עבודה 2 – Deep Reinforcement Learning

**הוראות הרצה – קבצי py**

- בכל קובץ, יצרנו פעולות של run של המודל, לכן על מנת להריץ את הקוד יש להריץ את אחד הקבצים הנבחרים.   
בנוסף, הפעולה run מחזירה 3 רשימות: רשימה של כל הEpisodes, Reward עבור כל Episode וערך הscore הממוצע עבור 100 Episodes רצופים.

חלק 1 –REINFORCE with value-function baseline

**שאלה 1 - תשובה***ה*Advantage *מייצג את התגמול הנוסף שניתן להרוויח מביצוע פעולה במצב מסוים, ביחס ל*baseline*, ומוגדר כהפרש בין ה*reward *המצטבר ל*Value-Function*.*

*כדאי לעקוב אחרי ה*gradient *אשר מחושב עם ה*advantage *מפני שבאלגוריתם* REINFORCE *ללא ה*advantage*, אנו מעדכנים את פרמטר המדיניות באמצעות עדכוני מונטה קרלו. זה גורם ל*gradients *רועשים, שעלולים להוביל ללמידה לא יציבה ואיטית, ולהטיית חלוקת המדיניות לכיוון לא אופטימלי. אך, הפחתת ה*reward *המצטבר מה*baseline *גורמת ל*gradients *קטנים יותר, ובכך לעדכונים קטנים ויציבים יותר.*

**שאלה 2 – תשובה**

נרצה להוכיח את השוויון הבא:

מתוך **ההנחה כי פונקציית הbaseline אינה תלויה בפעולה** , נקבל:

לכן, נפתח את הצד השמאלי באמצעות חוקי תוחלות ונקבל:

כעת, נפתח על פי הזהות הנ"ל:

כעת, קיבלנו אינטגרל על פונקציית צפיפות של התפלגות מותנית, אשר מסתכמת ל1, ונקבל:

כלומר, קיבלנו ש:

**ניסויים – השוואה**

REINFORCE**:**

**Architecture – PolicyNetwork** for regular **REINFORCE**

|  |  |  |
| --- | --- | --- |
| Activation | Units | Layer |
| - | 4 – Observation Space | Input |
| **Relu** | **12** | **Hidden** |
| Linear | 1 – Action Space | Output |

REINFORCE with Baseline**:**

**Architecture - PolicyNetwork** for **REINFORCE with Baseline**

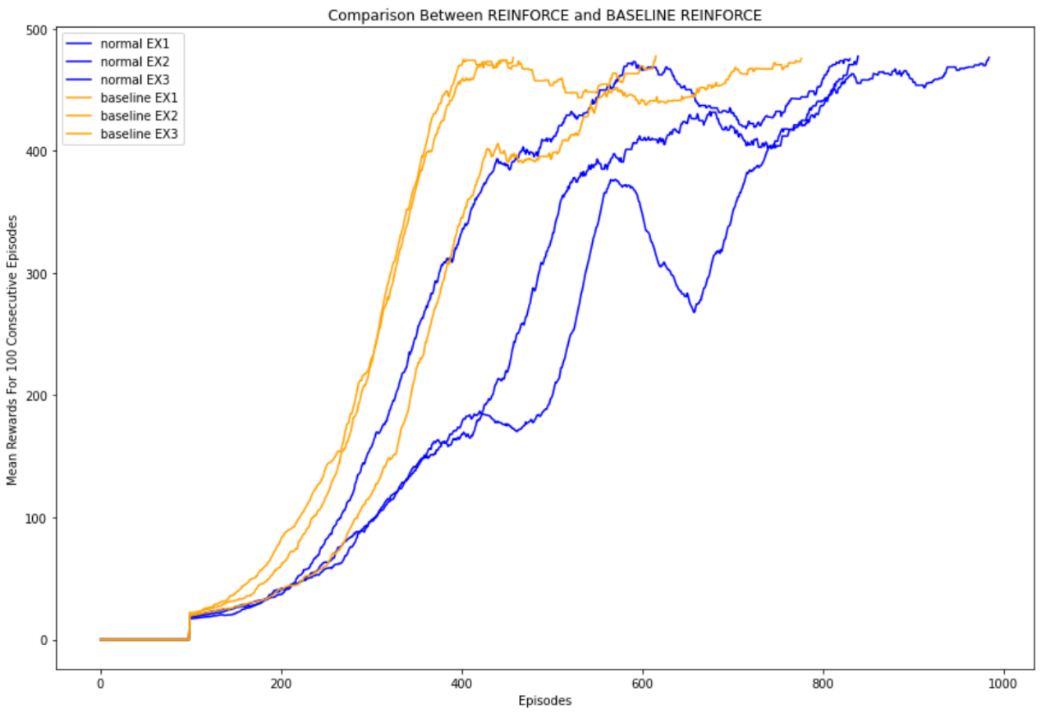
|  |  |  |
| --- | --- | --- |
| Activation | Units | Layer |
| - | 4 – Observation Space | Input |
| **Relu** | **12** | **Hidden** |
| Linear | 1 – Action Space | Output |

