CSC2621
Imitation Learning for Robotics

Florian Shkurti

Week 1: Behavioral Cloning vs. Imitation

New robotics faculty in CS

Jessica Burgner-Kahrs

Animesh Garg

Myself

Today’s agenda

- Administrivia
- Topics covered by the course
- Behavioral cloning
- Imitation learning
- Quiz about background and interests
- Identify first group of presenters for week 3

Administrivia

Administrivia

This is a graduate level seminar course

Course website: http://www.cs.toronto.edu/~florian/courses/imitation_learning/

Discussion forum + announcements: https://q.utoronto.ca (Quercus)

Request improvements anonymously: https://www.surveymonkey.com/r/LJJV5LY

Course-related emails should have CSC2621 in the subject

Prerequisites

Mandatory:

- Introductory machine learning (e.g. CSC411/ECE521 or equivalent)
- Basic linear algebra + multivariable calculus
- Intro to probability
- Programming skills in Python or C++ (enough to validate your ideas)

Recommended:

- Experience training neural networks or other function approximators
- Introductory concepts from reinforcement learning or control (e.g. value function/cost-to-go)

Prerequisites

Mandatory:

- Introductory machine learning (e.g. CSC411/ECE521 or equivalent)
- Basic linear algebra + multivariable calculus
- Intro to probability
- Programming skills in Python or C++ (enough to validate your ideas)

If you're missing any of these this is not the course for you.

You're welcome to audit.

Recommended:

- Experience training neural networks or other function approximators
- Introductory concepts from reinforcement learning or control (e.g. value function/cost-to-go)

If you're missing this we can organize tutorials to help you.

Grading

One assignment: 20%

Paper presentation in class: 20%

Course project: 60%

- Project proposal: 10%
- Midterm progress report: 10%
- Project presentation: 10%
- Final project report (6-8 pages) + code: 30%

Grading

One assignment: 20%

Individual submissions

Paper presentation in class: 20%

We will discuss 4 papers per lecture
7 students will be presenting per lecture
i.e. 1-2 students presenting each paper

Course project: 60%

- Project proposal: 10%
- Midterm progress report: 10%
- Project presentation: 10%
- Final project report (6-8 pages) + code: 30%

Grading

One assignment: 20%

Individual submissions

Paper presentation in class: 20%

Each group will give a practice talk to me on the Monday of the week they present

Course project: 60%

- Project proposal: 10%
- Midterm progress report: 10%
- Project presentation: 10%
- Final project report (6-8 pages) + code: 30%

Grading

One assignment: 20%

Individual submissions

Paper presentation in class: 20%

We will discuss 4 papers per lecture
7 students will be presenting per lecture
i.e. 1-2 students presenting each paper

Course project: 60%

- Project proposal: 10%
- Midterm progress report: 10%
- Project presentation: 10%
- Final project report (6-8 pages) + code: 30%

Groups of 2-3

Guiding principles for this course

Robots do not operate in a vacuum. They do not need to learn everything from scratch.

Guiding principles for this course

Robots do not operate in a vacuum. They do not need to learn everything from scratch.

Humans need to easily interact with robots and share our expertise with them.

Guiding principles for this course

Robots do not operate in a vacuum. They do not need to learn everything from scratch.

Humans need to easily interact with robots and share our expertise with them.

Robots need to learn from the behavior and experience of others, not just their own.

Main questions

How can robots incorporate others' decisions into their own?

How can robots easily understand our objectives from demonstrations?

How do we balance autonomous control and human control in the same system?

Main questions

How can robots incorporate others' decisions into their own?

How can robots easily understand our objectives from demonstrations?

How do we balance autonomous control and human control in the same system?

→

Learning from demonstrations
Apprenticeship learning
Imitation learning

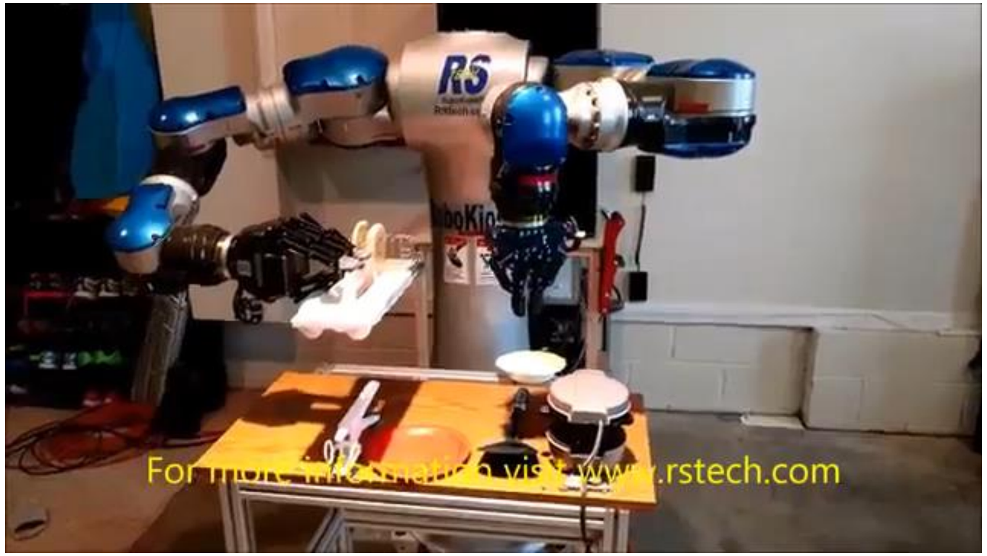
Reward/cost learning
Task specification
Inverse reinforcement learning
Inverse optimal control
Inverse optimization

Shared or sliding autonomy

Applications

Any control problem where:

- **writing down a dense cost function is difficult**
- there is a hierarchy of decision-making processes
- our engineered solutions might not cover all cases
- unrestricted exploration during learning is slow or dangerous



https://www.youtube.com/watch?v=M8r0gmQXm1Y

Applications

Any control problem where:

- **writing down a dense cost function is difficult**
- there is a hierarchy of interacting decision-making processes
- our engineered solutions might not cover all cases
- unrestricted exploration during learning is slow or dangerous

https://www.youtube.com/watch?v=Q3LXJGha7Ws

Applications

Any control problem where:

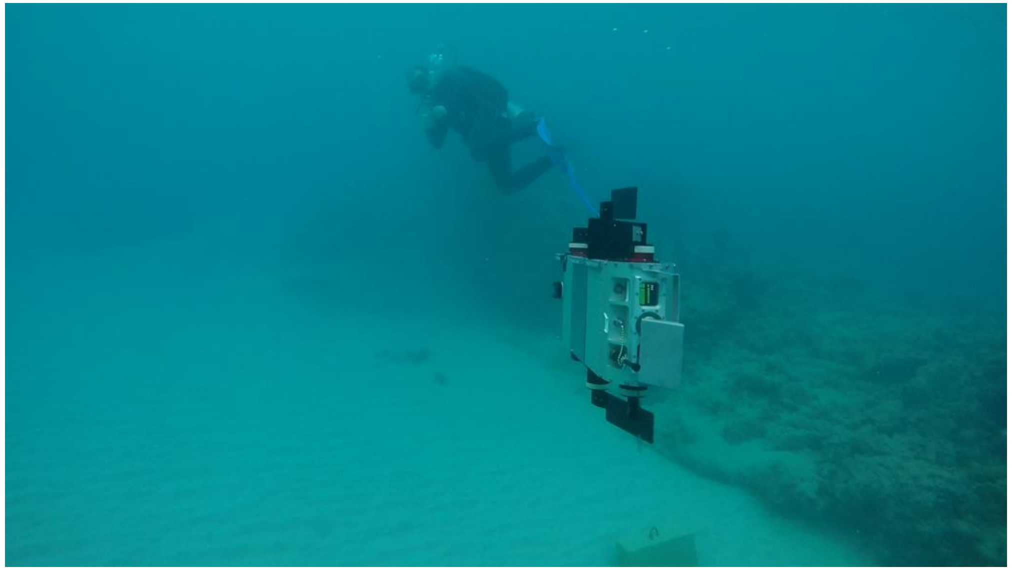
- **writing down a dense cost function is difficult**
- there is a hierarchy of interacting decision-making processes
- our engineered solutions might not cover all cases
- unrestricted exploration during learning is slow or dangerous

https://www.youtube.com/watch?v=RjGe0GiiFzw

Applications

Any control problem where:

- **writing down a dense cost function is difficult**
- there is a hierarchy of interacting decision-making processes
- our engineered solutions might not cover all cases
- unrestricted exploration during learning is slow or dangerous

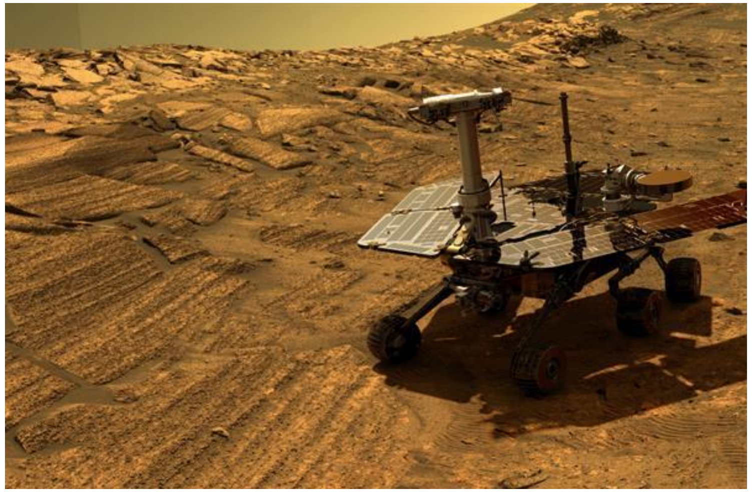


Robot videographer / documentarian

Applications

Any control problem where:

- **writing down a dense cost function is difficult**
- there is a hierarchy of interacting decision-making processes
- our engineered solutions might not cover all cases
- unrestricted exploration during learning is slow or dangerous



Robot explorer

Applications

Any control problem where:

- **writing down a dense cost function is difficult**
- there is a hierarchy of interacting decision-making processes
- our engineered solutions might not cover all cases
- unrestricted exploration during learning is slow or dangerous

https://www.youtube.com/watch?v=0XdC1HUp-rU

Back to the future

https://www.youtube.com/watch?v=2KMAAmkz9go

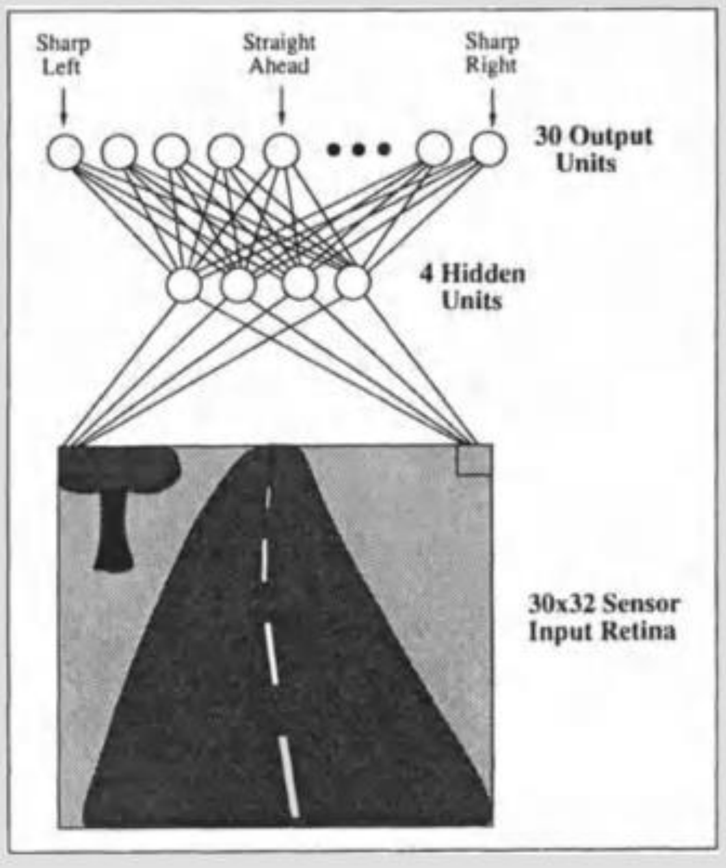
Navlab 1 (1986-1989)

