Improving the RL Baseline

IFT6757 - Fall 2020

By Étienne Boucher and Mélisande Teng.

Duckie control

Classical robotics methods

- work well when state information from camera feed is correct
- require careful tuning of the parameters depending on the duckie / the environment

RL approaches

- allows exploration of solutions that could not necessarily be found through classical methods or imitating existing expert behavior.
- computationally expensive
- data inefficient

Current RL Baseline: Deep Deterministic Policy Gradient agent

Recall Q learning: optimal policy π* = take the best action as defined by Q* at each time step.

$$Q^*(s,a) = E[R_{t+1} + \gamma \max_{a'} Q^*(s',a')] \qquad a^*(s) = \arg\max_{a}(Q^*(s,a))$$

DDPG concurrently learns a Q-function and a policy

- off-policy algorithm.
- deep Q-learning for continuous action spaces

reward = collision_avoidance_penalty
+ f(speed, lane pose)
+ penalty(lane deviation)

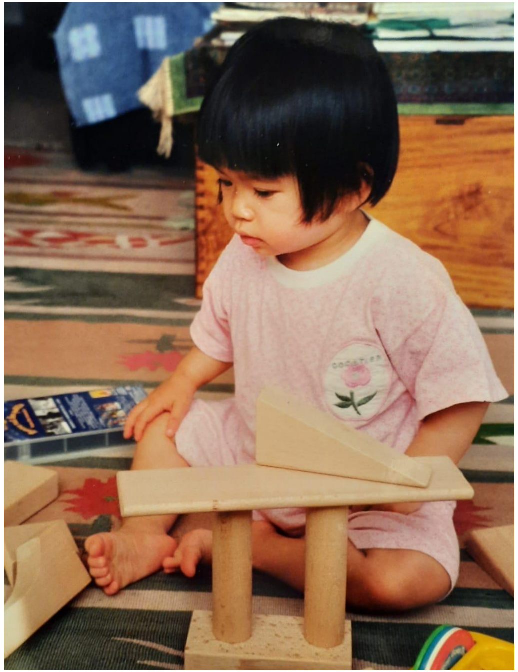
Introduction to DARLA: **D**isent**A**ngled **R**epresentation **L**earning **A**gent

Idea :

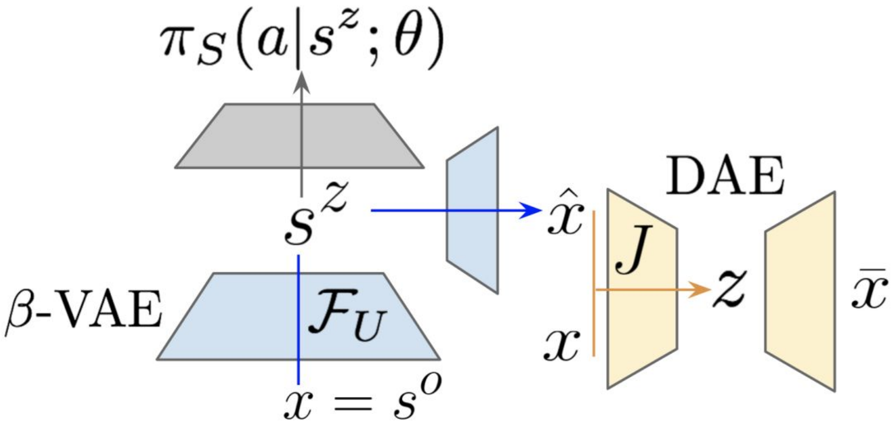
1. Learn to see (solve the perception task)
2. Learn to act (train RL model)
3. Transfer

Goal : learn a **disentangled representation of the environment** to be robust to domain shifts.
→ in Duckietown: different simulator maps, sim/real

Project agent observation state space to a latent state space expressed in terms of factorised data generative factors that are representative of the natural world.