**Architecture – StateValueNetwork** for **REINFORCE with Baseline**

|  |  |  |
| --- | --- | --- |
| Activation | Units | Layer |
| - | 4 – Observation Space | Input |
| **Relu** | **8** | **Hidden** |
| **Relu** | **8** | **Hidden** |
| Linear | 1 – Action Space | Output |

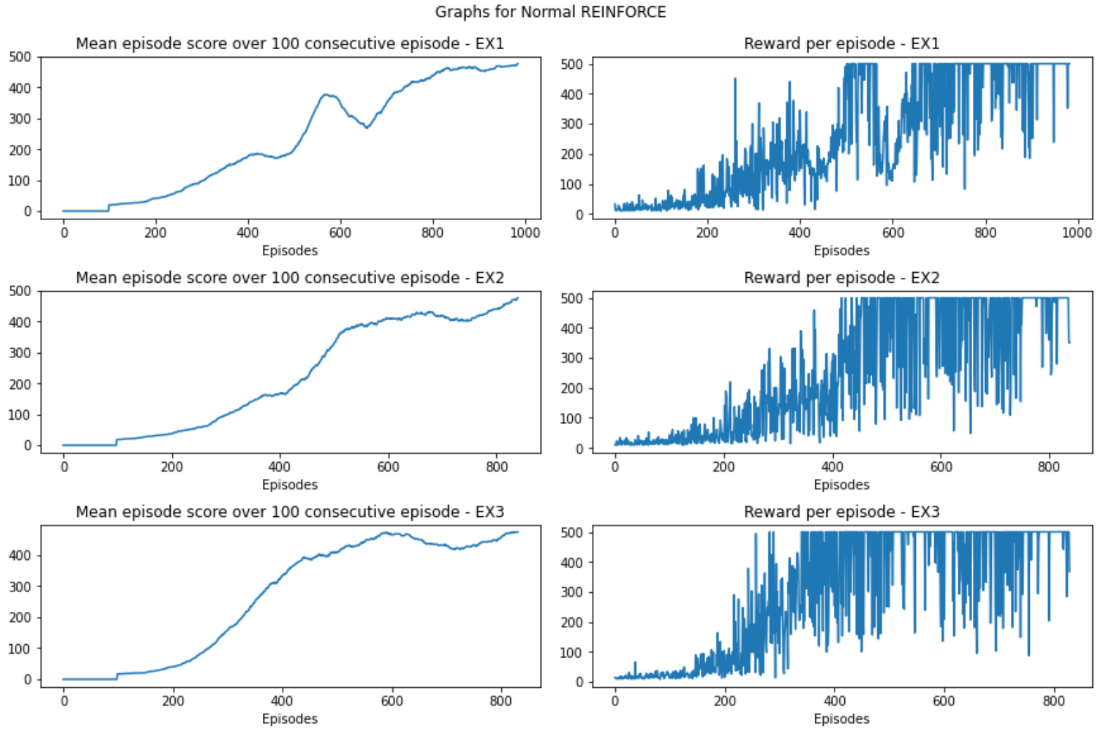
**Final Hyper-parameters**

*עבור הארכיטקטורה, הרצנו את המודל 3 פעמים בשביל להעריך את המודל עבור כמה הרצות שונות.*

