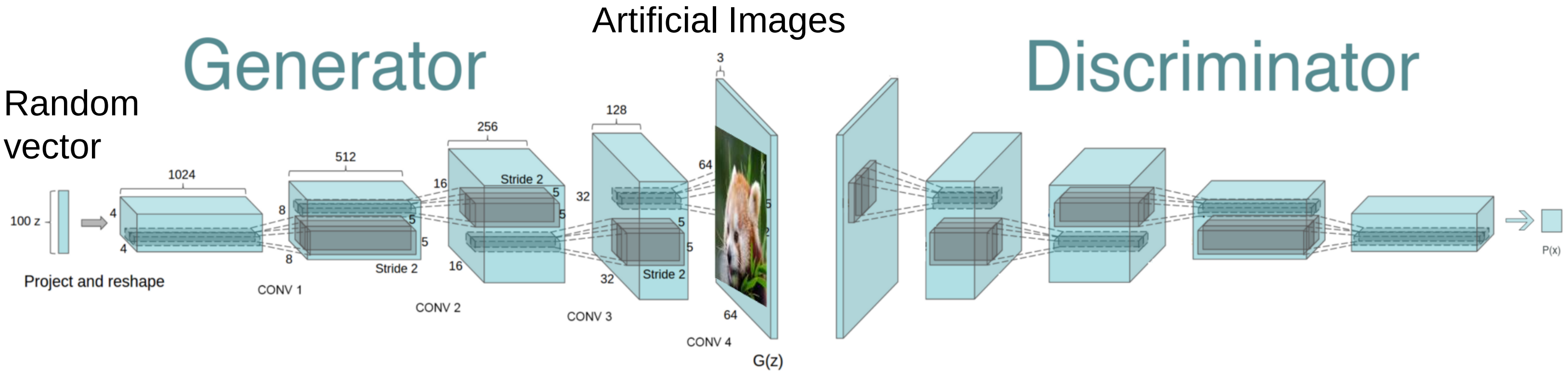
CS194/294-129: Designing, Visualizing and Understanding Deep Neural Networks

John Canny

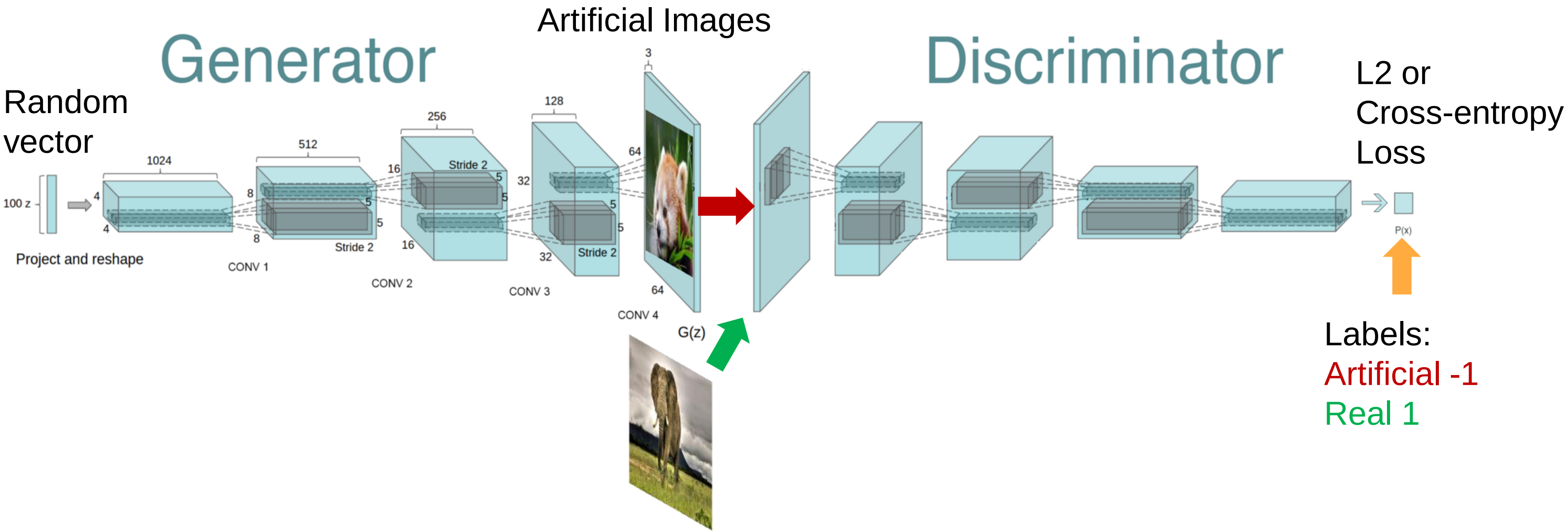
Spring 2018

Lecture 17: Imitation Learning

Last Time: Generative Adversarial Networks (GANs)

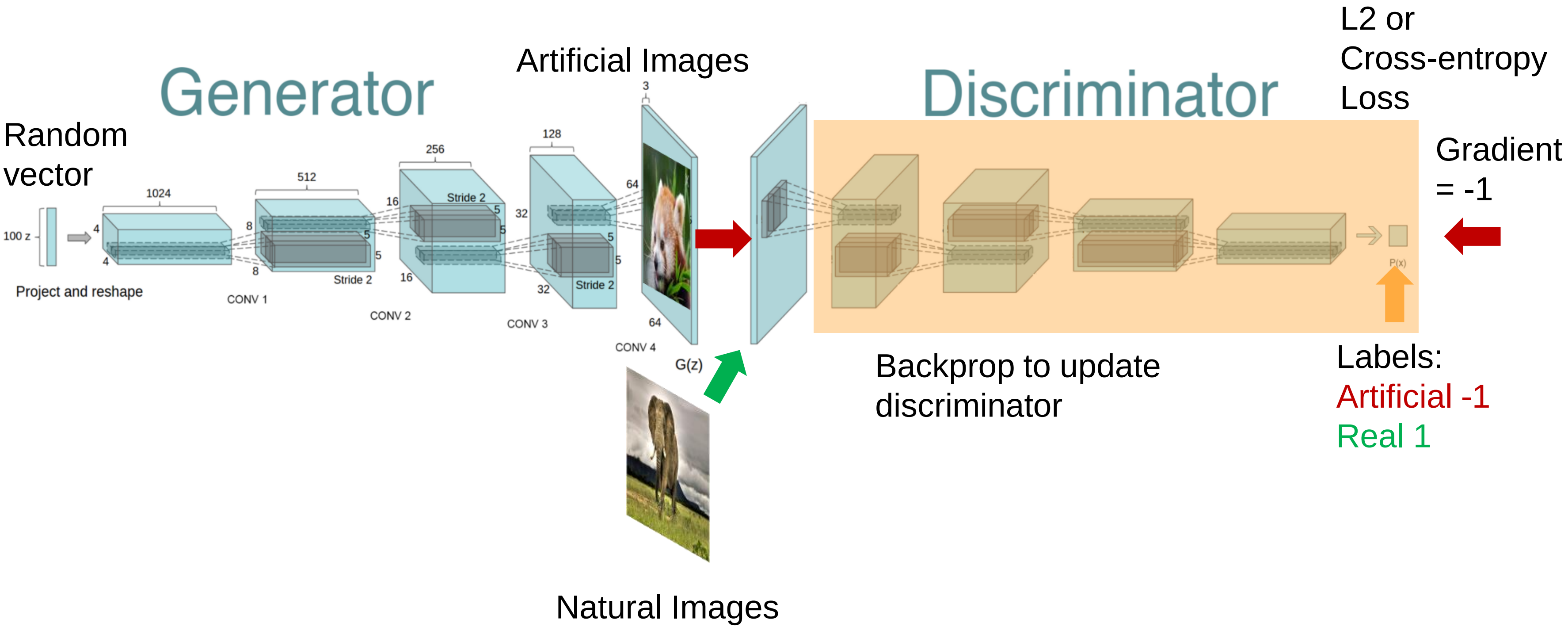


Last Time: GAN Discriminator Training



Natural Images

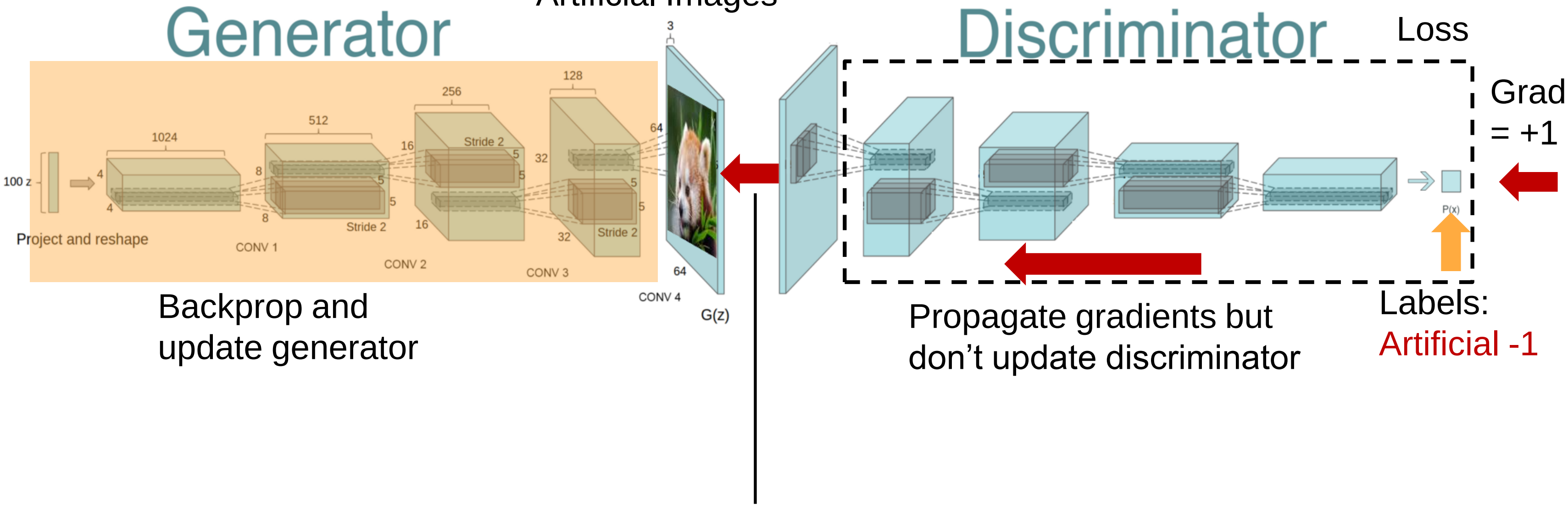
GAN Training: Minimize Discriminator classification loss



GAN Training: Train Generator to Fool the Discriminator

Artificial Images

L2 or
Cross-entropy
Loss



Gradient
= +1

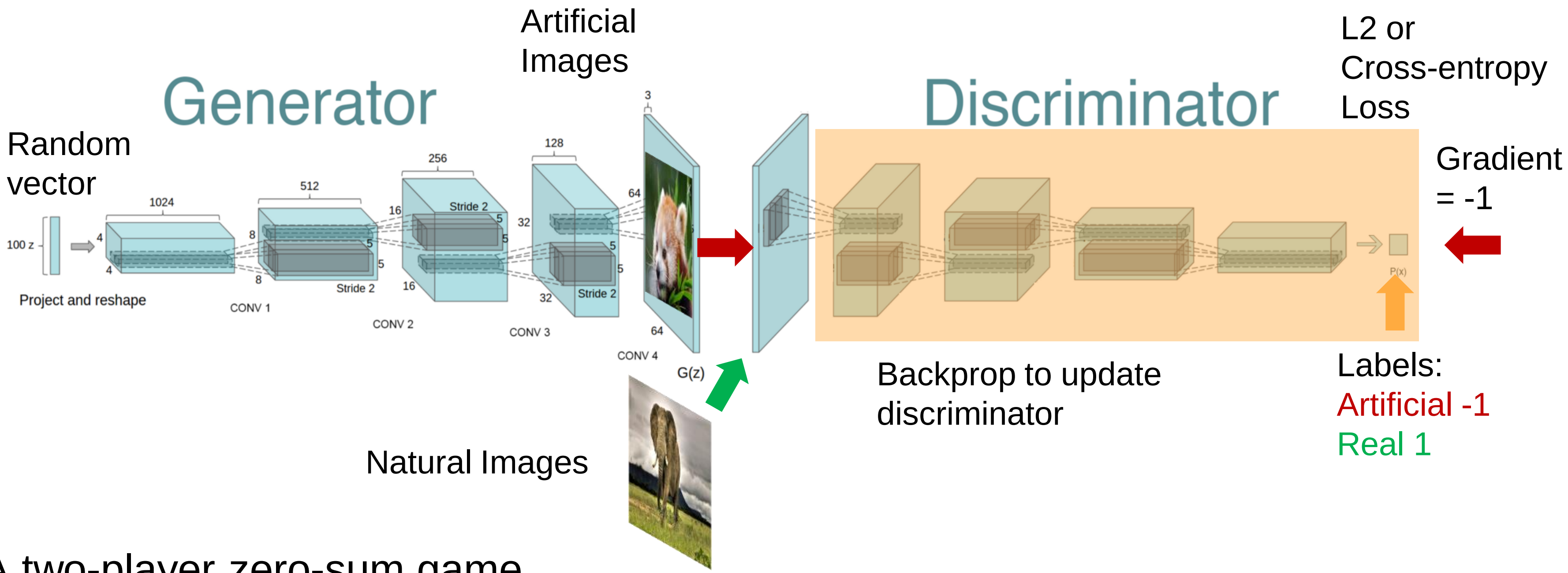
Backprop and
update generator

Propagate gradients but
don’t update discriminator

Labels:
Artificial -1

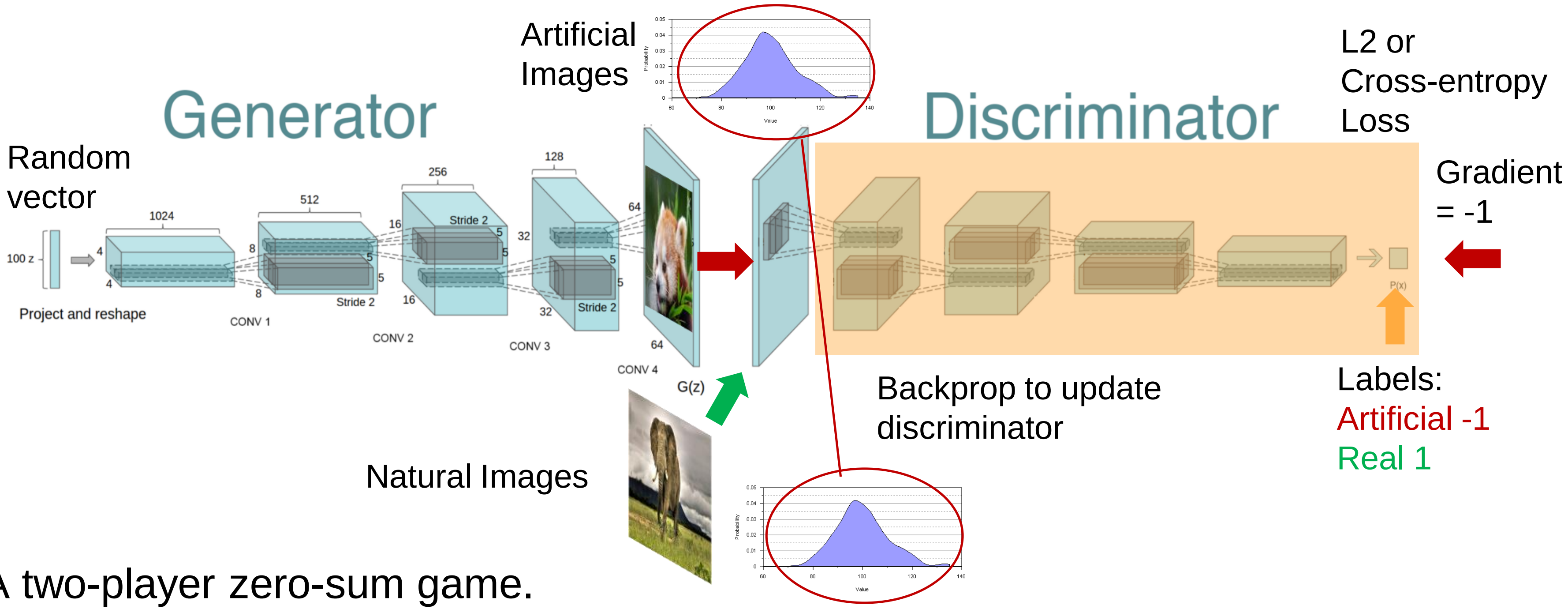
This gradient nudges the image
from “artificial” toward “natural”

GAN Training: Alternate Discriminator/Generator training



A two-player zero-sum game.

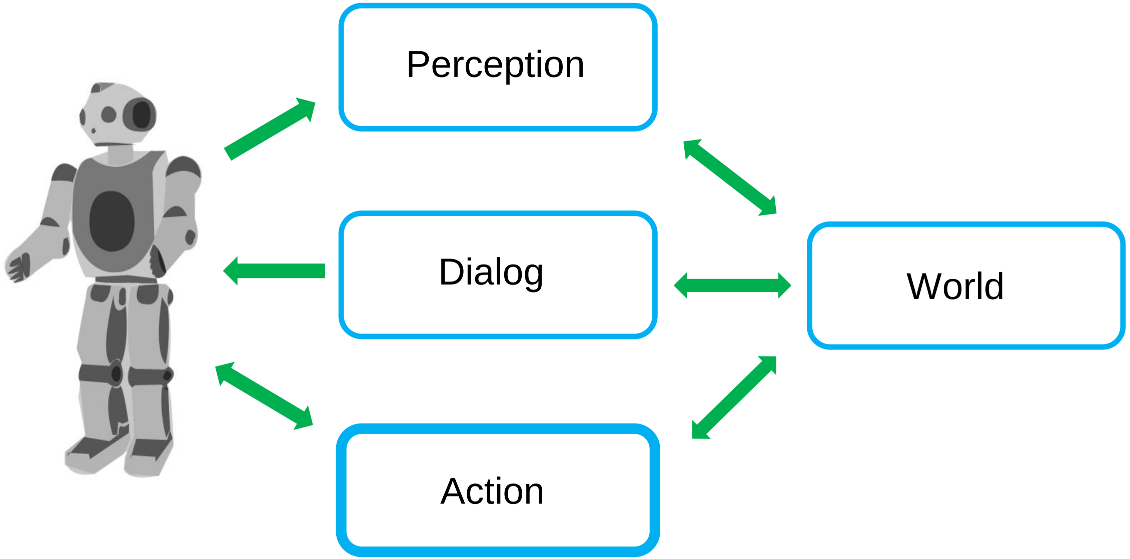
GAN Training: Alternate Discriminator/Generator training



A two-player zero-sum game.

Optimizing with minimax (alternating optimization) minimizes the difference (Jensen-Shannon divergence) between generator and natural image probability distributions.

This Time: Deep Control



Deep Control: First Idea: Imitate Human Actions

Supervised training of deep networks (with image category labels, captions, translations,…) from human data has worked well so far…

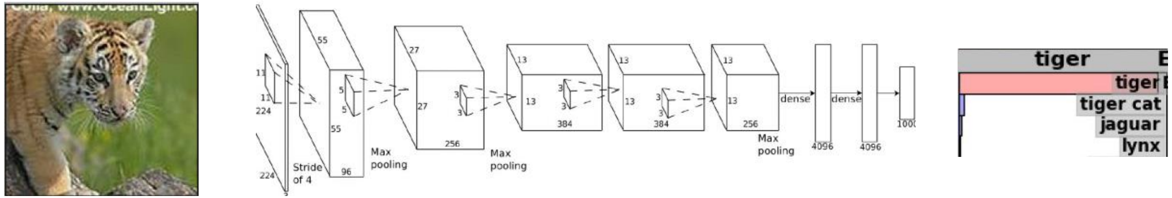
What about mimicking human control actions?

Imitation Learning via Behavior Cloning

This approach is called behavior cloning. Note that its not enough to record human actions, because humans are constantly adapting to the world.

