Deep-Q-Learning mit Super Mario Bros. und A3C

Jan Gaida
Angewandtes maschinelles Lernen
Hochschule Hof, Juni 2020

01

Einführung

Super Mario Bros. (1985)

- ➔ **Nintendo** Co. Ltd.
- ➔ Erstveröffentlichung:
  - **Japan**: 13. September 1985
  - **Westen**: 1986 bis 1987
- ➔ Platformen:
  - Famicom
  - Nintendo Entertainment System (NES)

Quelle: Datei:雨の日はファミコンで遊べる (15441664223).jpg – Wikipedia

Super Mario Bros. (1985)

- **Nintendo** Co. Ltd.
- Erstveröffentlichung:
  - **Japan**: 13. September 1985
  - **Westen**: 1986 bis 1987
- Platformen:
  - Famicom
  - Nintendo Entertainment System (NES)
- Medium:
  - **40KB** Steckmodul (max. 320 KB)
- Copyright / DMCA:
  - bis (mind.) 2080
  - Ausnahme(n):
    'educational setting or documentary purposes' [1]

Quelle: PicClickImg

[1] Quelle: Albright-Ip

Motivation

Quelle: KnowYourMeme.com

02

Environment

OpenAI Gym

Quelle: Velotio Technologies

Ziele:

- ➔ **Benchmark** von RL-Algorithmen durch eine große Kollektion von diversen Environments
- ➔ **Standardisierung** der Environment für bessere Vergleichbarkeit von RL-Algorithmen

Christian Kauten: Gym-Super-Mario-Bros

https://github.com/Kautenja/gym-super-mario-bros

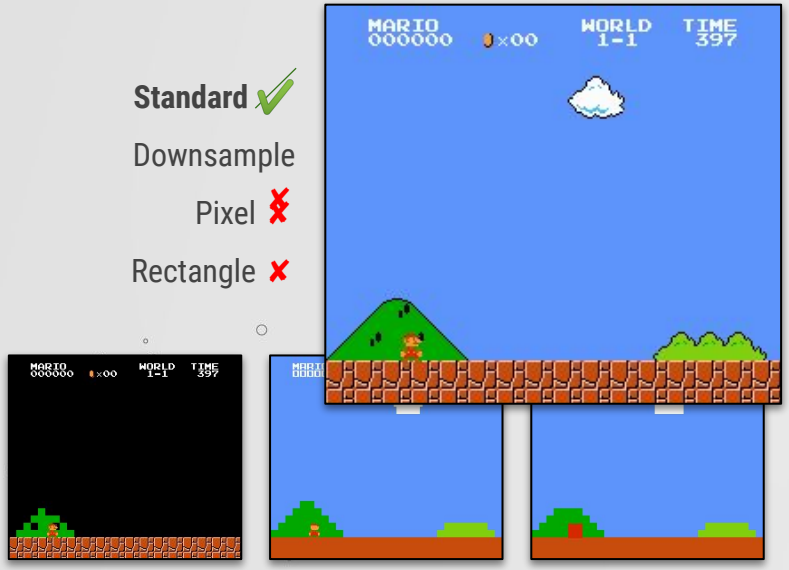
Render-Varianten:

Standard ✔

Downsample

Pixel ✘

Rectangle ✘



Vordefinierte Action-Spaces:

✘ Standard (256-Bit)

✔ **Complex**

✘ Simple

✘ Right-Only

Quelle: Github/Kautenja, Nintendo

Context des Environments

Action:
Array von codierten Aktionsnamen

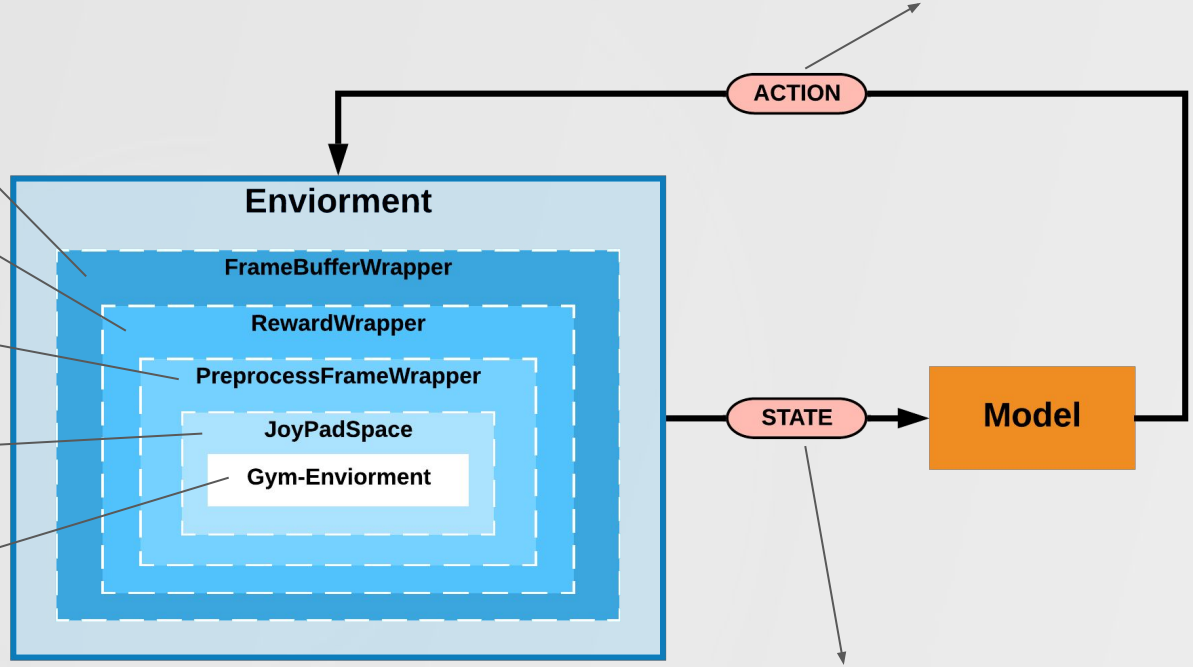
FrameBufferWrapper:
Sammeln mehrerer Frames