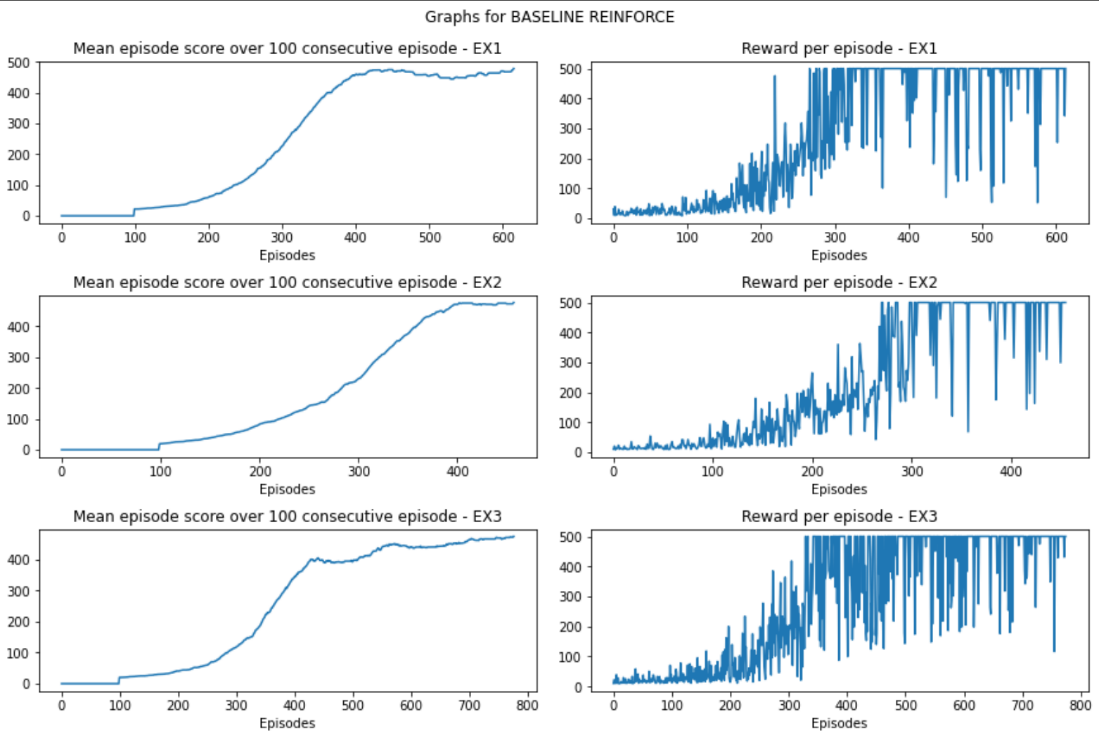
**

*התכנסות: אצלנו, אפשר להגיד שהמודל התכנס אם הוא הגיע למצב שממוצע 100 ה*Episodes *האחרונים שלו הוא מעל 475.  
לפי הגרף לעיל, אפשר לראות שהמודל עם ה* BASELINE *מתכנס יותר מהר מהמודל בלי ה* BASELINE*. עבור מודל* BASELINE *אפשר לראות ה*episode *הממוצע שלו להתכנסות הוא באזור ה600 לעומת המודל הרגיל שהוא מתכנס בממוצע אחרי 850.*

*בנוסף,* נציג את הגרפים עבור ערכי הReward בכל Episode וערך הscore הממוצע עבור 100 Episodes רצופים עבור הרצה של אלגוריתם REINFORCE ו- Baseline REINFORCE:



ניתן לראות כי יש מגמת עליה אחרי בערך 200 episodes וגם שהמודל מגיע לפרס המקסימלי אחרי בערך 400 episodes.



ניתן לראות כי במודל של ה BASELINE ההתכנסות יותר מהירה והוא מגיע לפרס מקסימלי כבר באזור ה 300 episodes שזה 100 פחות מהמודל הרגיל.

בנוסף, ניתן לראות שבמודל ללא הBaseline יש הרבה קפיצות קיצוניות בערכי הReward לעומת המודל עם הBaseline אשר עם ערכים קבועים יותר יחסית.

