64-424
Intelligent Robotics

https://tams.informatik.uni-hamburg.de/lectures/2018ws/vorlesung/ir

Marc Bestmann / Michael Görner / Jianwei Zhang

University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Technical Aspects of Multimodal Systems

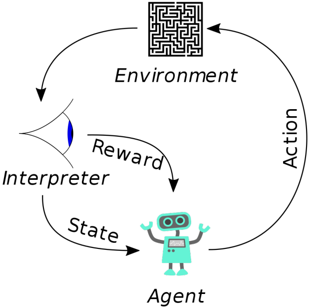
Winterterm 2018/2019

Outline

Outline

Reinforcement Learning

- Currently a hype topic in ML
  - Especially regarding robotics/motion
- Advances due to more investment
  - Most notably Google/OpenAI
- and common environments
  - OpenAI Gym
  - RoboSchool
  - PyBullet
- and baseline implementations
  - Open AI baselines
  - INRIA stable_baslines



https://en.wikipedia.org/wiki/Reinforcement_learning

Vanilla PPO

Video:
https://www.youtube.com/watch?v=hx_bgoTF7bs
Live Demo Roboschool

N. Heess et al, Emergence of Locomotion Behaviours in Rich Environments, 2017

Vanilla PPO - What is Learned

- Action: joint efforts
  - Planar walker: 9 DOF
  - Quadruped: 12 DOF
  - Humanoid: 28 DOF
  - Joints box constrained
- Observation proprioceptive
  - Joint angles and velocities
  - Velocimeter
  - Accelerometer
  - Gyroscope
  - Contact sensors at feet and leg
- Observation exteroceptive
  - Position in relation to center of the track
  - Profile of the terrain ahead

Vanilla PPO - What is Learned (cont.)

- Policy: two networks
  - Only proprioceptive observations
  - Only exteroceptive observations
  - Type of network not clear (probably MLP)
  - Somehow choose action together
- Reward
  - Forward velocity
  - Penalization for torques
  - Stay at center of track

N. Heess et al, Emergence of Locomotion Behaviours in Rich Environments, 2017

Vanilla PPO - How is it Trained

- Mujoco simulation
  - Physical parameters unknown
  - Rate of policy unknown
- PPO - Proximal Policy Optimization
  - Simplified version of TRPO - Trust Region Policy Optimization
  - Current policy is used to choose actions
  - After an episode, advantages of those actions are computed
  - The policy is updated so that good actions become more propable and vice versa
  - Updates are clipped to prevent the policy from leaving the area where it can explore senseful
- Distribution through workers
  - Each worker collects data and computes gradients
  - Batch results are processed by chief
  - New policy is distributed to workers

Deep Mimic

Video:
https://www.youtube.com/watch?v=vppFvq2quQ0

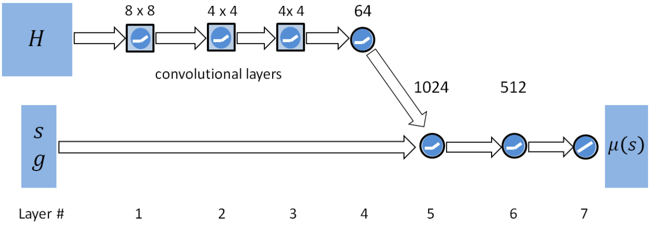
X. Peng et al., DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills, 2018

Deep Mimic - What is Learned

- Action: joint position
  - PD controllers compute effort
- "Observations" (States)
  - Relative pose of links to pelvis
  - Linear and angular velocity of links
  - Phase $\in [0, 1]$
  - Goal g
    - Target heading (walk)
    - Target position (kick, throw)
- Reward
  - $r_t = \omega^I r^I + \omega^G r^G$
  - Closeness to mocap and goal
  - $r^I = w^P r_t^P + w^v r_t^v + w^e r_t^e + w^c r_t^c$
  - Reward based on difference in joint position/velocity, end-effector position, CoM position

Deep Mimic - What is Learned (cont.)