RewardWrapper:
Berechnen des Rewards

PreprocessFrameWrapper:
Vorverarbeitung des Frames

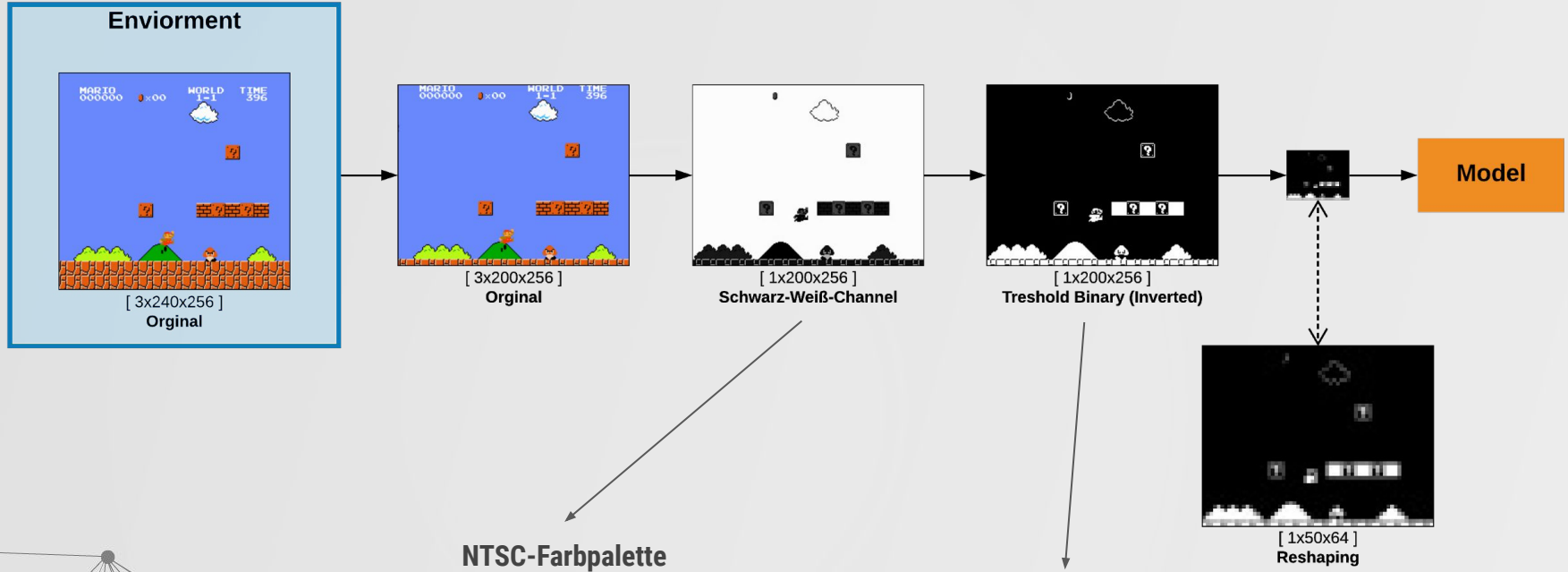
JoypadSpace:
OneHot-Encoding des ActionSpaces

Gym-Environment:
Ursprüngliches Environment



State:
Definiert den Zustand des Environments, hier:
Array<Frame>, Reward, Done-Indikator, Info-Dictionary

Preprocessing



NTSC-Farbpalette
→ Bestitz u.a. einen Helligkeits-Channel

Schwarz-*oder*-Weiß
(Threshold bei 64/256)

Reward

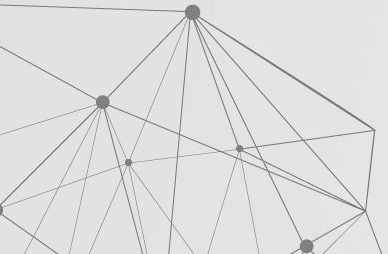
Reward Shaping ✔

- ➔ **Keine Generalisierung** des Problems
- ➔ **Abhängigkeit** von der Qualität der Reward-Funktion
- ➔ (Keine vollständige Exploration)

✘ Sparse Reward

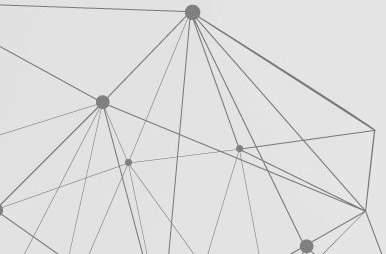
Aktuelles Forschungsgebiet in RL

(Curiosity-driven Exploration, Unsupervised Auxiliary Tasks, Hindsight Experience Replay)



