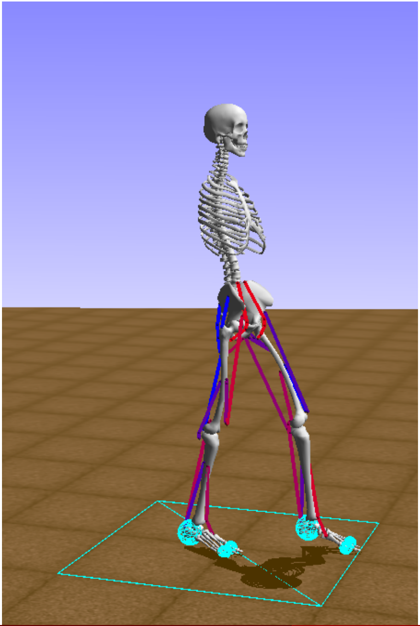
Autoencoder-based Deep Reinforcement Learning for Musculoskeletal models

Massimiliano Falzari

m.falzari@student.rug.nl

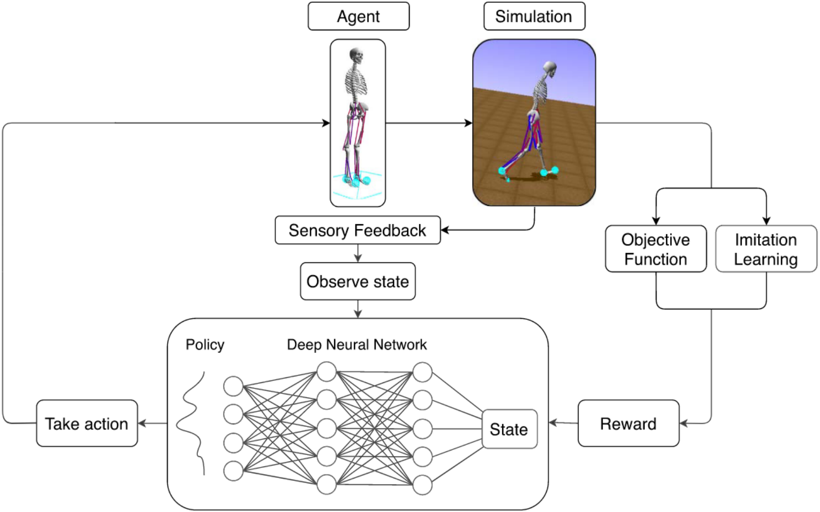
University of Groningen
The Nethelands

June 29, 2022



The simulation

- Opensim
- Healthy model (22 muscles)
- Goal $\rightarrow$ Normal walking


Agent
Simulation
Sensory Feedback
Observe state
Objective Function
Imitation Learning
Policy
Deep Neural Network
State
Reward
Take action

Main Problem

- Poor sample efficiency

Definition

Sample efficiency is the number of times an agent has to interact with the environment in order to learn a good policy/ learn a task.

How to improve it

- Change the RL algorithm (e.g model-based)
- Change the sampling method (e.g. importance sampling)
- Design better reward function
- **Learning good latent representation**

Main Problem

- Poor sample efficiency

Definition

Sample efficiency is the number of times an agent has to interact with the environment in order to learn a good policy/ learn a task.

How to improve it

- Change the RL algorithm (e.g model-based)
- Change the sampling method (e.g. importance sampling)
- Design better reward function
- **Learning good latent representation**

Background

Main Problem

- Poor sample efficiency

Definition

Sample efficiency is the number of times an agent has to interact with the environment in order to learn a good policy/ learn a task.

How to improve it

- Change the RL algorithm (e.g model-based)
- Change the sampling method (e.g. importance sampling)
- Design better reward function
- **Learning good latent representation**