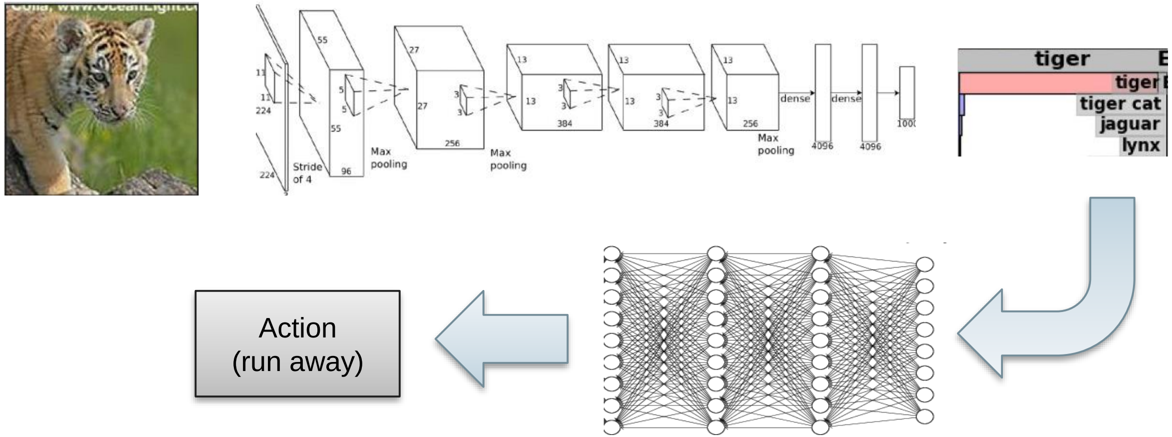
We need to learn a control loop from sensors to actuators.

Sensorimotor Learning



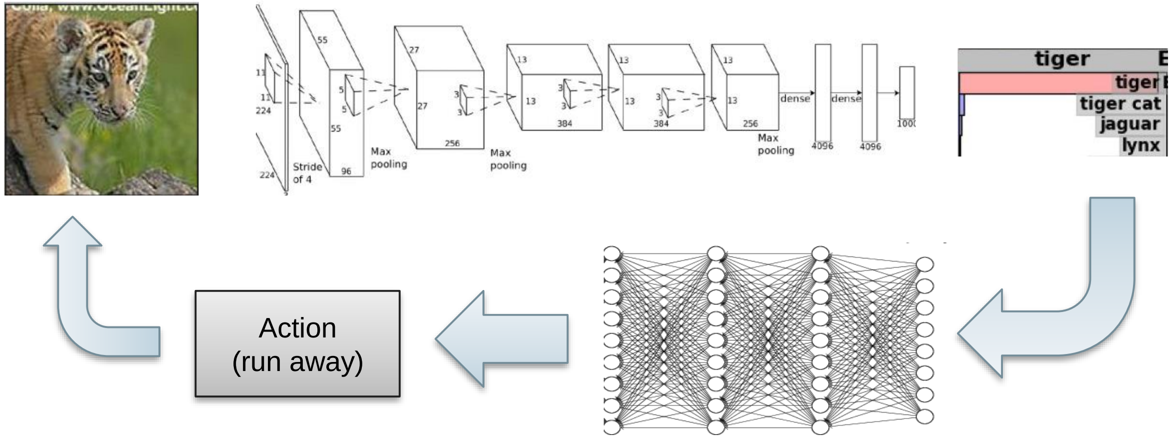
Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

Sensorimotor Learning

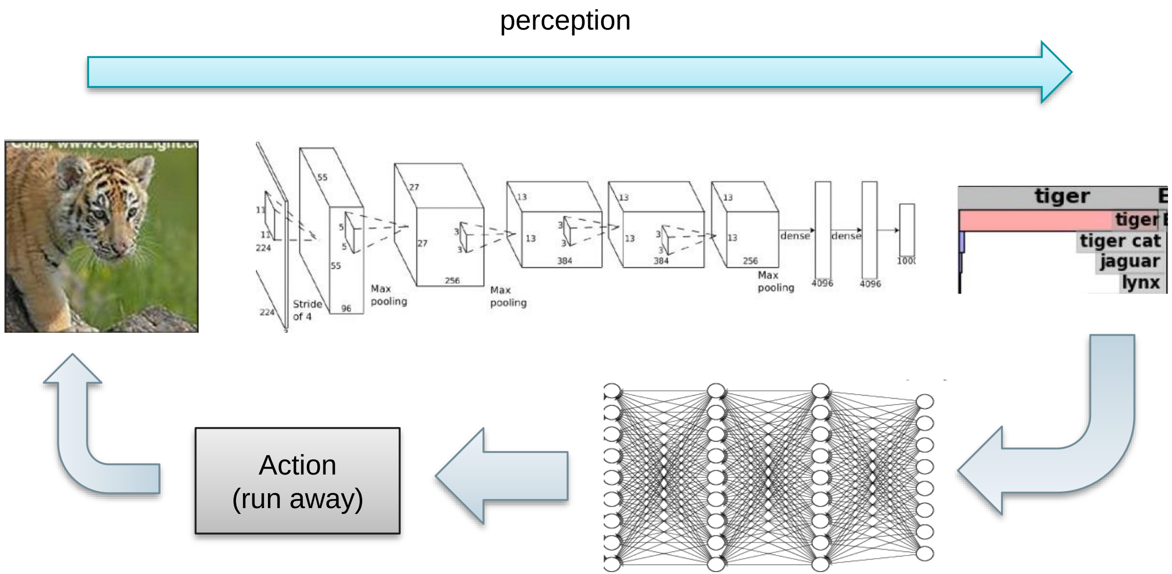


Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

Sensorimotor Learning

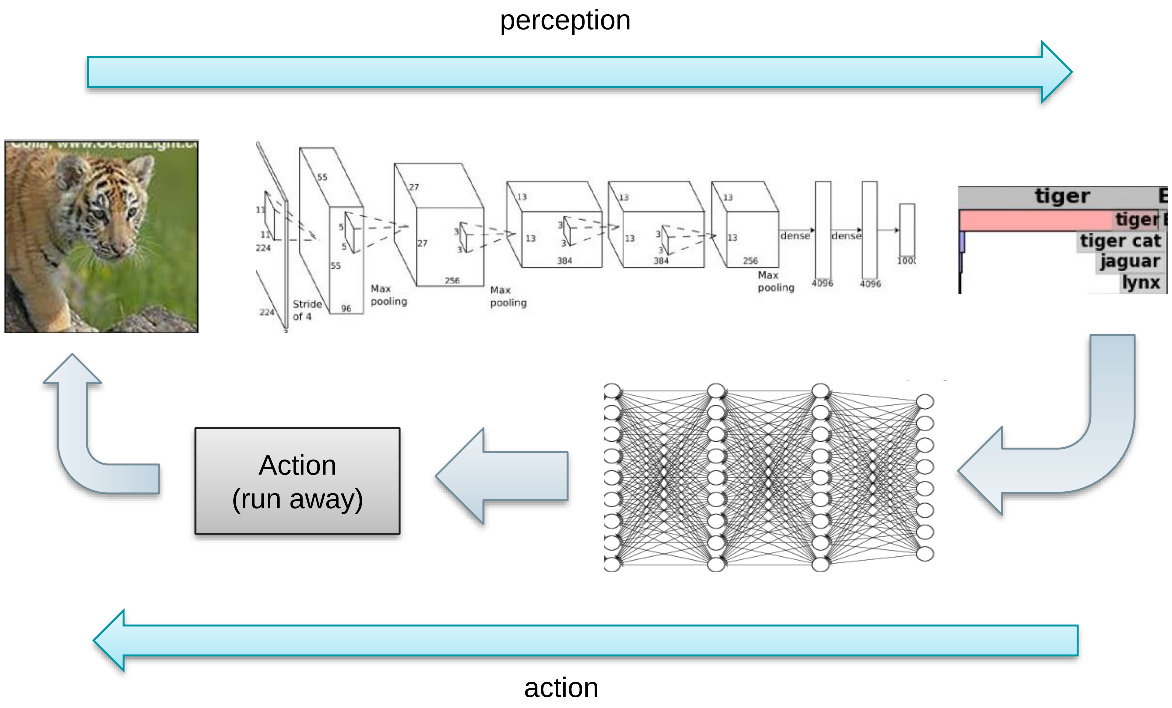


Sensorimotor Learning



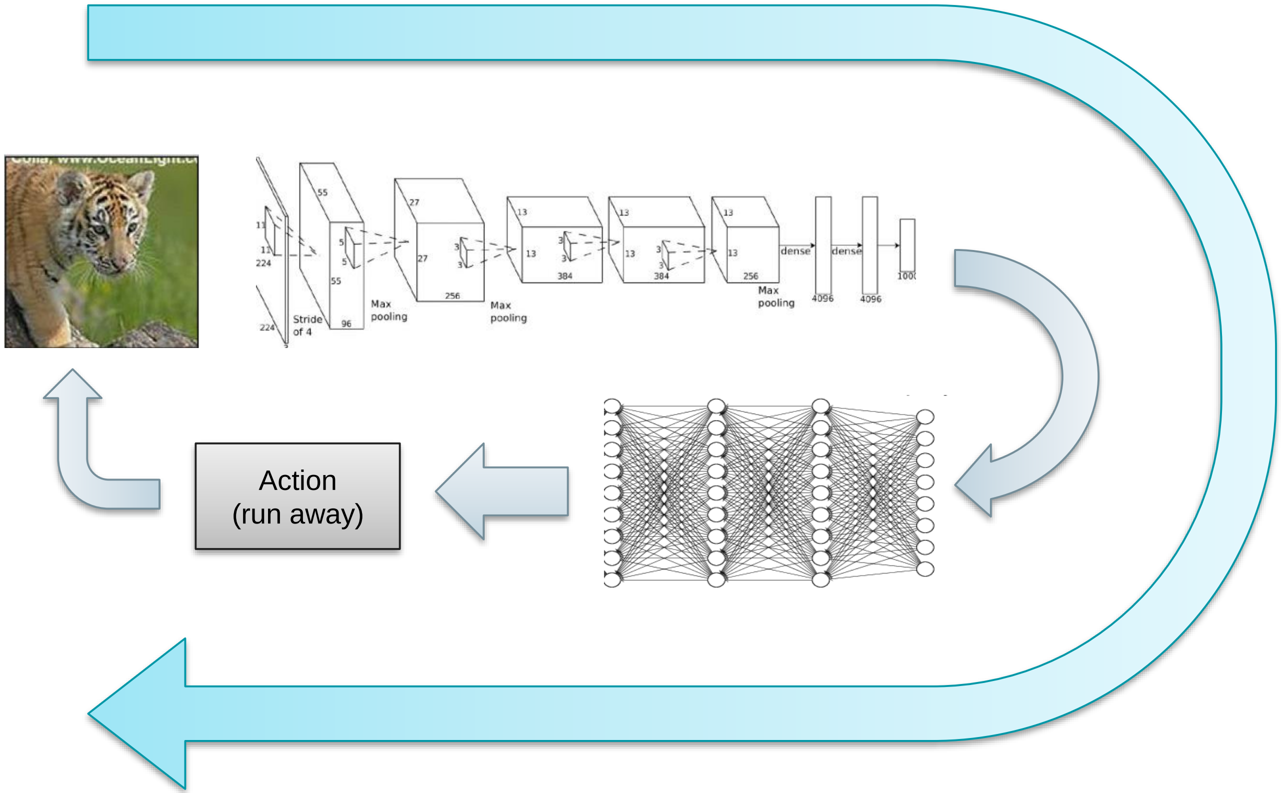
Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

Sensorimotor Learning



Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

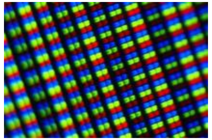
sensorimotor loop



Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

End-to-end vision

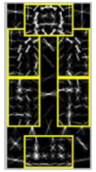
standard computer vision



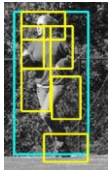
features (e.g. HOG)

mid-level features (e.g. DPM)

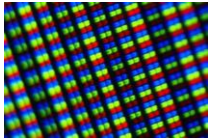
Felzenszwalb ‘08

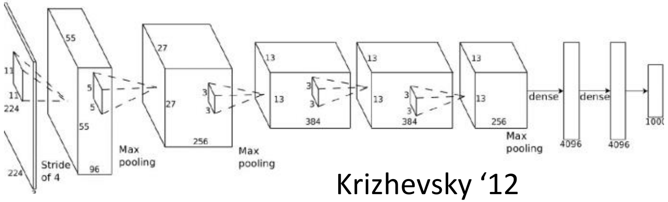


classifier (e.g. SVM)

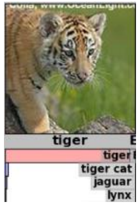


deep learning





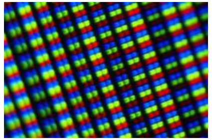
Krizhevsky ‘12



Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

End-to-end vision

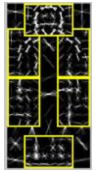
standard computer vision



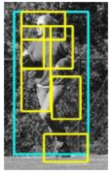
features (e.g. HOG)

mid-level features (e.g. DPM)

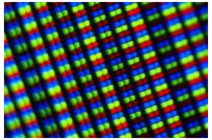
Felzenszwalb ‘08

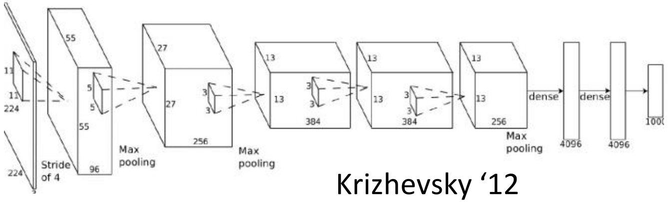


classifier (e.g. SVM)

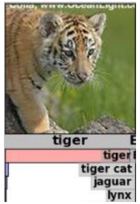


deep learning





Krizhevsky ‘12



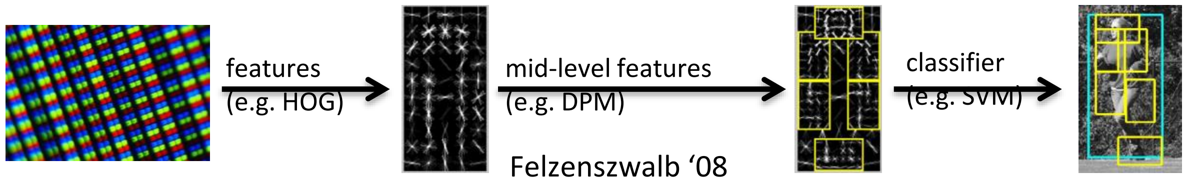
End-to-end control

standard robotic control

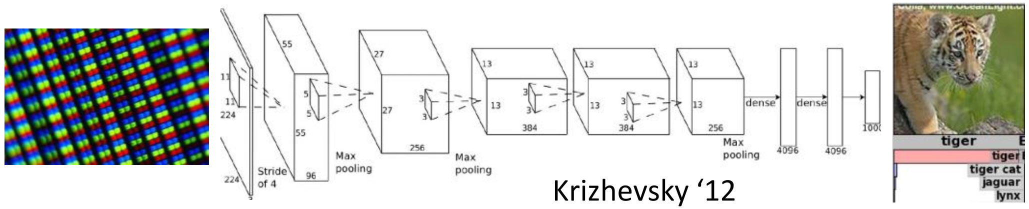
observations → state estimation (e.g. vision) → modeling & prediction → motion planning → low-level controller (e.g. PD) → motor torques

End-to-end vision

standard computer vision



deep learning



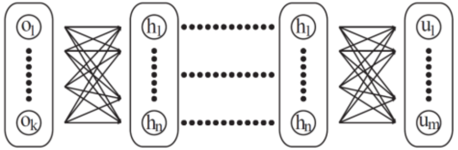
End-to-end control

standard robotic control

observations → state estimation (e.g. vision) → modeling & prediction → motion planning → low-level controller (e.g. PD) → motor torques