Reward

- ➔ X-Position (W = [len(buffer) * -1; len(buffer)])
  delta_x = x_1 - x_0
- ➔ Zeit (W = [0; (len(buffer)/10)])
  delta_time = min(t_1 - t_0, 0)
- ➔ Erreichtes Ziel (W = [0; 45])
  r_goal = 45 if goal_achieved else 0
- ➔ Verlorenes Leben (W = [-45; 0])
  r_life = -45 if life_lost else 0

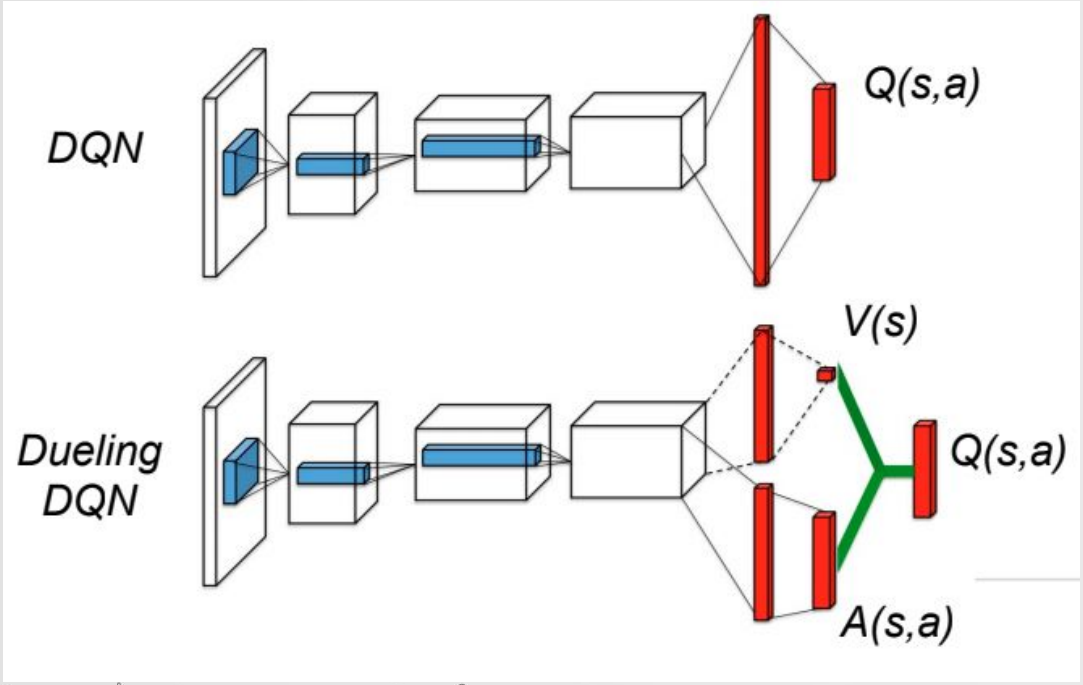


➔ **reward = (delta_x + delta_time + r_goal + r_life) / 10**

03

Advantage Actor-Critic

Warum Actor-Critic ?



➔ Erzeugen einer Q-Table von State's relativ zu der bestmöglichsten Action

➔ Erzeugen einer Q-Table von State's relativ zu der bestmöglichsten Action + Bewertung von aktuellen State

Quelle: Deep RL Bootcamp, Berkeley - Deep Q-Networks (Volodymyr Mnih)

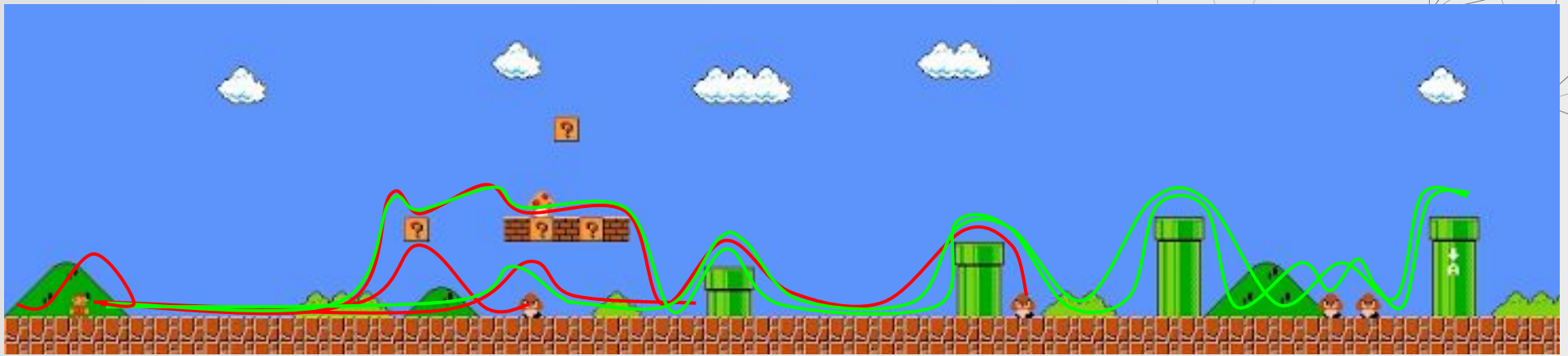
Warum Actor-Critic ?

ble von State's
öglichsten Action

ble von State's
öglichsten Action
tuellen State

Quelle: Deep RL Bootcamp, Berkeley - Deep Q-Networks (Volodymyr Mnih)

Warum Actor-Critic ?

