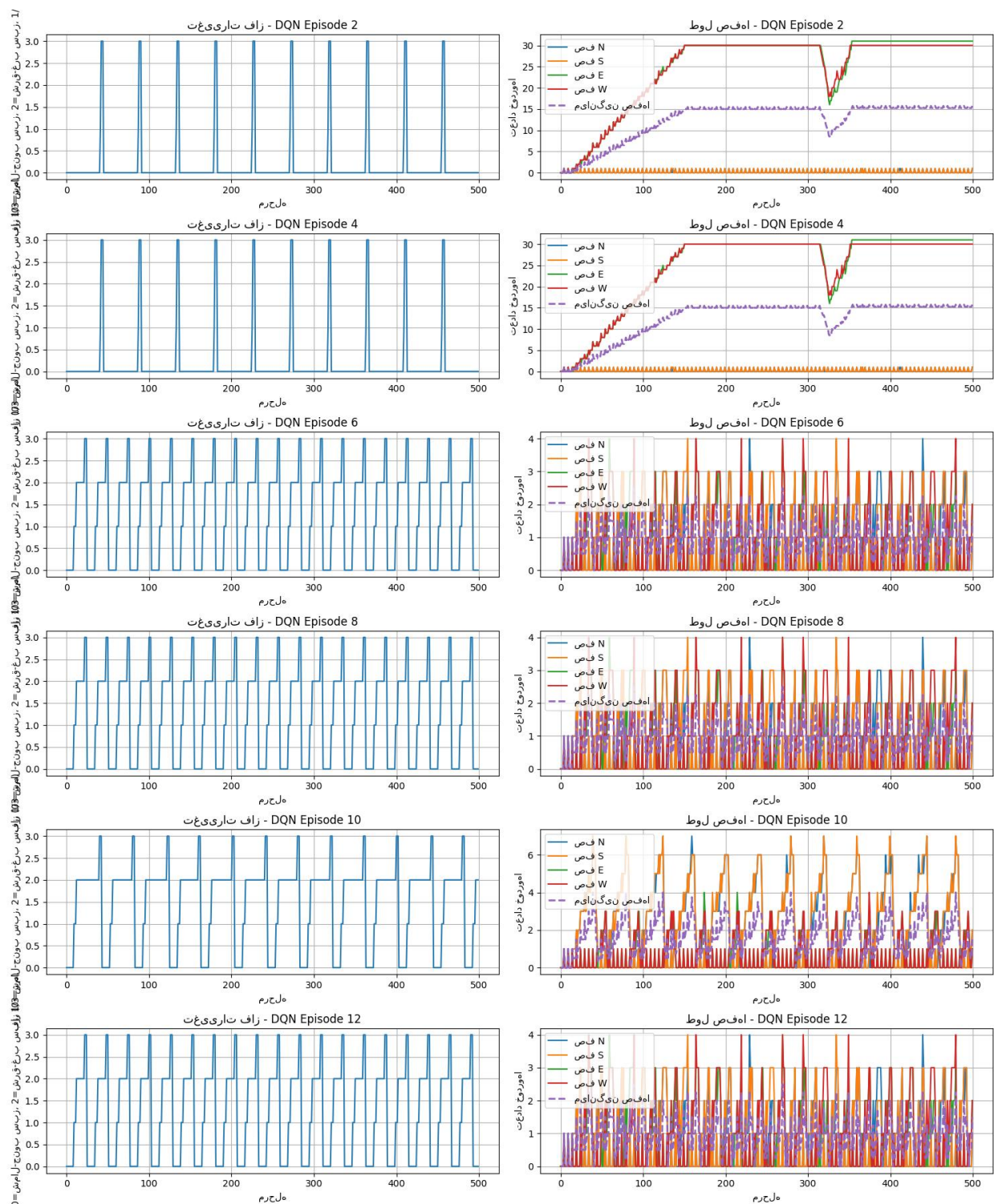


رفتار DQN: تغییرات فازهای چراغ راهنمایی و طول صف‌ها را در یک ران DQN نشان می‌دهد.





مقایسه رفتار سیستم DQN با روش‌های پایه (زمان ثابت و تصادفی) را در اپیزودهای مختلف نشان می‌دهد.



