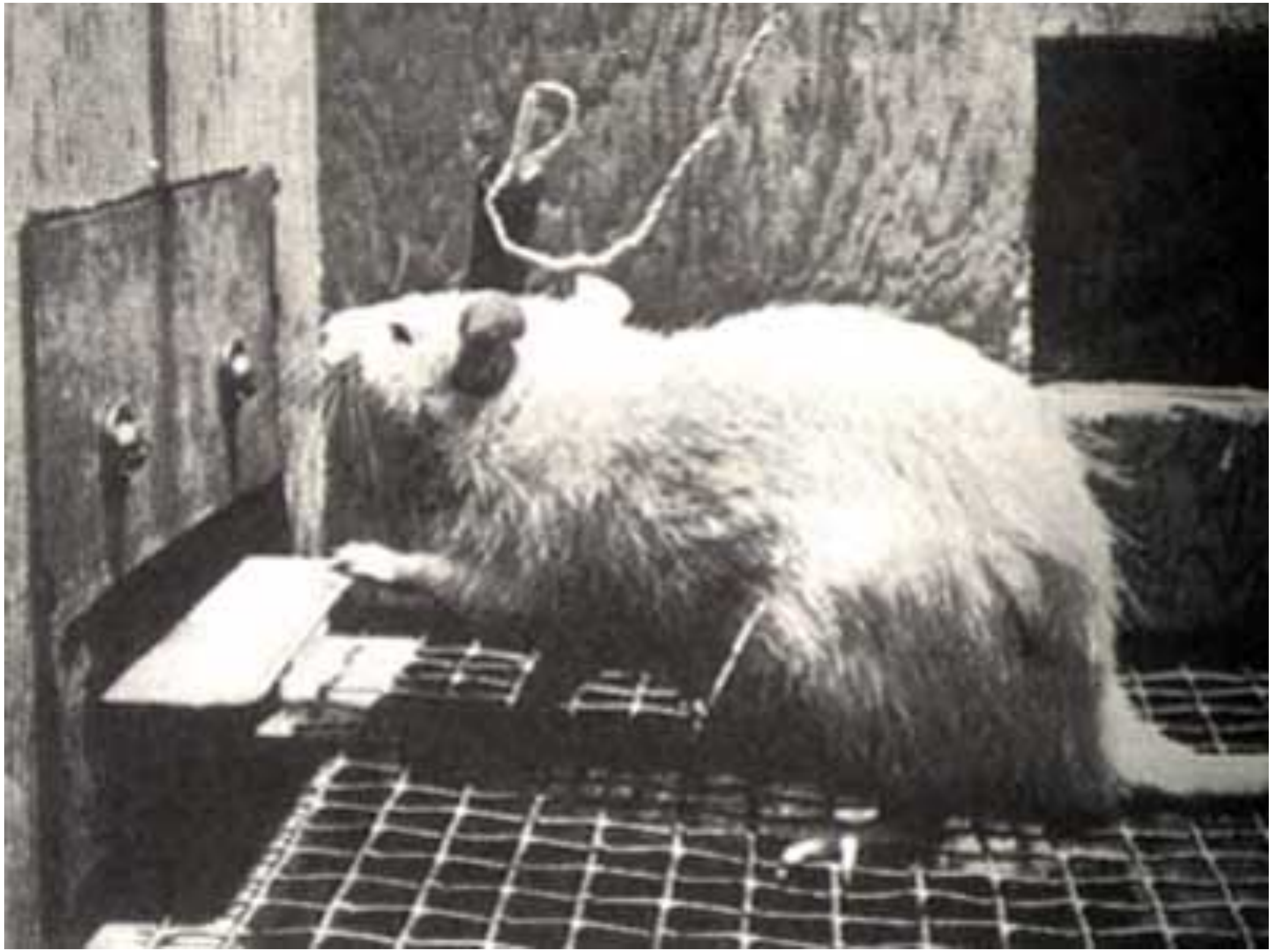
Reinforcement Learning



Slides L. Lazebnik and others

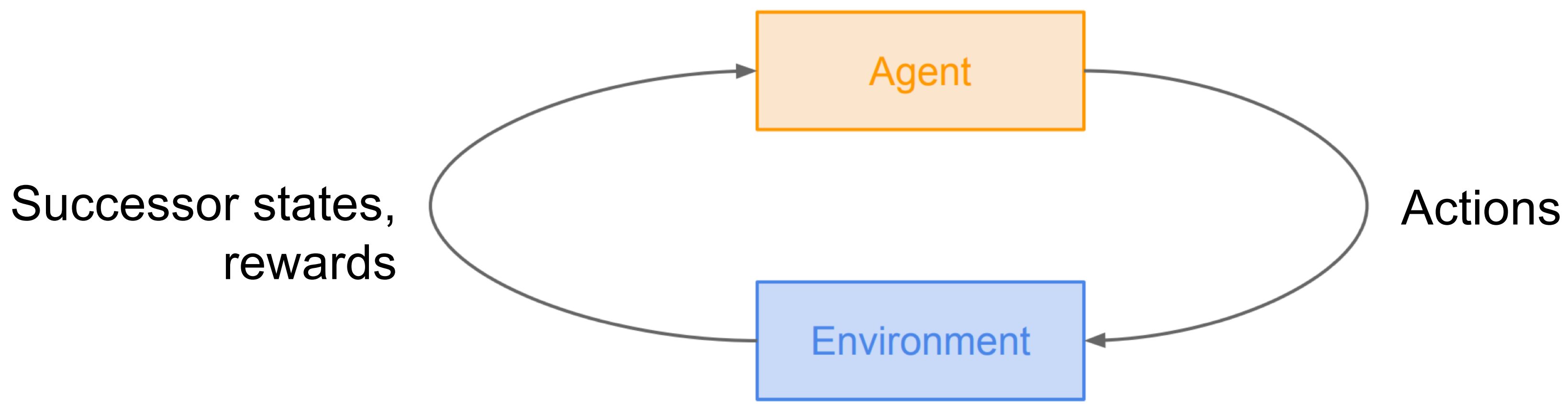
Previously

Supervised learning (classification, regression)

Unsupervised learning (e.g. metric learning)

Reinforcement learning (RL)

- Agent can take *actions* that affect the *state* of the environment and observe occasional *rewards* that depend on the state
- The goal is to learn a *policy* (mapping from states to actions) to maximize expected reward over time



RL vs. supervised learning

- Reinforcement learning loop
  - From state s, take action a determined by *policy* $\pi(s)$
  - Environment selects next state s' based on *transition model* $P(s'|s,a)$
  - Observe s' and reward $r(s')$, update policy

- Supervised learning loop