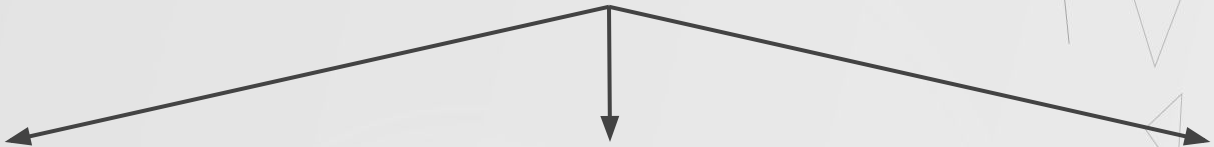


➔ *relativ* großer **ObservationSpace** (Output d. Env.)

➔ *relativ* großer **ActionSpace** (Input d. Env.)

Quelle: Vignette.Wikia

Idee hinter A3C



Asynchronous

Konkurrente Trainings-Prozesse

Advantage

.. unter Berücksichtigung einer geschätzten Bewertung des aktuellen States

Actor-Critic

.. ausgeführt durch einen Aktor in Abhängigkeit eines Kritikers

Keine Q-Tables !

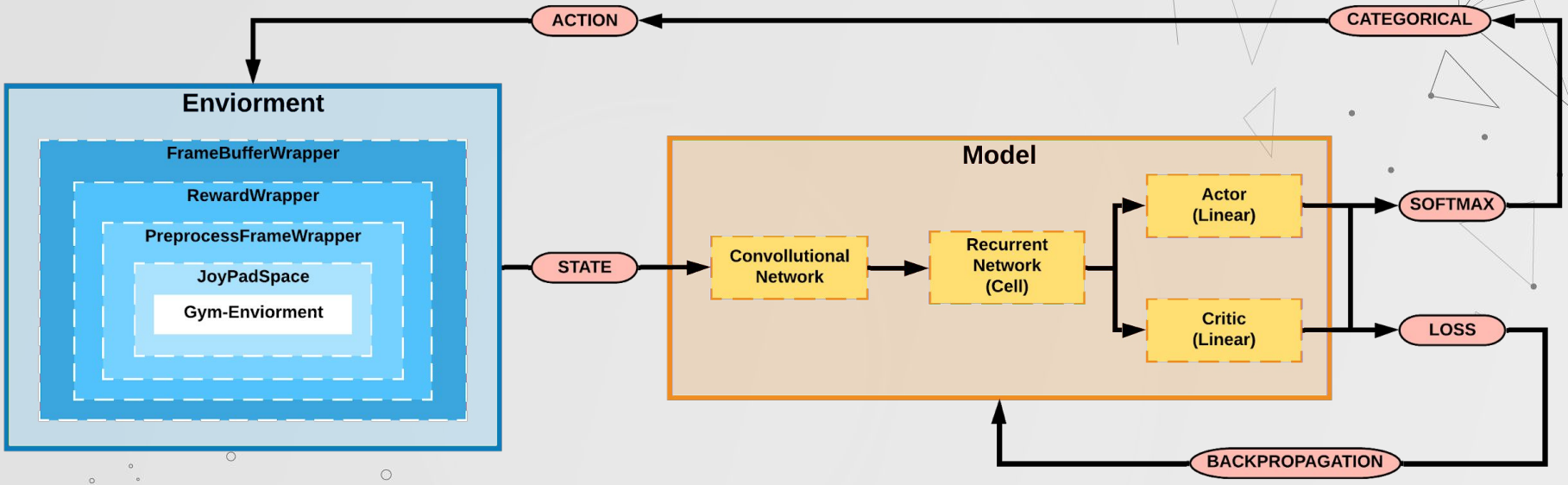
Idee hinter A3C

Beispiel:

[...] Let [us] imagine a small mischievous **child (actor)** [which] is discovering the amazing world around him, while his **dad (critic)** oversees him, to make sure that he does not do anything dangerous. Whenever the kid does anything good, his dad will praise and encourage him to repeat that action in the future. And of course, when the kid does anything harmful, he will get [a] warning from his dad. [...]