## آموخته‌ها، ایده‌ها و پیشنهادها

### آموخته‌ها

- الگوریتم DQN به‌طور مؤثری زمان انتظار و تعداد توقف‌ها را کاهش داد و عملکردی به‌مراتب بهتر از روش‌های پایه ارائه کرد.
- استفاده از Experience Replay و Target Network پایداری یادگیری را بهبود بخشید.
- تنظیم دقیق پارامترهای  $\gamma$ ، learning\_rate و  $\epsilon$ \_decay برای عملکرد بهینه حیاتی بود.
- مشکلات اولیه در روش تصادفی (عدم تصادفی بودن به دلیل محدودیت‌های apply\_action) و روش زمان ثابت (استفاده از فازهای پیش‌فرض SUMO به جای ۲۰۰ مرحله) با اصلاح کدها برطرف شدند.

### ایده‌ها

- استفاده از الگوریتم‌های پیشرفته‌تر مانند Double DQN یا Dueling DQN برای بهبود کارایی.
- افزودن معیارهای محیطی (مانند کاهش آلودگی یا مصرف سوخت) به تابع پاداش.
- استفاده از شبکه‌های عصبی عمیق‌تر یا معماری‌های پیچیده‌تر برای سناریوهای ترافیکی متنوع.

## پیشنهادهای مسیرهای آینده

- آزمایش با سناریوهای ترافیکی واقعی‌تر (مانند ساعات اوج ترافیک یا جریان‌های متغیر).
- گسترش سیستم به شبکه‌های ترافیکی پیچیده‌تر با چندین تقاطع.
- ادغام با داده‌های واقعی (مانند داده‌های دوربین‌های ترافیکی) برای کاربرد عملی.
- افزودن فازهای پیچیده‌تر مانند چپ‌گرد یا اولویت عابر پیاده.

## کارهای انجام شده

- پیاده‌سازی محیط: تعریف محیط شبیه‌سازی در `rl_environment.py` با فضای حالت (طول صف‌های نرمال شده، فاز، زمان فاز)، فضای عمل، و تابع پاداش.

- پیاده‌سازی DQN: پیاده‌سازی عامل DQN در `dqn_agent.py` با شبکه عصبی، Experience Replay و Target Network.

- آموزش مدل: آموزش مدل با ۲۰۰ اپیزود در `main.py` و ذخیره مدل‌ها در `.checkpoints/dqn_ep*.keras`.

- تست مدل: ارزیابی مدل آموزش دیده در `.test_model.py`.

- ارزیابی روش‌های پایه:

- `evaluation(fixed).py`: روش زمان ثابت با تغییر فاز هر ۲۰۰ مرحله.

- `evaluation(random).py`: روش تصادفی با انتخاب تصادفی فازها و تأیید توزیع فازها.

- `evaluation(dqn).py`: ارزیابی مدل DQN با  $\epsilon=0$ .

- تولید نمودارها: تولید نمودارهای مقایسه‌ای (`comparison_graphs.png`) و رفتار DQN (`dqn_behavior.png`) در `.plot.py`.

- رفع مشکلات:

- مشکل عدم ازسرگیری آموزش با استفاده از مدل معتبر `dqn_ep12.keras`.

- اصلاح روش زمان ثابت برای استفاده از تغییر فاز ۲۰۰ مرحله‌ای.

- اصلاح روش تصادفی با اطمینان از انتخاب تصادفی و چاپ توزیع فازها.

## نتیجه‌گیری

سیستم کنترل هوشمند چراغ راهنمایی مبتنی بر DQN با موفقیت پیاده‌سازی و ارزیابی شد. این سیستم عملکردی به مراتب بهتر از روش‌های زمان ثابت و تصادفی نشان داد، با کاهش ۹۸,۶٪ در میانگین زمان انتظار و تأخیر کل، و ۸۹,۷٪ در طول صف‌ها و تعداد توقف‌ها. نمودارهای مقایسه‌ای و رفتار سیستم این برتری را تأیید می‌کنند. با وجود چالش‌های اولیه مانند مشکلات تصادفی بودن و تنظیم فازها، اصلاحات انجام شده منجر به نتایج قابل اعتماد شد. پیشنهاد‌های ارائه شده برای بهبودهای آینده، مانند استفاده از الگوریتم‌های پیشرفته‌تر و سناریوهای پیچیده‌تر، می‌توانند این سیستم را برای کاربردهای عملی‌تر آماده کنند.