הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל ע"ש אנדרו וארנה ויטרבי

המעבדה לבקרה, לרובוטיקה וללמידה חישובית

**ספר פרויקט**

**אלגוריתם DQN**

**לבקרת רמזורים**

מבצעים:

נוואף סלמאן Nawaf Salman

נאהל עוידאת Nahel Awidat

מנחה:

דר.איל טייטלר Dr. Ayal Taitler

סמסטר רישום: חורף תשפ"ג

תאריך הגשה:

**תקציר**

ניהול יעיל של תנועה בצמתים עירוניים הוא מרכיב קריטי ביוזמות עיר חכמה מודרניות. פרויקט זה מתמקד באופטימיזציה של מדיניות הרמזורים בצומת יחידה לשיפור זרימת התנועה והפחתת עומסים.  
באמצעות סביבת הסימולציה SUMO (Simulation of Urban MObility) ושפת התכנות Python, יישמנו אלגוריתם DQN כדי ללמוד ולפתח מדיניות בקרה אופטימלית לרמזורים. באמצעות מודול TRACI (Traffic Control Interface), סוכן ה-DQN מתקשר עם סימולציית ה-SUMO, אוסף נתוני תנועה בזמן אמת ומתאים את תזמוני הרמזורים באופן דינמי.  
 המטרה המרכזית היא למזער את זמני ההמתנה הממוצעים ואת אורכי התורים של כלי הרכב בצומת.  
תוצאותינו מציגות שיפורים משמעותיים ביעילות התנועה, ומדגימות את הפוטנציאל של טכניקות למידת חיזוק בניהול מערכות תנועה. פרויקט זה סולל את הדרך לפתרונות נרחבים בבקרת תנועה חכמה, ותורם לרשתות תחבורה עירוניות יציבות יותר.

**Abstract**

Efficient traffic management at urban intersections is a critical component of modern smart city initiatives. This project focuses on optimizing traffic light policies at a single junction to enhance traffic flow and reduce congestion.   
Utilizing the SUMO (Simulation of Urban MObility) simulation environment and Python, we implemented a Deep Q-Network (DQN) algorithm to learn and develop an optimal traffic signal control policy.   
Through the TRACI (Traffic Control Interface) module, the DQN agent interacts with the SUMO simulation, gathering real-time traffic data and adjusting signal timings dynamically. The core objective is to minimize average waiting times and vehicle queue lengths at the intersection.   
Our results demonstrate significant improvements in traffic efficiency, showcasing the potential of reinforcement learning techniques in traffic management systems. This project paves the way for scalable solutions in smart traffic control, contributing to smoother and more sustainable urban transportation networks.

**מבוא**

ניהול תנועה יעיל בצמתים עירוניים מהווה אתגר מרכזי בערים המודרניות. עם העלייה המתמדת במספר כלי הרכב בכבישים, בעיות כמו עומסי תנועה, זמני המתנה ארוכים ופליטת מזהמים הולכות ומחריפות. בעיות אלו פוגעות באיכות החיים של התושבים, מגבירות את זיהום האוויר ומובילות לבזבוז זמן ואנרגיה. התמודדות עם האתגרים הללו והפחתת עומסי התנועה הפכה למשימה דחופה במסגרת יוזמות עיר חכמה ושיפור תחבורה בת קיימא.

הבעיה המרכזית בפרויקט זה היא ניהול ואופטימיזציה של רמזורים בצומת יחיד כדי לשפר את זרימת התנועה ולהפחית את זמני ההמתנה של כלי הרכב. צמתים אלו מהווים נקודות חנק בתשתיות התחבורה, ובקרה יעילה של רמזורים יכולה להביא לשיפור משמעותי בזרימת התנועה העירונית.

פתרונות קיימים לניהול רמזורים כוללים מערכות בקרת תנועה קבועות, אשר מבוססות על מחזורים ותזמונים מוגדרים מראש. מערכות אלו אינן גמישות ואינן מסוגלות להתאים את עצמן בזמן אמת לתנאי התנועה המשתנים, דבר שמוביל ליעילות נמוכה ולבעיות עומס. בנוסף, קיימות מערכות חכמות יותר המשתמשות בחיישנים ובאלגוריתמים פשוטים כדי להתאים את תזמוני הרמזורים, אך גם אלו לעיתים קרובות אינן מצליחות להתמודד בצורה מיטבית עם השינויים הדינמיים בתנועה.

**הגדרת בעיה**

לפתח מדיניות בקרה אופטימלית לרמזורים בצומת יחיד, שתוכל להתאים את תזמוני הרמזורים בזמן אמת ולשפר את זרימת התנועה. לשם כך, השתמשנו באלגוריתם DQN ובסימולציית SUMO כדי לבנות מדיניות חכמה לבקרת רמזורים.