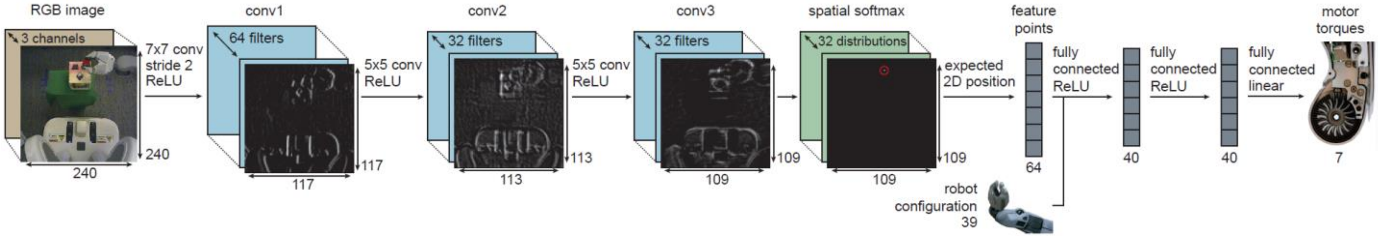
deep sensorimotor learning

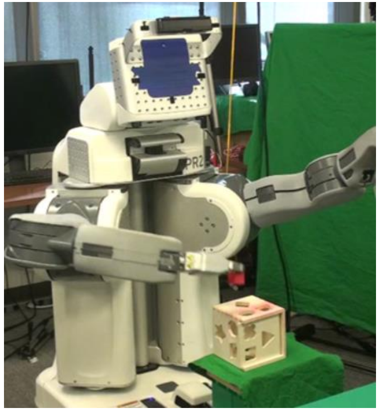
observations →



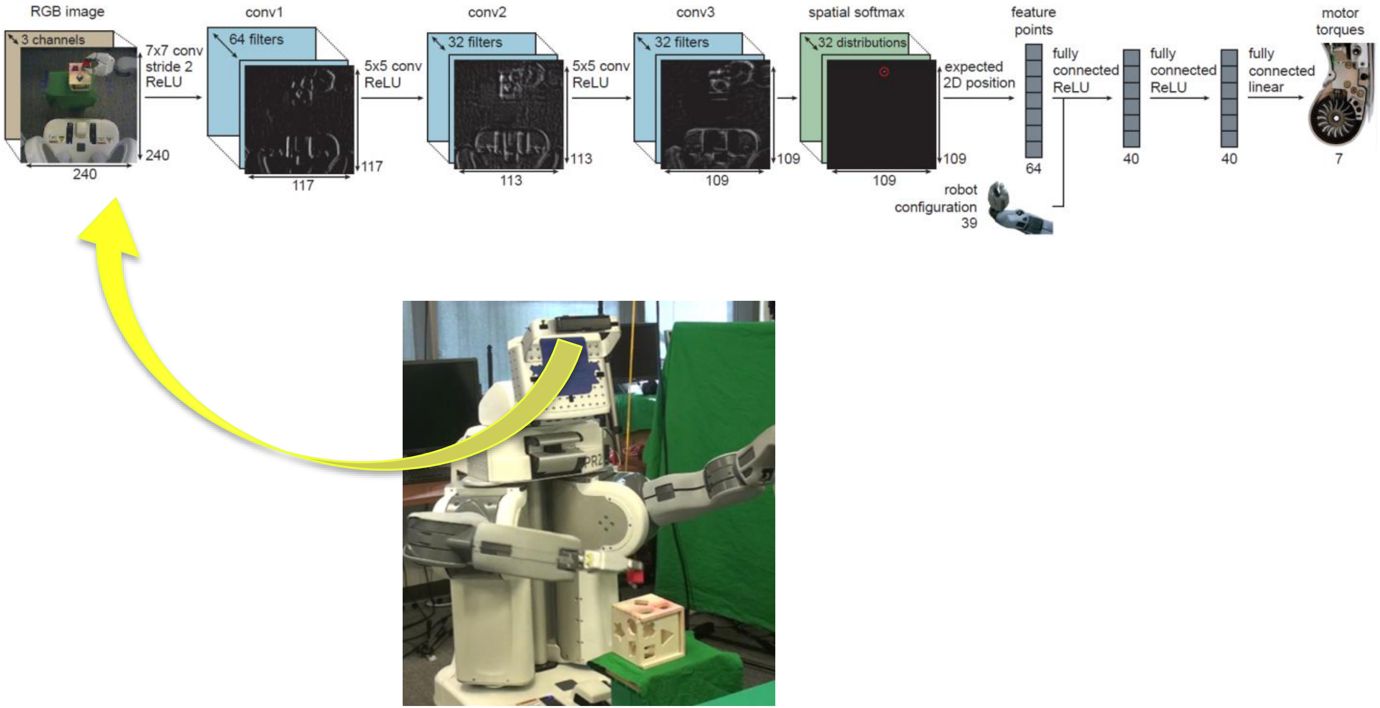
→ motor torques

Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

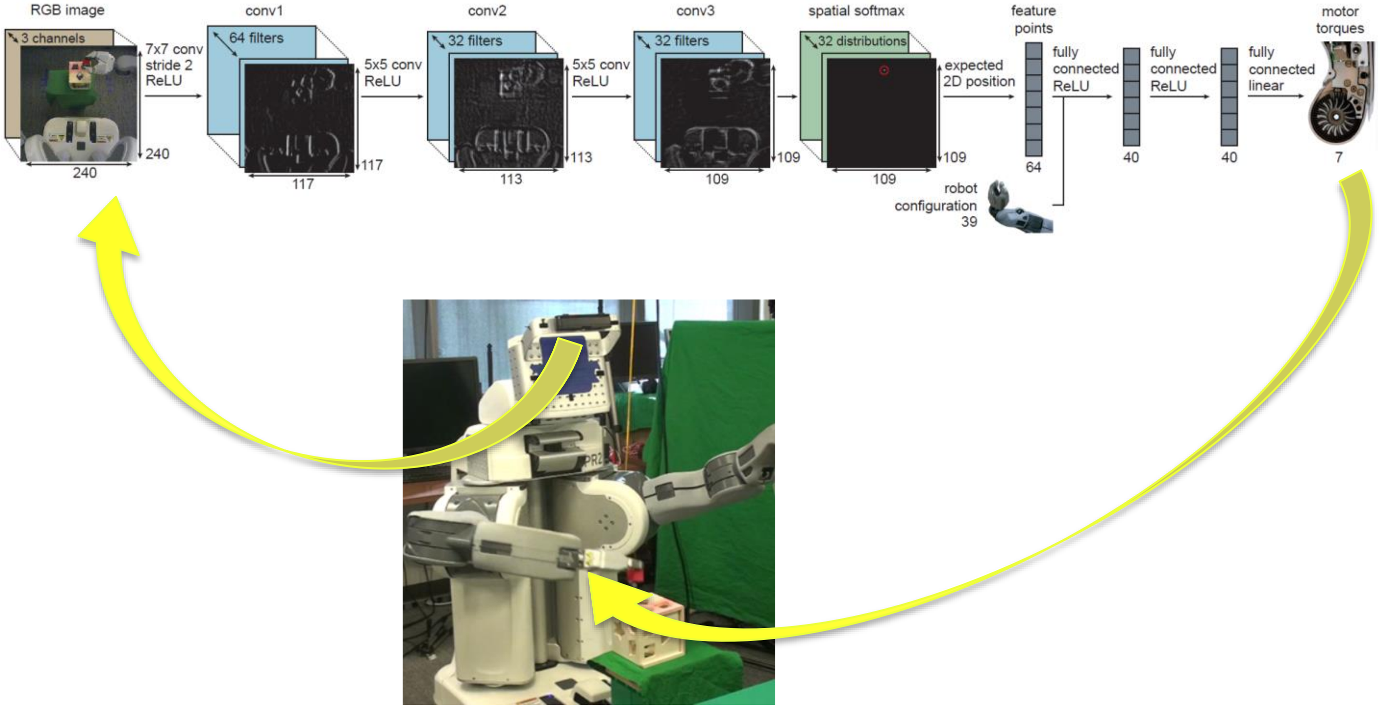




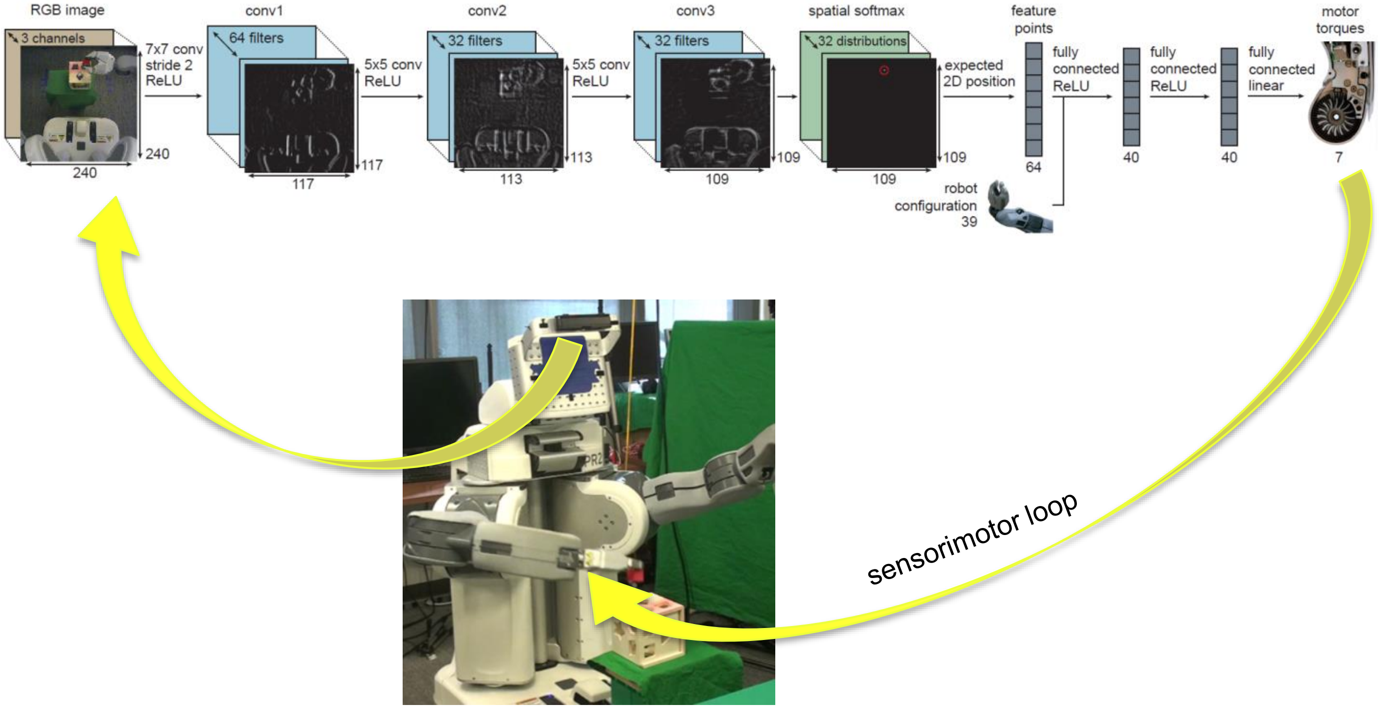
Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017



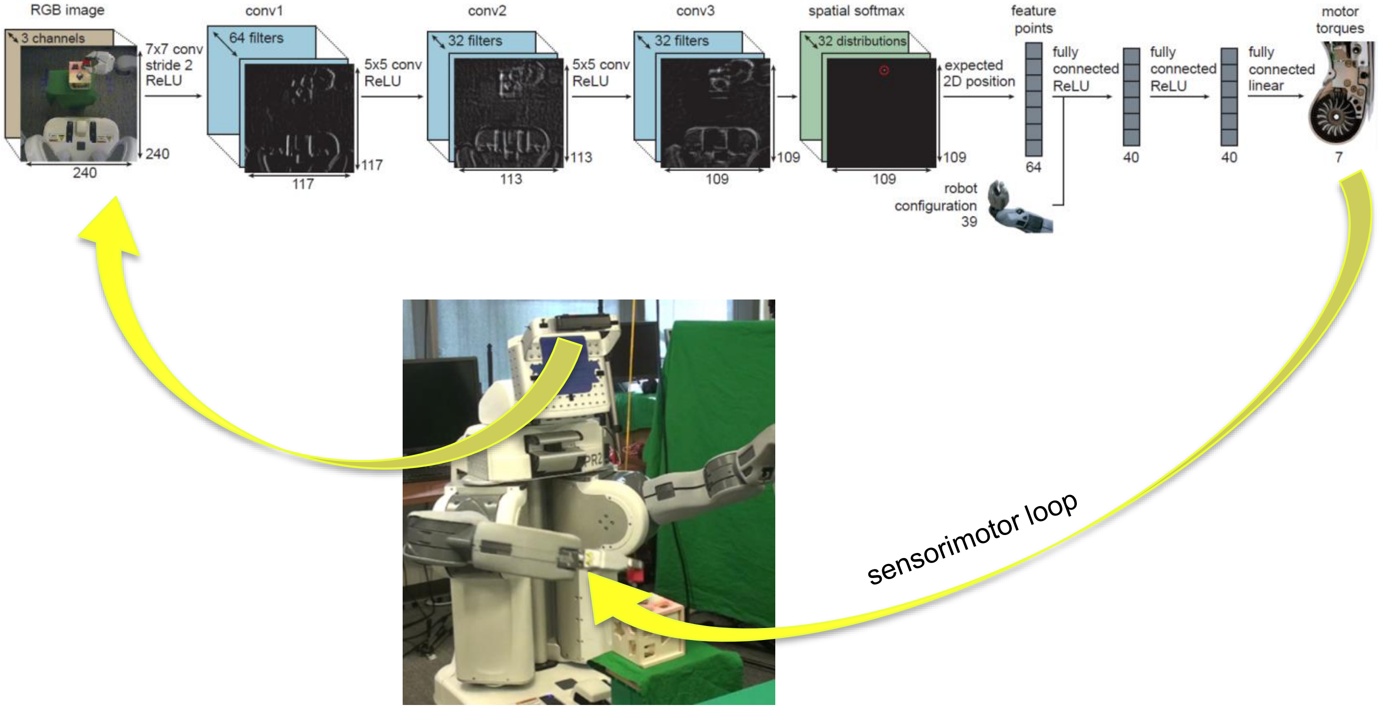
Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017



Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

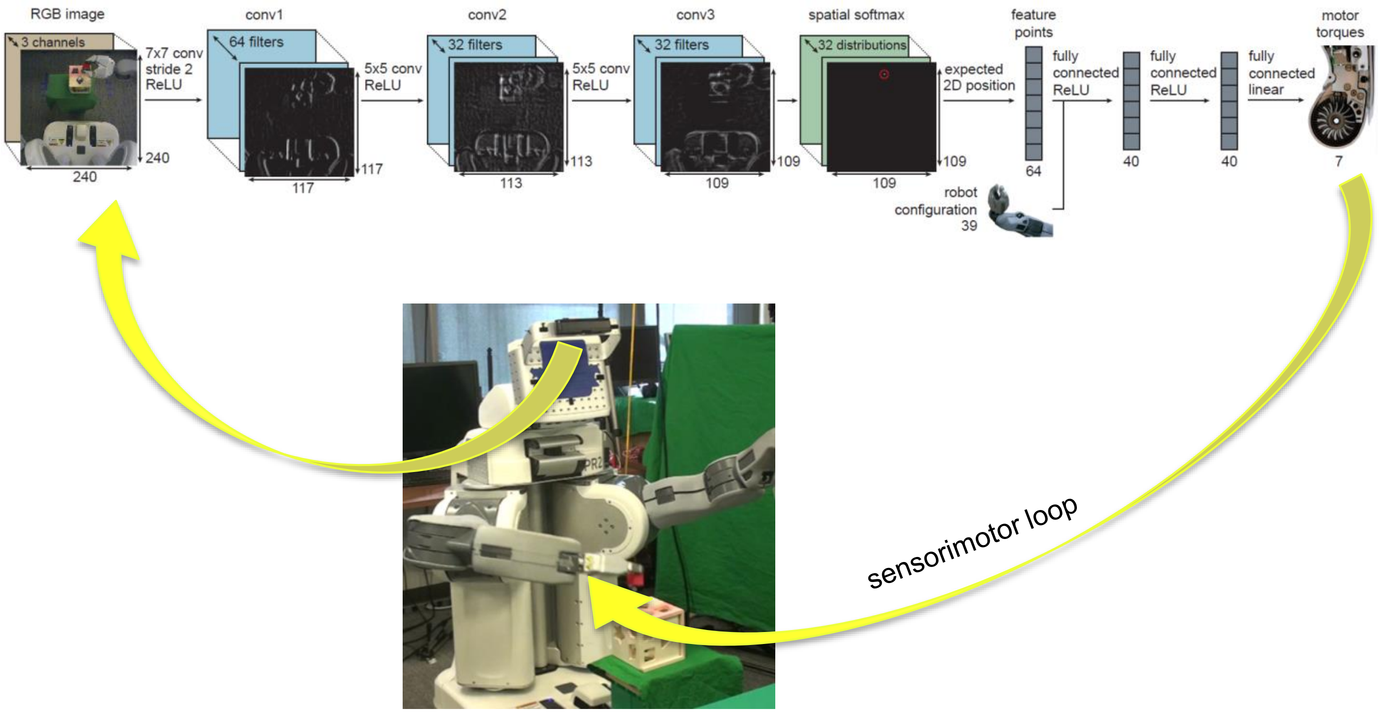


Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017



indirect supervision

Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017



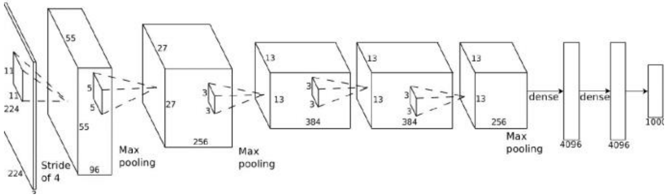
indirect supervision

actions have consequences

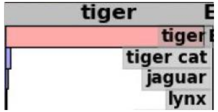
Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

Terminology & notation

$\mathbf{o}$

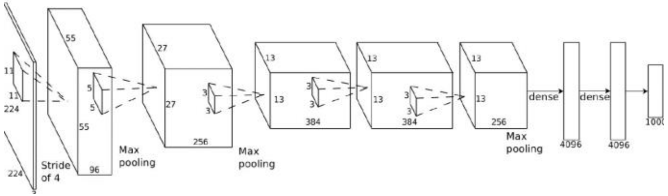


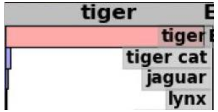
$\pi_\theta(\mathbf{u}|\mathbf{o})$



$\mathbf{u}$

Terminology & notation

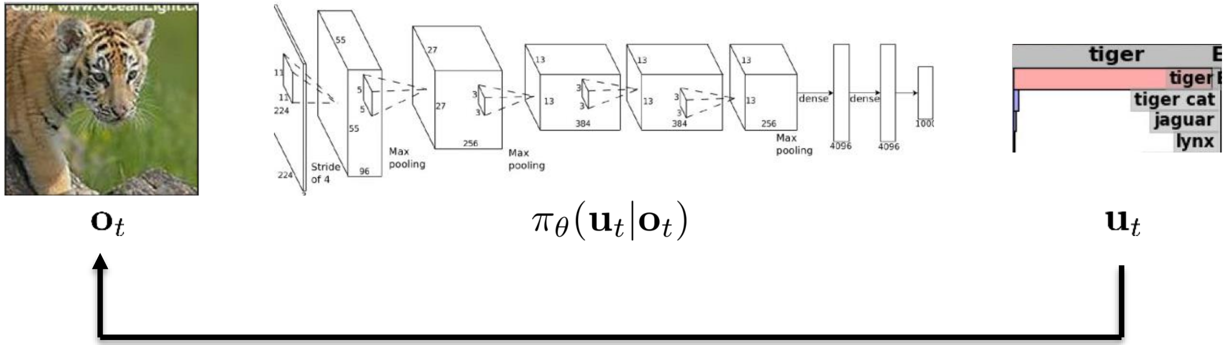




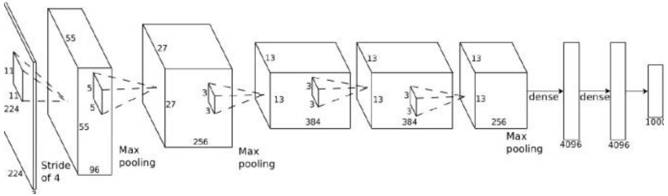
$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

Terminology & notation



Terminology & notation

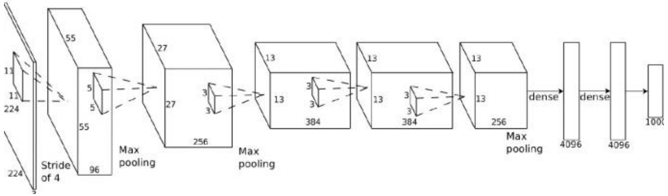


1. run away

$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

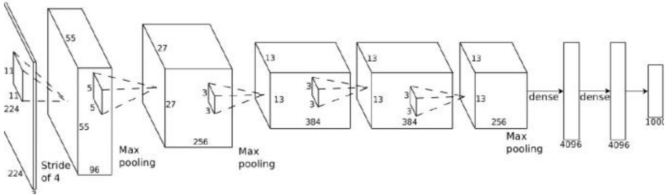
Terminology & notation



1. run away
2. ignore

$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

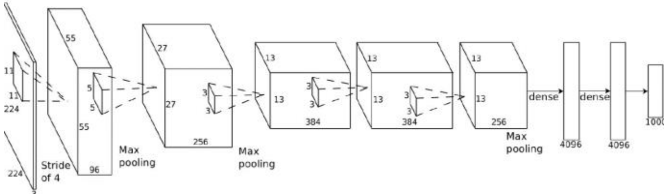
Terminology & notation

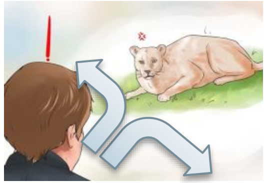


1. run away
2. ignore
3. pet

$\mathbf{o}_t$ $\qquad \pi_\theta(\mathbf{u}_t|\mathbf{o}_t) \qquad$ $\mathbf{u}_t$

Terminology & notation



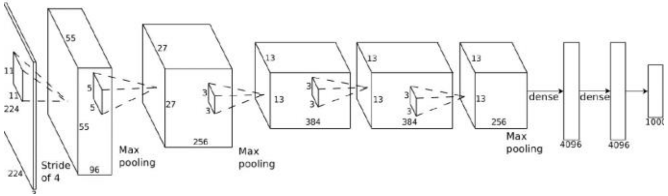


$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

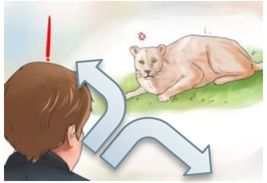
Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Terminology & notation

$\mathbf{o}_t$



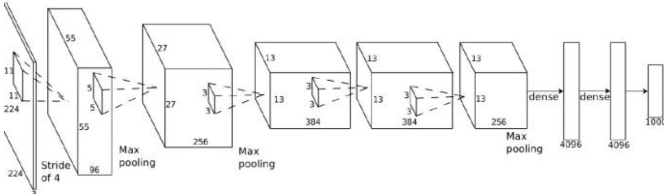
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

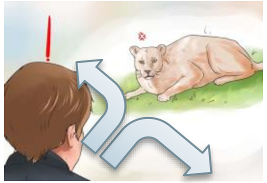


$\mathbf{u}_t$

$\mathbf{o}_t$ – observation

Terminology & notation



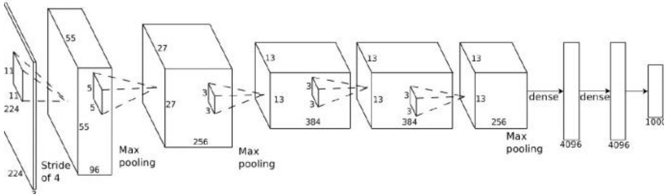


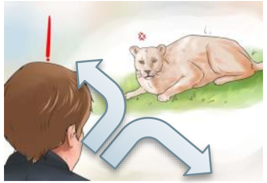
$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

$\mathbf{o}_t$ – observation

$\mathbf{u}_t$ – action

Terminology & notation





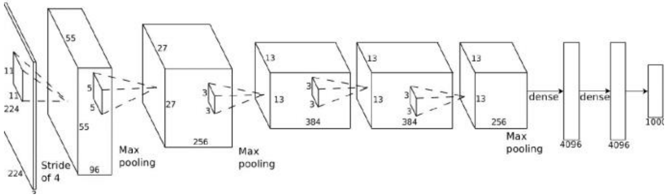
$\mathbf{o}_t$ $\qquad$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\qquad$ $\mathbf{u}_t$

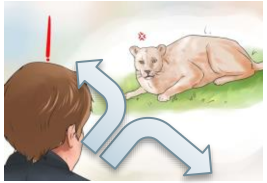
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Terminology & notation



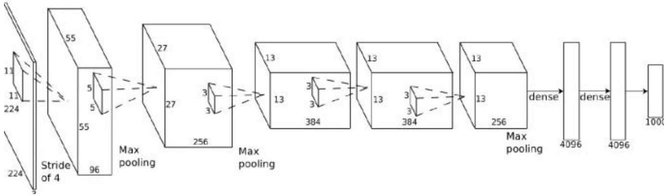


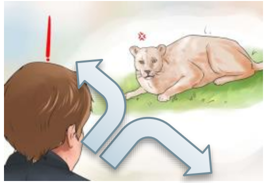
$\mathbf{o}_t$ $\qquad$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\qquad$ $\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