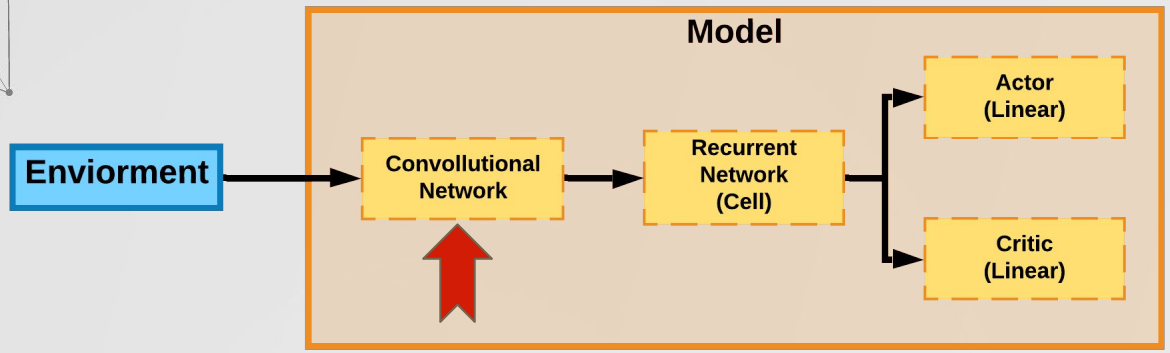
Quelle: github/uvipen

Context A3C


ACTION
CATEGORICAL
Enviorment
FrameBufferWrapper
RewardWrapper
PreprocessFrameWrapper
JoyPadSpace
Gym-Enviorment
STATE
Model
Convollutional Network
Recurrent Network (Cell)
Actor (Linear)
Critic (Linear)
SOFTMAX
LOSS
BACKPROPAGATION