מטרת הרשת היא לקבל מידע עדכני על מצב התנועה באופן דינמי, ולחזות את הפאזה האופטימלית של הצומת מסט פאזות מוגדר מראש.

**פתרונות אפשריים, והפתרון הנבחר**

בפרויקט בחנו כמה פתרונות לבקרת רמזורים: A2C ,אלגורתם Max Pressure, ופתרון מבוסס למידה מחיזוקים DQN.

Max Pressure: אלגוריתם Max Pressure הוא אלגוריתם קלאסי יותר לבקרת רמזורים והוא מתבסס על הפחתת הלחץ בצמתים. הלחץ על כל קשת בכביש מחושב על פי ההפרש בין מספר המכוניות הממתינות בכיוון הנכנס לבין מספר המכוניות הממתינות בכיוון היוצא.  
הסוכן בוחר את הפאזה שתפחית את הלחץ בצומת באופן המקסימלי.  
זה אלגוריתם פשוט ליישום, דורש עיבוד קטן יחסית. אבל החסרון שלו שהוא לא גמיש לתנאי תנועה משתנים, ופחות יעיל כאשר יש שינויים מהירים בתנועה, וכדי להיות יעיל דורש תנאי תנועה יציבים ורמת עומס נמוכה.

A2C: אלגוריתם A2C (Advantage Actor-Critic) הוא שיטה בלמידת חיזוק המורכבת משני מרכיבים עיקריים: "Actor" (שחקן) ו-"Critic" (מבקר). ה-Actor בוחר פעולות על פי מדיניות נוכחית, וה-Critic מעריך את ערך הפעולות שנבחרו. בתהליך האימון, ה-Actor בוחר פעולה, מבצע אותה, מקבל תגמול, וה-Critic מעריך את היתרון של הפעולה כדי לעדכן את המדיניות והערכת הפעולות.  
בבעיה שלנו מערכת רמזורים היא מערכת מורכבת עם הרבה מצבים ופעולות אפשריות, בגלל זה האלגוריתם A2C עלול להתקשות להתמודד עם כל המצבים והתרחישים המגוונים באופן יעיל. כמו כן, A2C דורש כוח חישובי רב וזיכרון כדי לעבד את כל המידע ולהתאים את המדיניות, וזה עשוי להיות בעייתי במיוחד בסביבות מורכבות כמו מערכות רמזורים בערים גדולות.

DQN: אלגוריתם DQN (Deep Q-Network) הוא שיטה ללמידת חיזוק שמשתמשת ברשתות נוירונים עמוקות כדי ללמוד ולשפר מדיניות לפתרון בעיות קבלת החלטות. DQN מציע יתרונות משמעותיים במונחים של פשטות, יציבות הלמידה, וניהול Replay Memory. אלו גורמים שהופכים אותו לבחירה מועדפת בבעיות רבות, במיוחד כאשר יש מרחב מצב גדול ונדרשת הערכה מדויקת של ערכי Q. במערכת רמזורים, היתרונות הללו יכולים להתבטא ביכולת טובה יותר להתמודד עם מצבים מורכבים ולהביא ללמידה יעילה יותר.

תהליך החלטה מרקובי:

תהליך החלטה מרקוב (MDP) הוא הגישה המתמקדת בפרקטיקה מתמטית שמשמשת למידול בעיות החלטה במקרים בהם התוצאה אינה ברורה. התהליך קרוי על שם המתמטיקאי הרוסי אנדרי מרקוב ונמצא בשימוש נרחב בתחומים רבים, כולל בבינה מלאכותית, מחקרי פעולות, כלכלה ותאוריה שליטה.

רכיביו המרכזיים של תהליך החלטה מרקובי הם:

-מצבים (S): המערכת או הסביבה יכולה להיות במגוון של מצבים או תנאים שונים. המצבים הללו מאפיינים את הקונפיגורציה הנוכחית של המערכת.

-פעולות (A): מקבלי ההחלטות הנקראים לעיתים קרובות "סוכנים ", יכולים לבצע פעולות המשפיעות על מעבר של הסביבה ממצב אחד למצב אחר. הפעולות הן ההחלטות או התנועות שסוכן יכול לבצע .

-מעברים (P): פונקציית הסתברות המעבר מתארת את סיכוי המעבר ממצב נתון למצב אחר בהינתן פעולה מסוימת. היא מייצגת את דינמיקת המערכת.

-תגמול (R): הסוכן מקבל רווח מספרי בהתאם לפעולה שבוצעה ולמצב שהתקבל בעקבותיה. המטרה היא בדרך כלל למקסם את הרווח הנצבר לאורך הזמן.

-מדיניות (π): מדיניות היא אסטרטגיה או סט של כללים המנחים את הסוכן בבחירת פעולות בכל מצב. היא מגדירה את המיפוי ממצבים לפעולות.

