Intelligence scores goals: Machine Learning for autonomous robots

Martin Riedmiller

Brainstomers: 'Our longterm goal is to build a soccer team where the decision making part is based completely on Reinforcement Learning (RL) methods'

(RC proceedings, 2000)

The promise of Reinforcement Learning (RL)

- RL: Learn by success and failure (Sutton, Barto, Watkins, 1983, 1989, Bertsekas, 1996),
- Neuro Dynamic Programming

- task: $\min_\pi J^\pi(s) = \min_\pi E(\sum_t^\infty \gamma^t c(s_t, \pi(s_t)) | s_0 = s)$

- Solve by: $J^{k+1}(s) \leftarrow \min_a \{c(s,a) + \gamma\, J^k(s')\}$

- Our approach: keep it simple and robust and efficient:
  - standardisation of c(.),
  - as few parameters as possible (gamma, lambda, epsilon)

- Brainstormers: a reinforcement learning way to play robotic soccer
- Towards Artificial General Control Ingelligence (AGCI)

1998: Paris

- Foundation of Brainstormer's team: soccer simulation 2D
  - Background: PhD thesis on neural reinforcement learning controllers

First matches: 0:19, 0:26, 0:23...

High expectations, deep fall: it's not so easy as it looks...

1999 Stockholm

- Software redesign: soccer as a multi-agent MDP
- Clean agent architecture:
  - Perception: from sensor to state estimation (Markov!)
  - Control: from state to actions in a modular architecture
  - Clear and simple, synchronous mainloop

2000 Melbourne

- First skill learned by RL with huge effect on competition team: Neuro-Kick
- 2nd place European championship in Amsterdam, 2nd place in Melbourne (behind FC Portugal)

Neuro Skill: RL Kicking

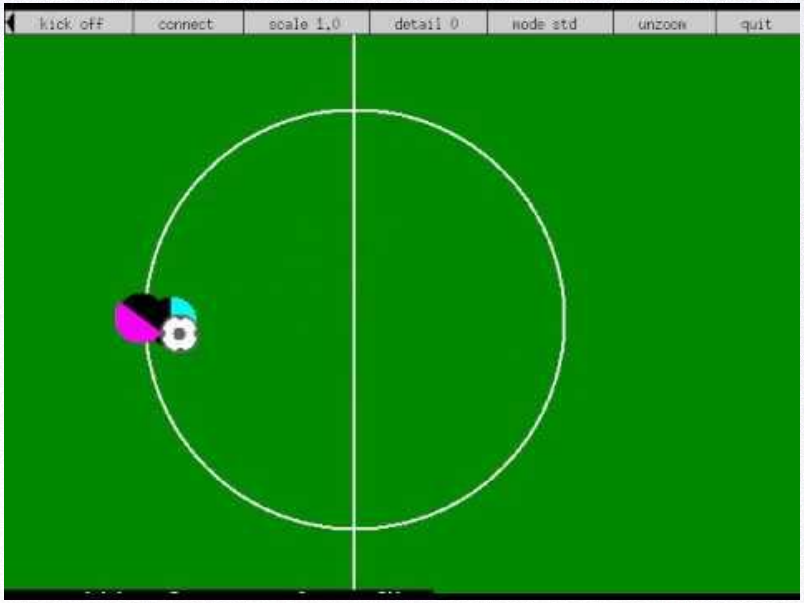
- Kicking is a multi-step controller
- NN represents value function, Learning rule adapted to NNs

$$\partial(y - (\min_a c(s,a) + \gamma J(f(s,a)))^2 / \partial w$$

- Success: desired speed in desired direction, failure: ball lost. Reduce time.
- Model-based RL
- 72 NNs (one for each target direction)
- Typical 2 hidden, 20 neurons each

Neuro Skill: RL Dribbling

- Q-learning, since opponent behaviour unknown
- Partial model encodes action (generalisation)



2001 Seattle

- Learned more RL skills: intercept, improved kick, …
- Learned a multi-agent attack for 7 vs 8 players by RL
- … another 2nd place (behind Tsinghuas)