Terminology & notation





$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

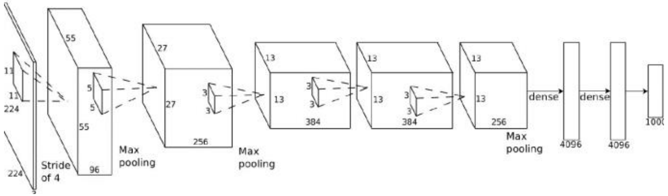
$\mathbf{x}_t$ – state

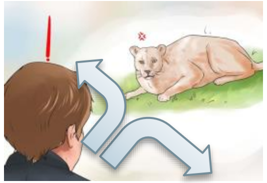
$\mathbf{o}_t$ – observation

$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

Terminology & notation





$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

$\mathbf{x}_t$ – state

$\mathbf{o}_t$ – observation

$\mathbf{u}_t$ – action

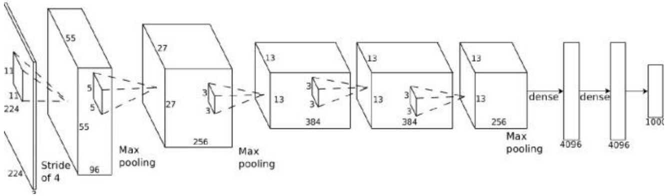
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

$\mathbf{o}_t$ – observation

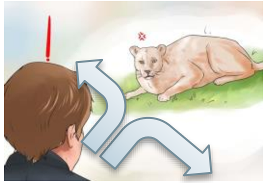
Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Terminology & notation

$\mathbf{o}_t$



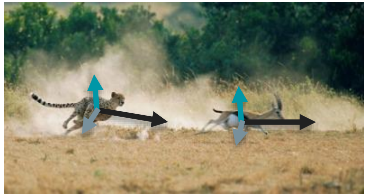
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$



$\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

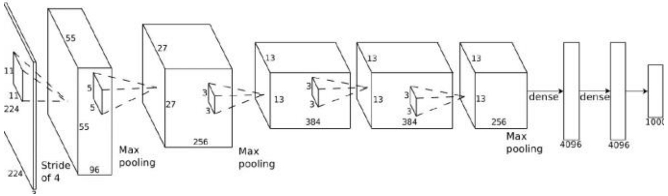


$\mathbf{o}_t$ – observation

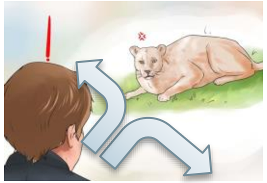
Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

Terminology & notation

$\mathbf{o}_t$



$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$



$\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

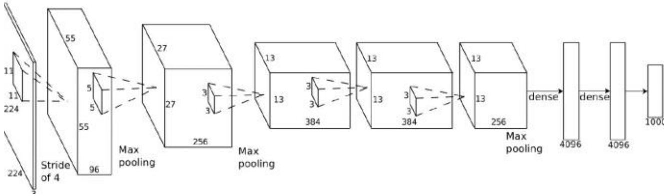
$\mathbf{o}_t$ – observation

$\mathbf{x}_t$ – state

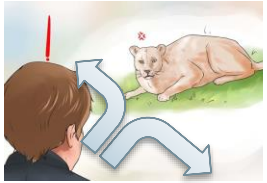
Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Terminology & notation

$\mathbf{o}_t$



$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$



$\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

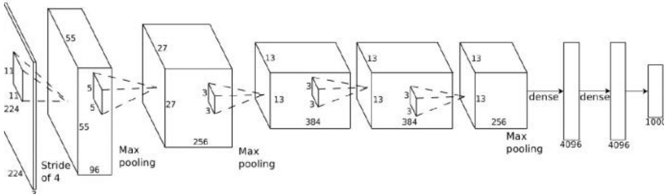
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

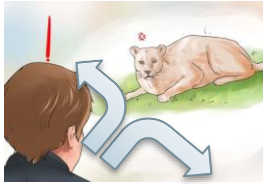
$\mathbf{o}_t$ – observation

$\mathbf{x}_t$ – state

Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Terminology & notation





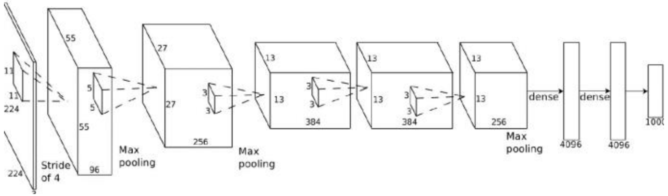
$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

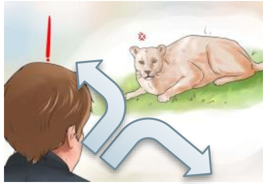
$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Terminology & notation

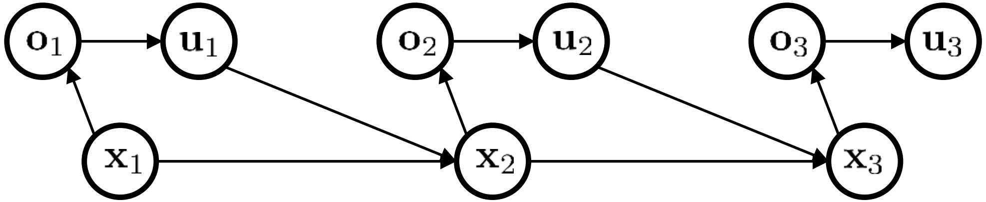




$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

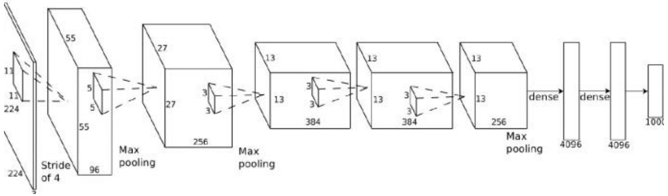
$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

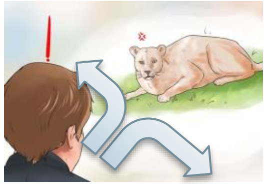
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy



Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Terminology & notation

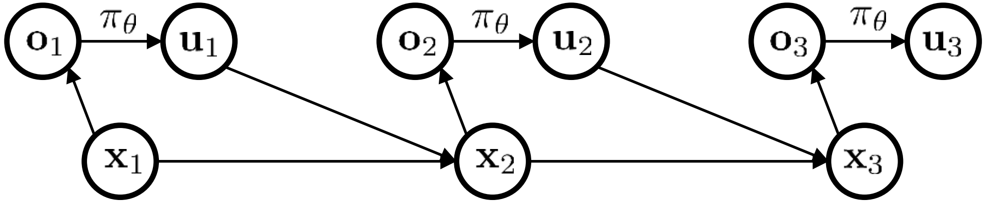




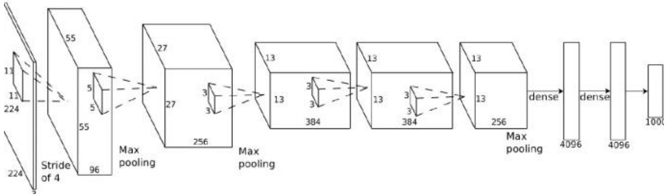
$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

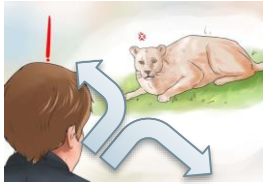
$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy



Terminology & notation

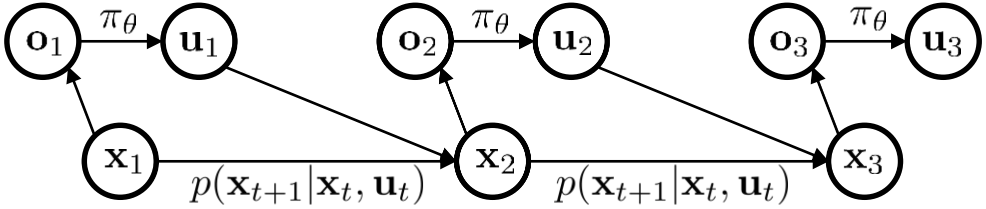




$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

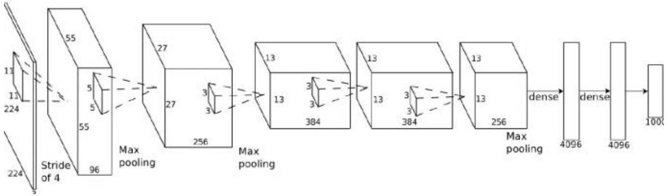
$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

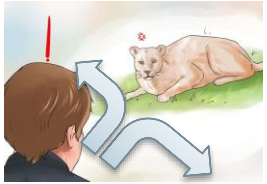
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy



Slide from “Deep Reinforcement Learning” CS294, S. Levine 2017

Terminology & notation

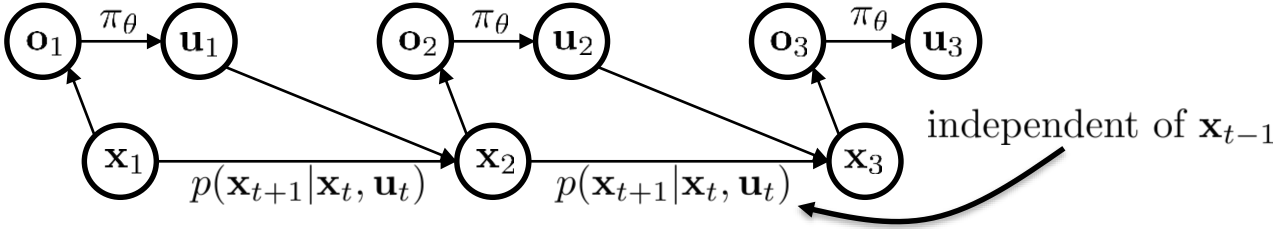




$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

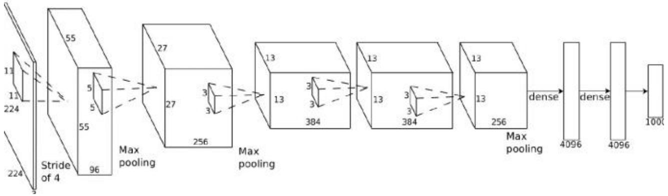
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy



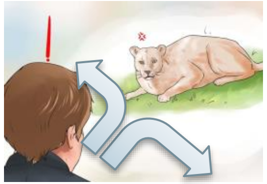
Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Terminology & notation

$\mathbf{o}_t$



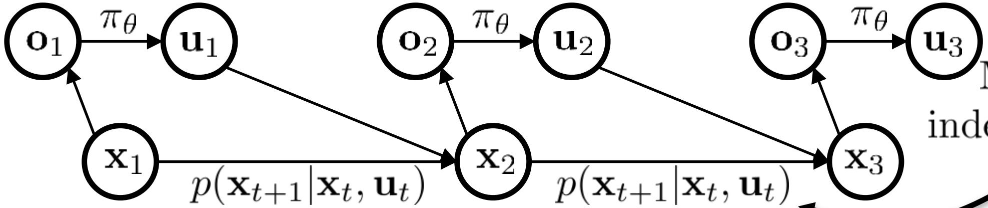
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$



$\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

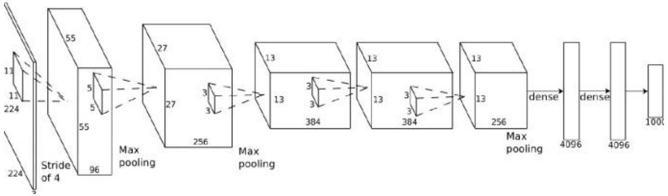


Markov property
independent of $\mathbf{x}_{t-1}$

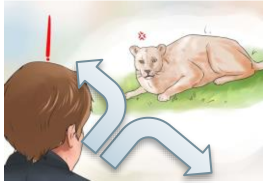
Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Terminology & notation

$\mathbf{o}_t$



$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$



$\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

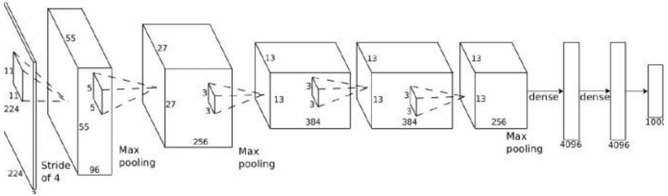
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

a bit of history…

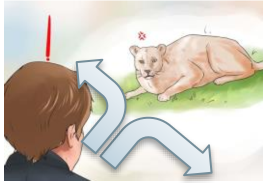
$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action

Terminology & notation

$\mathbf{o}_t$



$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$



$\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

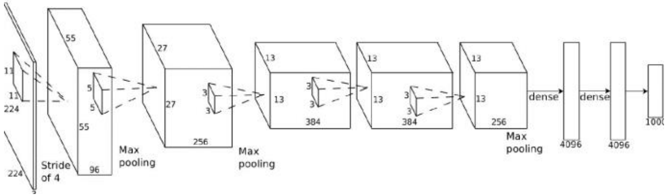
a bit of history…

$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action

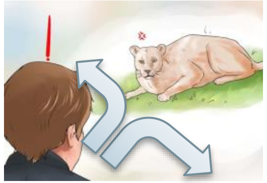
$\mathbf{s}_t$ – state
$\mathbf{a}_t$ – action

Terminology & notation

$\mathbf{o}_t$



$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$



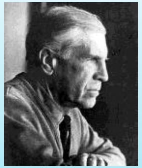
$\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

a bit of history…

$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action

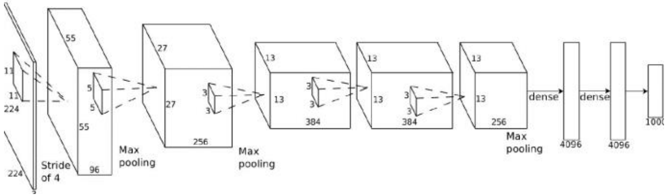


Lev Pontryagin

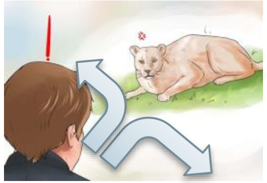
$\mathbf{s}_t$ – state
$\mathbf{a}_t$ – action

Terminology & notation

$\mathbf{o}_t$



$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$



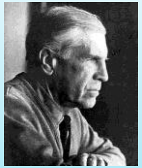
$\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

a bit of history…

$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action

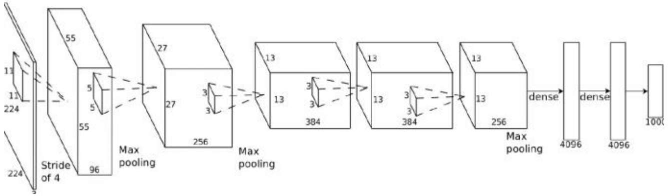


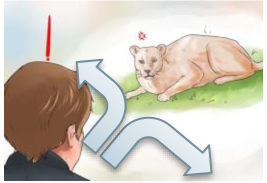
Lev Pontryagin

$\mathbf{s}_t$ – state
$\mathbf{a}_t$ – action

Richard Bellman

Terminology & notation





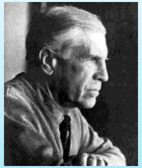
$\mathbf{o}_t$ $\quad$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\quad$ $\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

a bit of history…

$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action
управление

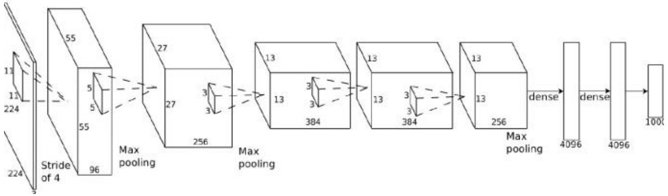


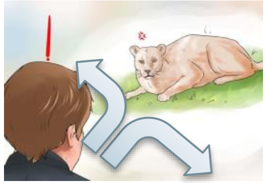
Lev Pontryagin

$\mathbf{s}_t$ – state
$\mathbf{a}_t$ – action

Richard Bellman

Terminology & notation





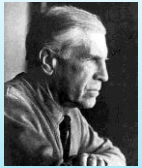
$\mathbf{o}_t$ $\qquad$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\qquad$ $\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

a bit of history…

$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action
управление



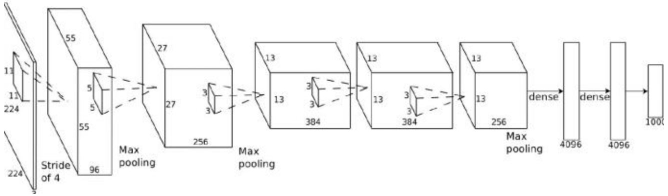
Lev Pontryagin

$\mathbf{s}_t$ – state
$\mathbf{a}_t$ – action

Richard Bellman

Imitation Learning

$\mathbf{o}_t$

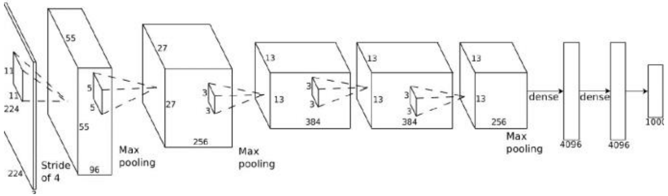


$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

$\mathbf{u}_t$

Imitation Learning

$\mathbf{o}_t$



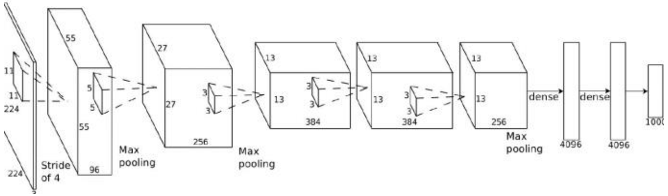
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

$\mathbf{u}_t$

Images: Bojarski et al. ‘16, NVIDIA

Imitation Learning

$\mathbf{o}_t$



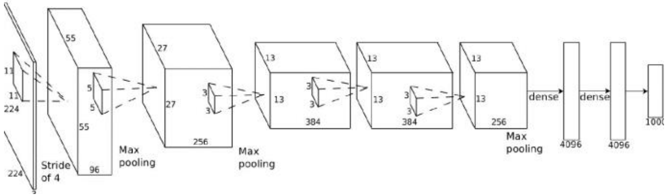
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

$\mathbf{u}_t$

Images: Bojarski et al. ‘16, NVIDIA

Imitation Learning

$\mathbf{o}_t$



$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

$\mathbf{u}_t$

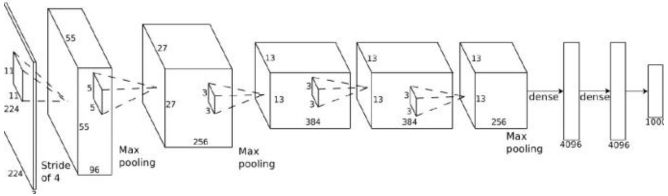
$\mathbf{o}_t$

$\mathbf{u}_t$

Images: Bojarski et al. ‘16, NVIDIA

Imitation Learning

$\mathbf{o}_t$

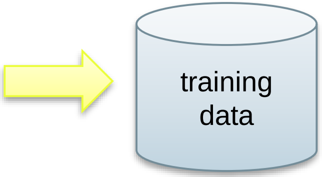


$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

$\mathbf{u}_t$

$\mathbf{o}_t$

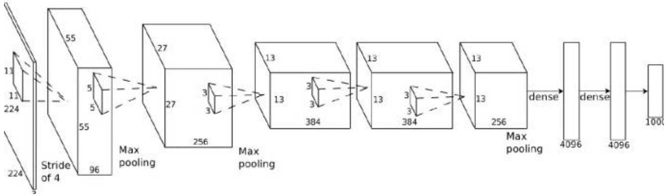
$\mathbf{u}_t$



Images: Bojarski et al. '16, NVIDIA

Imitation Learning

$\mathbf{o}_t$



$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

$\mathbf{u}_t$

$\mathbf{o}_t$
$\mathbf{u}_t$

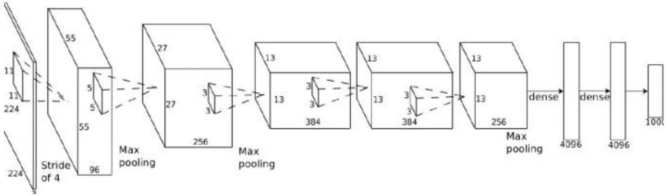
training data

supervised learning

Images: Bojarski et al. '16, NVIDIA

Imitation Learning

$\mathbf{o}_t$



$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

$\mathbf{u}_t$

$\mathbf{o}_t$

$\mathbf{u}_t$

→ training data → supervised learning → $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

Images: Bojarski et al. ‘16, NVIDIA

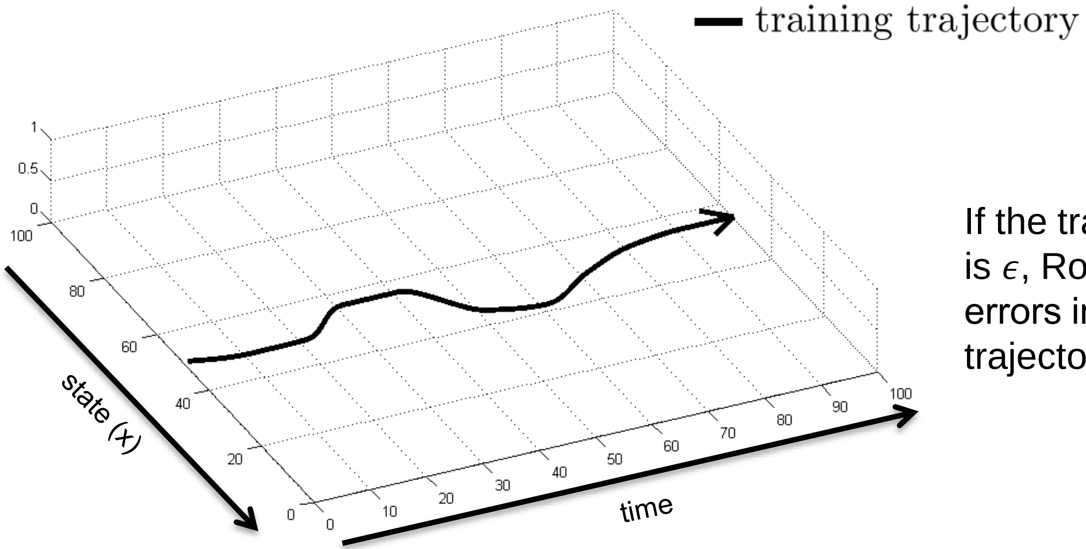
Does it work?