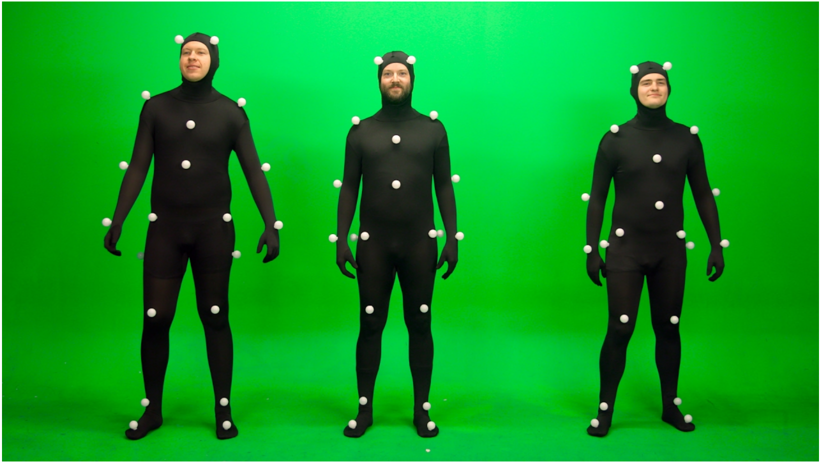
- Policy: single network
  - Input goal and state
  - 2 hidden layer with 1024 and 512 neurons
  - ReLU activation
  - Additional CNN for height map



Deep Mimic - How is it Trained

- Mujoco simulation
  - Physics parameter unknown
  - 30 Hz
- Initial state distribution
  - "Man muss immer umkehren" - Jacobi
  - Easier to learn starting from the back (backplay)
  - Reward clearer when near goal
  - Choosing initial state simple with mocap
- Early termination
  - Terminate episode if condition is reached
  - Classic for walking: head is below certain height
  - Reward for episode is set to zero
  - Further shapes reward function
  - Biases the data distribution to samples which are more favorable

Motion capture - small excursus



Motion capture - small excursus (cont.)

- Different ways to get the data
  - Infra red reflectors
  - LEDs blinking with different frequencies
  - IMUs on all links
  - Magnetic field based (hall sensor)
  - Exoskeleton measuring angles
- Pro
  - Faster learning
  - Less exploits of glitches
  - (Maybe) more useful on actual robot

Motion capture - small excursus (cont.)

- Contra
  - Expensive
  - A lot of work
  - Difficult to get data from animals, e.g. a tiger
  - You look kind of stupid while recording
  - Need to find a student which can do a round house kick
  - Need to bring student into hospital after failed round house kick

There has to be a better way!

Skills from Video

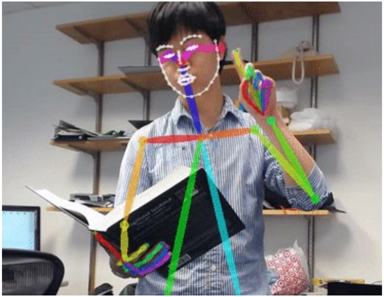
Video:
https://www.youtube.com/watch?v=4Qg5I5vhX7Q

SFV - What is Learned

- Pose estimation
  - 2D and 3D
  - 2 different estimators OpenPose and Human Mesh Recovery
- Motion reconstruction
  - Find optimal motion from single poses
  - Enforce temporal consistency to reduce jitter and glitches
- Learning of motion is similar to DeepMimic, just without goal
- $r = w^P r_t^P + w^v r_t^v + w^e r_t^e + w^c r_t^c$

SFV - How is it Trained

- Pose estimators
  - Supervised learning on single images
- Motion part
  - Similar to DeepMimic



http://www.cs.cmu.edu/ yaser/

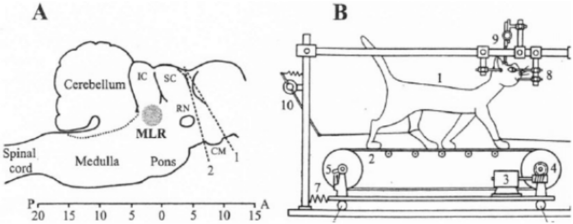
Do We Walk With Our Brain?