  - Get input x_i sampled i.i.d. from data distribution
  - Use model with parameters w to predict output y
  - Observe target output y_i and loss $l(w, x_i, y_i)$
  - Update w to reduce loss: $w \leftarrow w - \eta \nabla l(w, x_i, y_i)$

RL vs. supervised learning

- Reinforcement learning
  - Agent’s actions affect the environment and help to determine next observation
  - Rewards may be sparse
  - Rewards are not differentiable w.r.t. model parameters

- Supervised learning
  - Next input does not depend on previous inputs or agent’s predictions
  - There is a supervision signal at every step
  - Loss is differentiable w.r.t. model parameters

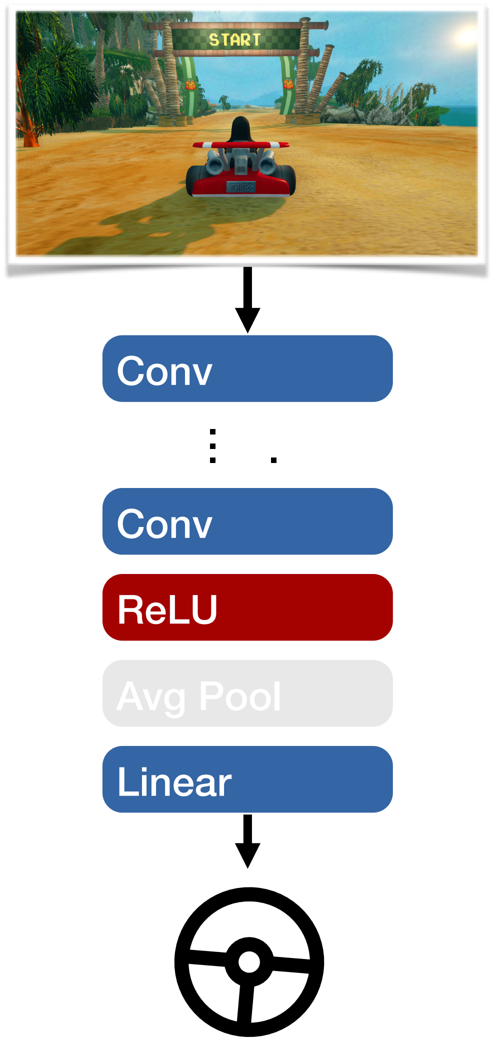
Deep learning for action

Input

- Observation

Output

- Action



How to backpropagate ?