https://www.youtube.com/watch?v=iIP4aPDTBPE

Navlab 2 + ALVINN (Dean Pomerlau's PhD thesis, 1989-1993)

30 x 32 pixels, 3 layer network, outputs steering command
~5 minutes of training per road type

ALVINN: architecture



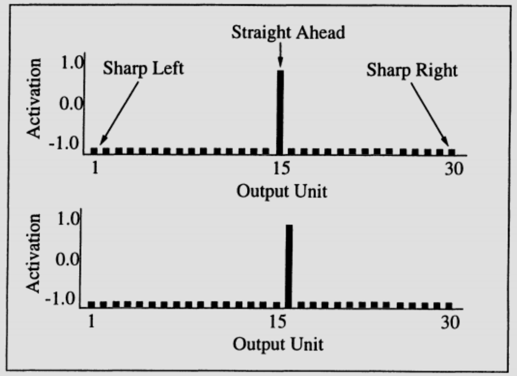


Figure 2.7: The representation of two steering directions using a "one-of-N" encoding. The top graph represents a straight ahead steering direction, since the middle output unit is activated. The bottom graph represents a slight right turn, since an output unit slightly right of center is activated.

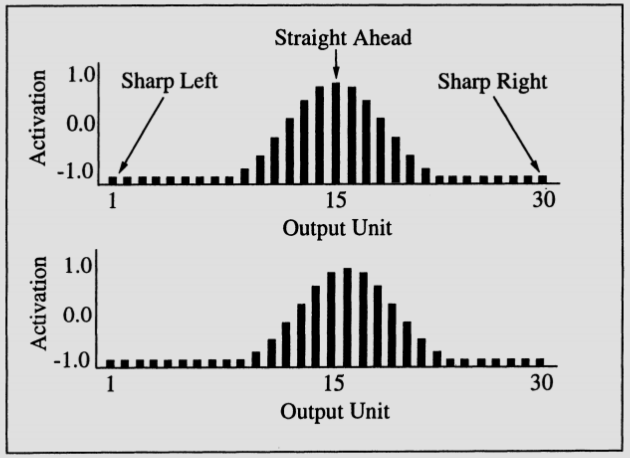


Figure 2.10: The representation of two steering directions using a gaussian output encoding. The top graph represents a straight ahead steering direction, since the gaussian "hill" of activation is centered on the middle output unit. The bottom graph represents a slight right turn, since the "hill" of activation is centered slightly right of the middle unit.

https://drive.google.com/file/d/0Bz9namoRIUKMa0pJYzRGSFVwbm8/view

Dean Pomerlau's PhD thesis

ALVINN: training set

To generate synthetic training data for the task of autonomous road following, I developed a program that generated aerial views of simulated stretches of roads and then used a model of the camera to back-project the aerial map into a 2D image of the road ahead. The simulated road image generator used nearly 200 parameters in order to generate a variety of realistic road images. Some of the most important parameters are listed in Figure 3.1.

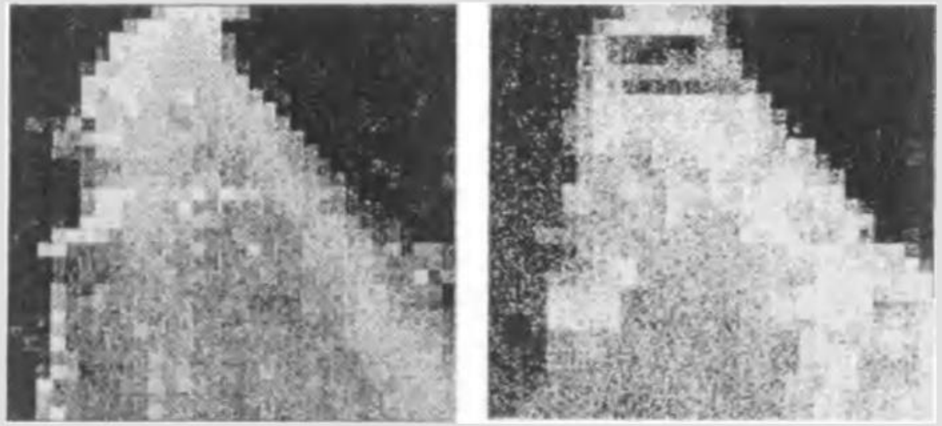
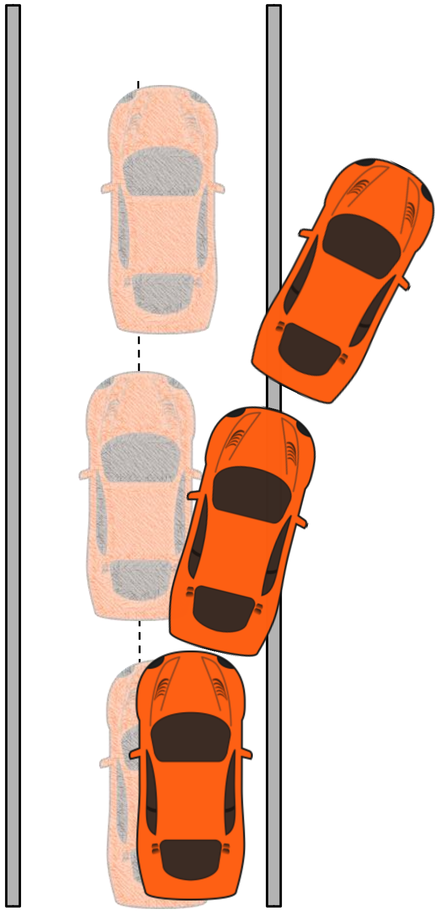


Figure 3.2: A low resolution video image of a single lane road (left) and an artificial single lane road image created by the road image generator (right).