No!

If the training trajectory error is ϵ, Ross et al. show that errors in the learned model's trajectory can be order $O(T^2\epsilon)$

Does it work?

No!

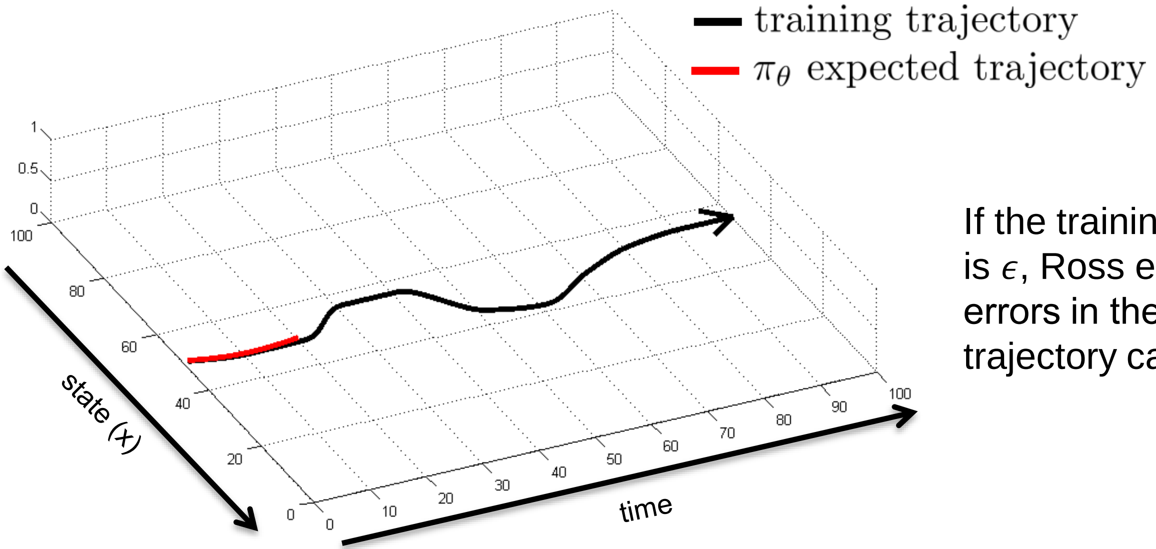


If the training trajectory error is ϵ, Ross et al. show that errors in the learned model's trajectory can be order $O(T^2\epsilon)$

Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Does it work?

No!

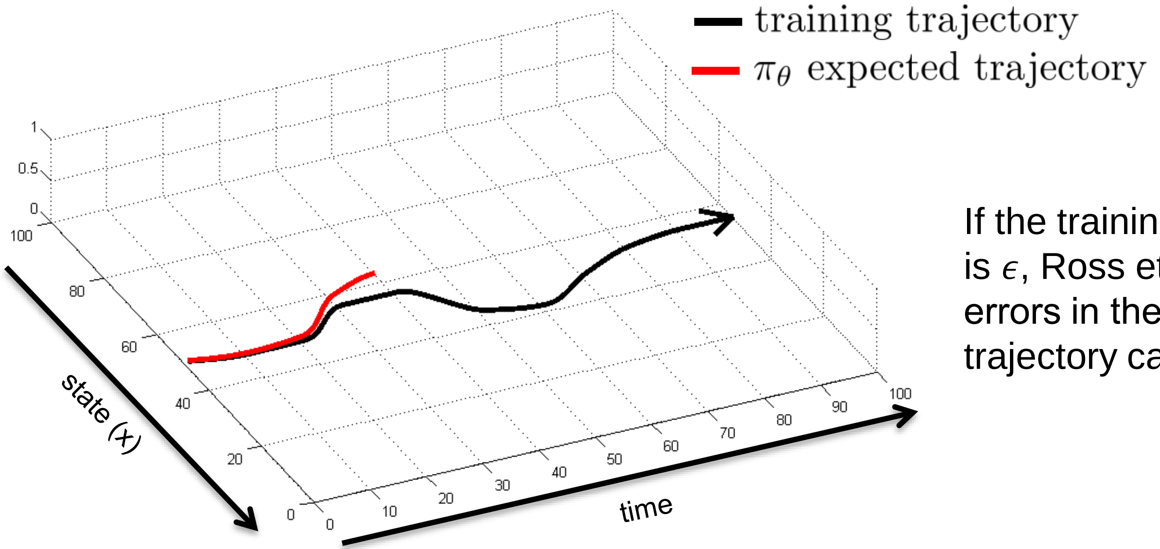


If the training trajectory error is ϵ, Ross et al. show that errors in the learned model's trajectory can be order $O(T^2\epsilon)$

Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Does it work?

No!

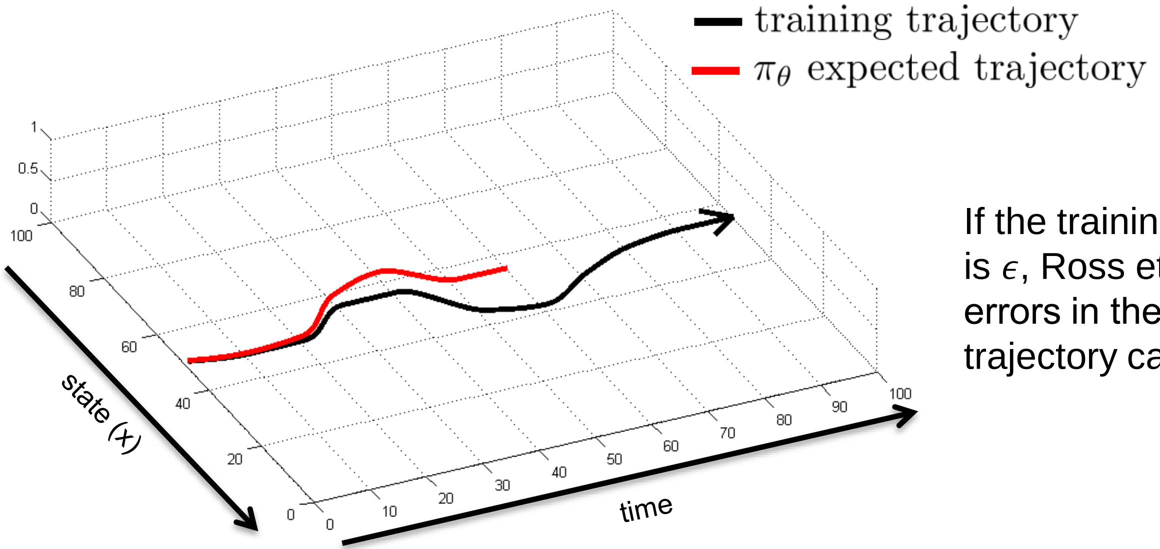


If the training trajectory error is ϵ, Ross et al. show that errors in the learned model's trajectory can be order $O(T^2\epsilon)$

Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Does it work?

No!

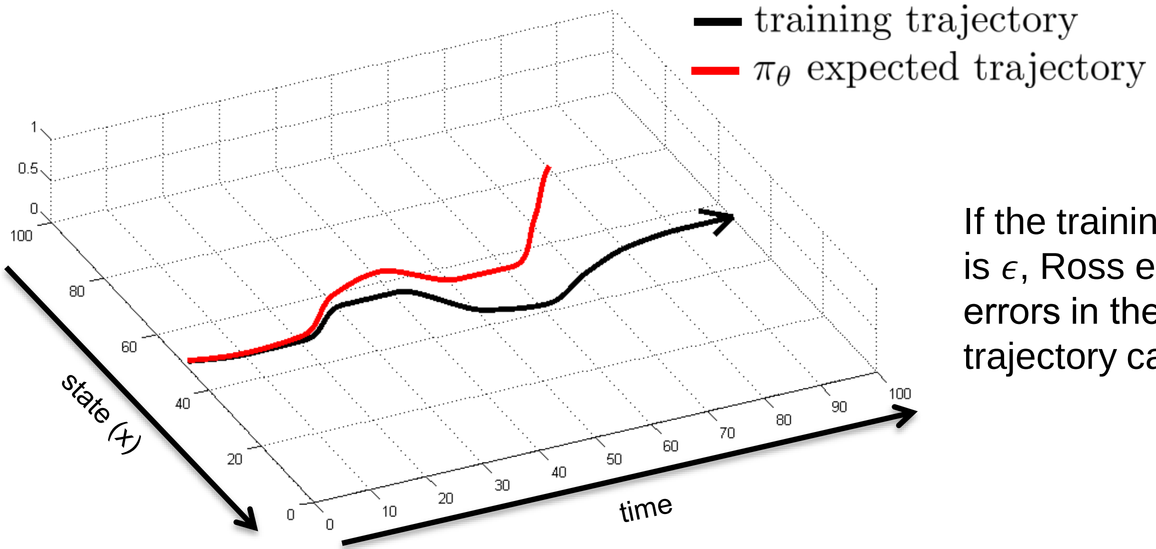


If the training trajectory error is ϵ, Ross et al. show that errors in the learned model's trajectory can be order $O(T^2\epsilon)$

Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Does it work?

No!

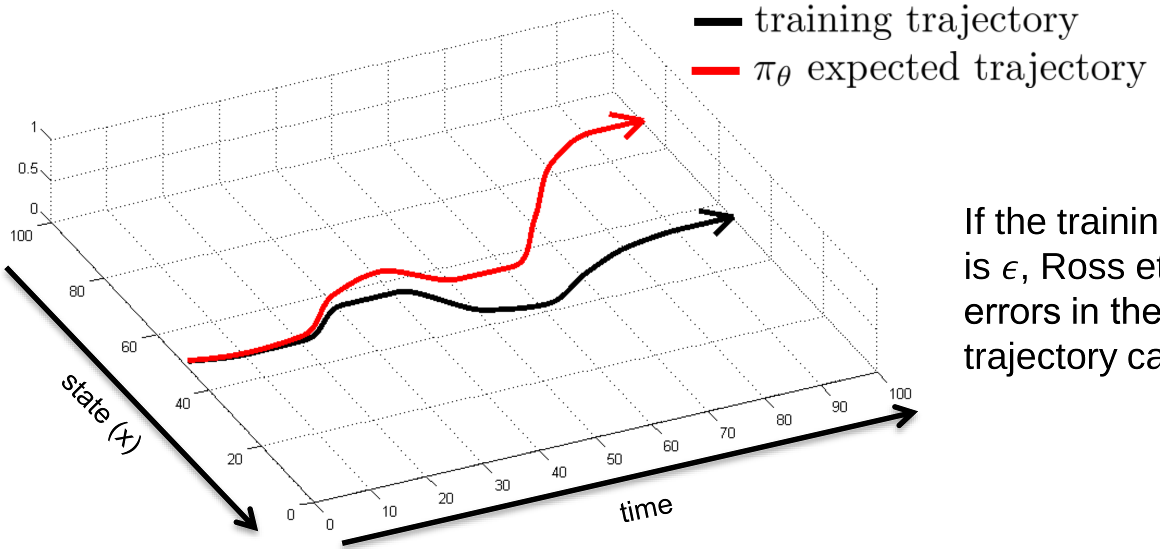


If the training trajectory error is ϵ, Ross et al. show that errors in the learned model's trajectory can be order $O(T^2\epsilon)$

Slide from "Deep Reinforcement Learning" CS294, S. Levine 2017

Does it work?

No!



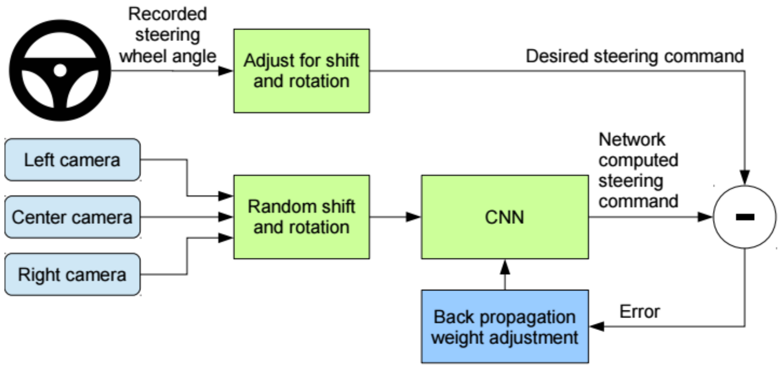
If the training trajectory error is ϵ, Ross et al. show that errors in the learned model's trajectory can be order $O(T^2\epsilon)$

Does it work?

Yes!

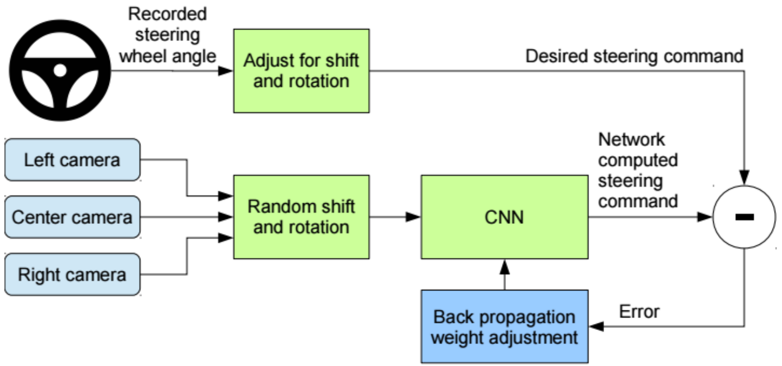
Video: Bojarski et al. ‘16, NVIDIA

Why did that work?



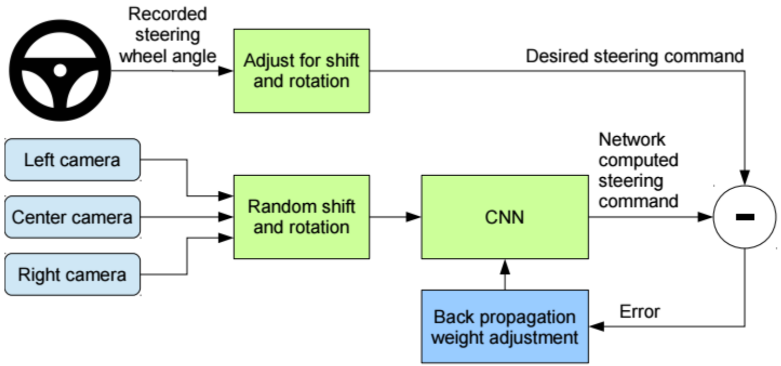
Bojarski et al. ‘16, NVIDIA

Why did that work?



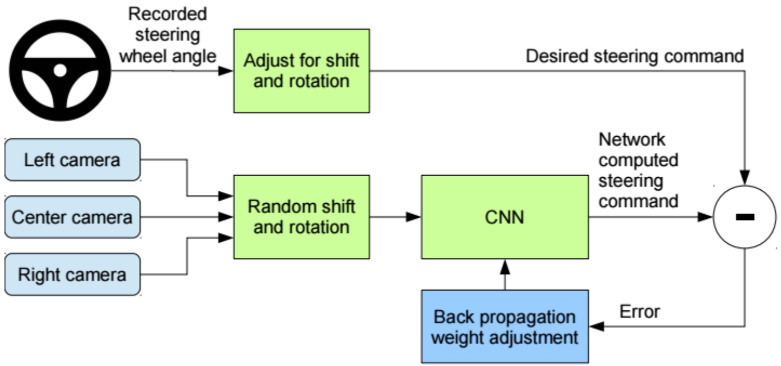
Bojarski et al. ‘16, NVIDIA

Why did that work?



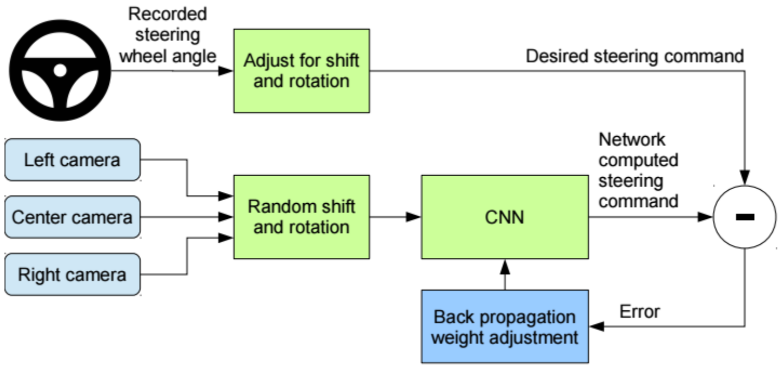
Bojarski et al. ‘16, NVIDIA

Why did that work?



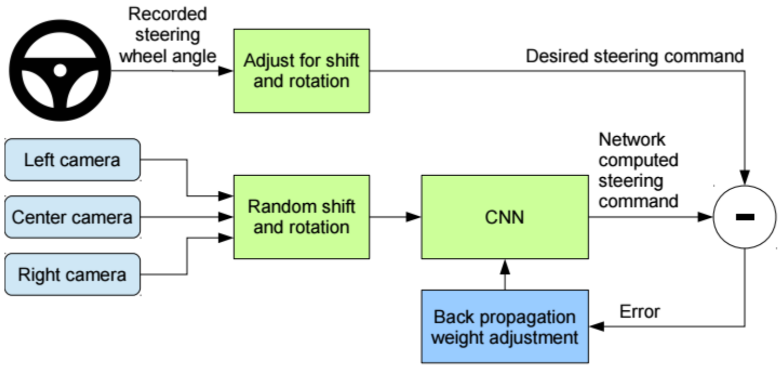
Bojarski et al. ‘16, NVIDIA

Why did that work?



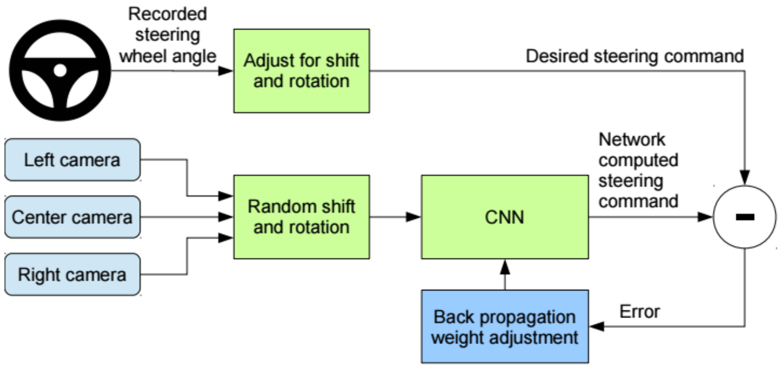
Bojarski et al. ‘16, NVIDIA

Why did that work?



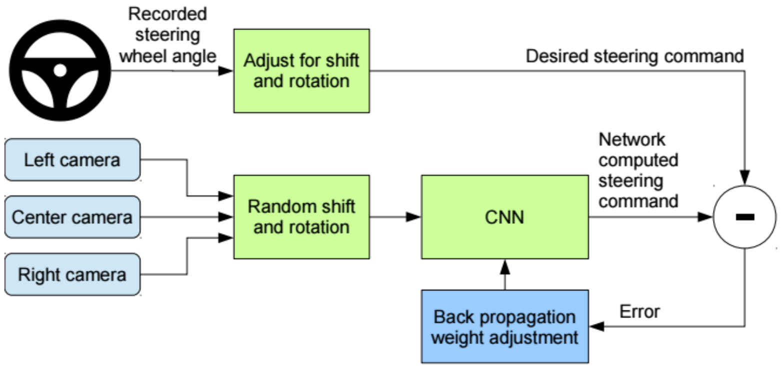
Bojarski et al. ‘16, NVIDIA

Why did that work?