- How do humans compute their walking?
- Chickens can run without head
- Legend of Störtebecker

http://www.neurologie.usz.ch/ueber-die-klinik/veranstaltungen/Documents/7_hirnstimulation.pdf

Central Pattern Generators

- Biological neural circuits in the spline
- Generate rhythmic output after being activated
- Used by humans for walking, breathing, swallowing, ...
- Models of this can be implemented for robots



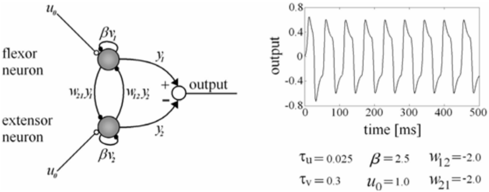
Central Pattern Generator, Mark L. Latash et al., Biomechanics and Motor Control, pp.157-174

Video

Show video

Central Pattern Generators

- Flexor and extensor neuron
- Activated with tonic (non-rhythmic) signal
- Different patterns with different weights



Central Pattern Generators for Gait Generation in Bipedal Robots, Almir Heralic et al., Humanoid Robots, New Developments, 2007

CPG - How is it learned

- It is not!
- Weights are hand crafted
- Learning would be possible either by direct parameter learning or RL

Evolutionary Approach

Video
https://www.youtube.com/watch?v=pgaEE27nsQw

Evolutionary Approach - What is Learned

- Muscle structure of the robot
- Parameters of FSM for leg state
- Target poses
- Force applied in relation to feedback
- Initial pose

Subject	**Parameters**
Muscle physiology	3–30 *
Muscle geometry	12–39 *
State transition	3
Target features	14
Feedback control	14–63 *
Initial character state	6

Geijtenbeek, Thomas, Michiel Van De Panne, and A. Frank Van Der Stappen. "Flexible muscle-based locomotion for bipedal creatures." ACM Transactions on Graphics (TOG) 32.6 (2013): 206.

Evolutionary Approach - How is it Trained

- Evolution approaches in general
  - Generate initial random population of parameter sets
  - Loop
    - Evaluate individuals based on fitness function
    - Pick best
    - Recombination / mutation
- Covariance matrix adaptation evolution strategy (CMA-ES)
  - Pairwise dependency between parameters is represented by covariance matrix
  - This matrix is updated to increase fitness
  - Good for ill-conditioned functions

Simulation Downsides - The Reality Gap

- Difference between simulation and reality
- Wrong models
  - Mass, inertia, size
  - Sensor noise non Gaussian
  - Actuator properties not correct
  - Change over time
  - No static values
- Friction / contact
- Soft bodies
- Environment model not correct
  - Changing lighting conditions
  - Cluttered background
  - Non static background

Simulation vs. Reality (cont.)

- Simulation physics not correct
  - Discrete approximation of continuous system
  - Simplifications due to performance bounds
  - Glitches

Bridging the Reality Gap

- What can we do?
- Improving simulation accuracy (smaller step size)
  - Not enough to bridge reality gap
- Adding sensor noise
  - Noise is not perfectly Gaussian
  - Needs noise model, which can have errors
- Domain randomization
  - Currently the most used approach
  - Simulated variability in training time to make model generalize
  - Implementation depends on the scenario

J. Tobin et al., Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World, 2017

Learning Dexterity

Video
https://www.youtube.com/watch?v=jwSbzNHGflM&t=1s

Learning Dexterity - What is Learned

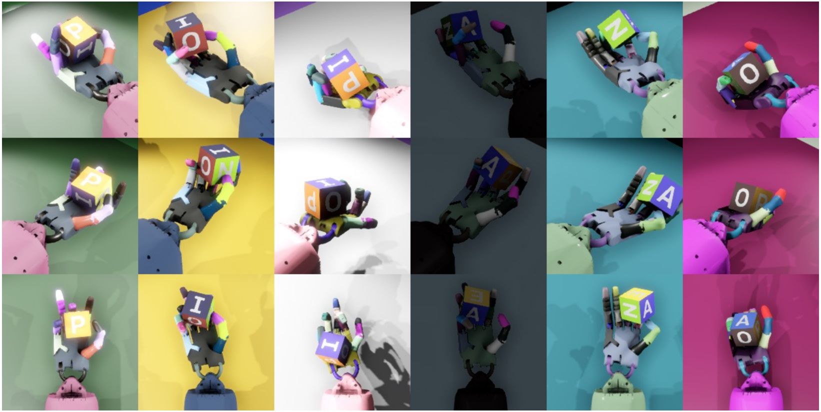
- CNN for object pose detection
  - Based on three camera inputs
- LSTM for finger actions given finger and object pose
- Both networks are concatinated

