

Datenanalyse für die Lageaufklärung

Künstliche Intelligenz für die Lagedarstellung aus Videosequenzen kommerzieller Flugroboter im Deutschen Rettungsrobotik-Zentrum

Autor: Hartmut Surmann

Algorithm 1 Asynchronous one-step Q-learning - pseudocode for each actor-learner thread.

```

// Assume global shared  $\theta$ ,  $\theta^-$ , and counter  $T = 0$ .
Initialize thread step counter  $t \leftarrow 0$ 
Initialize target network weights  $\theta^- \leftarrow \theta$ 
Initialize network gradients  $d\theta \leftarrow 0$ 
Get initial state  $s$ 
repeat
  Take action  $a$  with  $\epsilon$ -greedy policy based on  $Q(s, a; \theta)$ 
  Receive new state  $s'$  and reward  $r$ 
   $y = \begin{cases} r & \text{for terminal } s' \\ r + \gamma \max_{a'} Q(s', a'; \theta^-) & \text{for non-terminal } s' \end{cases}$ 
  Accumulate gradients wrt  $\theta$ :  $d\theta \leftarrow d\theta + \frac{\partial(y - Q(s, a; \theta))^2}{\partial \theta}$ 
   $s = s'$ 
   $T \leftarrow T + 1$  and  $t \leftarrow t + 1$ 
  if  $T \bmod I_{target} == 0$  then
    Update the target network  $\theta^- \leftarrow \theta$ 
  end if
  if  $t \bmod I_{asyncUpdate} == 0$  or  $s$  is terminal then
    Perform asynchronous update of  $\theta$  using  $d\theta$ .
    Clear gradients  $d\theta \leftarrow 0$ .
  end if
until  $T > T_{max}$ 

```

Motivation

Kommerzielle Flugroboter liefern umfangreiche Bild- und Videodaten in Such- und Rettungseinsätzen. Diese Daten müssen schnell verarbeitet, ausgewertet und für das Lagebild aufbereitet werden. Neue Algorithmen aus der KI-Forschung, insbesondere aus dem Bereich des „Deep Learnings“ versprechen hier eine verbesserte Unterstützung der Helfer. Voraussetzung dafür ist allerdings ein tiefgehendes mathematisch/informationstechnisches Verständnis der Funktionsweise der Netze, um Fehler zu vermeiden.

Pseudo-Code eines „Deep Learning Actor-Critic Netz“ für das Reinforcement-Learning. Der Roboter lernt mithilfe des Netzwerkes selbständig in unbekanntem Umgebungen sicher zu navigieren.