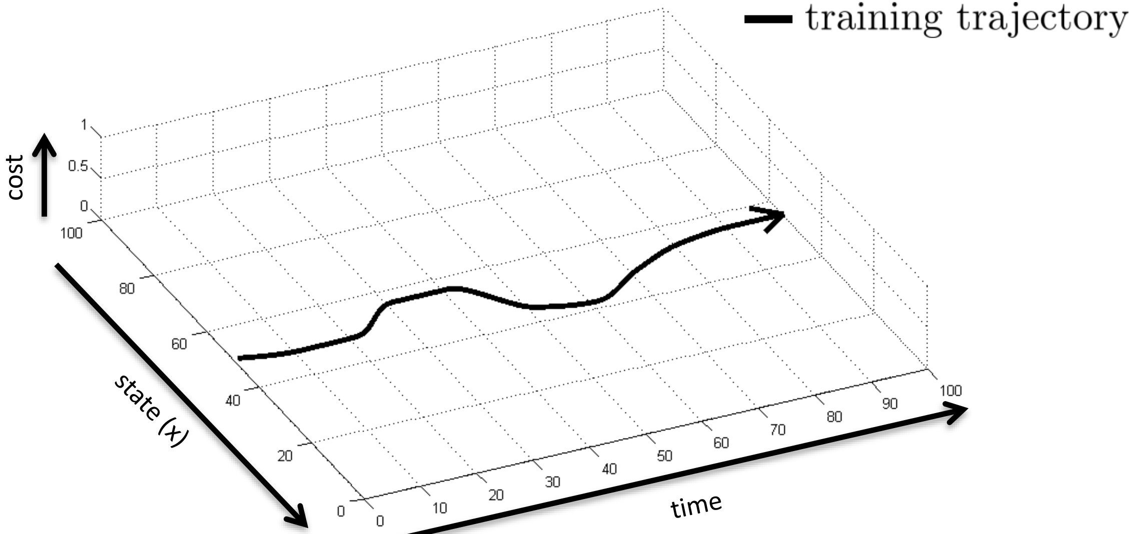
Bojarski et al. ‘16, NVIDIA

Why did that work?

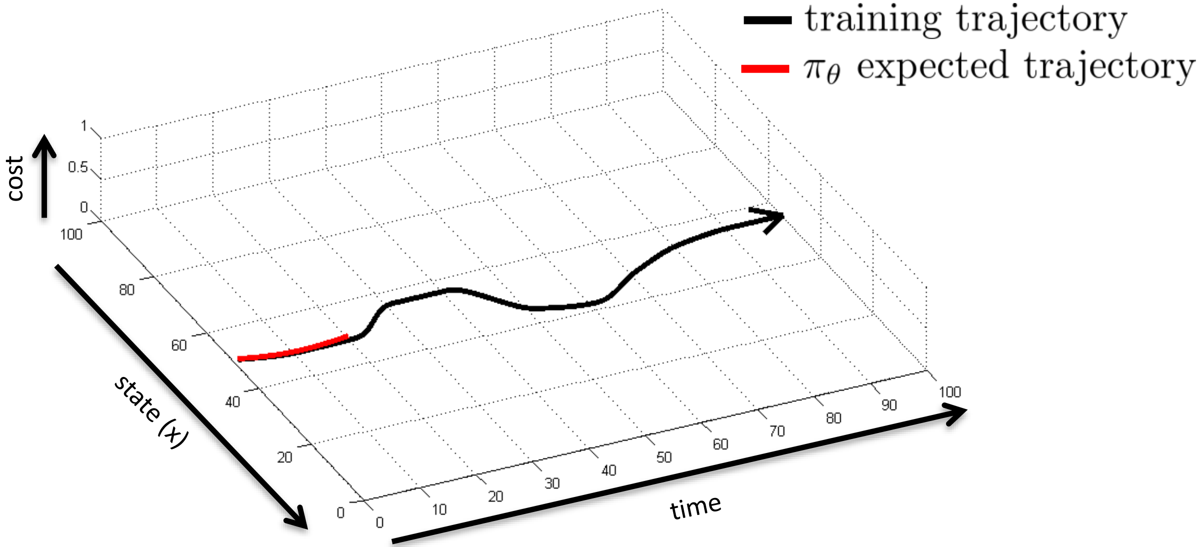


Bojarski et al. '16, NVIDIA

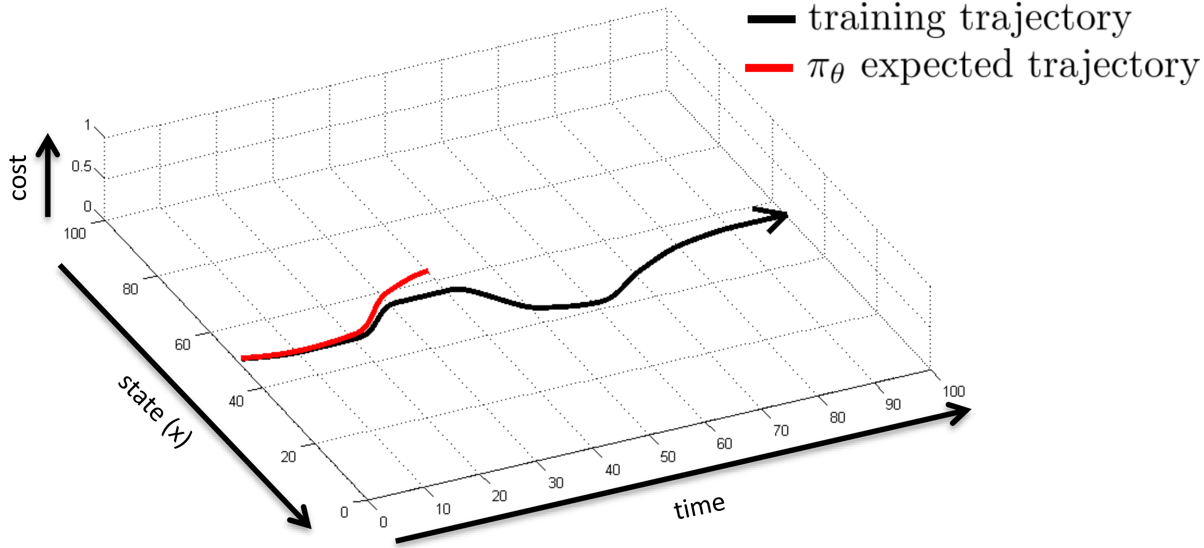
Can we make it work more often?



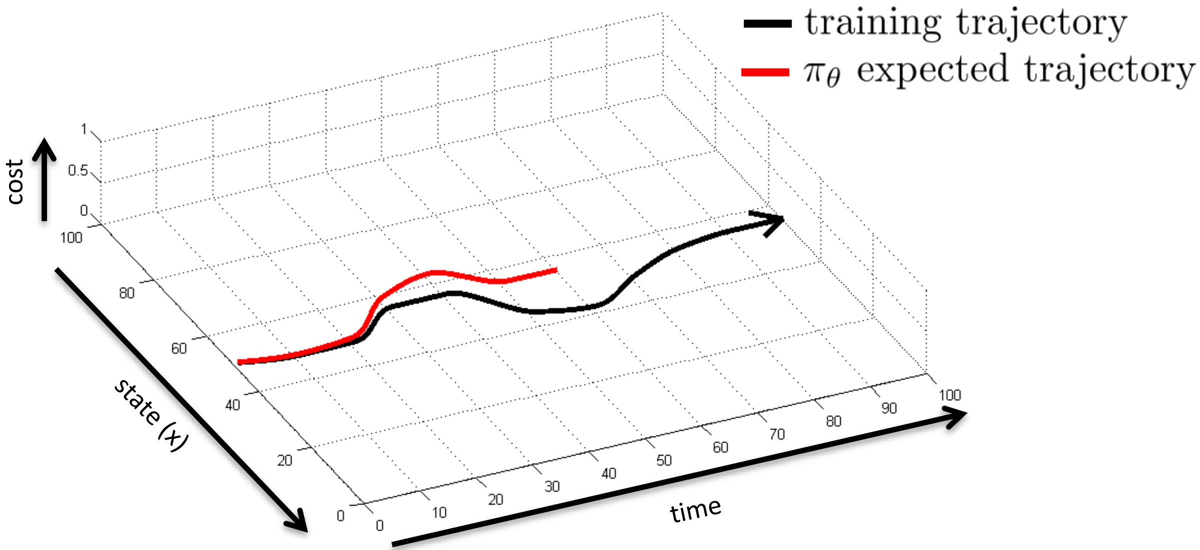
Can we make it work more often?



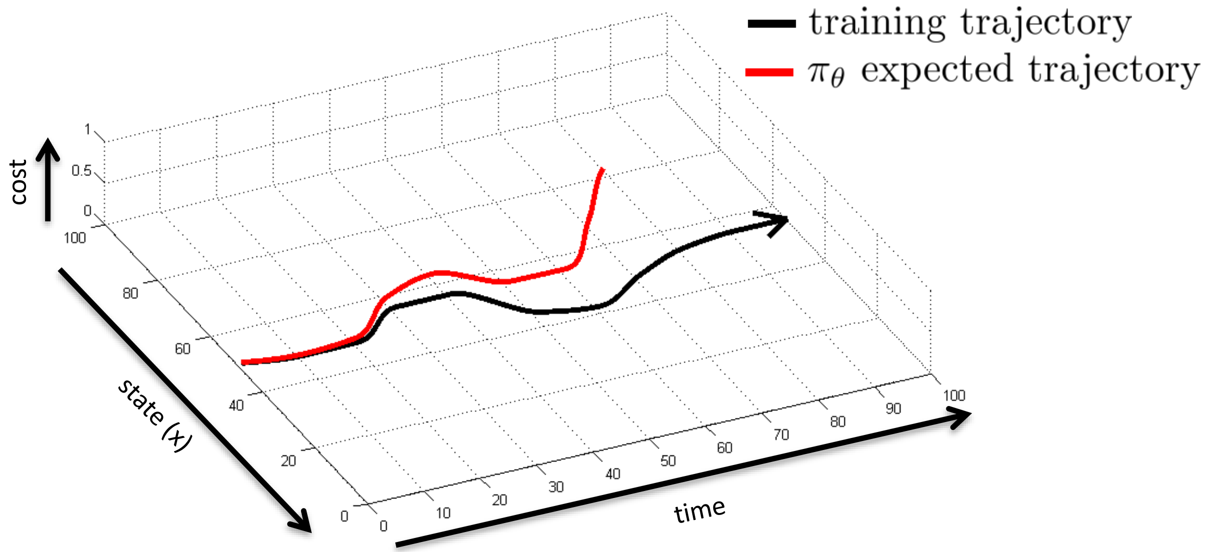
Can we make it work more often?



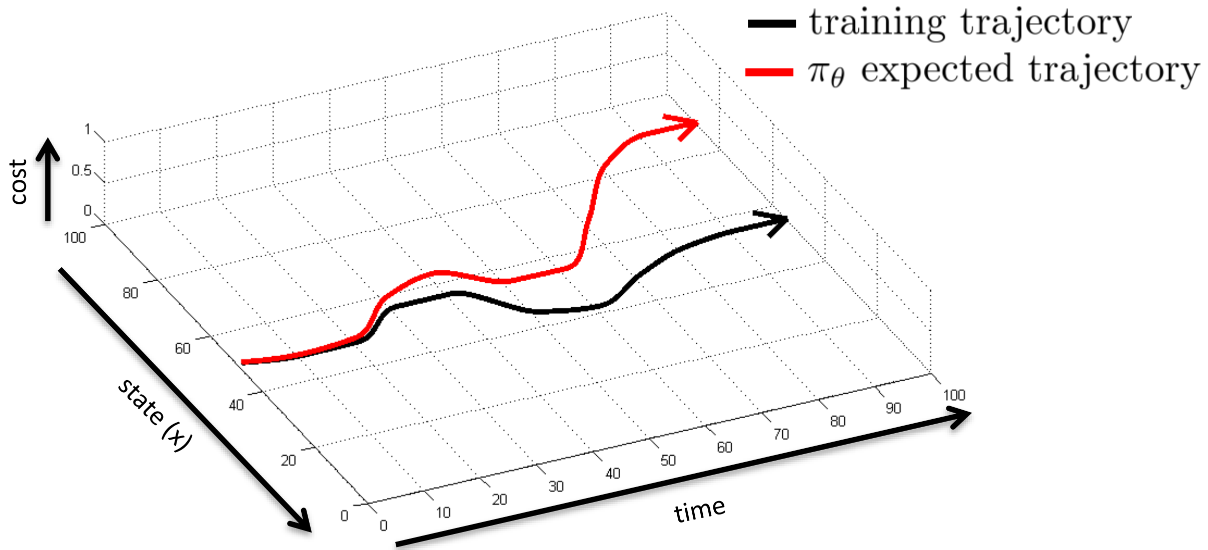
Can we make it work more often?



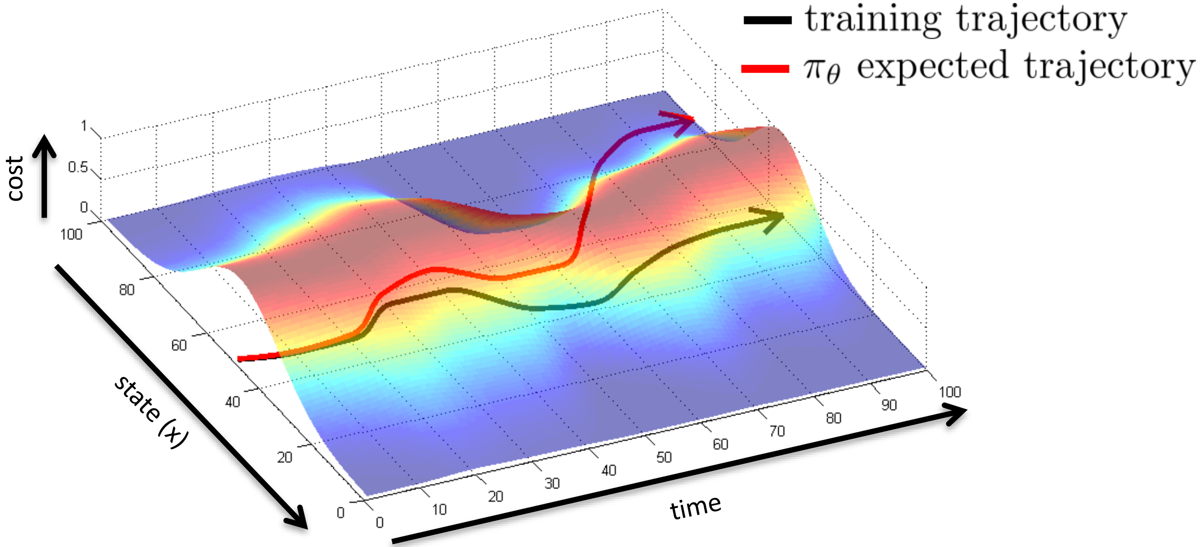
Can we make it work more often?



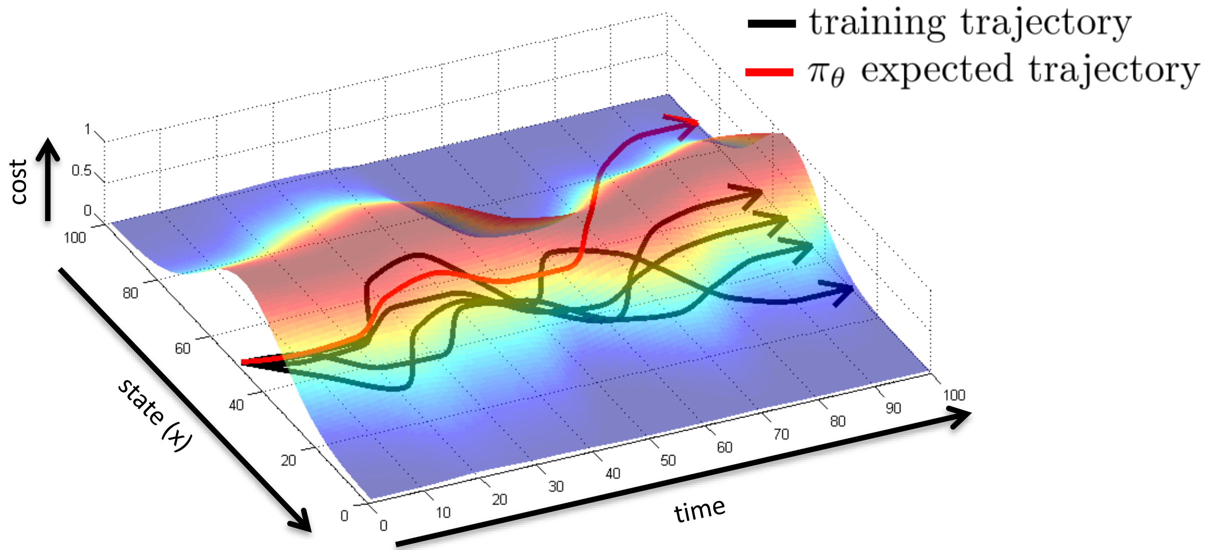
Can we make it work more often?



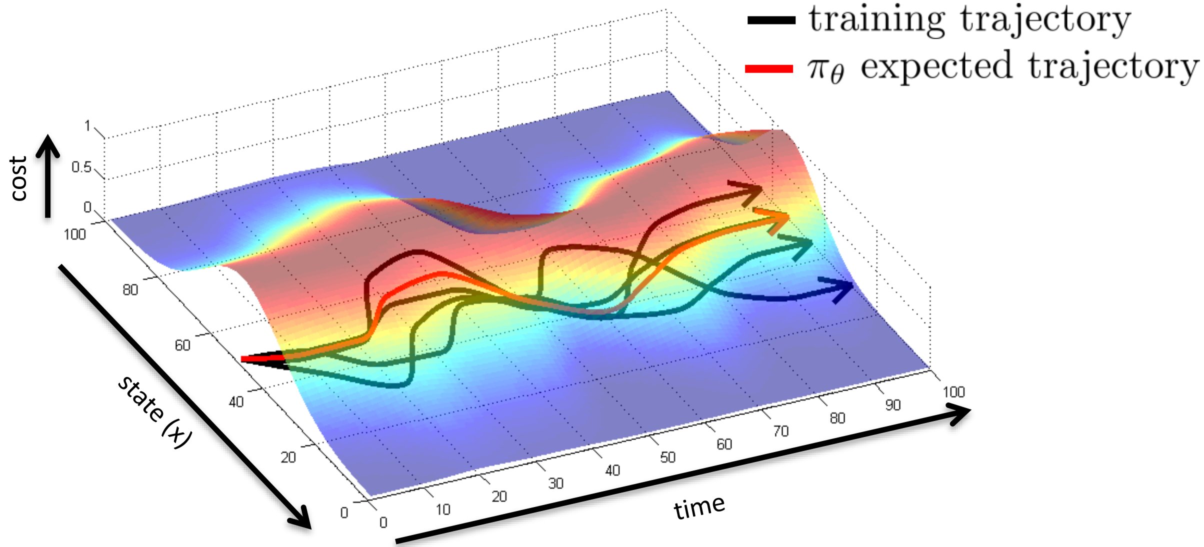
Can we make it work more often?



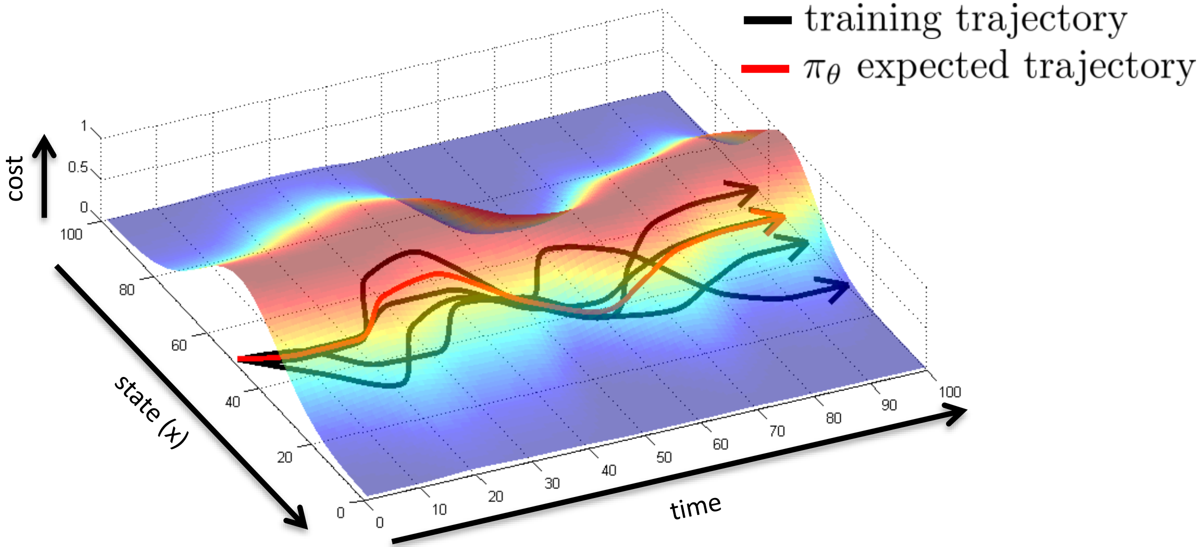
Can we make it work more often?