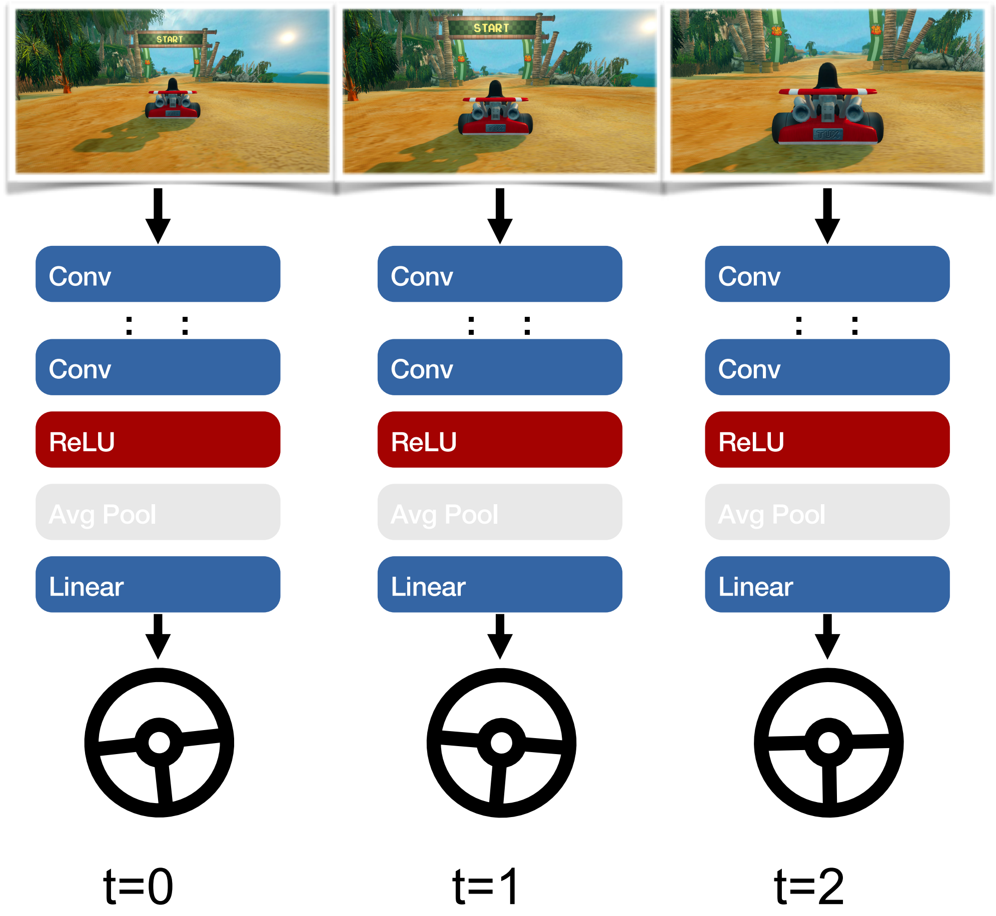
What is the loss ?

How to train such networks ?

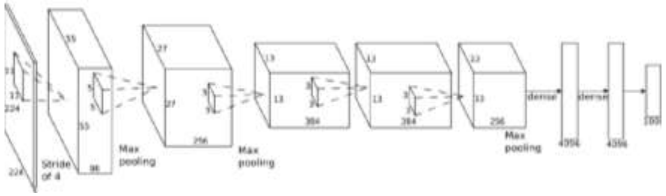
Acting in an environment

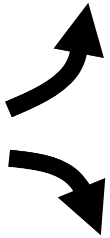
Action changes that state the of the world

- Non-differentiable
- Often non-repeatable
- Long-range dependencies
- Time matters



Learn policies





o_t $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ a_t

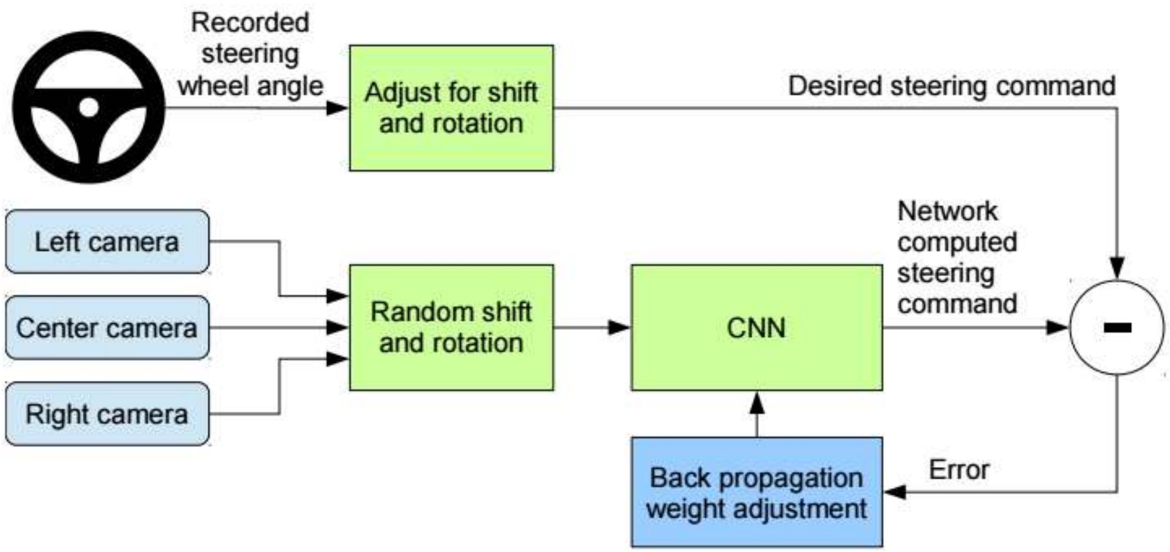
Supervised learning paradigm training data o_t a_t