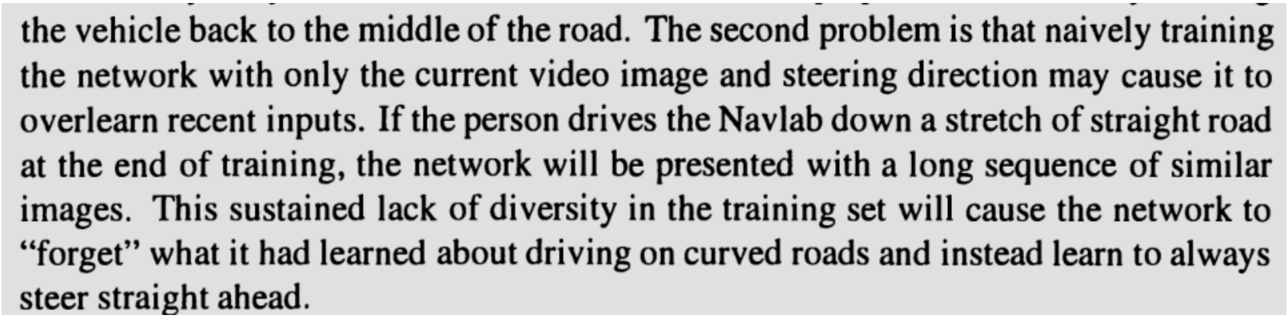
3.2 Training "on-the-fly" with Real Data

Online updates via backpropagation

Problems Identified by Pomerlau



Test distribution is different from training distribution (covariate shift)


the vehicle back to the middle of the road. The second problem is that naively training the network with only the current video image and steering direction may cause it to overlearn recent inputs. If the person drives the Navlab down a stretch of straight road at the end of training, the network will be presented with a long sequence of similar images. This sustained lack of diversity in the training set will cause the network to "forget" what it had learned about driving on curved roads and instead learn to always steer straight ahead.

Catastrophic forgetting

(Partially) Addressing Covariate Shift

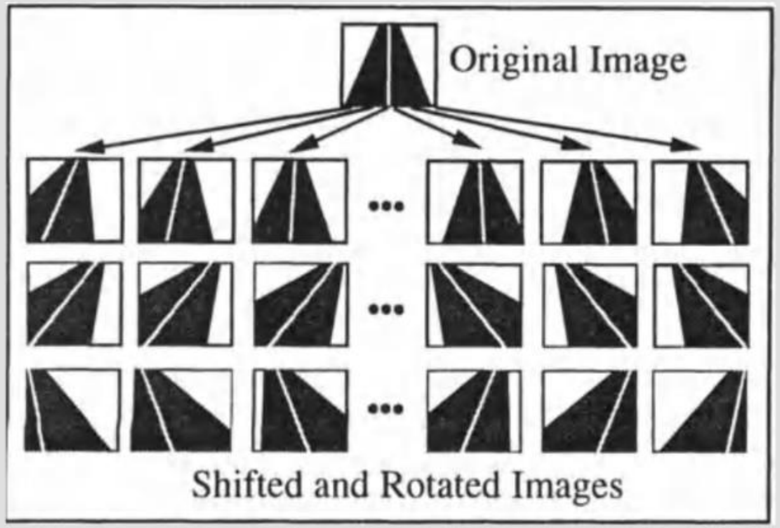
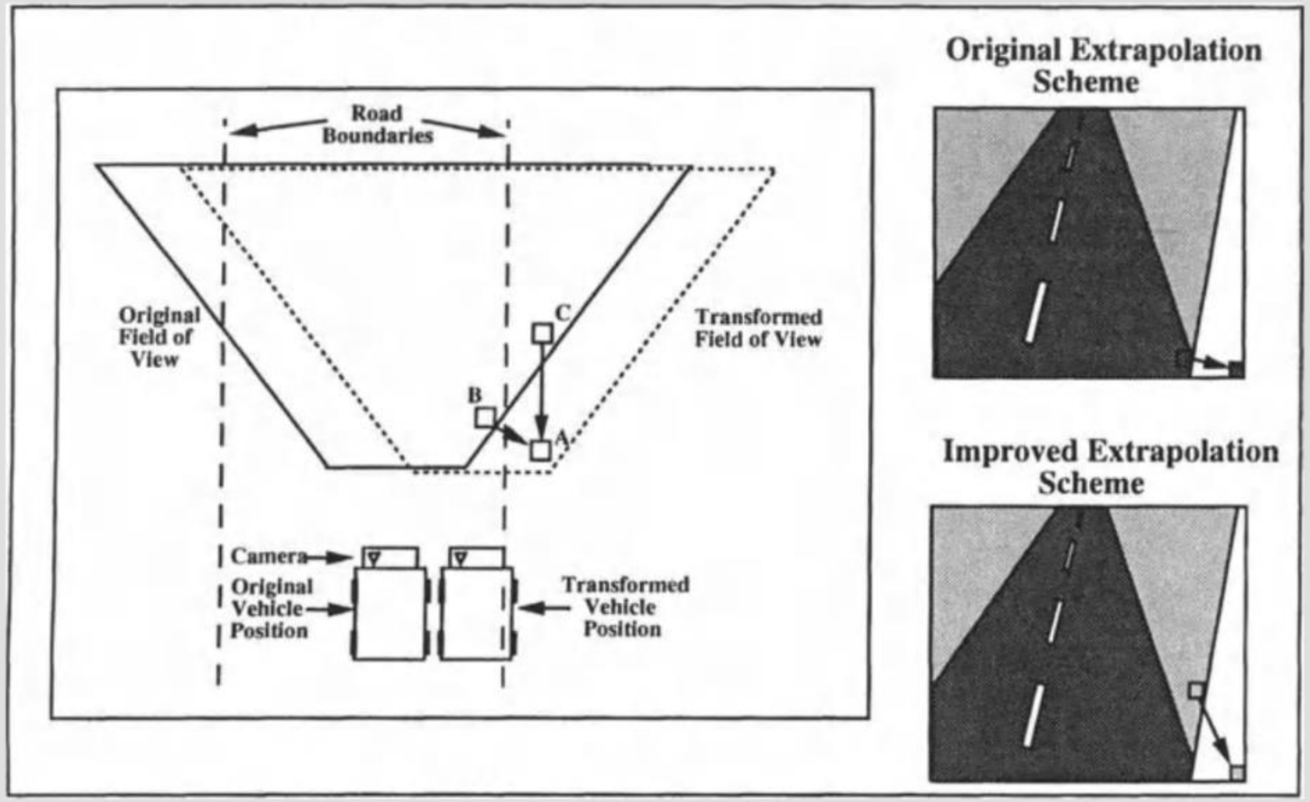