Can we make it work more often?

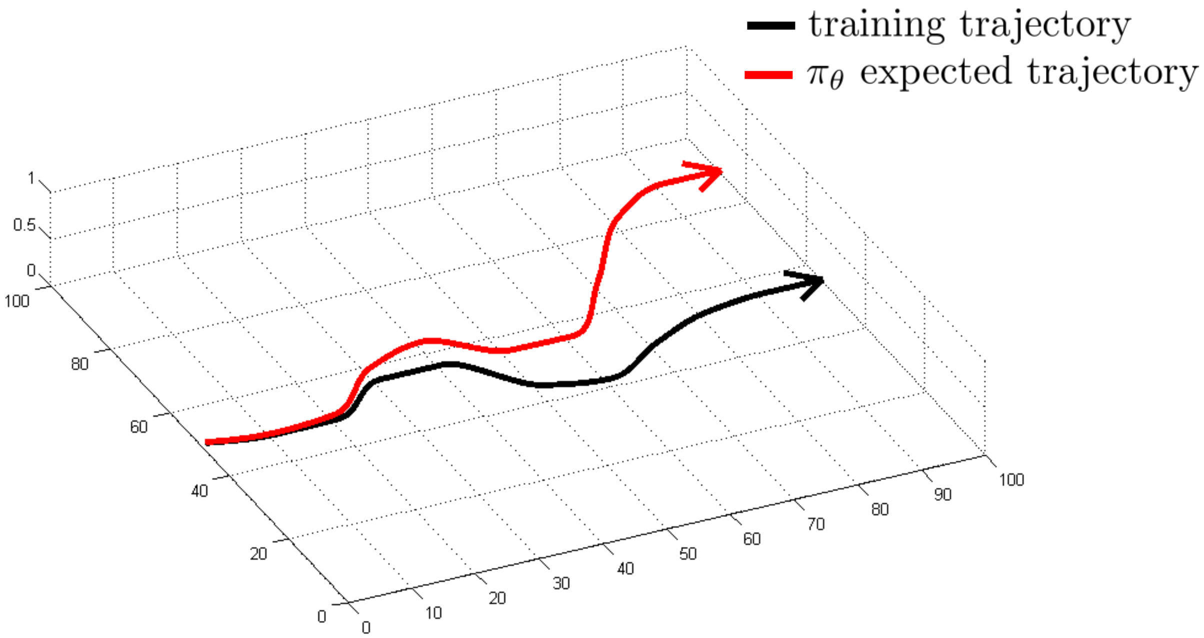


Can we make it work more often?

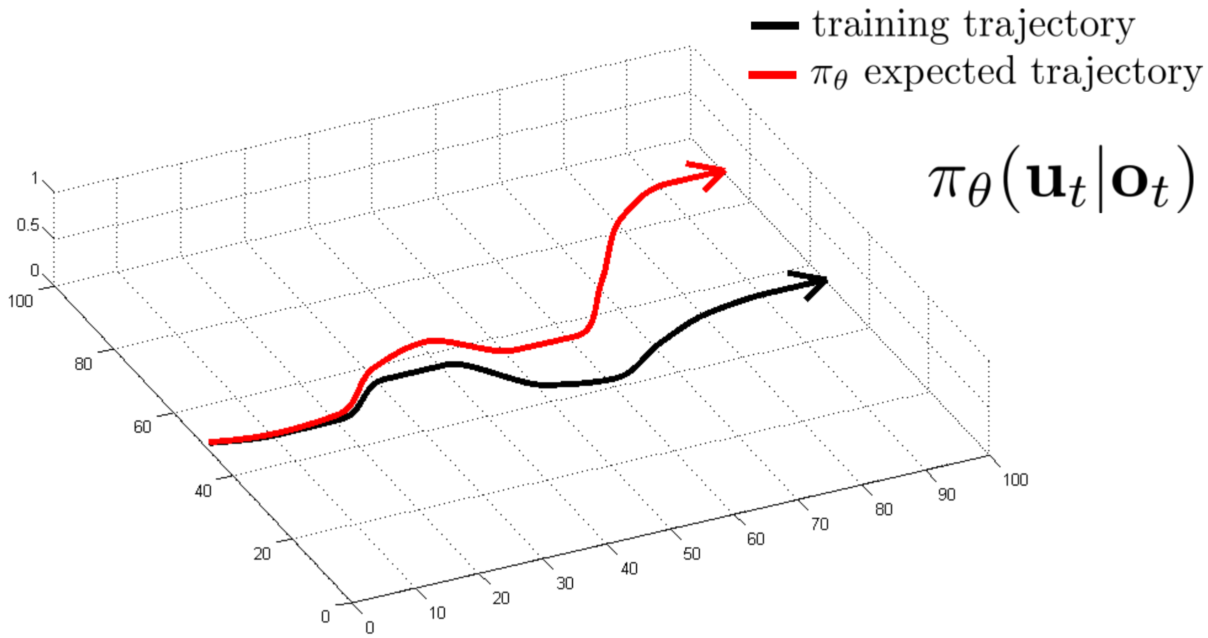


stability

Can we make it work more often?

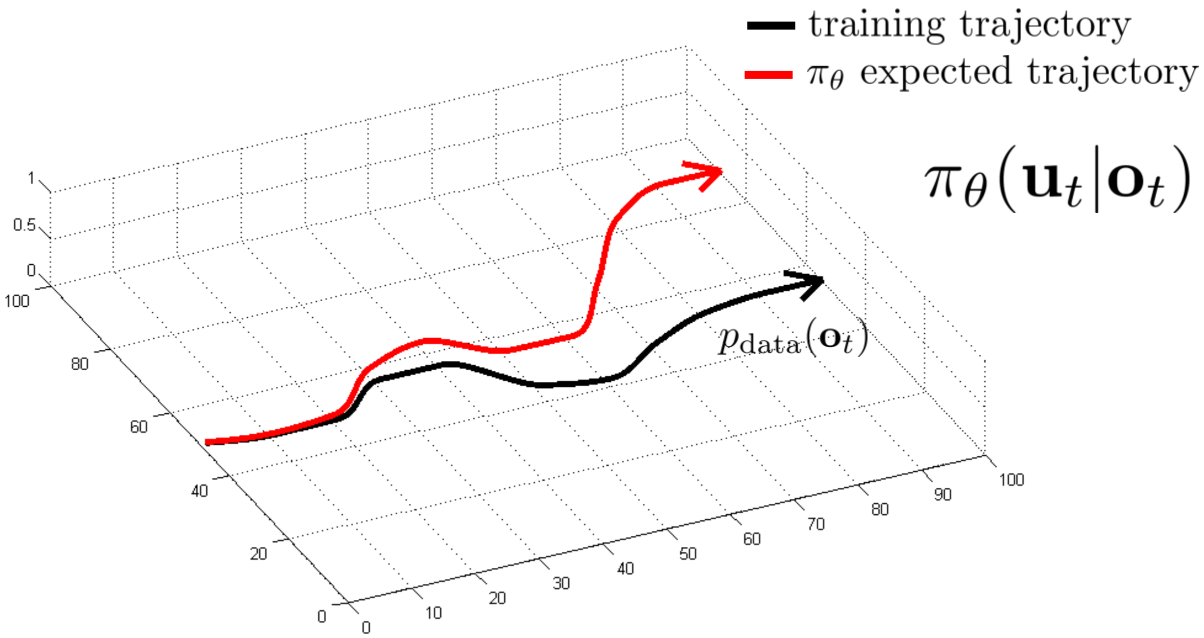


Can we make it work more often?

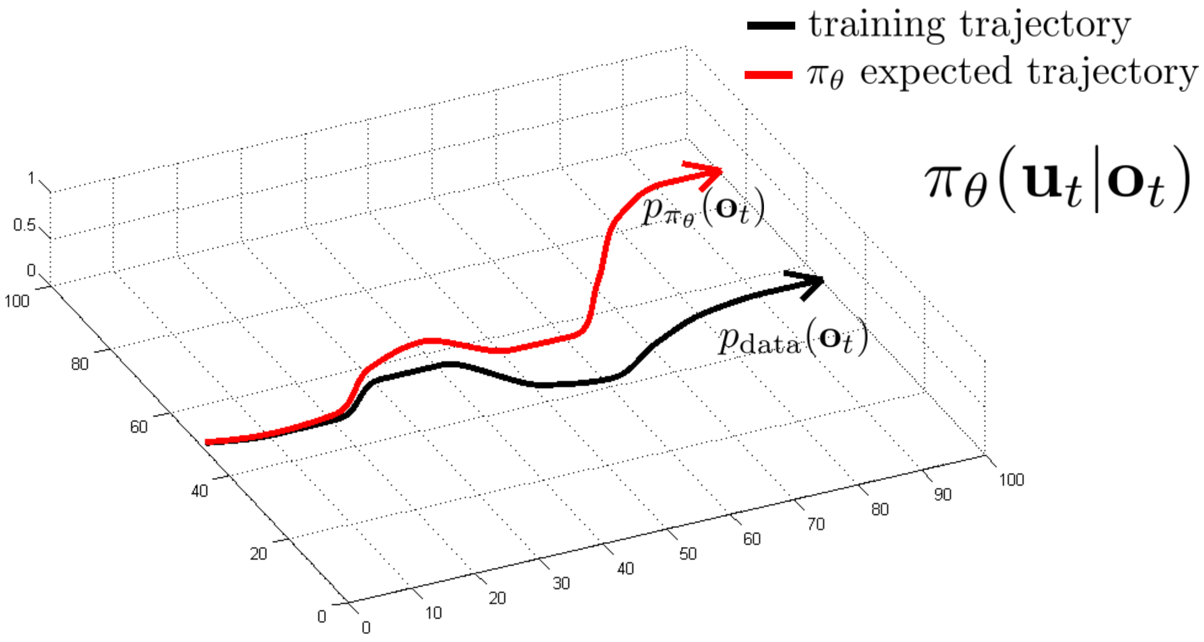


$$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$$

Can we make it work more often?

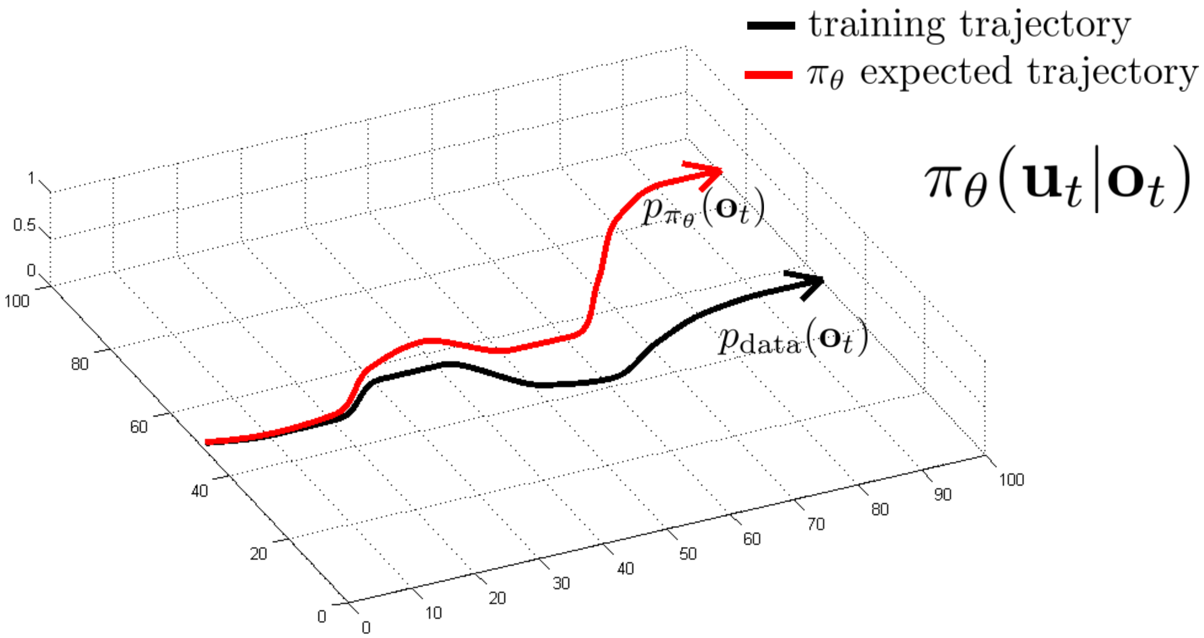


Can we make it work more often?



$$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$$

Can we make it work more often?

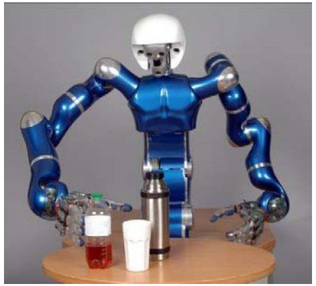


$$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$$

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

More Terminology

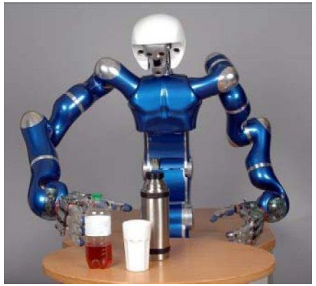
Behavior Policy: The policy $\pi_\theta(u|o)$ that the agent uses to act in the world.



Target Policy: A policy $\pi^t_{\theta^t}(u|o)$ the agent is learning.

More Terminology

On Policy: Agent learns from its own experience, so target policy = behavior policy.



Off Policy: Target policy $\neq$ behavior policy. More general. Can use experience from other agents

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

Current policy

$\pi_\theta(u_t|o_t)$

“Expert” actions

u_t

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

Current policy

$\pi_\theta(u_t|o_t)$

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

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

Current policy

$\pi_\theta(u_t|o_t)$

Ross et al. ‘11

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

DAgger: **D**ataset **Agg**regation

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

Current policy

$\pi_\theta(u_t|o_t)$

Ross et al. '11

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

DAgger: **D**ataset **Agg**regation

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

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

Current policy

$\pi_\theta(u_t|o_t)$

Ross et al. ‘11

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

DAgger: **D**ataset **Agg**regation

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

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

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

Current policy
$\pi_\theta(u_t|o_t)$

Ross et al. ‘11

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

DAgger: **D**ataset **Agg**regation

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

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

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

1. train $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{u}_1, \ldots, \mathbf{o}_N, \mathbf{u}_N\}$

Current policy $\pi_\theta(u_t|o_t)$

“Expert” actions u_t

Ross et al. ‘11

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

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 $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

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

1. train $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{u}_1, \ldots, \mathbf{o}_N, \mathbf{u}_N\}$
2. run $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$

Current policy

$\pi_\theta(u_t|o_t)$

“Expert” actions u_t

Ross et al. ‘11

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

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 $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

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

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

Current policy

$\pi_\theta(u_t|o_t)$

"Expert" actions u_t

Ross et al. '11

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

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 $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

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

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

Current policy $\pi_\theta(u_t|o_t)$

“Expert” actions u_t

Ross et al. ‘11

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

DAgger: **D**ataset **Agg**regation

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

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

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

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

Current policy
$\pi_\theta(u_t|o_t)$

Ross et al. '11

DAgger Example

Ross et al. '11

What’s the problem?

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

Ross et al. ‘11

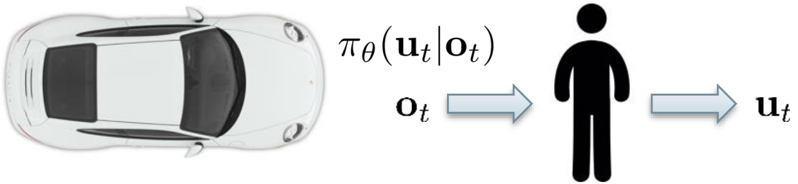
What's the problem?

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

Ross et al. '11

What's the problem?

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

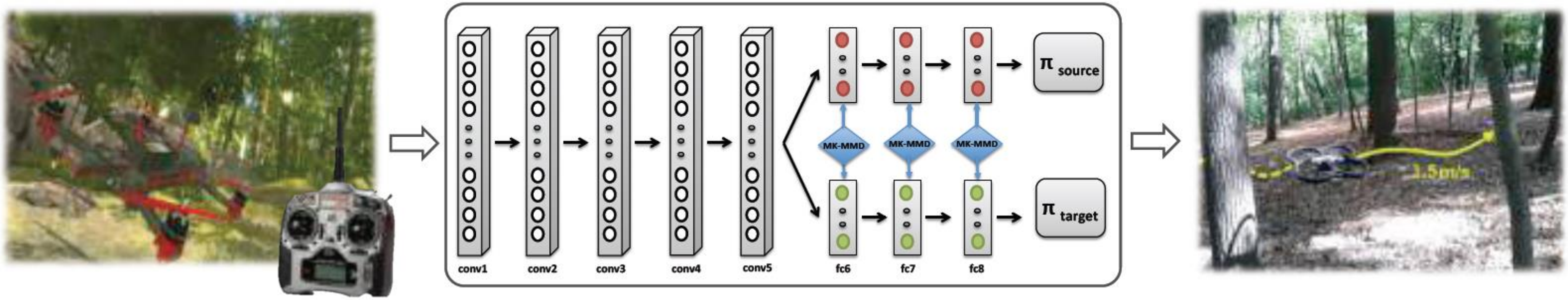


Ross et al. '11

Domain Adaptation: Learning Reactive Controls for an MAV

Challenge: It's often much easier to get human training data in an environment different from the target environment (e.g. in simulation).

Developing a controller for the target domain after training in a different domain is a domain adaptation challenge.



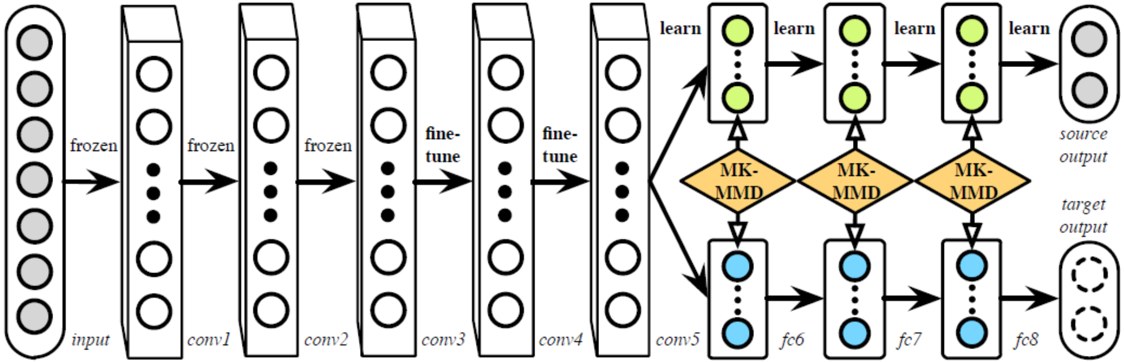
Training Domain

Target Domain

Shreyansh Daftry, J. Andrew Bagnell, and Martial Hebert, "Learning Transferable Policies for Monocular Reactive MAV Control" 2016

Domain Adaptation: Learning Reactive Controls for an MAV

The domain adaptation network shares early layers, fine-tunes last CNN layers, and replicates FC layers:



Hidden layer activations (layers 6 – 8)

It minimizes a composite loss:

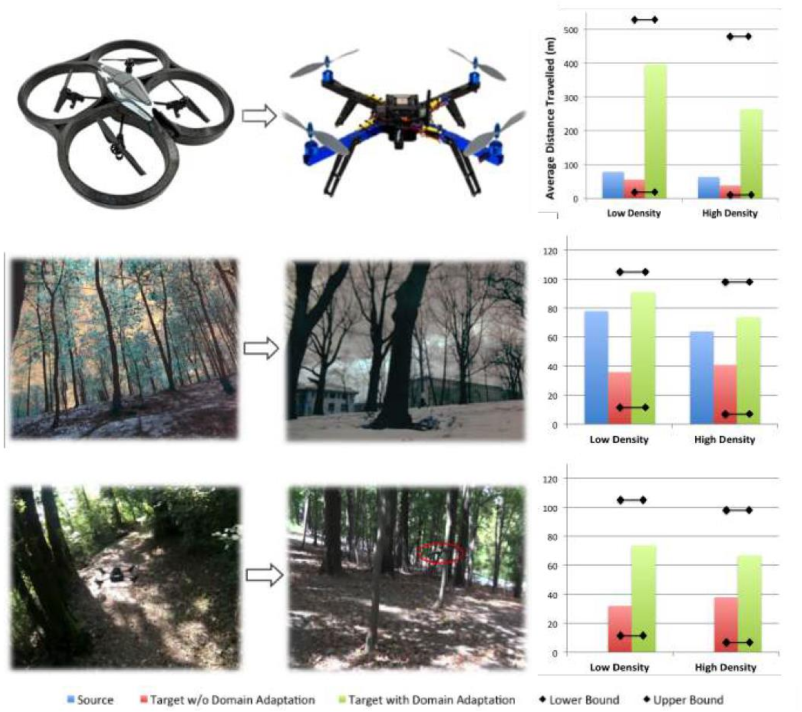
$$\min_{\Theta} \underbrace{\frac{1}{n_s} \sum_{i=1}^{n_s} J(\theta(x_i^s), y_i^s)}_{\text{Label loss}} + \lambda \underbrace{\sum_{l=l_1}^{l_2} d_k^2(\mathcal{D}_s^l, \mathcal{D}_t^l)}_{\text{Distribution loss}}$$

Train on labeled source data, and unlabeled target data.