04

Netzwerk Architektur



Approach B

Naiv

Approach B

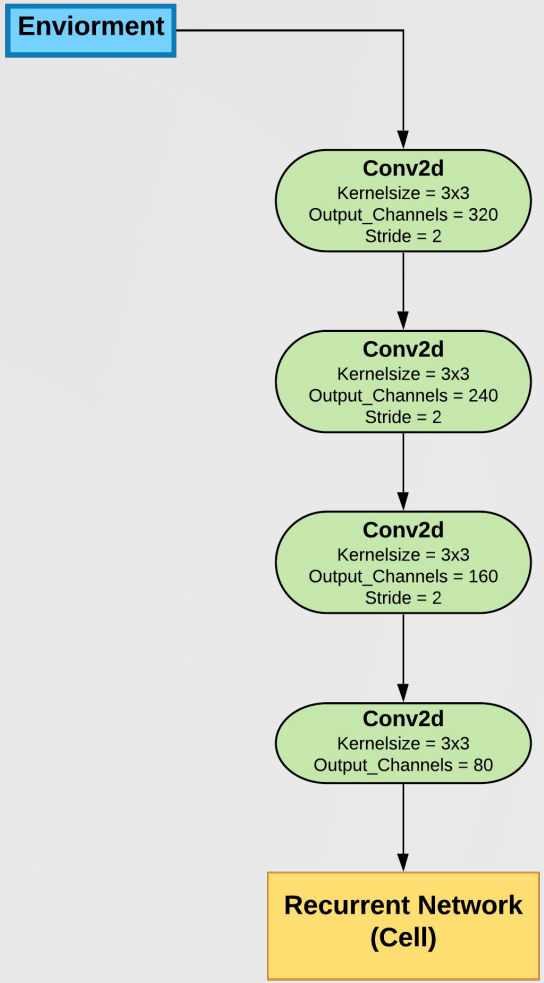
Deep-Convolutional

➔ Auswertung der Bilderfolge

➔ Approach B ist inspiriert von Google_ResNet-Modulen

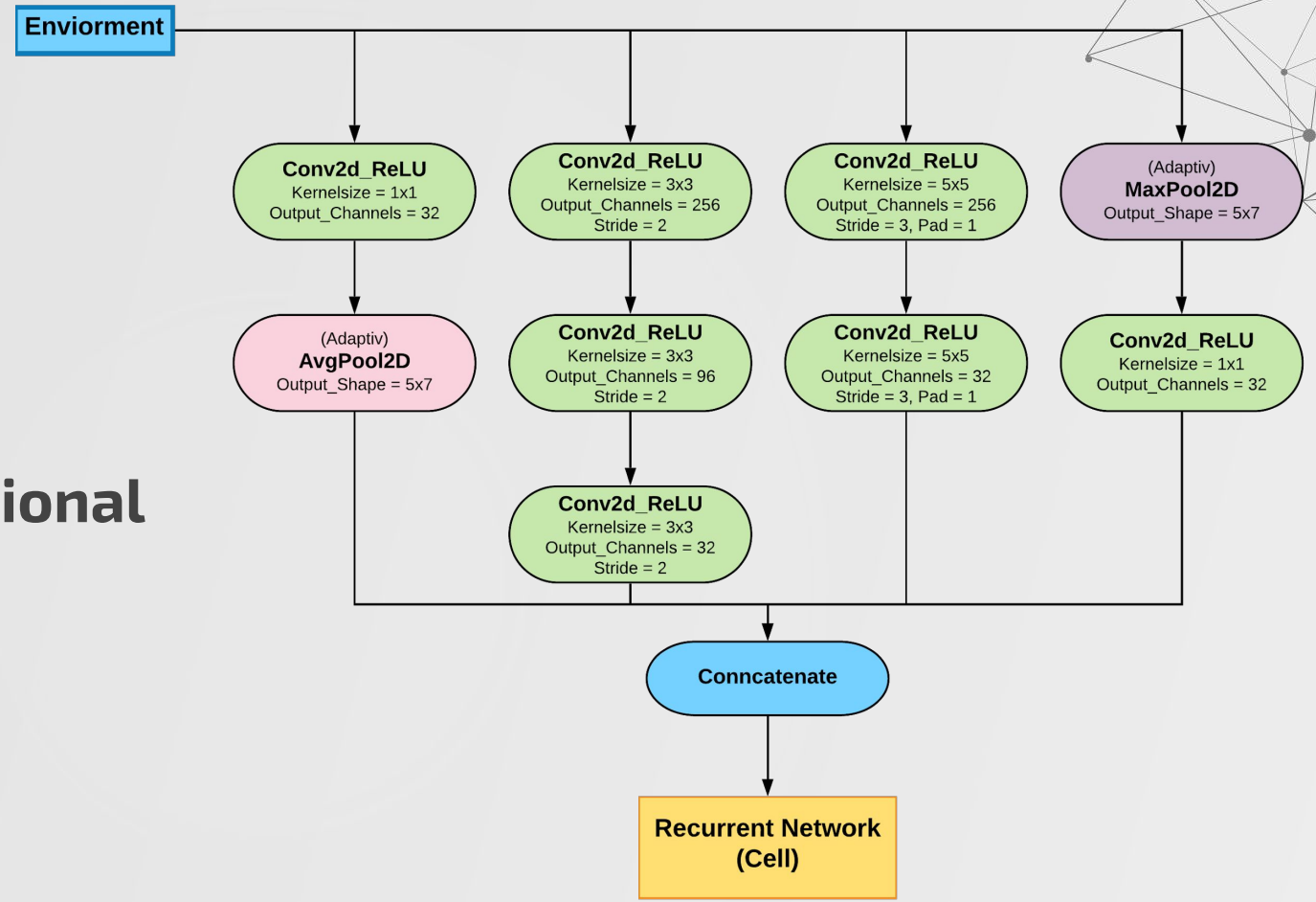
Approach A

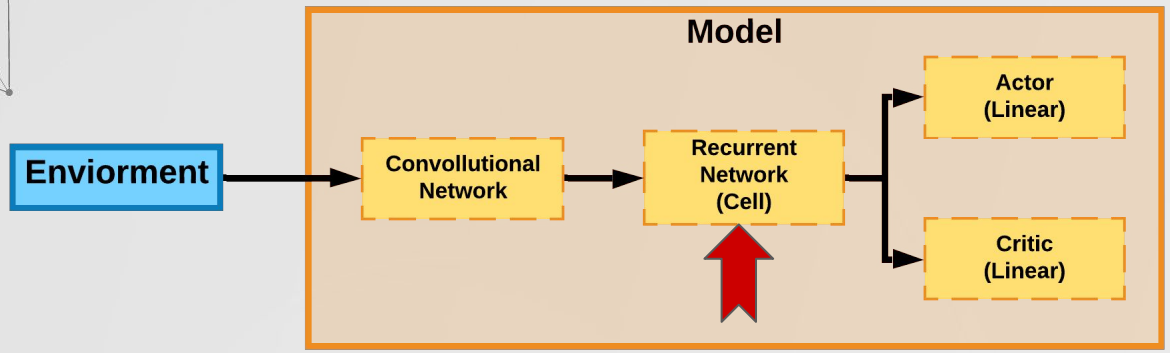
Naiv



Approach B

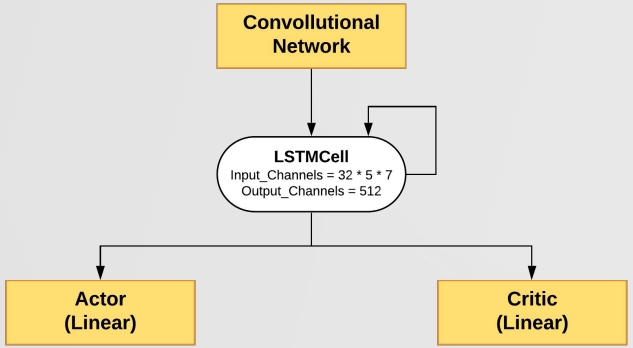
Deep-Convolutional





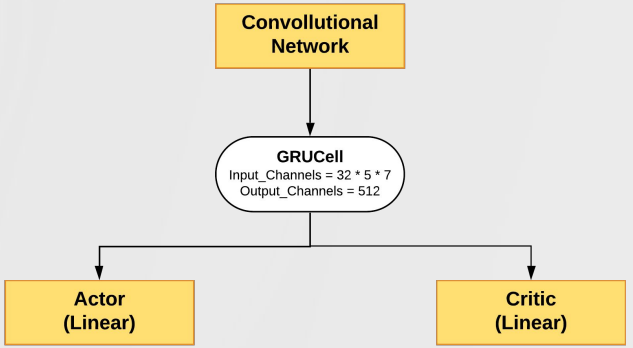
Approach A

LSTM-Zelle



