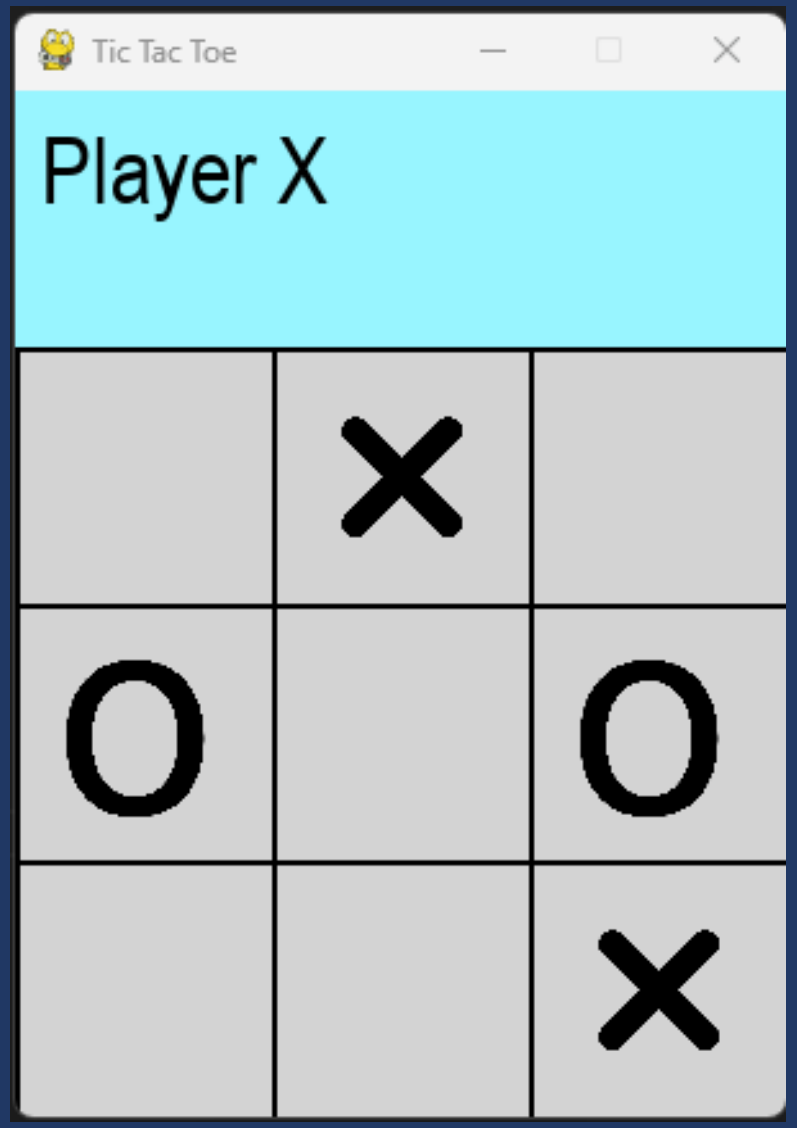


גלעד מרקמן

Entropy Regularization

קישור ל GitHub

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

Exploration v. Explotation

- דנו בהרצאות הקודמים בנושא של חקר Exploration לעומת ניצול ידע קיים Explotation.
  - לנצל (Exploit) לבחור את הפעולה שנראית הכי טובה, על בסיס הידע הקיים.
  - לחקור (Explore) לבחור פעולות פחות בטוחות כדי לגלות אם ישנן אפשרויות טובות יותר.
- אם הסוכן יפעל רק לפי הידע הקיים אצלו הוא לא יתקדם ולא יגלה פתרון טוב יותר.
- באלגוריתם DQN עשינו שימוש ב e-greedy. בחרנו מספר קטן e כך שבהסתברות קטנה נבחר צעד אקראי ואילו בשאר המקרים נבחר את הצעד הטוב ביותר הידוע לנו.

חקר ב Policy Gradient

- באלגוריתמים המבוססים על Policy Gradient החקר (exploration) הינו חלק אינהרנטי של המודל, שהינו הסתברותי.
- בחירת הפעולה נעשית על ידי דגימה הסתברותית מרשימת ההתפלגויות, ולא על ידי בחירה של הפעולה עם ההתפלגות הכי גבוהה.
- כך לדוגמה, אם יש לנו שלוש פעולות עם התפלגות: [0.3, 0.1, 0.6] אנחנו נבחר 60% מהפעמים את הפעולה השלישית, 30% את הפעולה הראשונה ו- 10% את הפעולה השניה.

בעיית החקר ב Policy Gradient

- הבעיה באלגוריתם שתוך כדי אימון ההסתברויות משתנות כך שפעולה אחת מקבלת הסתברות גבוהה מאוד ואילו שאר הפעולות מקבלות הסתברויות נמוכות מאוד.
- אם, לדוגמה, לאחר אימון אנחנו מגיעים להסתברויות של: [0.995, 0.003, 0.002], הסיכוי שתבחר פעולה שאינה הראשונה הוא קלוש.
- ב Policy Gradient אין אפשרות להשתמש ב e-greedy , ולכן עלינו למצוא דרך אחרת להגביר את החקר.
- לצורך כך נעשה שימוש ב entropy.

