# 3 Theorie

1. **Ist A2C ein On- oder Off-policy Lerner? Warum?**

A2C ist Teil der on-policy Familie, da die Value Funktion anhand der Policy gelernt wird, während man dieser folgt. Anders ausgedrückt kann die Value Funktion nicht gelernt werden, wenn man einer anderen Policy folgt.
Alternativ Formuliert:
Da keine Experience Replay zum lernen genutzt wird, die ihre Werte aus einer anderen Policy bezieht, ist der A2C ein On-Policy Lerner.Würden Samples benutzt werden, die vom Netz **vor** einem Update erzeugt wurden, würde es sich um einen Off-Policy Lerner handeln, da diese nicht von der aktuellen Policy generiert worden wären.
Im Fall des A2C werden die Samples der Policy vor einem Update des Netzes jedoch verworfen, weshalb es sich um einen On-Policy Lerner handelt.

2. **Welche Mechanik von A2C bricht die Korrelation der einzelnen Beobachtungen?**

Da es beim A2C mehrere, unabhängige Agenten initialisiert werden, die die Sample Daten erzeugen, mit denen gelernt wird, wird die Korrelation der Sample-Daten gebrochen. Ein Update erfolgt immer auf Basis des Fehlers eines Batches, wobei dieser aus mehreren Messwerten von verschiedenen unabhängigen Agenten bestehen.

3. **Warum muss die Advantage im Policy-Loss eine Konstante sein?**

Da der Policy Loss sowohl von der Advantage, als auch von den aktuellen Wahrscheinlichkeit der Aktionen abhängt, sind hier potenziell Anpassungen und Fehlerberechnung von beiden Arten der Parameter möglich und würden zu einer Anpassung des Values und der Policy Wahrscheinlichkeiten führen, wenn die Advantage nicht als Konstante behandelt werden würde. Dieses Verhalten ist jedoch unerwünscht, der Fehler im vorhergesagten Value soll im Rahmen des Value Losses bestimmt werden. Würde man diese Trennung nicht vornehmen, so läge eine Korrelation von Policy und Value vor, welche nach Möglichkeit minimiert oder verhindert werden sollte, sonst können Feedbackloops auftreten.

4. **Warum benötigt A2C keine explizite Explorationsstrategie (bspw. $\epsilon$-greedy) mehr? Wie funktioniert Exploration hier? Erklären Sie anhand eines kleinen (Rechen-)Beispiels.**

Dadurch, dass jeder Agent sich unabhängig (mit eigener Environment) von den anderen bewegt wird automatisch exploriert und die Notwendigkeit für eine Experience Replay fällt weg.

Da die gewählten Aktionen durch Wahrscheinlichkeiten im Netz initialisiert werden, gilt am Anfang:

da 8 Aktionen: W'keit pro Aktion ca 12.5%

nach einigen Gradient Ascents hat Agent gelernt, dass eine Aktion besser ist als die anderen
z.B. wenn Ziel NO:
W'keit N 20%, NO 30%, O 19%, SO 10%, S 8%, SW 4%, W 9%

wählt der Agent nun (unwahrscheinlicher, aber möglicher Weise) W und erreicht das Ziel somit nicht in 5 Schritten, so ist der diskontierte Reward für diese Aktion:
$-0.1 + 0.99(-0.1) + 0.99^3(-0.1) + 0.99^4(-0.1) + 0.99^5(-0.1) + V(s_5)$
Dadurch wird der Policy Loss maximal (da dieser der schlechtest mögliche Outcome für die 5 betrachteten Schritte ist)
Die Gewichte im Netz werden dadurch angepasst, dass im nächsten Schritt eine geringere W'keit für die Aktion W ausgegeben wird.

Die andere Mechanik, die für Exploration sorgt, ist der Entropie Loss

```
H = -1* K.sum(actor_predicted * K.log(actor_predicted),
axis=-1)
entropy_loss = -1* K.mean(H)
```

⇒ durch das maximieren (−H) der Entropy bewegen wir die Policy weg vom deterministischen Zustand
Zu Beginn sorgt die Entropie für eine stärkere Exploration, nach einiger Zeit überwiegt der Policy-Loss (fällt stärker ins Gewicht) wodurch eine Exploitation erfolgt.