Approach B

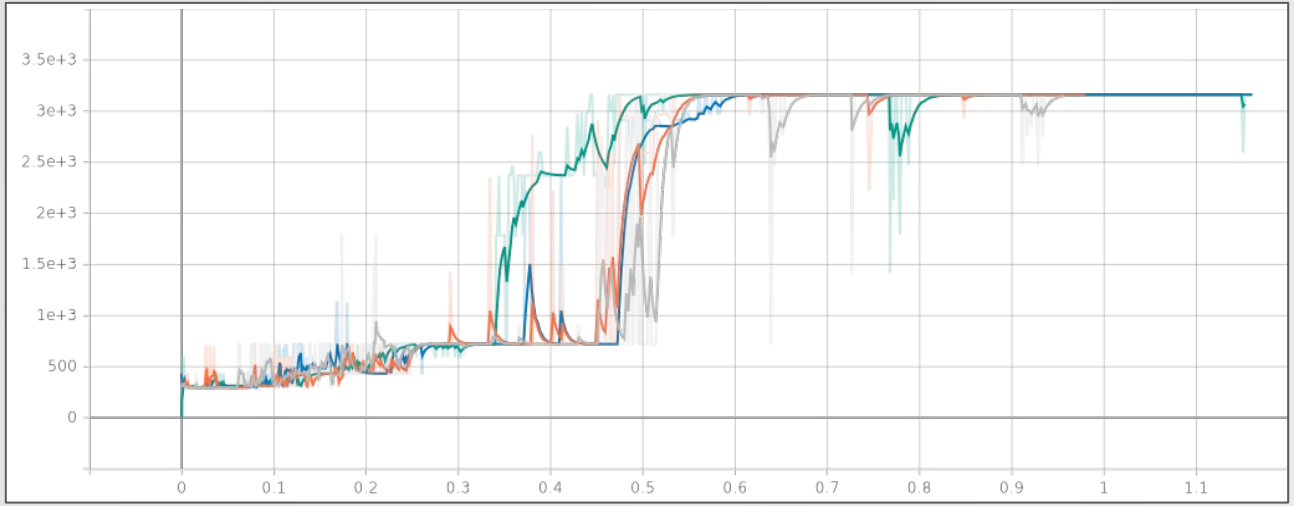
GRU-Zelle



05
Ergebnisse

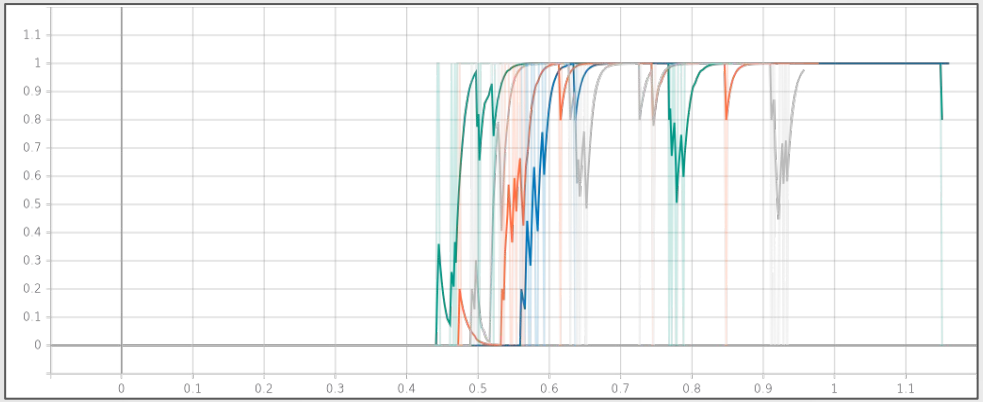
Erreichte X-Position

Stage 1 World 1
3500 Ep x 5 Threads
Smoothingfaktor: 0.8



DCN (GRU)
CN (GRU)
CN (LSTM)
DCN (LSTM)

Flagge



GRU: Convolutional vs DeepConvolutional

CN

1500 Ep x5

2000 Ep x5

2500 Ep x5

3500 Ep x5

DCN

https://docs.google.com/presentation/d/13Ps1Q0eDhY0vCg19zyqMB4chFpsJy2QdCVsPpUVhHl0/edit?usp=sharing