M. Andrychowicz et al., Learning Dexterous In-Hand Manipulation, 2018

Learning Dexterity - How is it Trained

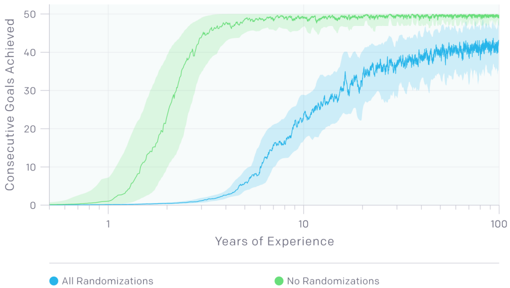
- PPO
- Domain randomization
  - Object dimensions
  - Object and finger masses
  - Surface friction coefficients
  - Robot joint damping coefficients
  - Actuator controller P term (proportional gain)
  - Joint limits
  - Gravity vector
  - Colors in simulation

Learning Dexterity - Domain Randomization



Learning Dexterity - Impact of Domain Randomization

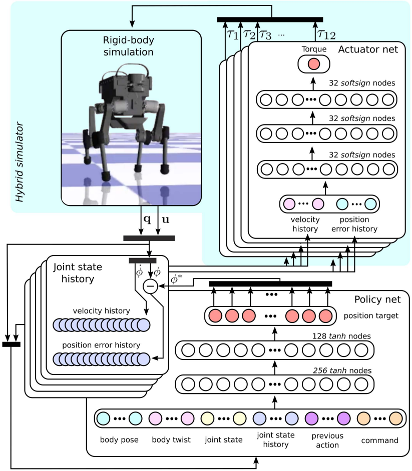
- Median number of successes
  - Without: 0
  - With: 11.5
- Training simulated time
  - Without: 3 years
  - With: 100 years



Learning Dynamic Skills

Video
https://www.youtube.com/watch?v=aTDkYFZFWug

Learning Dynamic Skills - Overview



Learning Dynamic Skills - What is Learned

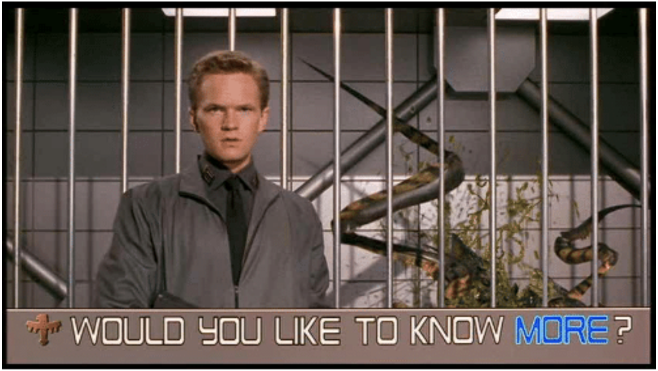
- Policy Network
  - MLP 2 hidden layers 256, 128 nodes
  - Joint angles, velocities
  - Joint state history
  - Body height estimation (filtered forward kinematics)
  - Body pose, twist (IMU)
  - Previous action
  - Command
- Actuator Network
  - MLP 3 hidden layers with 32 nodes
  - Velocity history
  - Position error history

J. Hwangbo et al., Learning agile and dynamic motor skills for legged robots, Science Robotics 2018

Learning Dynamic Skills - How is it Trained

- Randomized simulator model
  - Links masses
  - CoM positions
  - Joint positions
- Policy Network
  - RL with TRPO
- Actuator Network
  - Supervised learning
  - Data collection on robot with simple walk algorithm
  - Joint Position error, Velocity, and Torque

More Information



https://me.me/i/would-you-like-to-know-more-none-905fe16f7eaf4a3b96e292ff789f1e04

More Information

- We only had a quick overview, here are some further information
- ML Foundations
  - Machine learning lecture next semester
  - Arxiv Insights - Youtube channel
  - R. Sutton - Reinforcement Learning, an Introduction (free)
  - Berkeley - Deep RL http://rail.eecs.berkeley.edu/deeprlcourse/
- Current advances
  - Open AI blog - https://blog.openai.com/
  - reddit.com/r/MachineLearning
  - CORL conference (open access)
- If you have some good sources, tell me!

Discussion

What would you teach a robot?

Masterproject
Thesis