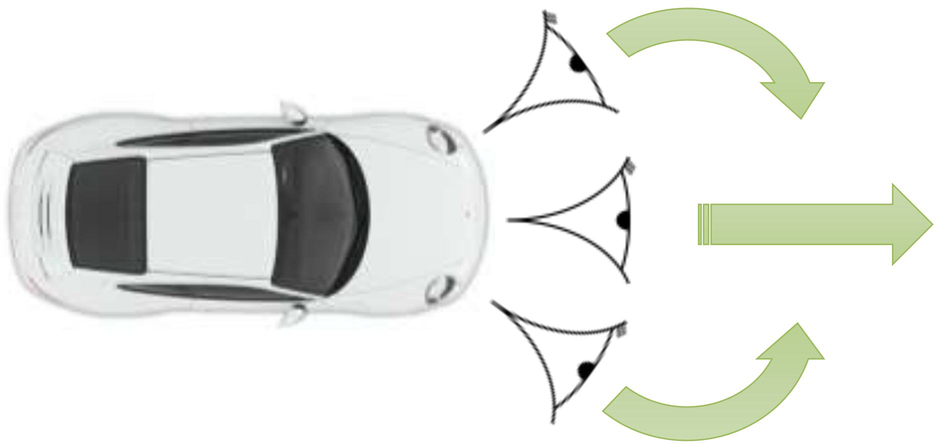
Learn the policy $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

Does is work ?

Supervised Learning – Imitation Learning

Bojarski '16 NVIDIA. End to End Learning for Self-Driving Cars





Dagger

- How to handle distribution shift
- Gather more training data using initial policy
-

DAgger: **D**ataset **A**ggregation

goal: collect training data from $p_{\pi_\theta}(\mathbf{o}_t)$ instead of $p_{\text{data}}(\mathbf{o}_t)$

how? just run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

but need labels $\mathbf{a}_t$!

1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \dots, \mathbf{o}_N, \mathbf{a}_N\}$
2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \dots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{a}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