-discount factor (GAMMA): + התיחסות לעתיד

דינמיקת MDP מאופיינת במאפיין המרקובי, לפיו המצב הבא תלוי רק במצב הנוכחי ובפעולה שבוצעה, ולא ברצף המצבים והפעולות שקדמו לו. המטרה ב-MDP אם כך היא למצוא מדיניות אופטימלית שממקסמת את הרווחים הצפויים לאורך הזמן. היבט זה מתבטא בפונקציית הערך, המייצגת את סכומי הרווח הצפויים לאורך הזמן בהתאם למצב נתון ולמדיניות ספציפית.

פתרון של MDP כולל מציאת מדיניות אופטימלית או פונקציית ערך אופטימלית. אלגוריתמים פופולריים לפתרון של MDP כוללים תכנות דינמי, שיטות מונטה קרלו, וטכניקות למידה מחיזוקים.

למידה מחיזוקים:

למידה מחיזוקים Reinforcement Learning היא פרדיגמת למידת מכונה השואב ת השראה מפסיכולוגיה התנהגותית, ובה סוכן לומד לק בל החלטות על ידי אינטראקציה עם סביבה וקבלת משוב בצורת פרס או עונש. הרעיון המרכזי הוא לאפשר לסוכן ללמוד התנהגויות אופטימליות דרך ניסיון וטעיה, ולא על ידי תכנות מפורש.

ב-RL, תהליך הלמידה מתקיים כאשר סוכן פועל בסביבה, נוקט בפעולות על פי המצב הנוכחי שלה ומקבל משוב בצורת אות פרס. מטרתו של הסוכן היא ללמוד מדיניות – מיפוי מצבים לפעולות – הממקסמת את סכום הפרסים הצפויים לאורך זמן. הסביבה, במקביל, מגיבה לפעולות של הסוכן ועוברת למצבים חדשים. זהו בעצם יישום של תהליך החלטה מרקובי.

הסוכן חוקר פעולות שונות כדי להבין את ההשלכות שלהן, וכך משפר את אסטרטגיית קבלת ההחלטות שלו. האתגר המרכזי בלמי דה מחיזוקים הוא בחלוקה נכונה בין exploration ל- exploitation. חקירת הסביבה נחוצה כדי שהסוכן יגלה פעולות אופטימליות שאולי טרם גילה, וה-exploitation לעומתה נחוץ לשם בחירת פעולות ידועות על מנת למקסם רווח מיידי.

יכולת הסוכן ללמוד ולהתאים את עצמו למגוון רחב של מצבים ולסביבות דינמיות, הופכת את הלמידה מחיזוקים לכלי עוצמתי בסביבות מורכבות ולא ודאיות, ומדגישה את פוטנציאל השיטה ליישומים בעולם האמיתי.

DQN:

אלגוריתם DQN (Deep Q-Network) הוא שיטה ללמידת חיזוק שמשתמשת ברשתות נוירונים עמוקות כדי ללמוד ולשפר מדיניות לפתרון בעיות קבלת החלטות.

**איך DQN עובד?**

**פונקציית Q:** פונקציית Q מעריכה את הערך של ביצוע פעולה מסוימת במצב מסוים. המטרה היא למצוא את הפונקציה שתיתן את ההערכה הטובה ביותר לפעולה בכל מצב, כך שהמדיניות תהיה אופטימלית.

**רשת נוירונים:** ב-DQN משתמשים ברשת נוירונים עמוקה להערכת פונקציית Q. הרשת מקבלת כקלט את המצב הנוכחי ומחזירה את ערכי Q לכל הפעולות האפשריות במצב הזה.

**זיכרון חוויות**: במהלך האימון, שומרים בזיכרון את החוויות מהאינטראקציה עם הסביבה (מצב, פעולה, תגמול, מצב הבא). זיכרון זה מאפשר לאלגוריתם ללמוד מהניסיון בצורה יעילה.

**עדכון משוקלל (Target Network):** בנוסף לרשת הראשית, יש רשת מטרה שנמצאת בעדכונים פחות תכופים ומשמשת להבטחת יציבות הלמידה.

**חקר וניצול: DQN** משתמש בגישה הנקראת "חקר וניצול" (Exploration and Exploitation) ישנם זמנים שבהם האלגוריתם בוחר פעולה אקראית (חקר) וזמנים שבהם בוחר את הפעולה הטובה ביותר לפי הרשת הנוכחית (ניצול).

**תהליך האימון**

**איסוף חוויות**: האינטראקציה עם הסביבה מביאה לאיסוף של חוויות (מצב, פעולה, תגמול, מצב הבא).

**עדכון פונקציית Q:** שימוש בחוויות שנאספו כדי לעדכן את משקלי הרשת הניורונית בהתאם לנוסחת ה-Q-Learning.

**עדכון רשת המטרה**: עדכון רשת המטרה במרווחים קבועים כדי לשמור על יציבות הלמידה.