LSTM: Convolutional vs DeepConvolutional

CN

1500 Ep x5

2000 Ep x5

2500 Ep x5

3500 Ep x5

DCN

DCN - LSTM

Stage 1 World 3
2500 Ep x 5 Threads
Smoothingfaktor: 0.8

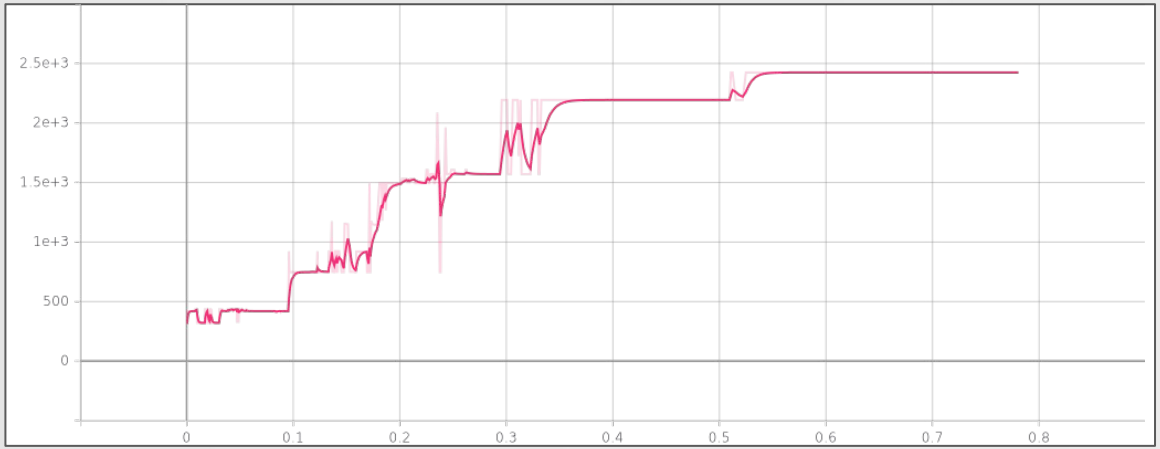
500 Ep x5

1000 Ep x5

2000 Ep x5

2500 Ep x5

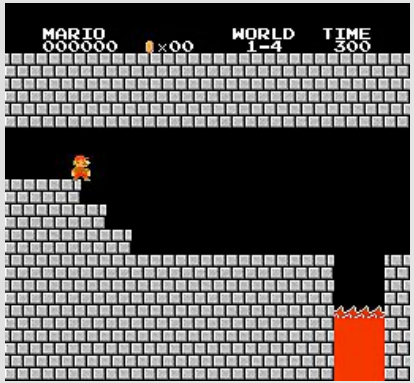
X-Position



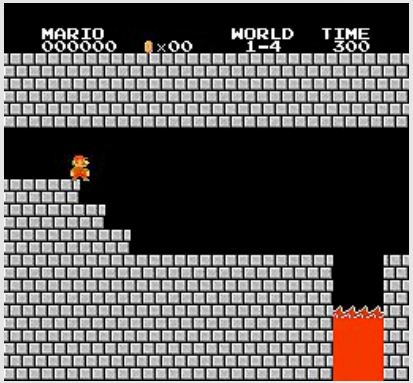
Quelle: Wikipedia

DCN - LSTM

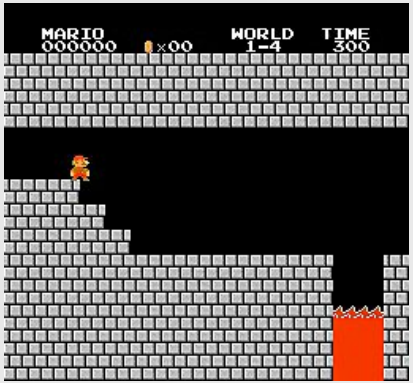
Stage 1 World 3
7500 Ep x 5 Threads
Smoothingfaktor: 0.8



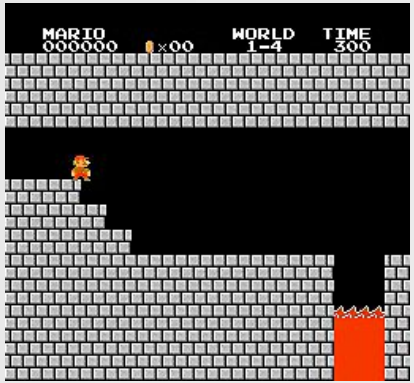
1000 Ep x5



2000 Ep x5

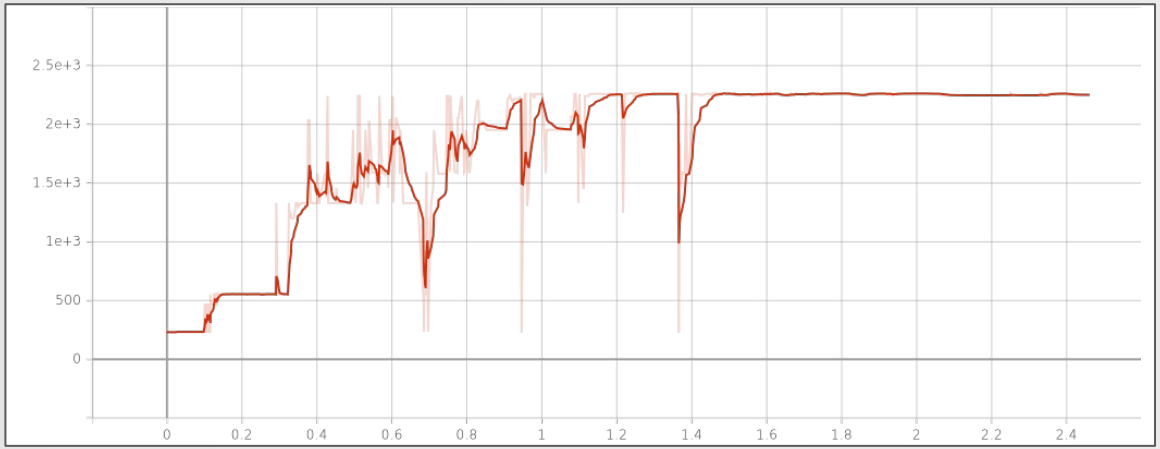


3000 Ep x5



7500 Ep x5

X-Position



Quelle: Wikipedia

06

Resümee

- Super Mario ist für einige RL-Algorithmen eine große Herausforderung
  → ‘The Mario AI Competition (2009-2012)’
  → MarI/O, Intrinsic Curiosity Module, ...

- ‘Do not trust a (learning) robot using GRU’
  → Stabile Lernerfolge sind für manche Probleme Key
  → LSTM > GRU

- ‘We must go deeper’
  → Deep-CNN > CNN

DANKE FÜR IHRE AUFMERKSAMKEIT

Ask me Anything

Repository

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.