Daftry et al. 2016

Domain Adaptation: Learning Reactive Controls for an MAV

Examples:



Experiments and Results for (Row-1) Transfer across physical systems from ARDrone to ArduCopter, (Row-2) Transfer across weather conditions from summer to winter and (Row-3) Transfer across environments from Univ. of Zurich to CMU.

Daftry et al. 2016

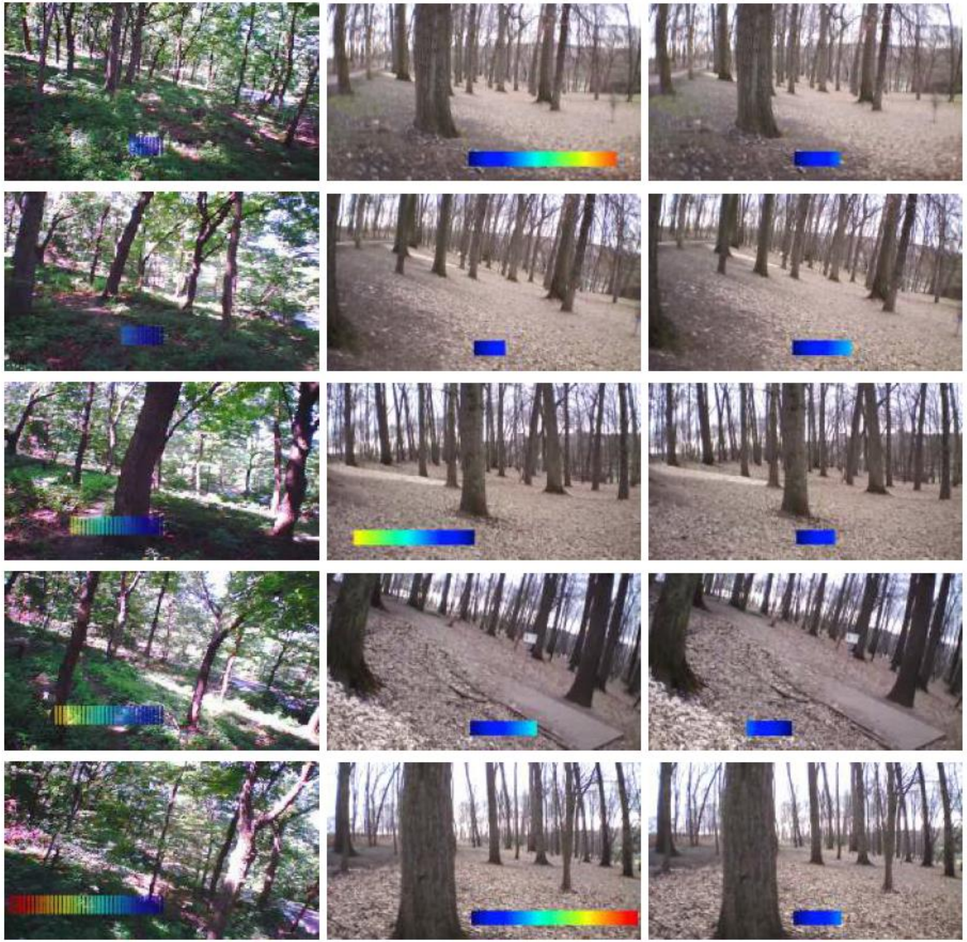
Reactive MAV Controls

Qualitative visualization of an example flight in dense forest.

The training data was collected from the same environment during summer season (Col-1) and tested during the winter season (Col-2).

The image sequence of MAVs on-board view is chronologically ordered from top to bottom and overlaid with color-coded commands issued by the policy learned using our proposed approach.

Additionally, we also compute the commands that would have been generated by the policy without domain adaptation (Col-3), for qualitative comparison.

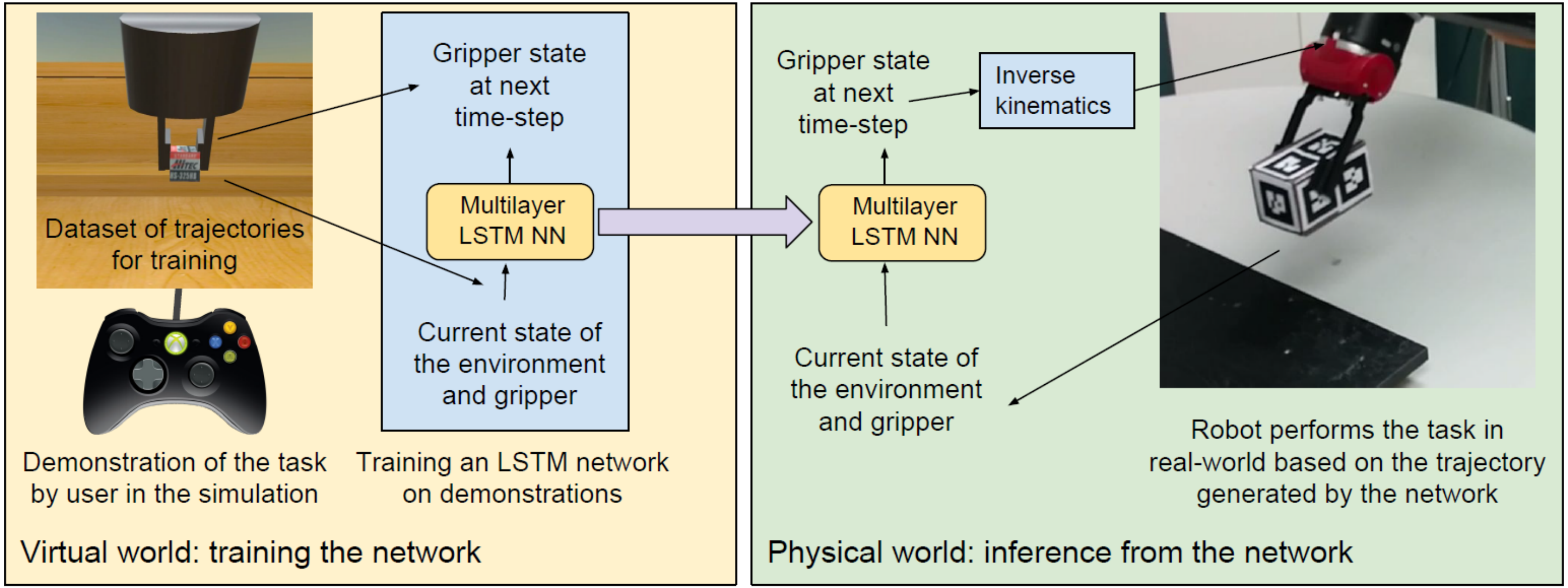


Daftry et al. 2016

Domain Adaptation: Learning Reactive Controls for an MAV

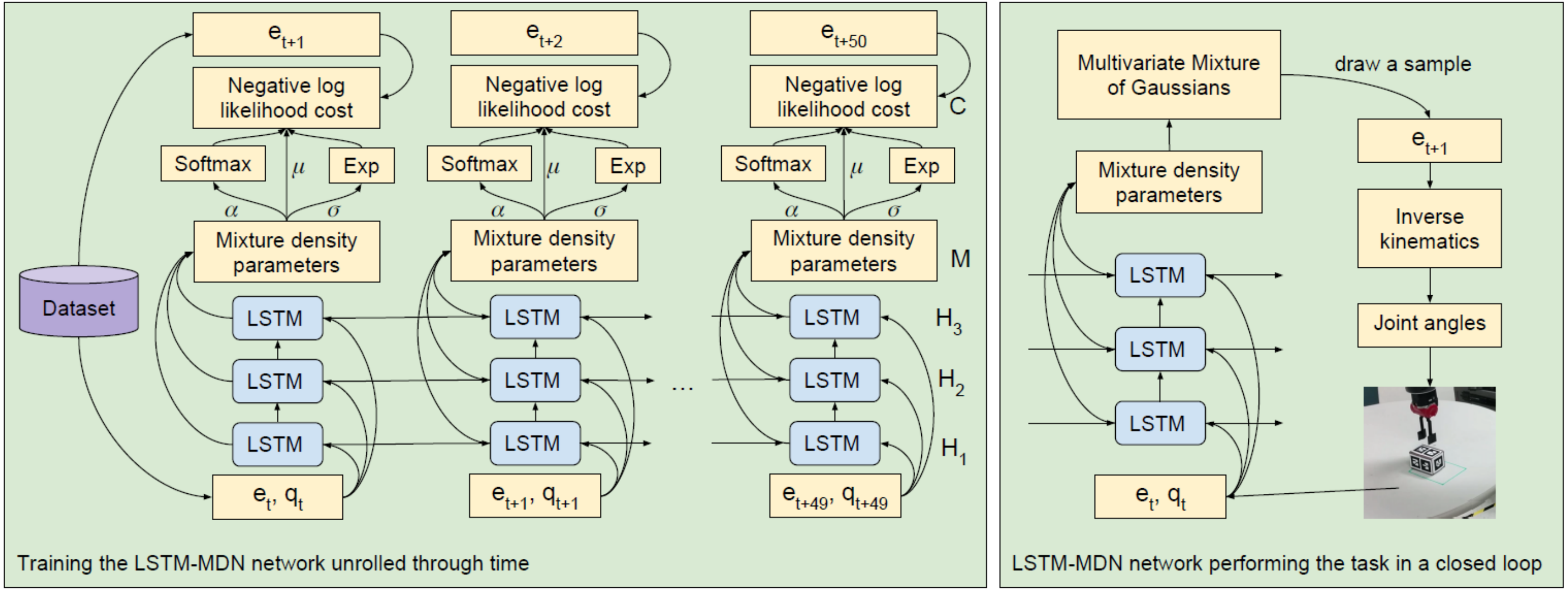
Video: https://www.youtube.com/watch?v=Jvx0DWxTXAE

Error Recovery: Learning ADL tasks with an LSTM



Rouhollah Rahmatizadeh, Pooya Abolghasemi, Aman Behal, Ladislau Boloni, "From Virtual Demonstration to Real-World Manipulation Using LSTM and MDN" 2016

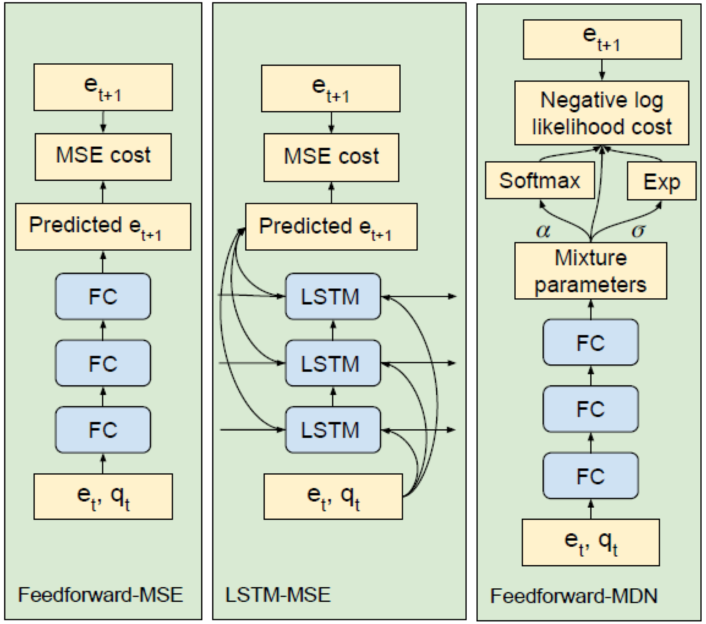
Error Recovery: Learning ADL tasks with an LSTM



Rahmatizadeh et al 2016

Error Recovery: Learning ADL tasks with an LSTM

Experiments: Comparison with baseline models (success rate):



Controller	Pick and place	Push to pose
Feedfoward-MSE	0%	0%
LSTM-MSE	85%	0%
Feedforward-MDN	95%	15%
LSTM-MDN	**100%**	**95%**

Environment	Pick and place	Push to pose
Virtual world	100%	95%
Physical world	80%	60%

Rahmatizadeh et al 2016

Error Recovery: Learning ADL tasks with an LSTM

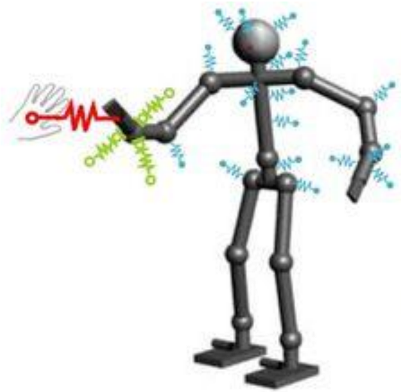
Video: https://youtu.be/9vYlIG2ozaM

Rahmatizadeh et al 2016

GAIL: Generative Adversarial Imitation Learning

Rather than trying to mimic the user blindly, try to solve the same control problem that the user is solving. i.e. estimate the user's cost function, and then optimize the cost by training.

This is called Inverse Reinforcement Learning (IRL).



Jonathan Ho and Stefano Ermon, "Generative Adversarial Imitation Learning"

GAIL: Generative Adversarial Imitation Learning

Inverse Reinforcement Learning (IRL) is under-constrained, and often uses regularization heuristics.

Entropy regularization: define $H(\pi) \triangleq \mathbb{E}_\pi[-\log \pi(a|s)]$

Estimating the cost function is: $\underset{c \in \mathcal{C}}{\text{maximize}} \left(\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)] \right) - \mathbb{E}_{\pi_E}[c(s,a)]$

Then the imitation learning problem is: $\mathrm{RL}(c) = \underset{\pi \in \Pi}{\arg\min} -H(\pi) + \mathbb{E}_\pi[c(s,a)]$

Jonathan Ho and Stefano Ermon, “Generative Adversarial Imitation Learning”

GAIL: Generative Adversarial Imitation Learning

Inverse Reinforcement Learning (IRL) is under-constrained, and often uses regularization heuristics.

Entropy regularization: define $H(\pi) \triangleq \mathbb{E}_{\pi}[-\log \pi(a|s)]$

Entropy is highest for a random policy (random actions at every step).

Entropy is lowest (0) for a deterministic policy that takes a single action at each step.

Jonathan Ho and Stefano Ermon, "Generative Adversarial Imitation Learning"

GAIL: Generative Adversarial Imitation Learning

Inverse Reinforcement Learning (IRL) is under-constrained, and often uses regularization heuristics.

Entropy regularization: define $H(\pi) \triangleq \mathbb{E}_{\pi}[-\log \pi(a|s)]$

Expert trajectory cost

Estimating the cost function is: $\underset{c \in \mathcal{C}}{\text{maximize}} \left(\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s,a)] \right) - \mathbb{E}_{\pi_E}[c(s,a)]$

Then the imitation learning problem is: $\mathrm{RL}(c) = \underset{\pi \in \Pi}{\arg\min} -H(\pi) + \mathbb{E}_{\pi}[c(s,a)]$

Jonathan Ho and Stefano Ermon, “Generative Adversarial Imitation Learning”

GAIL: Generative Adversarial Imitation Learning

Inverse Reinforcement Learning (IRL) is under-constrained, and often uses regularization heuristics.

Entropy regularization: define $H(\pi) \triangleq \mathbb{E}_\pi[-\log \pi(a|s)]$

Estimating the cost function is: $\underset{c \in \mathcal{C}}{\text{maximize}} \left(\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)] \right) - \mathbb{E}_{\pi_E}[c(s,a)]$

Learned policy cost

Then the imitation learning problem is: $\mathrm{RL}(c) = \arg\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)]$

Jonathan Ho and Stefano Ermon, "Generative Adversarial Imitation Learning"

GAIL: Generative Adversarial Imitation Learning

MaxEnt IRL looks for a cost function which assigns low cost to the expert policy, and high cost to other policies.

Estimating the cost function is: $$\underset{c\in\mathcal{C}}{\text{maximize}}\left(\min_{\pi\in\Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s,a)]\right) - \mathbb{E}_{\pi_E}[c(s,a)]$$

Jonathan Ho and Stefano Ermon, “Generative Adversarial Imitation Learning”

GAIL: Generative Adversarial Imitation Learning

GAIL then uses an adversary to discriminate the expert and learned policies by their state occupancy functions ρ_π and ρ_{π^E}. Don't worry about TRPO for now…

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \tag{17}$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

$$\begin{aligned} &\hat{\mathbb{E}}_{\tau_i}\left[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)\right] - \lambda \nabla_\theta H(\pi_\theta), \\ &\text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}] \end{aligned} \tag{18}$$

6: **end for**

Jonathan Ho and Stefano Ermon, "Generative Adversarial Imitation Learning"

GAIL: Generative Adversarial Imitation Learning

https://www.youtube.com/watch?v=0hw0GD3lkA8

Imitation learning: recap

$\mathbf{o}_t$

$\mathbf{u}_t$

training data

supervised learning

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

Imitation learning: recap

$\mathbf{o}_t$

$\mathbf{u}_t$

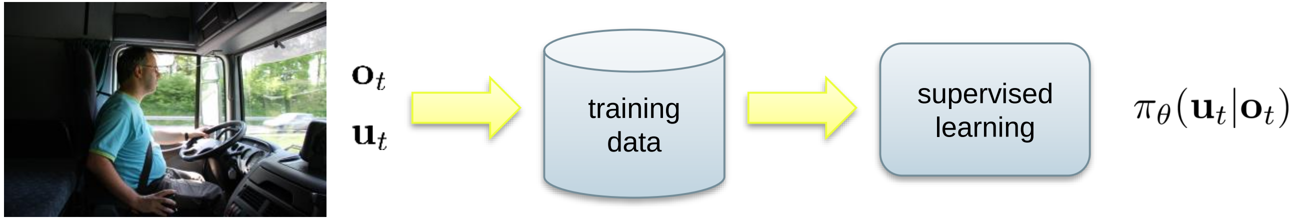
training data

supervised learning

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

Usually (but not always) insufficient by itself

Imitation learning: recap



Usually (but not always) insufficient by itself

Distribution mismatch problem

Imitation learning: recap

$\mathbf{o}_t$ $\mathbf{u}_t$ → training data → supervised learning → $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

Usually (but not always) insufficient by itself

Distribution mismatch problem

Sometimes works well

Imitation learning: recap

$\mathbf{o}_t$

$\mathbf{u}_t$

training data

supervised learning

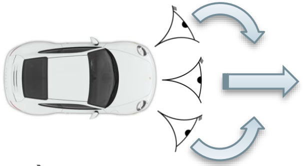
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

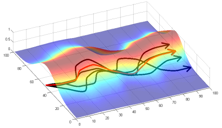
Usually (but not always) insufficient by itself

Distribution mismatch problem

Sometimes works well

Micro-models (e.g. image and control transformations)





Imitation learning: recap

$\mathbf{o}_t$

$\mathbf{u}_t$

training data

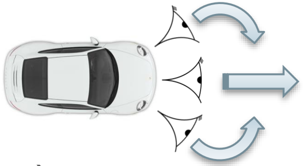
supervised learning

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

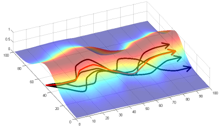
Usually (but not always) insufficient by itself

Distribution mismatch problem

Sometimes works well



Micro-models (e.g. image and control transformations)



Add more **on-policy** data, e.g. using Dagger

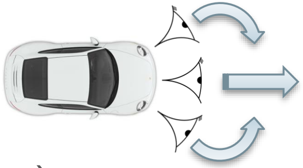
Imitation learning: recap

$\mathbf{o}_t$ $\mathbf{u}_t$ → training data → supervised learning → $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

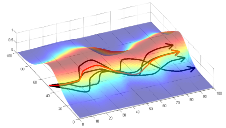
Usually (but not always) insufficient by itself

Distribution mismatch problem

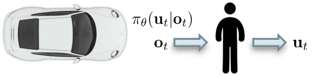
Sometimes works well



Micro-models (e.g. image and control transformations)



Add more **on-policy** data, e.g. using Dagger



Domain adaptation and error recovery