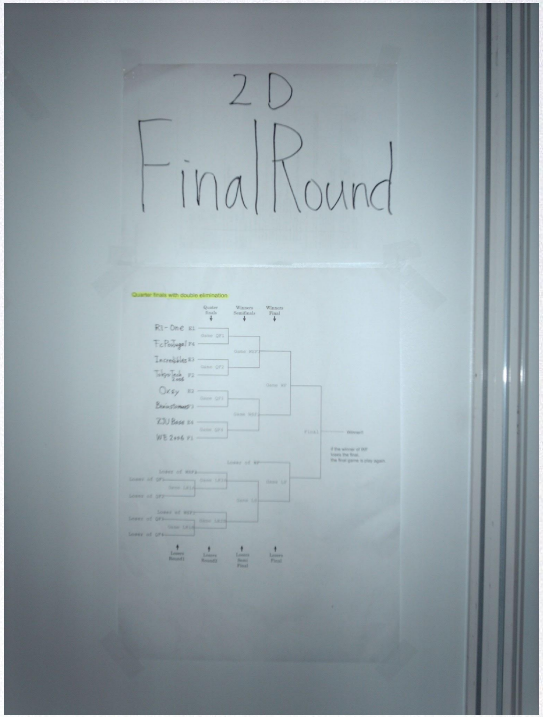
MAS RL: 2vs2

- Success: goal, failure: ball lost, minimum time
- Temporal abstraction (e.g. pass, dribble, moving to positions)
- Uses an abstract model to predict next situation
- Neural network learns a shared valuefunction (>30 state dim)
- Combination with planning
- Decides where to pass or where to run

2002-2008: Fukuoka to Suzhou

Co-Team Lead from 2005: Thomas Gabel

- 2000-2004: 2nd, 2nd, 3rd, 3rd, 2nd -> ‘eternal best looser’
- 2005: Worldchampion! Finally!
- 2006: Lost final in Bremen against Wright Eagle
- 2007: Trained neuro-RL defense behaviour against Wright Eagle: Worldchampion again (Gabel et al, 2007)
- 2008: Worldchampion!



2003 - 2008: MiddleSize League

Co-Team Leads: Martin Lauer, Sascha Lange, Roland Hafner, Artur Merke

- 2003: learned low level controllers
- 2004: learned intercept (simulator)
- 2006: NeuroDribble - better than any human coded Tribots behaviour before
- Worldchampion 2006 and 2007.
- Others: first game against humans (Atlanta 2007), first software transfer to different robot in one night (Suzhou 2008), open source platform (with HARTING, 2007-10)

Learning to Dribble: Neuro RL

- Neural Fitted Q iteration (Riedmiller, 2005)
  - Full-batch update over all transitions (s,a,s')
  - RL: series of supervised problems
  - Rprop as weight update algorithm
  - 'Data-efficient RL' (1 mio trials -> 300)
- 'Growing batch'
- Learned directly on real robot < 1hour interaction
- Applied in competition team from 2007

Overview of learned behaviours

	'00	'01	'02	'03	'04	'05	'06	'07	'08
Simulation League									
NeuroKick	●	●	●	●	●	●	●	●	●
NeuroIntercept	●	●	●	●		○	○		
NeuroGo2Pos	●	●	●	●	●				
NeuroADB								●	●
NeuroAttack	○	●	●	●	●		●	●	●
NeuroPenalty				●	●	●	●	●	●
Rank	2	2	3	3	2	1	2	1	1
MidSize League									
NeuroMotorSpeed								○	○
NeuroGo2Pos							○	○	○
LmapIntercept							●	●	●
NeuroDribble								●	●
Rank							1	1	3

Inspirations from RoboCup

‘Set up learning system - make holidays for 6 months - come back, take agents, become worldchampion’ (Brainstormers, 2001)

- Did we fulfill our promise? How far can we push learning? Can we get rid of programming at the end?
- Is our architecture right, in particular splitting perception from control?
  - Are the representations that we compute always optimal for the actual purpose? E.g. Middle size league: do we actually need to know the position of the ball in centimeters to decide to go there? Wouldn’t a task specific representation serve the purpose of control much better?

Control team@DeepMind : solve AGCI

Artificial general control intelligence (AGCI): Learn to control when only target is given

From scratch, from raw sensors, with raw actions

The RACE route to AGCI

- Robustness & Reliability
  - Reliable control and learning performance independently of initial conditions
- Autonomy
  - No prior task knowledge required to set up agents (features, reward, actions)
- Complexity
  - Applicable to complex tasks (DOFs, horizon, nonlinear dynamics)
- Efficency
  - Applicable to real world systems

This talk: focus on autonomy

Increasing Autonomy:

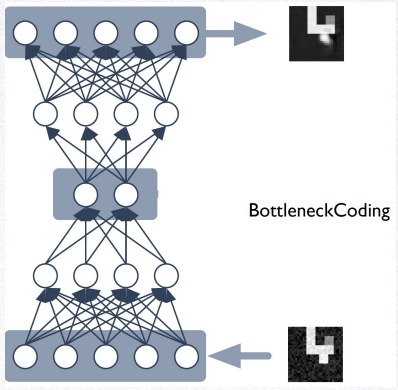
How to get from raw sensors to meaningful representations?

Or: resolving the split between perception and action.

