Advanced Robot Control and Learning

Prof. Sami Haddadin

Part 2

Machine Learning in Robotics

Chapter 1: Overview of Machine Learning

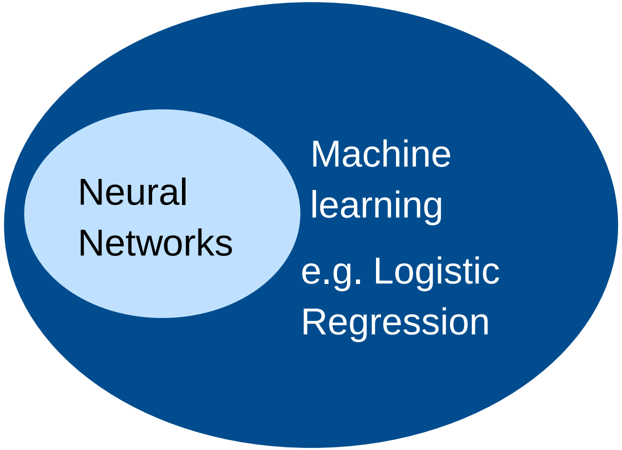
What is Machine Learning?

Machine learning (ML) is a set of computer algorithms that improves through ‘experience’ from data. All types of ML algorithms are trying to approximate some unknown underlying function from available observations.

The most prominent type of ML algorithms is Neural Network.

Other types of ML algorithms include:

- Linear regression
- Support Vector Machines
- Decision trees
- Random Forests

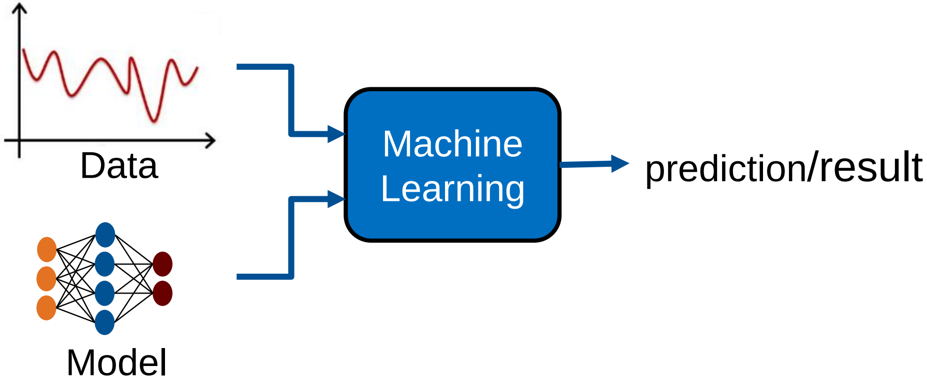


Neural Networks

A Neural Network (NN) is a connected graph, i.e. a collection of connected units called artificial neurons. The connections, also called edges, transmit a signal to other neurons. Each edge has a weight associated with it that adjusts the strength of the transmitted signal.

The arrangement of the neurons is called topology of the network. The prototypical examples are layers of neurons.

Neural networks are universal function approximators, i.e. any continuous function can be approximated by a neural network (c.f. Universal Approximation Theorem).



What Neural Networks can do?

For example:

- Approximate (arbitrarily complex) functions
- Extract ‘most important’ features from data (in the sense of principal component analysis)
- Find optimal parameters for a specific problem

Some concrete applications:

- Face recognition in pictures, $F\colon \text{image} \mapsto \begin{cases} \text{true} & \text{if face is present} \\ \text{false} & \text{otherwise} \end{cases}$
- Parameter tuning for an impedance controller
- Speech recognition
- Spam-filter in email clients

What Neural Networks cannot do?

Theoretical limitations:

- Although neural networks are **universal function approximators**, the provided **proofs are not constructive** regarding the number of neurons, weights or topology.
- **No Free Lunch theorems**: These theorems (shown 1996 by David Wolpert) state that for every pattern a learner is good at learning, there is another pattern the same learner is very bad at.

Practical limitations:

- **Data preparation:** input data needs to be pre-processed to fit the requirements of the used algorithm.
- **Model assumption:** many algorithms need some assumption about the data they are given, e.g. convolutional neural networks for images.
- **Parameters:** the parameters (weights) are learned during training which is an optimization problem. The performance relies purely on the learned parameters.