Problems

Non-markovian

Multi-model behavior

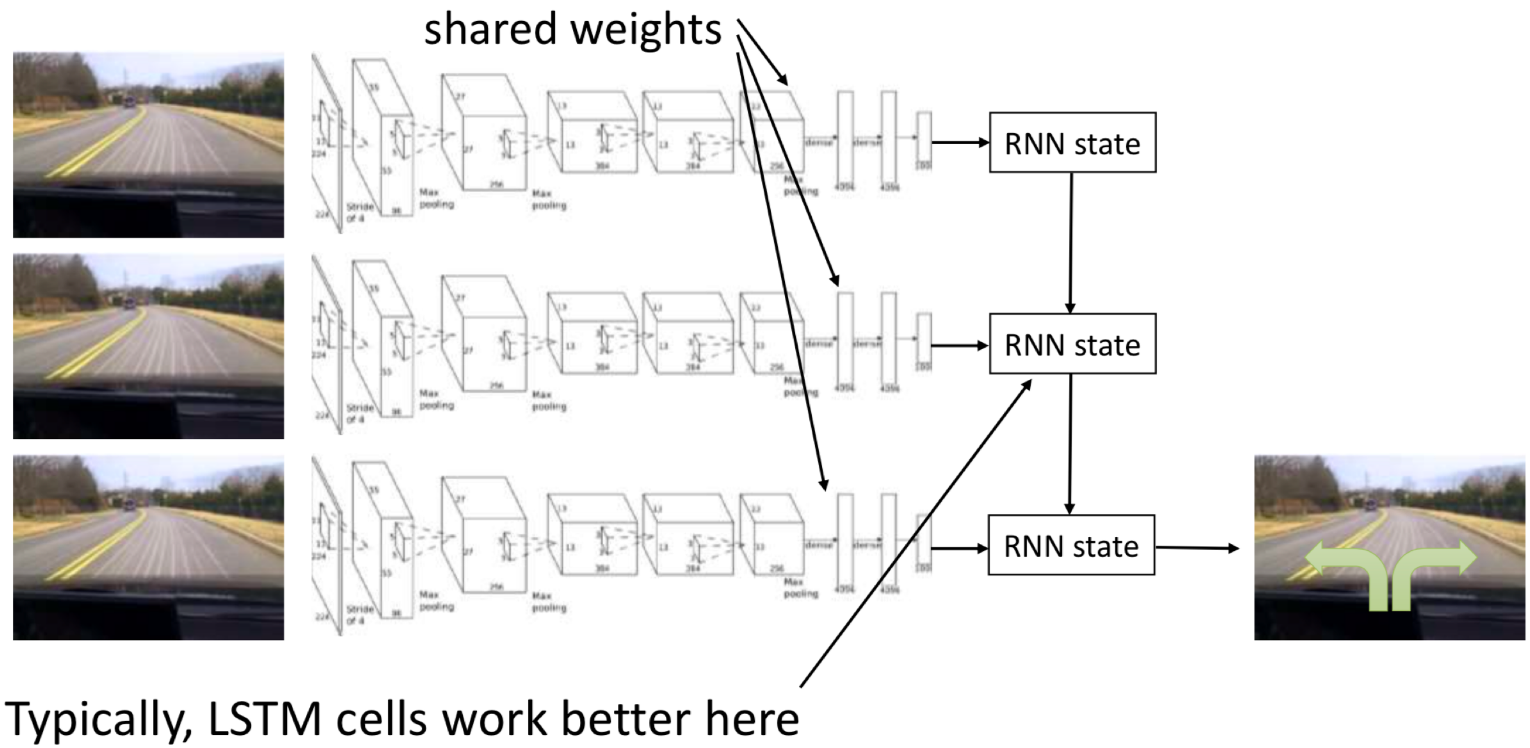
Output mixtures of Guassians

Implicit density models

Latent variable models (how to use noise effectively)

Auto-regressive discretization

How to use the history



Case studies

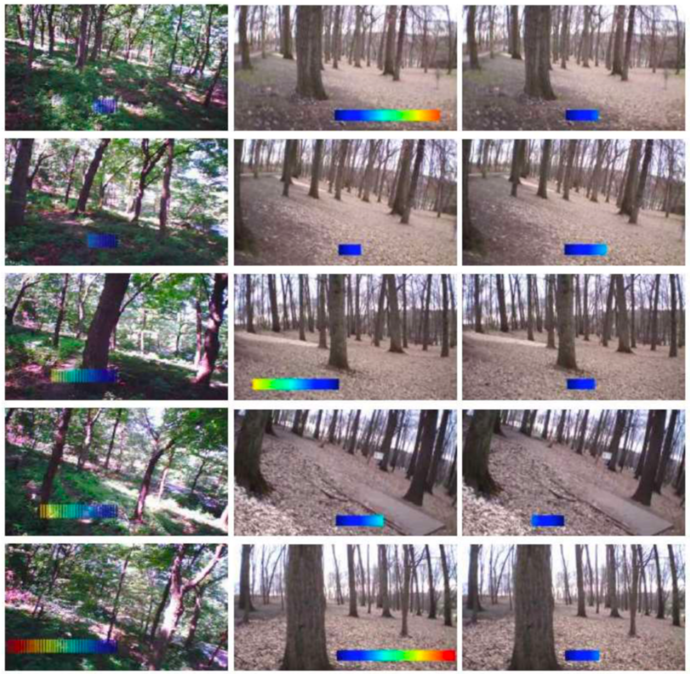
- Trail following as classification

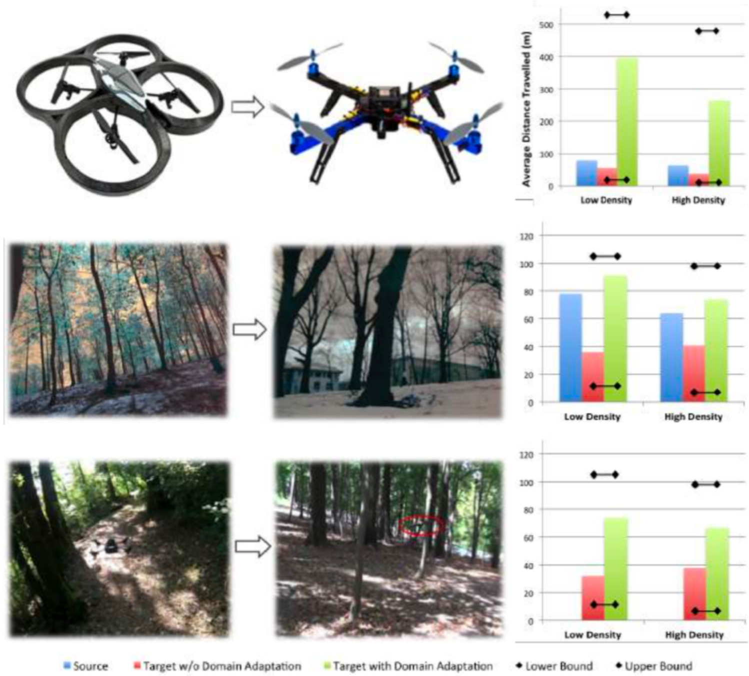
A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots, A. Guisti et al