חלק 2 – Advantage Actor-Critic

**שאלה 1 – תשובה**

error TD *היא ההפרש בין התגמול העתידי הצפוי לאומדן הנוכחי של פונקציית הערך. זה ניתן על ידי המשוואה:*

*פונקציית היתרון היא אומדן של כמה טובה פעולה נתונה בהשוואה לפעולה הממוצעת במצב נתון. היא מוגדרת כשגיאת TD הצפויה עבור אותה פעולה, ניתנת על ידי:*

*אפשר להגיע לפונקציית היתרון על ידי נטילת התוחלת לשגיאת TD על פני כל הפעולות האפשריות:*

ראש הטופס

*לכן, השימוש בשגיאת TD של פונקציית הערך עבור עדכון פרמטרי רשת המדיניות זהה לשימוש הערכת היתרון.*

**שאלה 2 - תשובה**

ה policy-function היא פונקציית הActor, וה value-function היא פונקציית הCritic.

* תפקידו של הActor הוא ללמוד את הPolicy שתמקסם את התגמול הצפוי על ידי בחירת הפעולה הטובה ביותר בכל מדינה.
* תפקידו של הCritic, לעומת זאת, הוא ללמוד את הvalue-function, שמעריכה את התגמול העתידי הצפוי עבור מצב נתון.

**ניסויים – השוואה**

* הארכיטקטורה עבור האלגוריתמים REINFORCE ו- REINFORCE with Baseline מוגדרים כפי שהצגנו בחלק 1.

Actor-Critic**:**

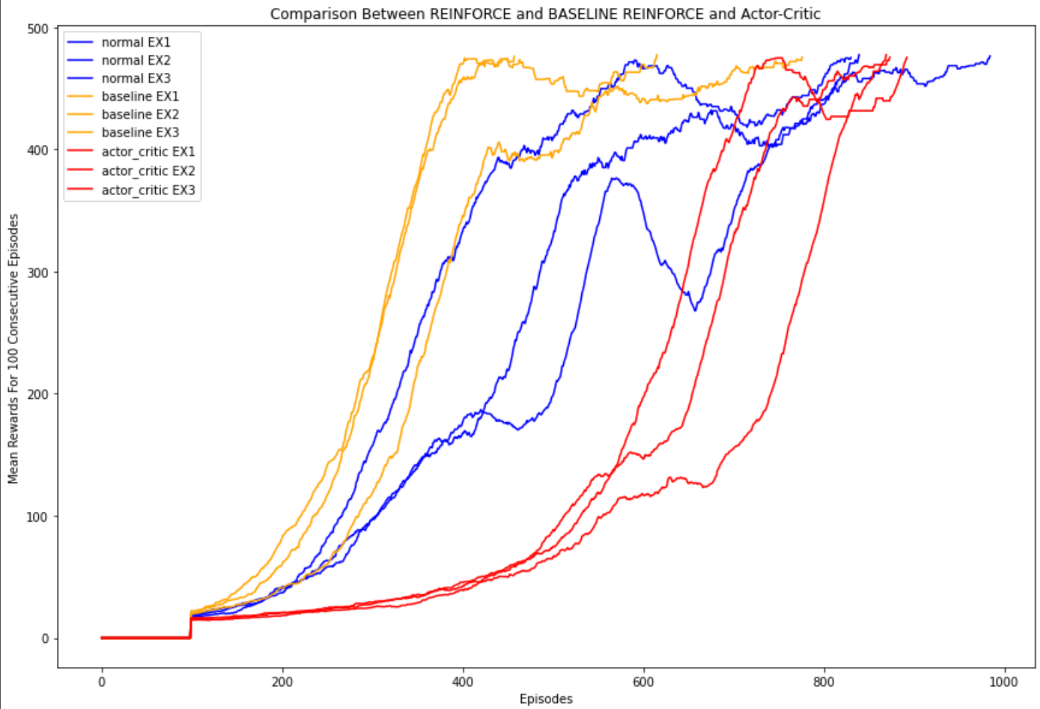
**Architecture - PolicyNetwork** for **Actor-Critic**

|  |  |  |
| --- | --- | --- |
| Activation | Units | Layer |
| - | 4 – Observation Space | Input |
| **Relu** | **12** | **Hidden** |
| Linear | 1 – Action Space | Output |