3 Categories

Machine Learning can be divided into three main categories of problems

- **Supervised Learning:** is the task of learning a function that maps an input to an output based on example input-output pairs.

- **Unsupervised Learning:** looks for previously undetected patterns in a data set with no pre-existing labels and with a minimum of human supervision.

- **Reinforcement Learning:** is concerned with how agents ought to take actions in an environment in order to maximize the notion of cumulative reward.

Supervised Learning

Supervised learning is a direct approximation method

- **the data** consists of labeled example pairs (x_i, y_i),
- **the task** is to find a function f, such that $f(x_i) = y_i$ and that it extrapolates well to unseen data.

(E.g. face recognition.)

It further subdivides to

- Classification: where y_i are elements of a discrete space (e.g. {true, false}, {cat, dog, elephant}).
- *Regression:* where y_i are elements of a continuous space (e.g. price, age, $\mathbb{R}$).

Unsupervised Learning

Unsupervised learning tries to extract properties from given data

- **the data** is any collection of elements
- **the task** is to find a function f and possibly it's inverse f^{-1}, such that $f(x) = x'$ and $f^{-1}(x') \approx x$.

Some examples are:

- *Data compression* – autoencoders (in the sense of principal component analysis)
- *Clustering* – grouping data into 'similar' groups

a) *Anomaly detection* – identification of elements which differ significantly from the majority of the data

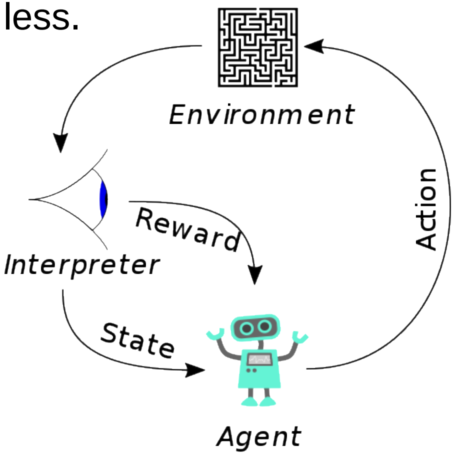
Reinforcement Learning

In reinforcement learning the world is split into an agent and an environment. The time is discretized and at each time step the agent takes an action in the environment. The environment is interpreted into a reward and a representation of the state, that are fed back to the agent. The goal of the agent is to maximize the cumulative discounted reward, discounted means that the future rewards matter less.

- **the data** consists of state and reward pairs (s_t, r_t),
- **the task** is to find a function $\pi: s_t \mapsto a_t$, such that $\sum_t (\gamma)^t r_t(s_t, a_t)$ is maximal.

  Note that the reward $r_t(s_t, a_t)$ is dependent on the state and action taken.

  The function $\pi: s_t \mapsto a_t$ is referred to as policy.

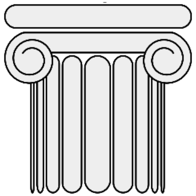


Some examples are:

- Video game bots (state is current frame, reward is game score)
- Power-consumption optimization

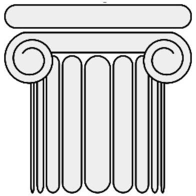
Machine Learning Landscape

Supervised Learning



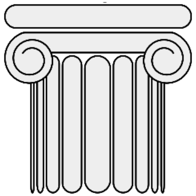
Logistic Regression,
Linear Regression,
Support Vector Machines (SVMs),
Decision Trees

Unsupervised Learning



Autoencoders,
Principal Component Analysis (PCA),
Generative Adversarial Networks (GANs),
k-Means

Reinforcement Learning



REINFORCE,
Q-Learning,
Actor-Critic,
TRPO

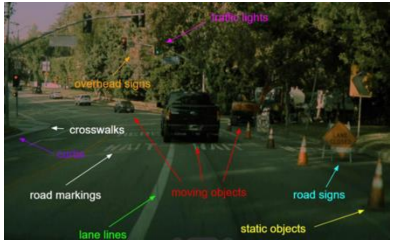
Which of these disciplines are important for Robot Learning?