- Video

Case studies

Learning transferable policies for monocular reactive control of MAV, Daftry, Bagnell, Hebert





Problems

Non-markovian

Multi-model behavior

Output mixtures of Gaussians

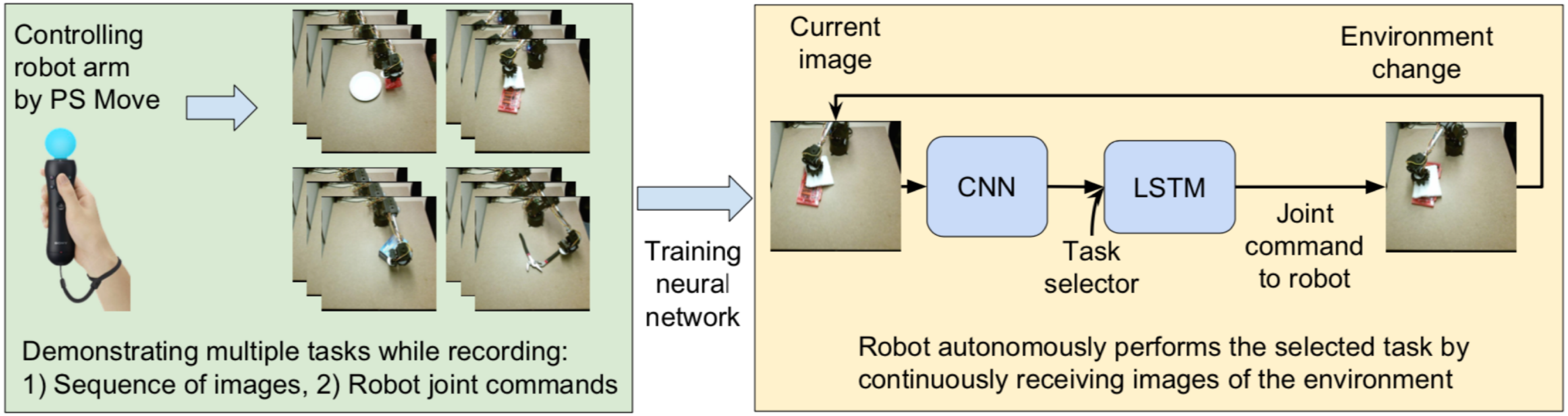
Implicit density models

Auto-regressive discretization

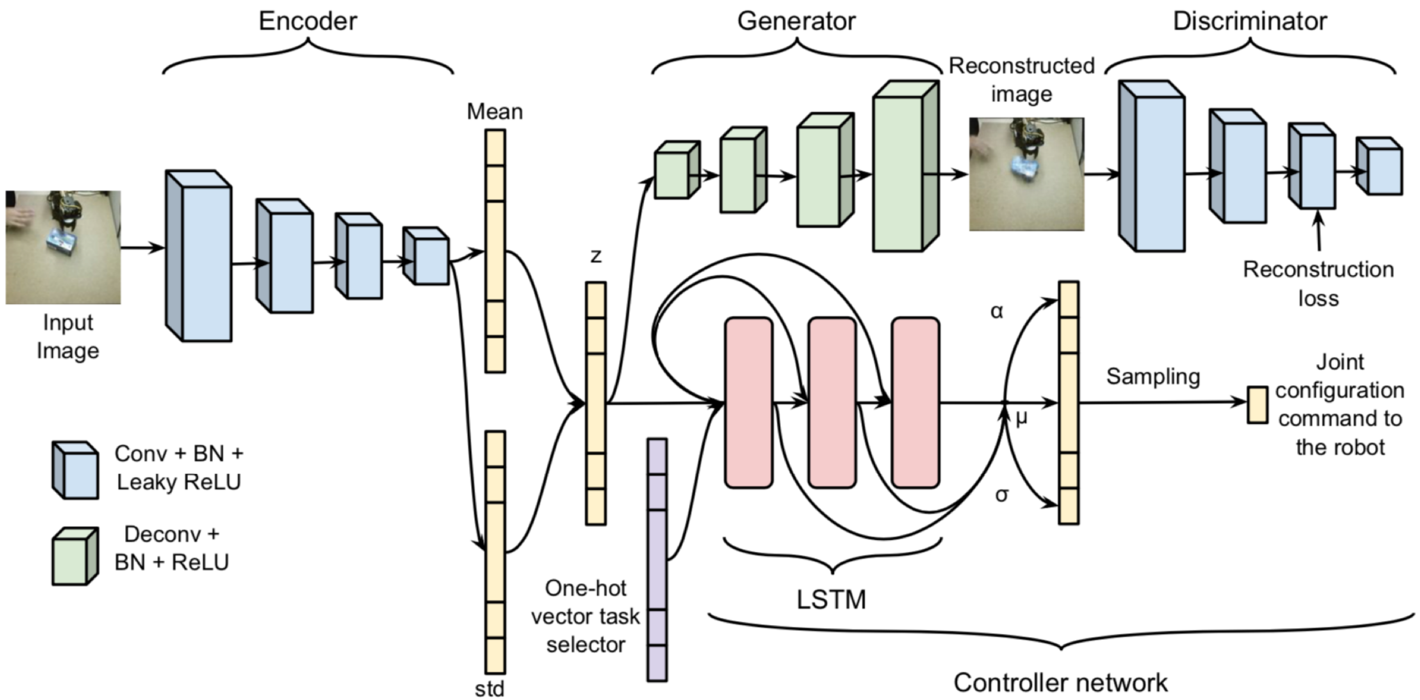
Case studies

Vision-based multi-task manipulation for inexpensive robots using end-to-end demonstrate, Rouhollah Rahmatizadeh, Pooya Abolghasemi, Ladislau Bölöni, and Sergey Levine.

- Video



Controlling
robot arm
by PS Move
Demonstrating multiple tasks while recording:
1) Sequence of images, 2) Robot joint commands
Training
neural
network
Current