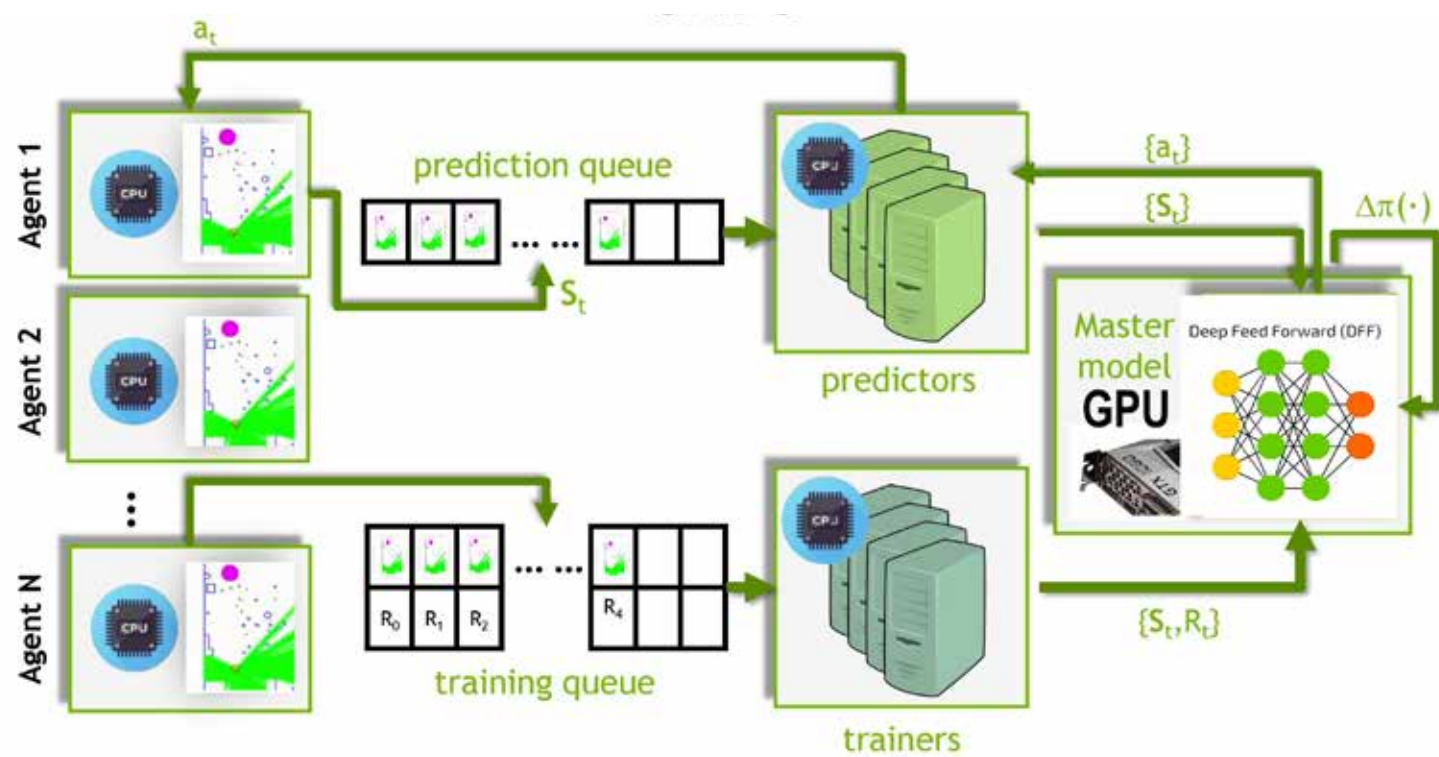


Abb. 1: Architekturbild eines dreifach parallelisierten „Asynchronous Advantage Actor-Critic“-Modells. Die Simulation verwendet Advanced Vector Extensions zur gleichzeitigen Simulation unterschiedlicher Agenten (threads) für das Training eines GA3C verteilt auf der GPU.

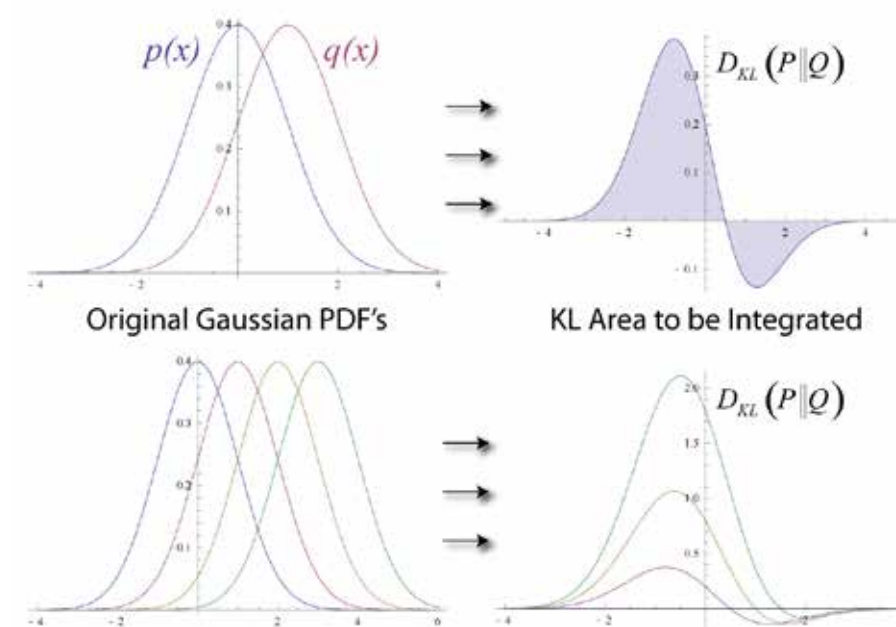


Abb. 2: Illustration der Kullback-Leibler-Divergenz/Entropie zweier Gaußverteilungen. Die KL-Divergenz wird zur Beurteilung der Lernfähigkeit und Performance des neuronalen Netzwerkes verwendet.



Abb. 3: Beispiel für die Szenenklassifikation eines konvolutionalen neuronalen Netzes (Mask-RCNN).



Abb. 4: Erkennung der menschlichen Pose in einem RGB-Kamerabild während eines Deckenfeuers.