Figure 3.4: The single original video image is shifted and rotated to create multiple training exemplars in which the vehicle appears to be at different locations relative to the road.



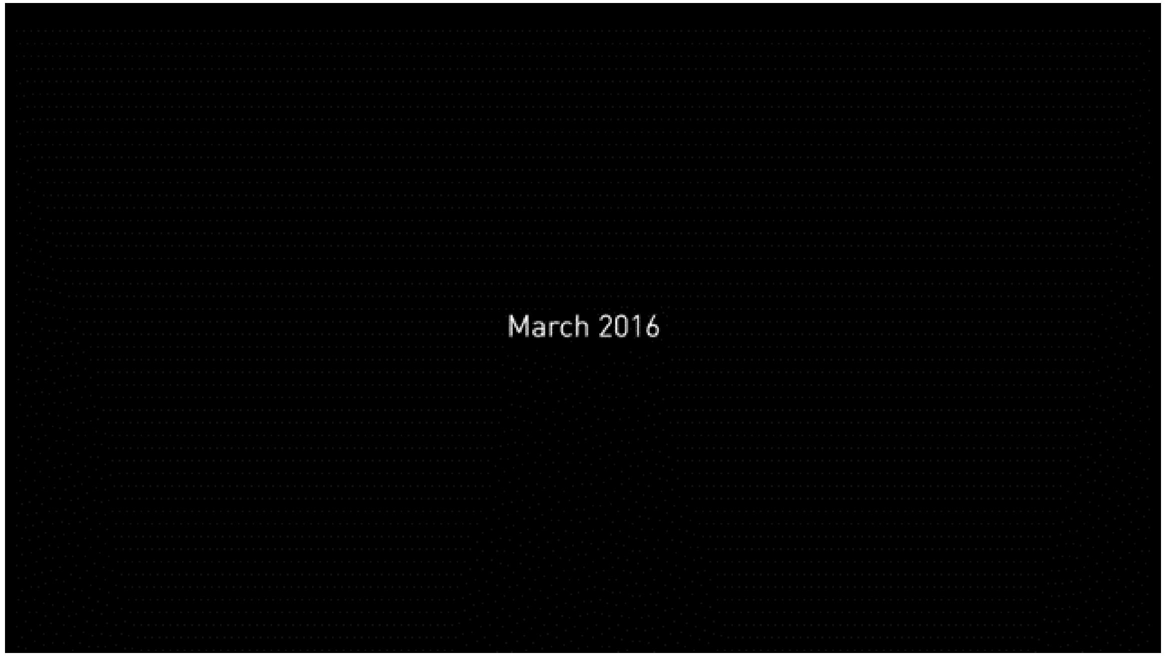
(Partially) Addressing Catastrophic Forgetting

1. Maintains a buffer of old (image, action) pairs

2. Experiments with different techniques to ensure diversity and avoid outliers

Behavioral Cloning = Supervised Learning

25 years later



https://www.youtube.com/watch?v=qhUvQiKec2U

How much has changed?

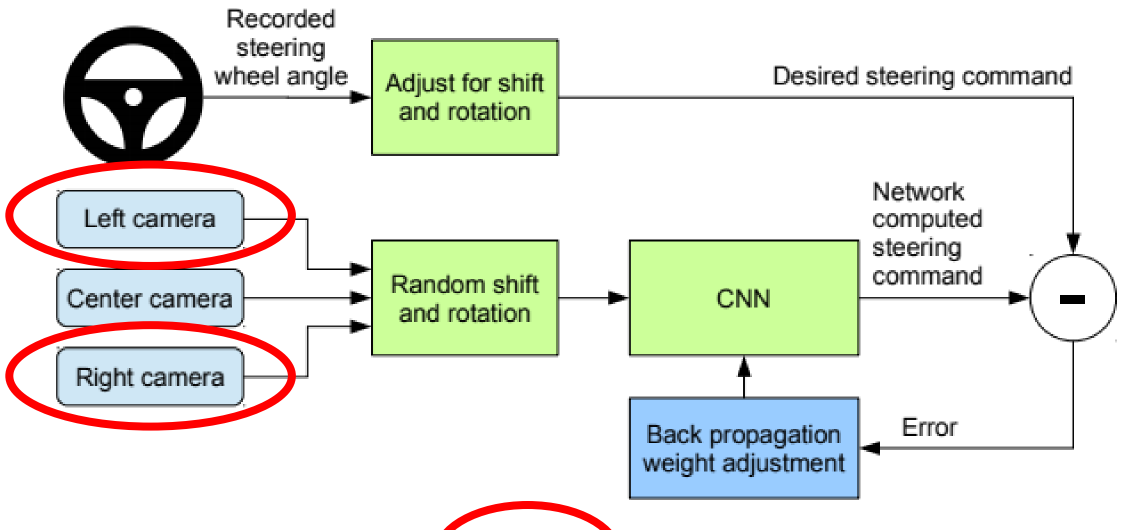


Figure 2: Training the neural network.