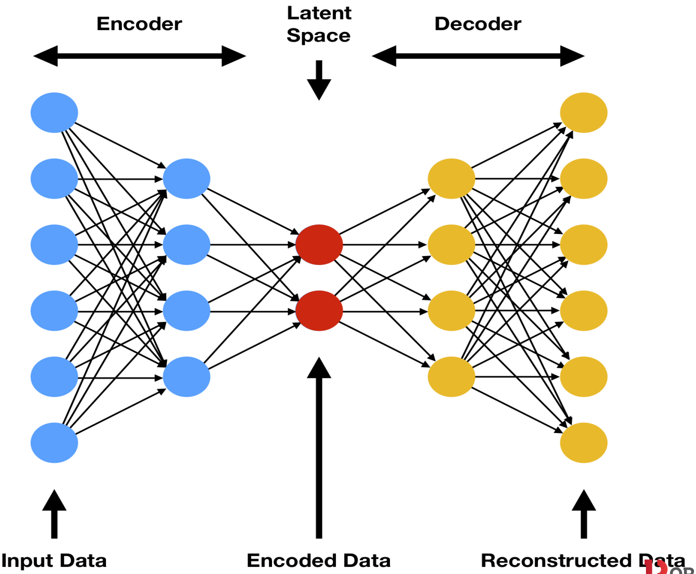
Standard Methods

- PCA
- LDA
- LLE
- IsoMap

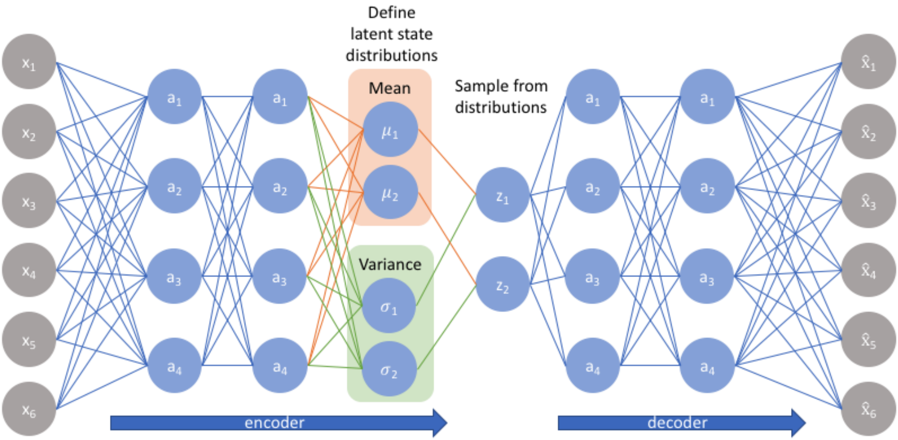
Neural Methods

- Autoencoders

Autoencoder



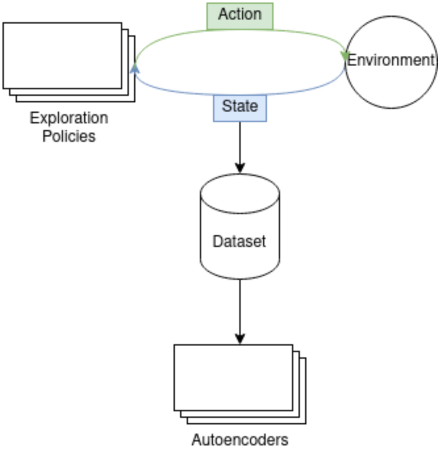
Variational Autoencoder



Learn to see

Learn a latent state representation using Autoencoders. The observations used during training will be collected using 2 different exploration policies:

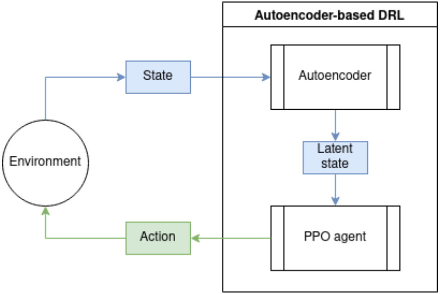
- random policy
- RL agent



Methods

Learn to act

PPO based agent will be trained in the environment to learn a walking policy. The trained autoencoder will encode the observations into a latent state.



Results/Expectation

university of groningen

Main Challenges

- Collect a good distribution of data over the environment
- Select the most appropriate number of latent dimension

Expectation

- Better performance in term of reward
- Higher sample efficiency (number of iterations needed to achieve a good performance)
- Lower computational resources required

ROBOTICSLAB

university of groningen

Example

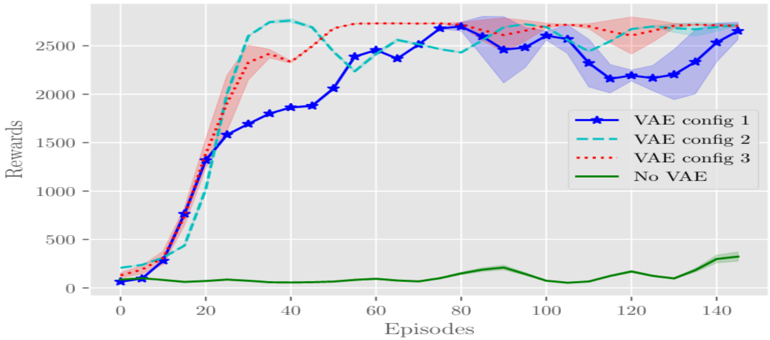


Figure: On the use of Deep Autoencoders for Efficient Embedded Reinforcement Learning

ROBOTICSLAB

Thanks for your attention

for more info about the project visit
`https://172.104.159.41/thesis/summary.html`