Supervised Learning in Robotics (Computer Vision)

Computer Vision is concerned with how computers can gain understanding from digital images or videos

Autonomous Driving (Tesla)

- 8 cameras around the car
- NN uses HydraNets architecture with shared backbone
- NN estimates 50+ tasks simultaneously on a small computer

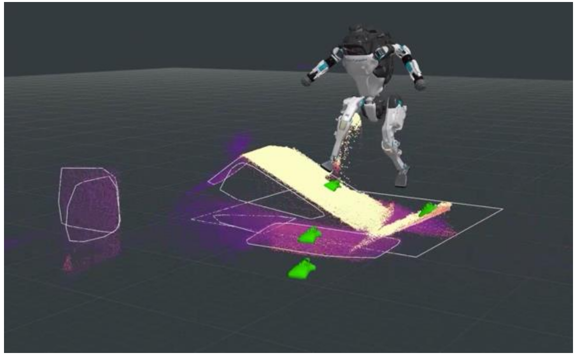


Humanoid Locomotion (Boston Dynamics)

- Hydraulically powered
- Uses model predictive control (MPC)
- Leverages precomputation (trajectory libraries)
- Uses geometric segmentation for perception (local planar regions and geometric primitives)

space exploration (NASA)

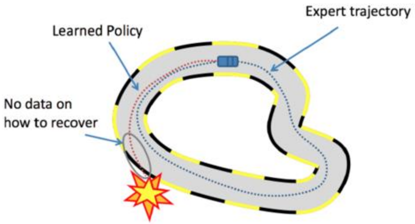
- Uses stereo camera for vision



Reinforcement Learning in Robotics

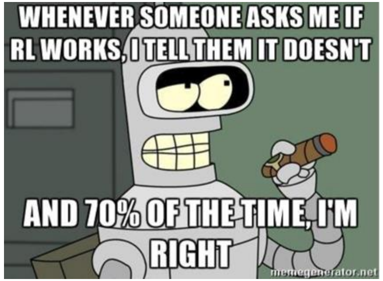
Imitation Learning

- Tries to mimic an expert behaviour from demonstrations
- Has advantages in situations when designing reward function is difficult (e.g. robot grasping)
- The simplest algorithm is as follows:
    - collect trajectories, i.e. state-action pairs (s_t, a_t)
    - train policy π with loss function $\mathrm{L}(a_t, \pi(s_t))$
- Used for driving vehicles (ALVINN algorithm)
- Shortcoming is that the collected state-action pairs are not i.i.d. (independent and identically distributed)



Reinforcement learning

- Used in complex problems without obvious or easily programmable solution (game playing - AplhaGo, control problem – elevator scheduling)
- Uses reward function to indicate if task was accomplished
- Example: Learning Policy Improvements with Path Integrals
    - Learned optimal performance of 10-DoF robot in 2D
- RL algorithms work well in simulated environments, but are not yet good in real world applications.
- Shortcomings are that curse of dimensionality, long training time, instability, and many others



Machine Learning in Robotics

Additional examples of Machine Learning in Robotics:

- Recognizing objects and their shapes for grasping (supervised learning)
- Detecting faces and interpret the mimic (supervised learning)
- Imitation Learning: a specific skill is demonstrated and the robot uses this information to reproduce it (supervised learning)
- Learning optimal trajectories (supervised or reinforcement learning)
- Learning optimal controller parameters for a particular problem (supervised learning)

Following are selected research videos

1. Learning Agile and Dynamic Motor Skills for Legged Robots, Robotic Systems Lab, ETH Zurich, 2019
2. RMPflow - A Computational Graph for Automatic Motion Policy Generation, NVIDIA, 2019
3. TossingBot: Learning to Throw Arbitrary Objects with Residual Physics, Princeton, 2019
4. Transferring End-to-End Visuomotor Control from Simulation to Real World for a Multi-Stage Task, Imperial College London, 2017
5. Guided Uncertainty-Aware Policy Optimization, UC Berkeley & Stanford, 2019
6. Solving Rubik's Cube with a Robot Hand, OpenAI, 2019

   "For the Rubik's cube task, we use 8 × 8 = 64 NVIDIA V100 GPUs and 8 × 115 = 920 worker machines with 32 CPU cores each. … The cumulative amount of experience ... is roughly 13 thousand years."


Control PC
State
measurement

Training GUAPO is sample efficient

Start of training

...15 minutes later