offline

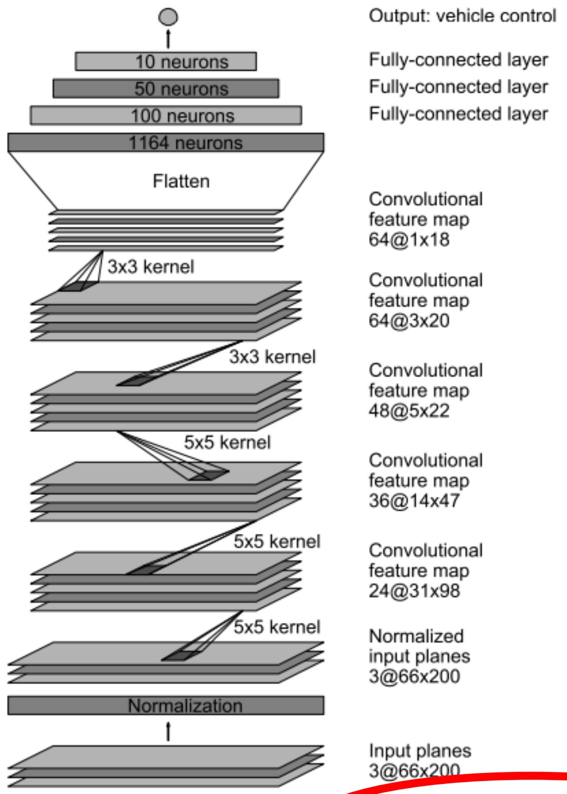


Figure 4: CNN architecture. The network has about 27 million connections and 250 thousand parameters.

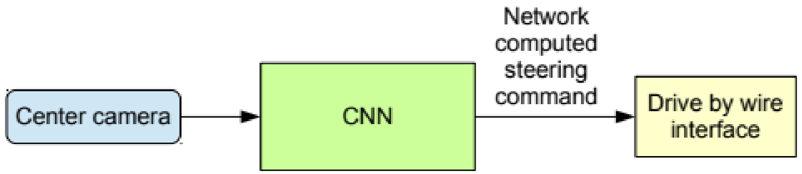


Figure 3: The trained network is used to generate steering commands from a single front-facing center camera.

End to End Learning for Self-Driving Cars, Bojarski et al, 2016

How much has changed?

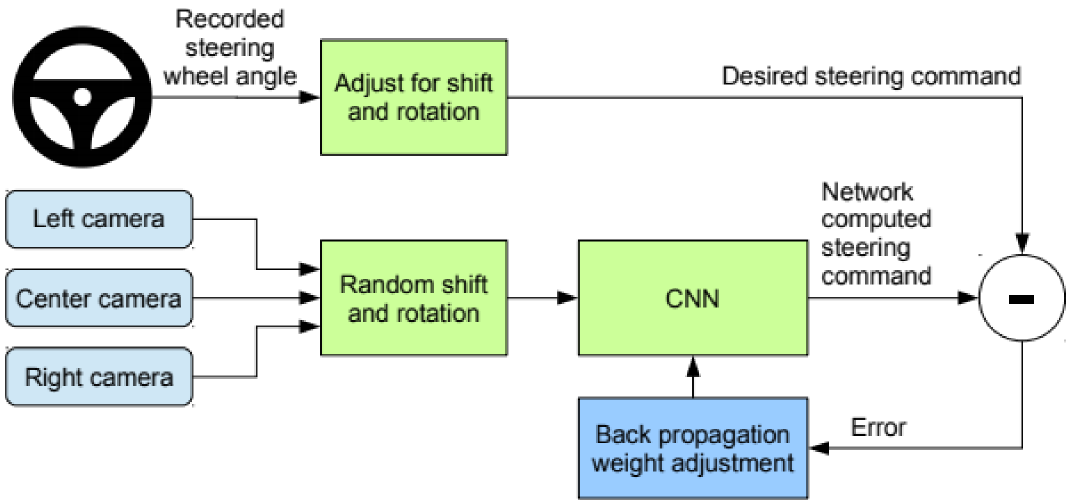


Figure 2: Training the neural network.

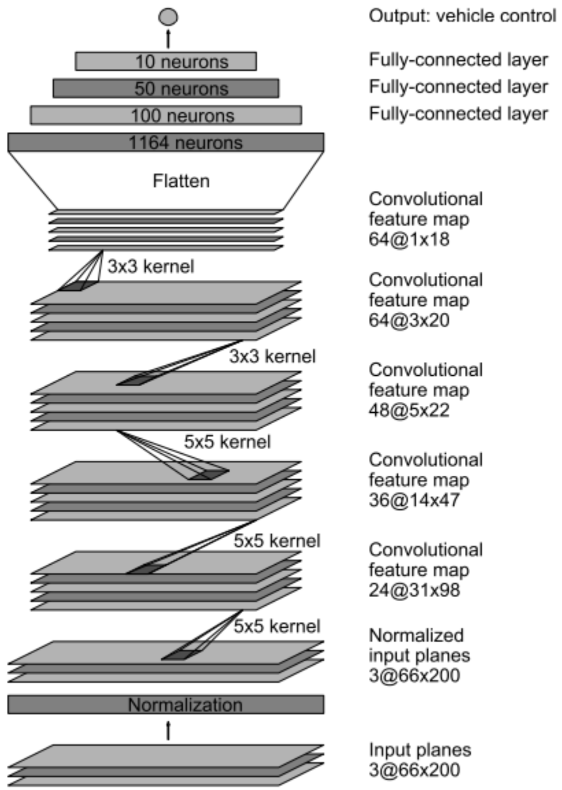


Figure 4: CNN architecture. The network has about 27 million connections and 250 thousand parameters.