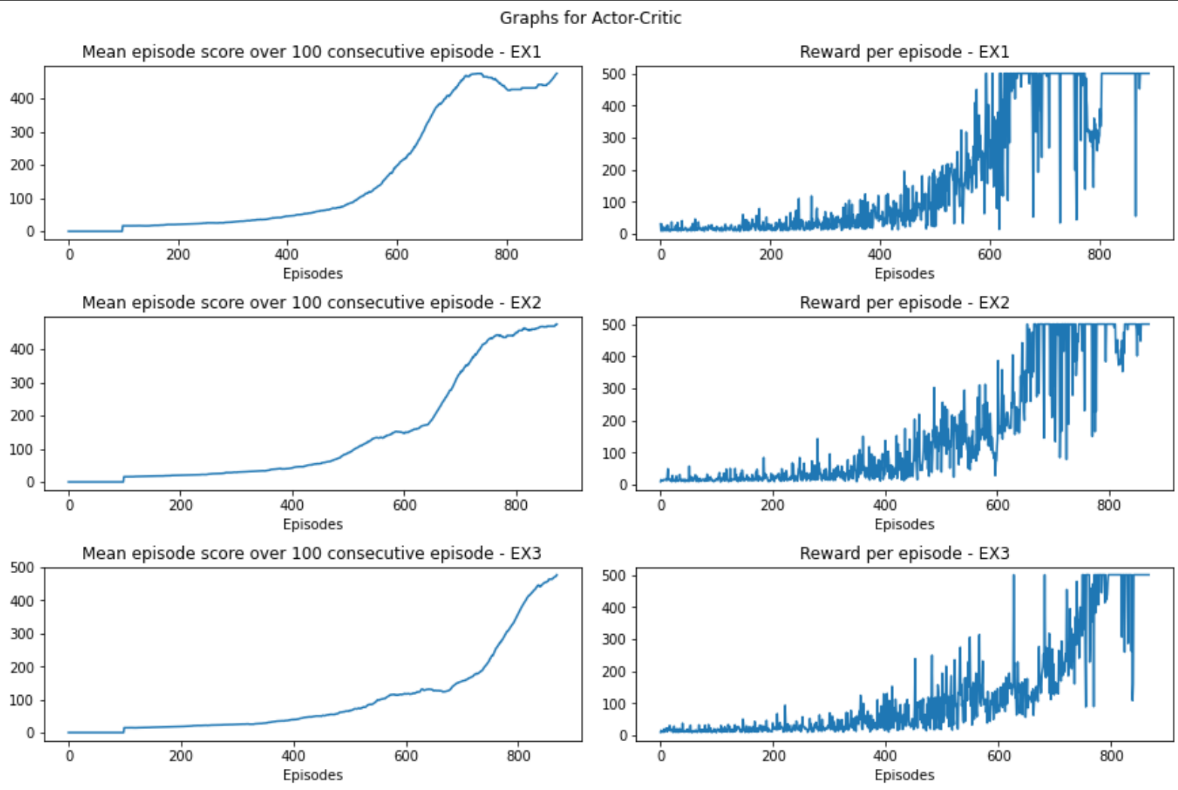
**Architecture – StateValueNetwork** for **Actor-Critic**

|  |  |  |
| --- | --- | --- |
| Activation | Units | Layer |
| - | 4 – Observation Space | Input |
| **Relu** | **64** | **Hidden** |
| **Relu** | **32** | **Hidden** |
| Linear | 1 – Action Space | Output |

**Final Hyper-parameters**

*עבור הארכיטקטורה, הרצנו את המודל 3 פעמים בשביל להעריך את המודל עבור כמה הרצות שונות.*

*אחרי שהרצנו מספר ניסויים ושיחקנו עם פרמטרים ראינו שהביצועים של מודל actor-critic הם פחות טובים מהביצועים של* BASELINE REINFORCE.  
באופן ממוצע הוא מתכנס לאחר 800 episodes בדומה לBASE REINFORCE שמתכנס לאחר 850. אבל המגמת עלייה שלו מתבצעת מאוחר יותר באזור ה 600 episodes.

נציג את הגרפים עבור ערכי הReward בכל Episode וערך הscore הממוצע עבור 100 Episodes רצופים עבור הרצה של אלגוריתם Actor-Critic:

ניתן לראות כי המודל מגיע לפרס מקסימלי לראשונה בממוצע באזור ה 750 episodes שזה כמעט פי 2 ממודל ה BASELINE REINFORCE.