image
Environment
change
CNN
LSTM
Task
selector
Joint
command
to robot
Robot autonomously performs the selected task by
continuously receiving images of the environment


Encoder
Generator
Discriminator
Reconstructed image
Mean
z
std
Input Image
Reconstruction loss
α
μ
σ
Sampling
Joint configuration command to the robot
Conv + BN + Leaky ReLU
Deconv + BN + ReLU
One-hot vector task selector
LSTM
Controller network

Issues with supervised learning

Human needs to provide training data

Need a lot of data, some important training data is hard to obtain

Topics : interaction and active learning

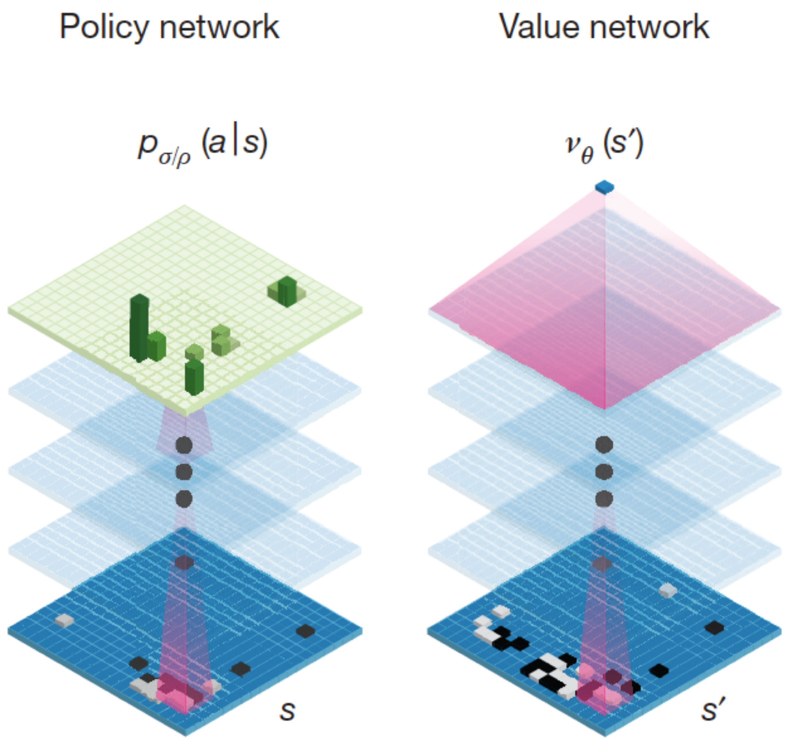
Humans can learn without this level of supervision