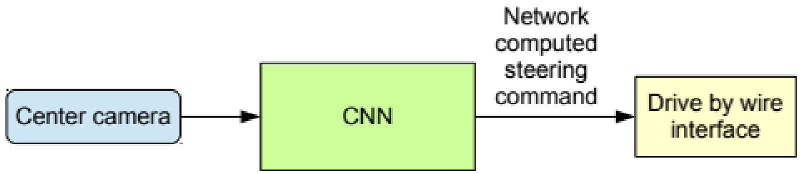


Figure 3: The trained network is used to generate steering commands from a single front-facing center camera.

"Our collected data is labeled with road type, weather condition, and the driver's activity (staying in a lane, switching lanes, turning, and so forth)."

End to End Learning for Self-Driving Cars, Bojarski et al, 2016

How much has changed?

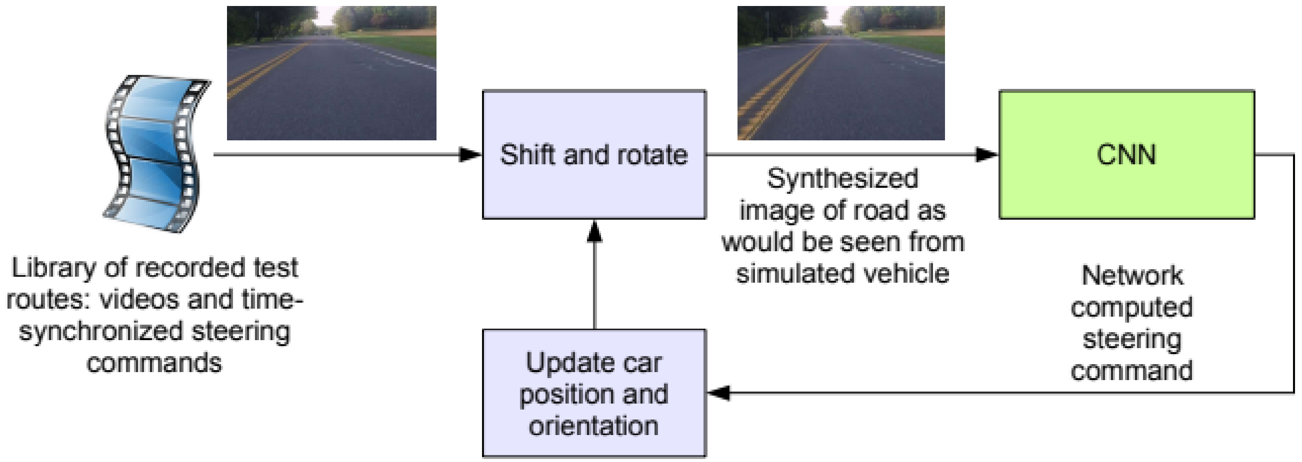


Figure 5: Block-diagram of the drive simulator.

How much has changed?

A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots, Giusti et al., 2016
https://www.youtube.com/watch?v=umRdt3zGgpU

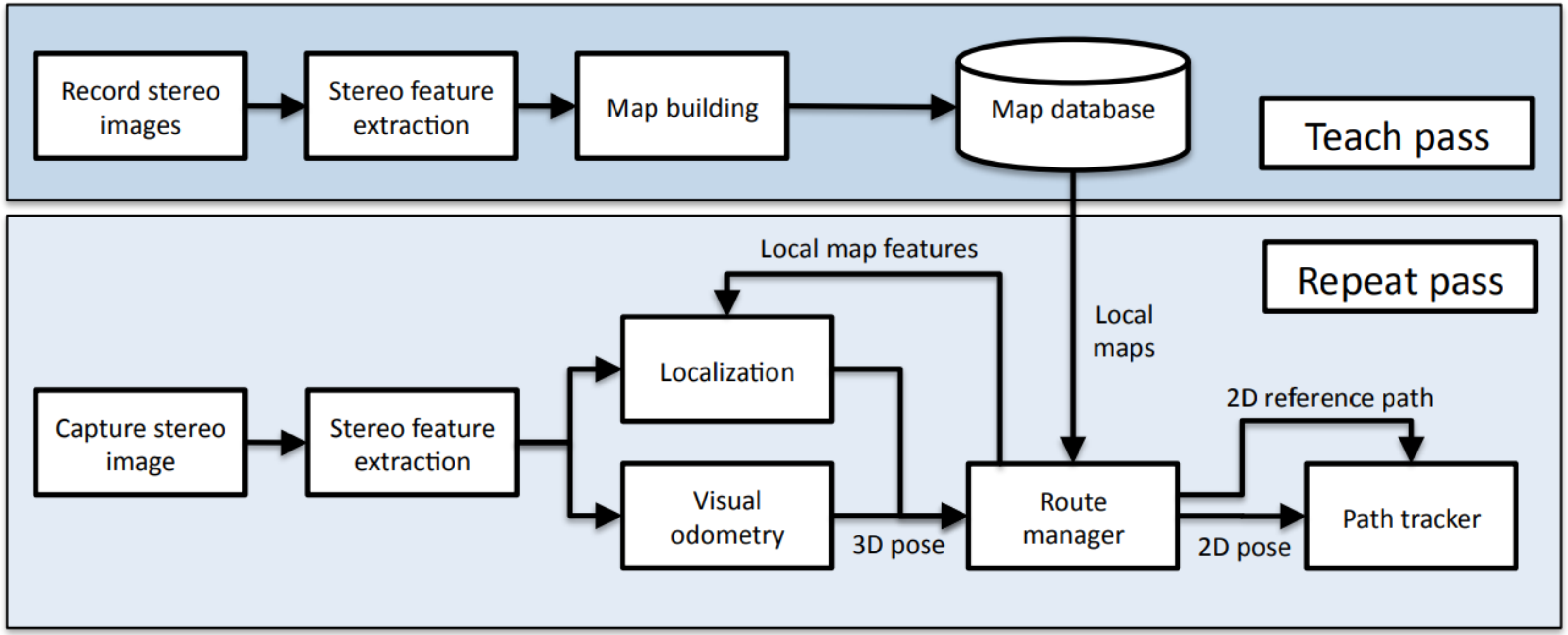
How much has changed?