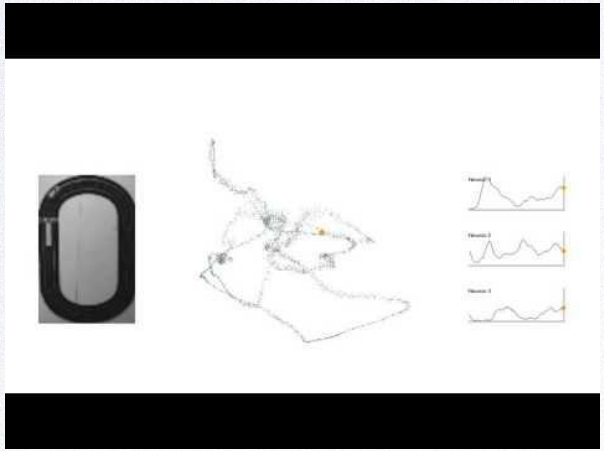
Deep Fitted Q (DFQ)

Idea: use convolutional autoencoders to learn low-dimensional representations from raw pixels

- Deep Fitted Q iteration (DFQ, Lange & Riedmiller, ESANN 2010, Riedmiller, Voigtländer and Lange, IJCNN 2012)
- approach: layer-wise training of autoencoders using deep convolutional networks
- Data-efficient
- Robustness, Autonomy: control might fail if representation is not good enough



Example: Slot Car Racing from raw pixels



- One of the first deep RL systems, fully trained in real world, from scratch
- Convolutional, 19 layers, >1 Mio weights
- Data-efficient: 7,000 images for pretraining, less than 30 min of training on real track

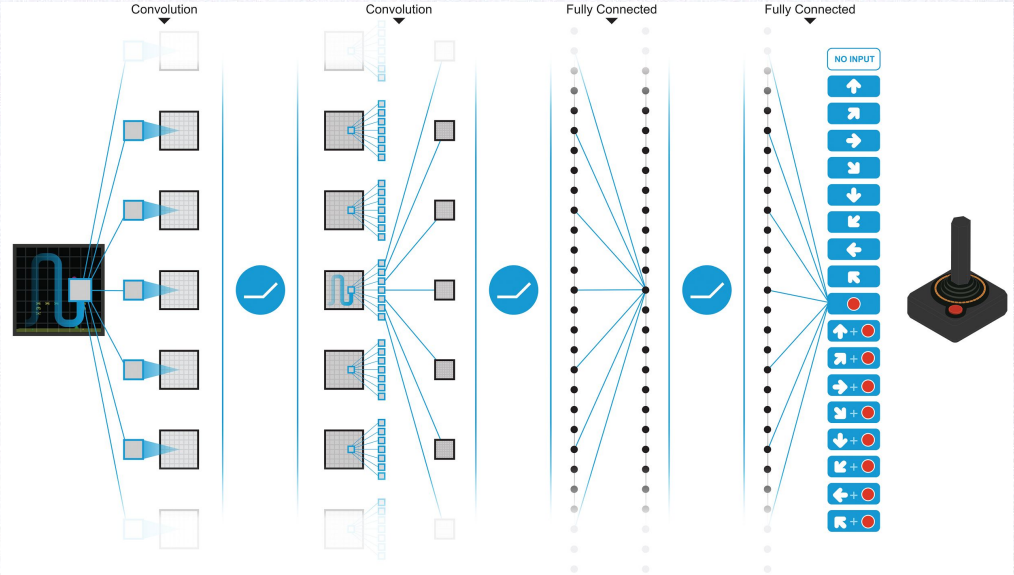
Learning task specific representations

Increasing Autonomy: learn representation from pixels to Q-function (or policy) in one network end-to-end

- DQN (Mnih et al, Nature 2015) learned to play Atari games from pixels
- DDPG (Lillicrap et. al, arXiv, 2015): extends DQN by using Q-gradients to learn policy (cf NFQCA, Hafner & Riedmiller, 2011, DPG, Silver et.al, ICML 2014, Hausknecht and Stone, ICLR 2016)
- Using mini-batch updates allows to deal with large amounts of training data
- Typically millions of transitions needed for proper training: data-efficiency suffers
- Recent developments: Async RL -> reduce wall-clock time drastically