From their own experience, feedback through rewards, improving

Back to -> Reinforcement learning

Applications of deep RL

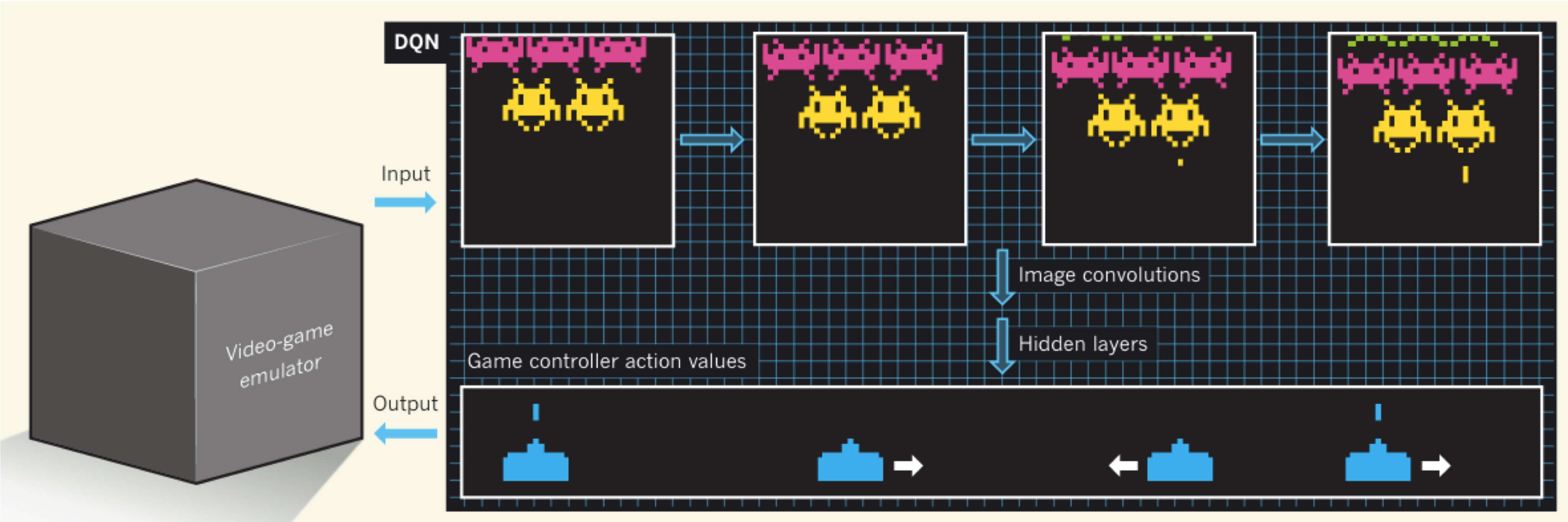
- AlphaGo and AlphaZero



https://deepmind.com/research/alphago/

Applications of deep RL

- Playing video games



[Video](#)

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, [Human-level control through deep reinforcement learning](#), *Nature* 2015

Applications of deep RL

- End-to-end training of deep visuomotor policies

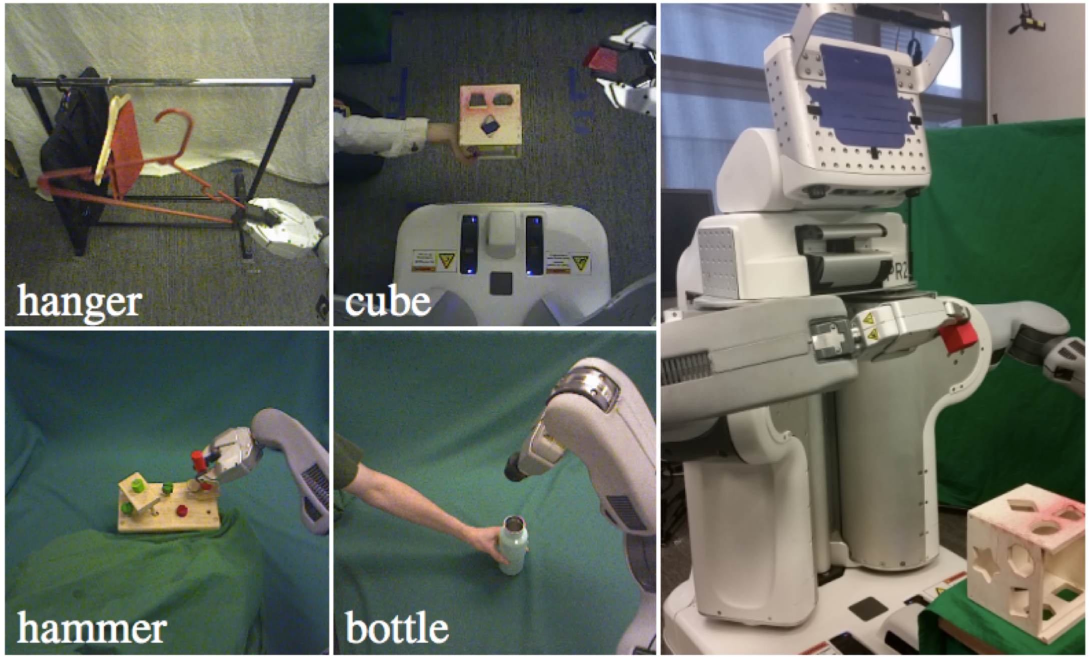


Fig. 1: Our method learns visuomotor policies that directly use camera image observations (left) to set motor torques on a PR2 robot (right).

Video

Sergey Levine et al., Berkeley