Deep Reinforcement Learning for Robotics Using DIANNE
Tim Verbelen, Steven Bohez, Elias De Coninck, Sam Leroux, Pieter Van Molle
Bert VanKeirsbilck, Pieter Simoens, Bart Dhoedt
sam.leroux@ugent.be
How can we build robots that are able to execute complex tasks without programming them explicitly?
Kuka Youbot

- 5 axis arm
  - Length: 66 cm
- Gripper
- Battery operated
- Omnidirectional wheels
  - Max speed: 0.8 m/s
- Embedded PC
Reinforcement learning

Agent

Environment
Reinforcement learning
Reinforcement learning

Agent → Action → Environment
Reinforcement learning

Agent → Reward → Environment
Deep Reinforcement learning

- The actor needs to process high dimensional observations to determine the next action.
- Our favorite processing block: deep neural networks
How can we train without destroying our robot?
V-REP simulator
Multiple simulator instances gathering experience on CPU
Multiple simulator instances gathering experience on **CPU**

**GPU** system training the model
How can we evaluate our models on the robot?
Brain transplantation!
How can we connect the different components?
Dianne

- Modular software framework for designing, training and evaluating neural networks.
- Distributed training and evaluation
- Java based
- Easy integration (service based architecture)
- GUI
- Open source (AGPL 3)
Experience Pool

Deployed agent

Deployed agent
Experience Pool

Deployed agent

Training

Deployed agent

Repository
Deep Reinforcement learning algorithms
DQN

“Playing Atari with Deep Reinforcement Learning” (Mnih et al, 2013)

raw laser scanner measurements (512 values)

Expected future return for each possible action

Q Values
DDPG

Continuous control with Deep Reinforcement Learning (Lillicrap, et al. 2015)

raw laser scanner measurements (512 values)

Actor network

Continuous action

Expected future return

Critic network
Visit dianne.intec.ugent.be for more information
embracing a better life
Abstraction layer with ROS