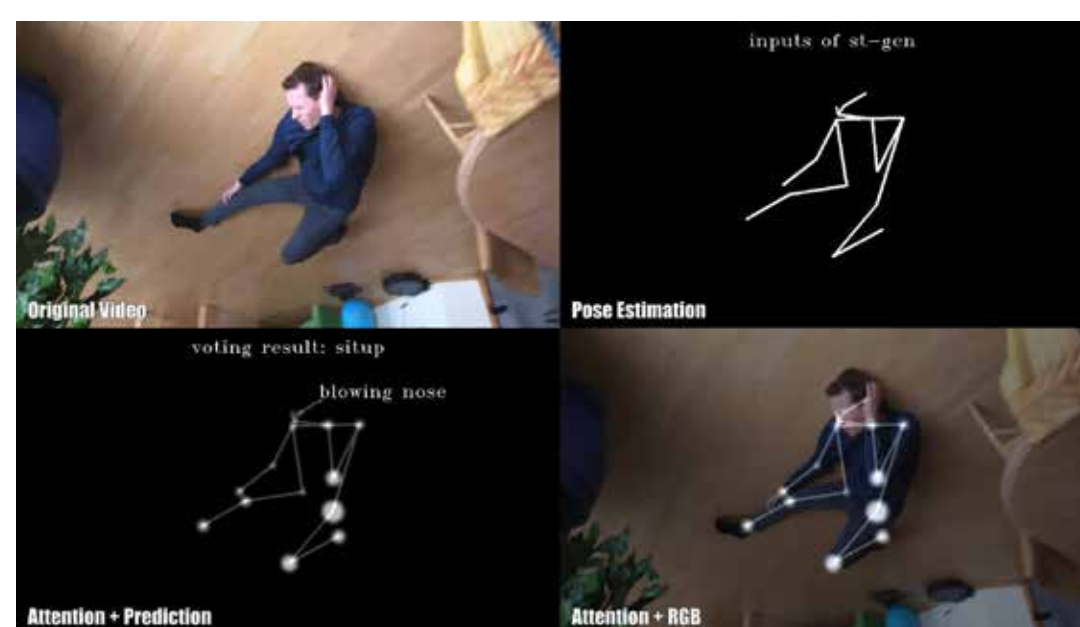


Abb. 5: Beispiel eines „Spatial Temporal Graph Convolutional Networks“ (ST-GCN) zur menschlichen Aktionsanalyse. Mangelnde Trainingsdaten führen zur Fehlklassifikation.



Abb. 6: 360°-Bild einer Mavic Pro mit einer ONE X 360 beim Einflug in ein Gebäude durch ein Fenster.

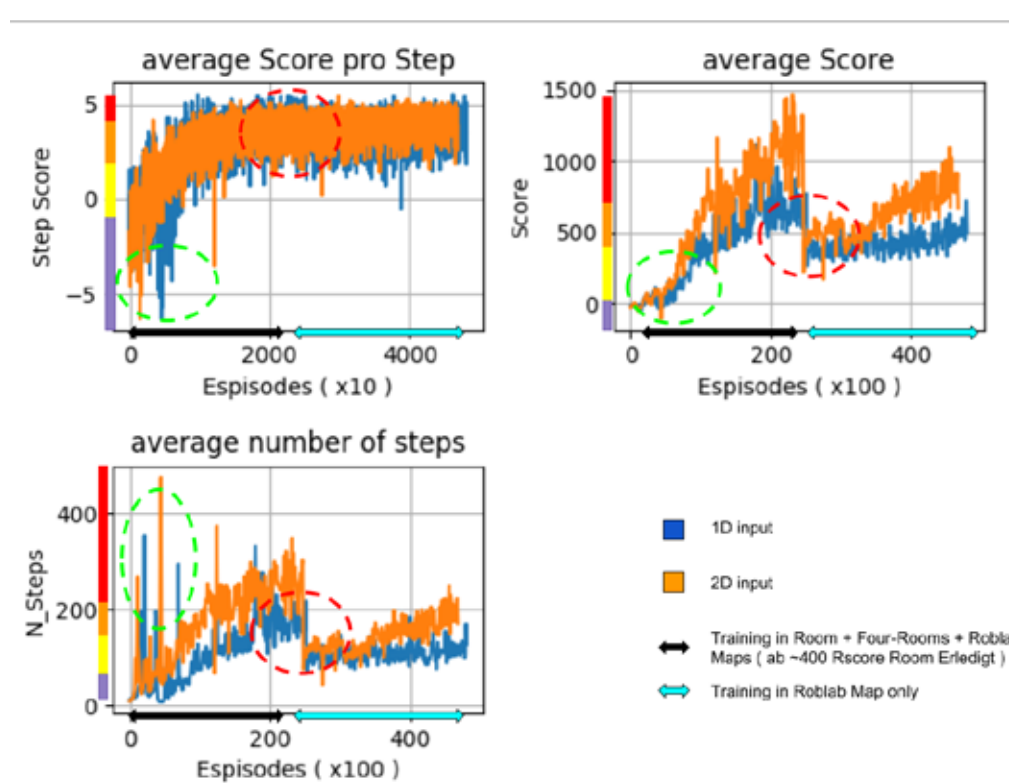


Abb. 7: Lernerfolg eines „Asynchronous Advantage Actor-Critic Networks“ mittels „Reinforcement Learning“ auf einer GPU. Gelernt wurde ein Roboternavigators basierend auf 2D/3D-Abstandsdaten. Anomalien, die beim Lernen auftreten können, sind markiert und gut zu erkennen.



Abb. 8: Nicht nur die KI's müssen trainiert werden! Übung mit den Rettungskräften.



Abb. 9: Beispiel für eine Benutzerschnittstelle. Auf einem Foto werden die Bereiche markiert, die durch die kleinen Flugroboter selbständig erkundet werden sollen.

Videos auf dem Robotikkanal der WH
(<https://www.youtube.com/user/RoblabFhGe>)