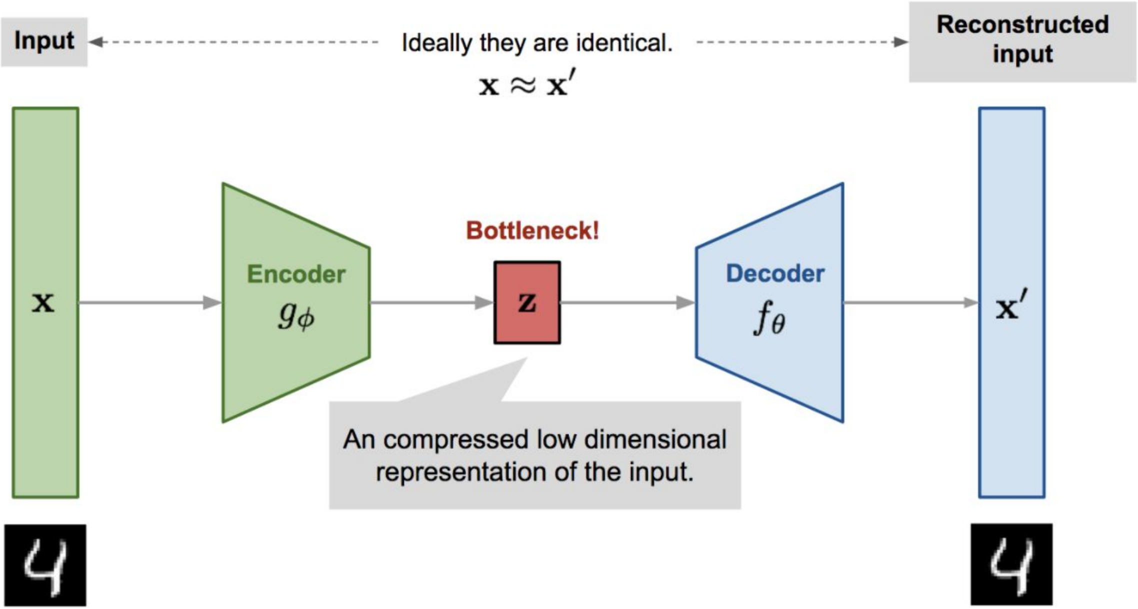
Introduction to DARLA



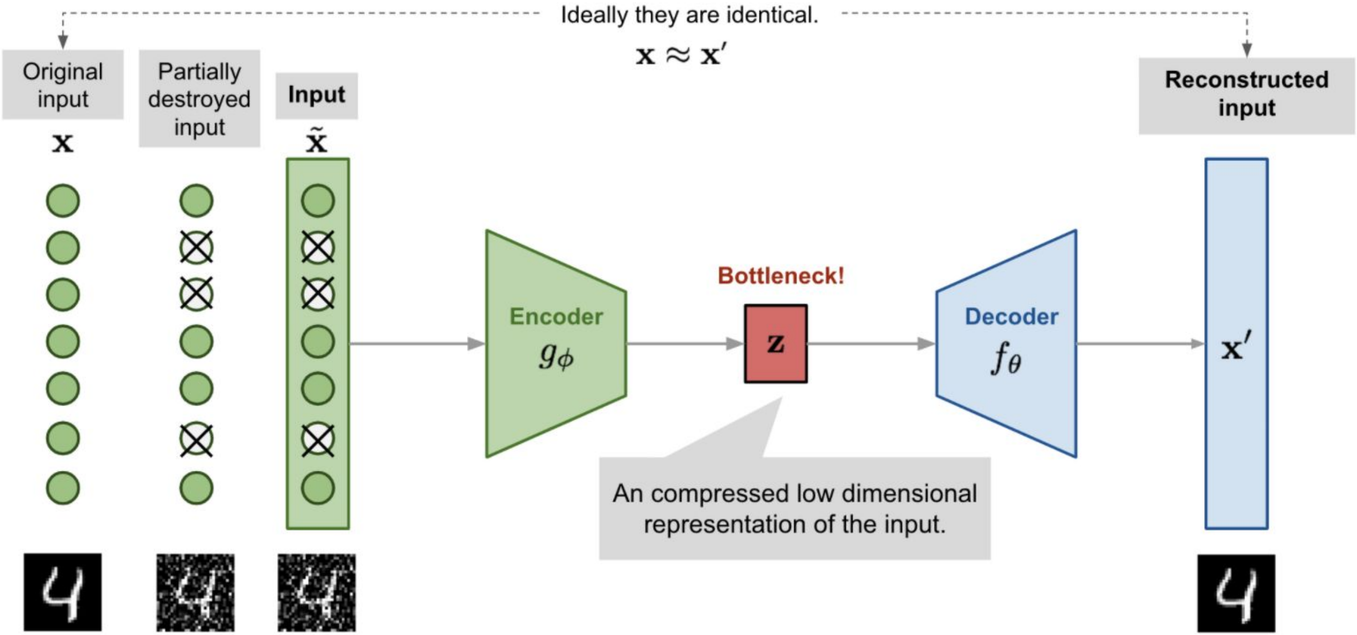
Perceptual model:

1. Train DAE model to get targets in feature space
2. Train β-VAE model using targets from 1.

Perceptual Model: Autoencoders



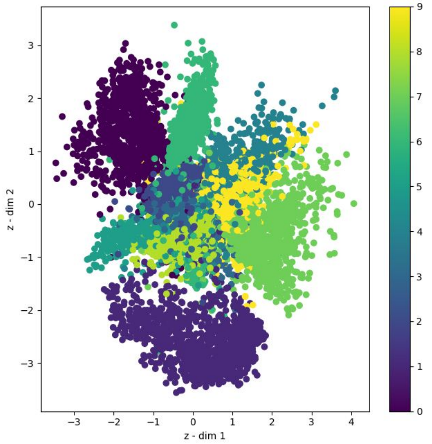
Perceptual Model: DAE

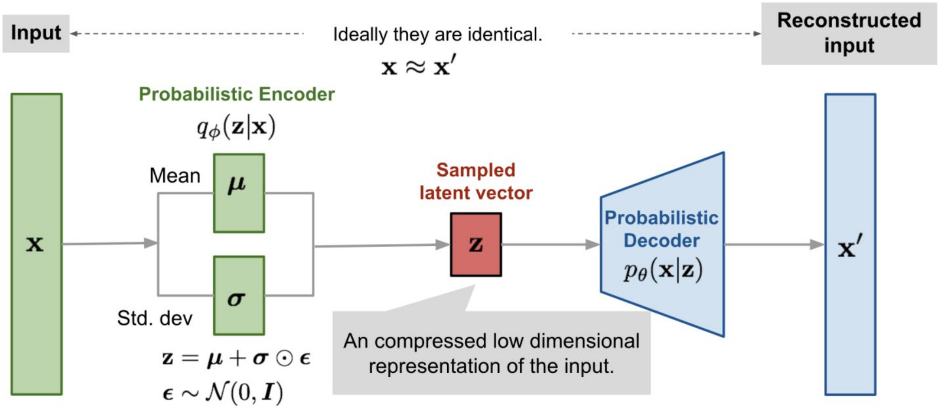


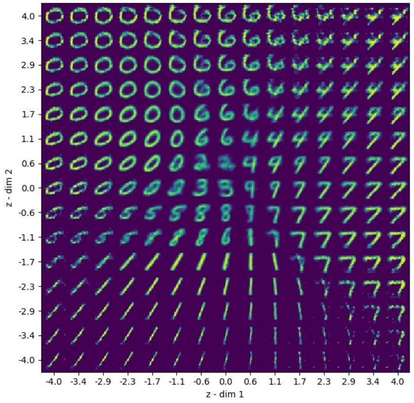
Perceptual Model: VAE

Idea : a more regular, continuous latent space
- Map the input to a distribution
- Add regularization

Objective: $$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}))$$







Perceptual Model: Beta-VAE_DAE

Objective:

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta\, D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

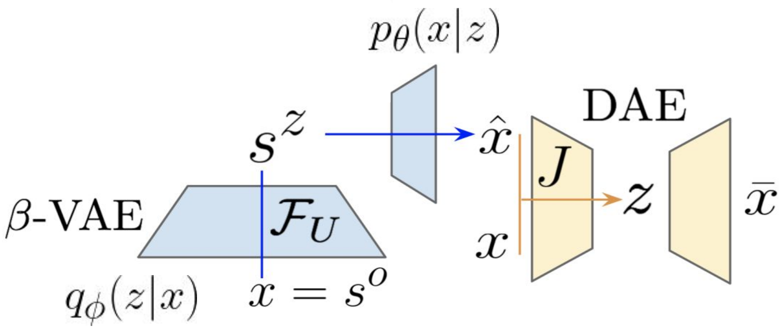
Increase β parameter to encourage more disentangled latent representation.

In DARLA:

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\, \|J(\hat{\mathbf{x}}) - J(\mathbf{x})\|_2^2 - \beta\, D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

J is passing the input image in DAE up to a chosen layer.

Train with perceptual similarity loss



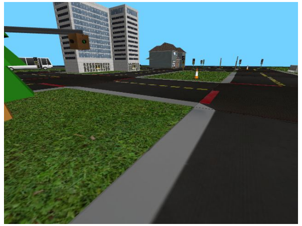
That’s where we are at!

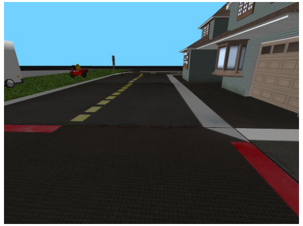
Dataset:

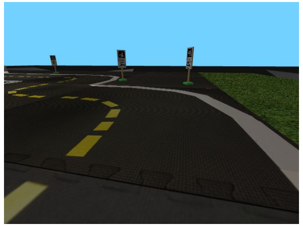
- 6000 images
- random positions and orientations
- every object meshes
- every type of tile

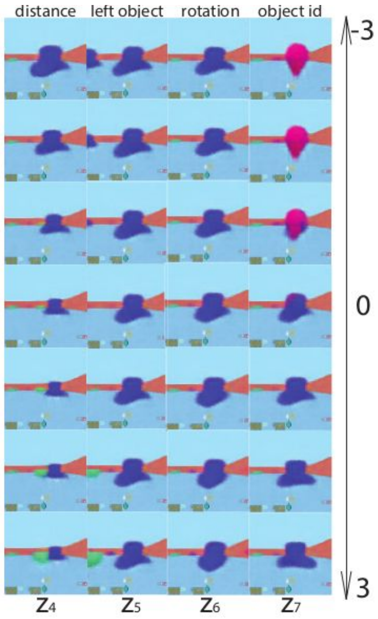
DARLA model implemented

Have yet to see the traversals









Next steps: RL agent and transfer

- Train the RL agent on top of the perceptual model
- Try the model without further training on a real robot

Thanks for your attention !

References

Continuous control with deep reinforcement learning, Lillicrap et al.

DARLA: Improving Zero-Shot Transfer in Reinforcement Learning, Higgins et al.

Nice auto encoder illustrations from https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html