Entropy – אי וודאות

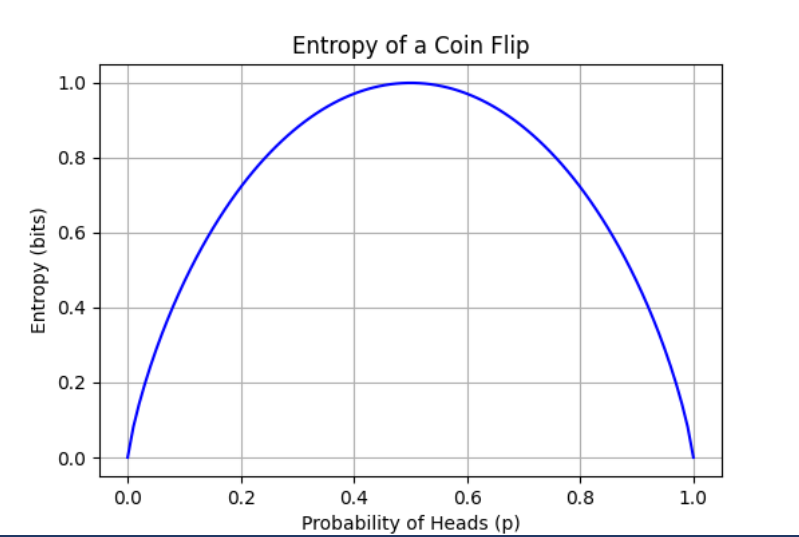
- ה entropy הוא מדד מתמטי לאי ודאות או אקראיות בהתפלגות בדידה.
- מדד entropy גבוה מציין מידת אי ודאות גבוהה. היכולת שלנו לצפות את התוצאה קטנה. לעומת זאת, מדד entropy נמוך אי הודאות קטנה, ואנחנו יכולים לצפות את התוצאה בסיכוי גבוה.
- הנוסחה המתמטית לחישוב entropy היא:

$$H(X) = -\sum_{i} p_i * \log p_i$$

הסבר הנוסחה ל entropy

נסביר את הנוסחה באמצעות דוגמה: $H(X) = -\sum_i p_i * \log p_i$

- נניח יש לנו מטבע תקין שההסתברות לראש וזנב היא [0.5,0.5].
  - מידת אי הודאות היא: $-(0.5 * \log 0.5 + 0.5 * \log 0.5) = 1$
- נניח הפעם המטבע תמיד יוצא ראש. ההתפלגות היא קרוב ל- [1,0].
  - מידת הודאות היא: $-(1 * \log 1 + 0 * \log 0) = 0$



- נשרטט הגרף של הפונקציה כאשר x הוא משתנה 1 (השני הוא 1-x).
- כאשר p=0.5 מידת אי הודאות היא הכי גדולה.
- כאשר p=0 או p=1 אין אי ודאות כלל.

שימוש ב entropy באלגוריתם

- אנחנו מעוניינים להגדיל את אי הודאות (entropy) של ההסתברויות, כך שלא תהיה פעולה שתשלוט על ההסתברויות.
- במחלקה distribution יש פעולה המחשבת את ה entropy:

Entropy = dist.entropy()

- אם נדאג ש entropy של הסתברות הפעולות לא יהיה קטן מידי, אזי אנחנו דואגים לקיים חקר. הפעולה dist.sample() תבחר מידי פעם פעולות שונות.

Entropy Regularization

```python
def learn(self):

    states, actions, rewards = self.memory.get_tensors()

    G = self.compute_G(rewards)

    dist = self.get_masked_dist(states)
    log_probs = dist.log_prob(actions)

    # Compute the loss using vectorized operations
    policy_loss = -(G * log_probs).sum()

    # Add Entropy regularization
    entropy = dist.entropy().mean()
    loss = policy_loss - self.entropy_coe * entropy


    # backward
    self.policy.optimizer.zero_grad()
    loss.backward()

    self.policy.optimizer.step()

    #### for logging
    self.sum_loss += loss.detach()
    self.sum_entropy += dist.entropy().detach().mean()
```

- נשנה את פונקציית הלמידה ונוסיף את נושא ה entropy.
- נשים לב שה entropy מוסף עם מינוס.
- חישוב ה entropy הוא חלק מגרף החישוב, ולכן כאשר אנחנו מנסים להקטין את ה loss אנחנו למעשה מגדילים את ה entropy.
- עדכון הפרמטרים מצד אחד יגדיל את ההסתברות של הפעולה העדיפה, אולם מנגד ינסה להגדיל את ה entropy ובכך ימנע הגדלה רבה מידי של הפעולה הנבחרת.
- ב entropy coe הוא פרמטר קטן [0.2-0.01] שנועד להגביל את ה entropy אחרת הסוכן יהיה יותר מידי רנדומלי.