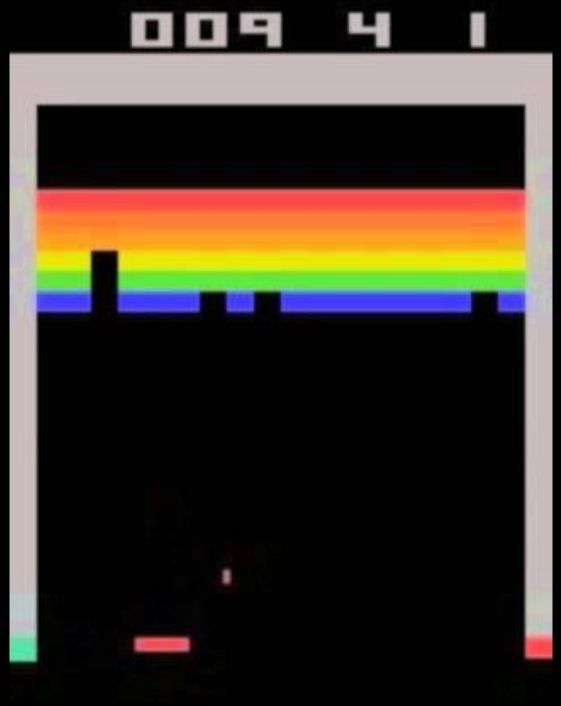
DQN

- End-to-end: from pixels to actions
- Trained only on reward (score)
- One agent for 50 games
- Super-human level on most games



Space Invaders

Breakout



General Atari Player

DDPG example

- Actor-critic: uses dQ/da to update critic (cf NFQCA)
- Mini-batches to cope with large transition sets
- Able to learn directly from pixels
- Data-efficency not a focus

(Improved) NFQ efficiency, scaling: Cheetah

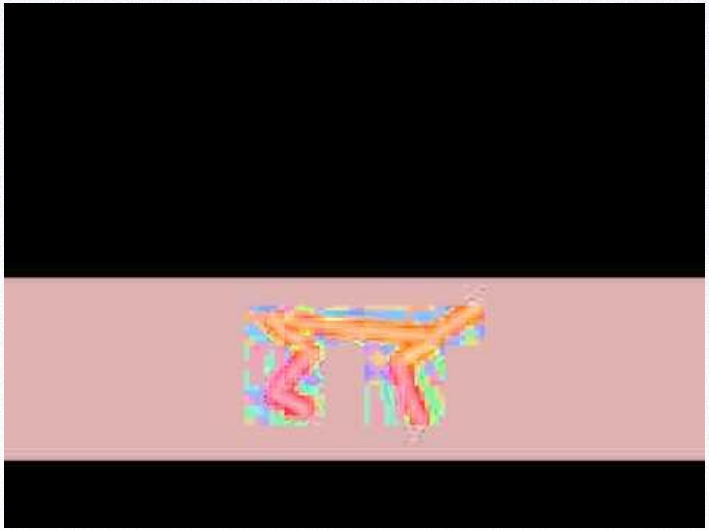
Cheetah: 6 joints, springs

Input: 18 dims

Actions: 6 dims

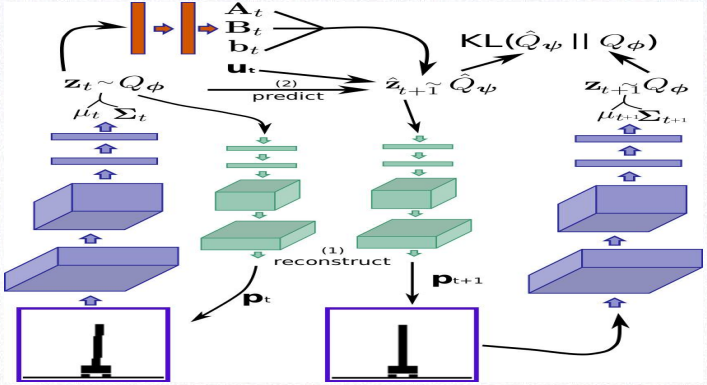
Results: 400 episodes, 5s each

Reward: maximizing forward velocity, discounting



What's next?

- Combining autonomy, efficiency, robustness
  - Idea: double exploit of transitions:
    - learn (latent) state space model to predict next state observations. Regularisation helps to find useful representations, e.g. Embed to Control E2C (Watter et. al, NIPS 2015, Assael et. al, arXiv, 2015)
    - Reuse transitions to learn control
- Attacking complex systems: AlphaGo (Silver et. al, 2016)



Conclusions

- Machine Learning and AG(C)I:
  - Huge progress in recent years: images, speech, DQN for Atari, AlphaGo
  - Important aspect is the generality of agents

- RoboCup Lessons:
  - Competition (and cooperation) is great for advancing a field at speed
  - Fascinating balance between engineering and science to progress state of the art
  - Made lots of friends for life ;-)

THANK YOU

Credits

Thanks to my Brainstormers, my colleagues at DeepMind and at Freiburg University: Thomas Gabel, Roland Hafner, Sascha Lange, Martin Lauer, Arthur Merke (and many students), Tobias Springenberg, Joschka Bödecker, Jan Wülfing, Manuel Watter, Manuel Blum, Thomas Lampe, Arne Voigtländer, Yuval Tassa, Tom Erez, Tim Lillicrap, David Silver, Nicolas Heess

Additional Credits