Not a lot for learning lane following with neural networks.

But, there are a few other beautiful ideas that do not involve end-to-end learning.

Visual Teach & Repeat

Human Operator or
Planning Algorithm



Visual Path Following on a Manifold in Unstructured Three-Dimensional Terrain, Furgale & Barfoot, 2010

Visual Teach & Repeat

Key Idea #1: Manifold Map

Build local maps relative to the path. No global coordinate frame.

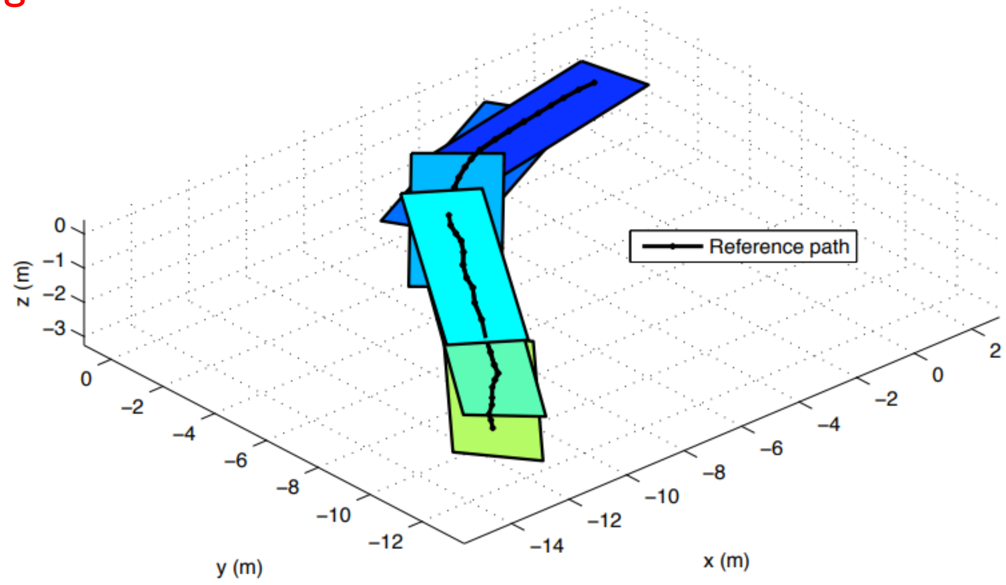


Fig. 5. A view of six overlapping submaps with the reference path plotted above.

Visual Path Following on a Manifold in Unstructured Three-Dimensional Terrain, Furgale & Barfoot, 2010

Visual Teach & Repeat

Key Idea #1: Manifold Map

Build local maps relative to the path. No global coordinate frame.

Key Idea #2: Visual Odometry

Given two consecutive images, how much has the camera moved? Relative motion.

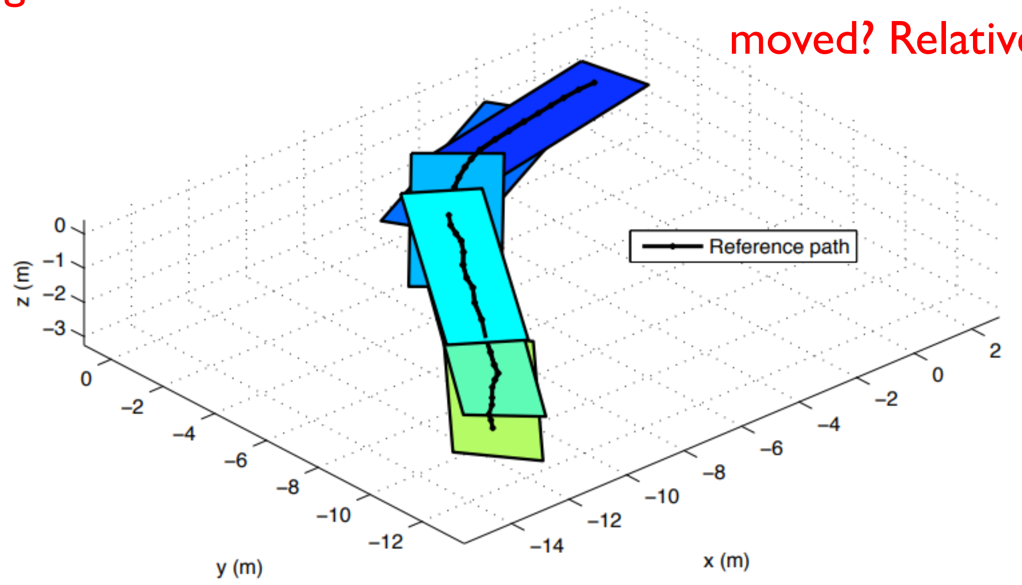


Fig. 5. A view of six overlapping submaps with the reference path plotted above.

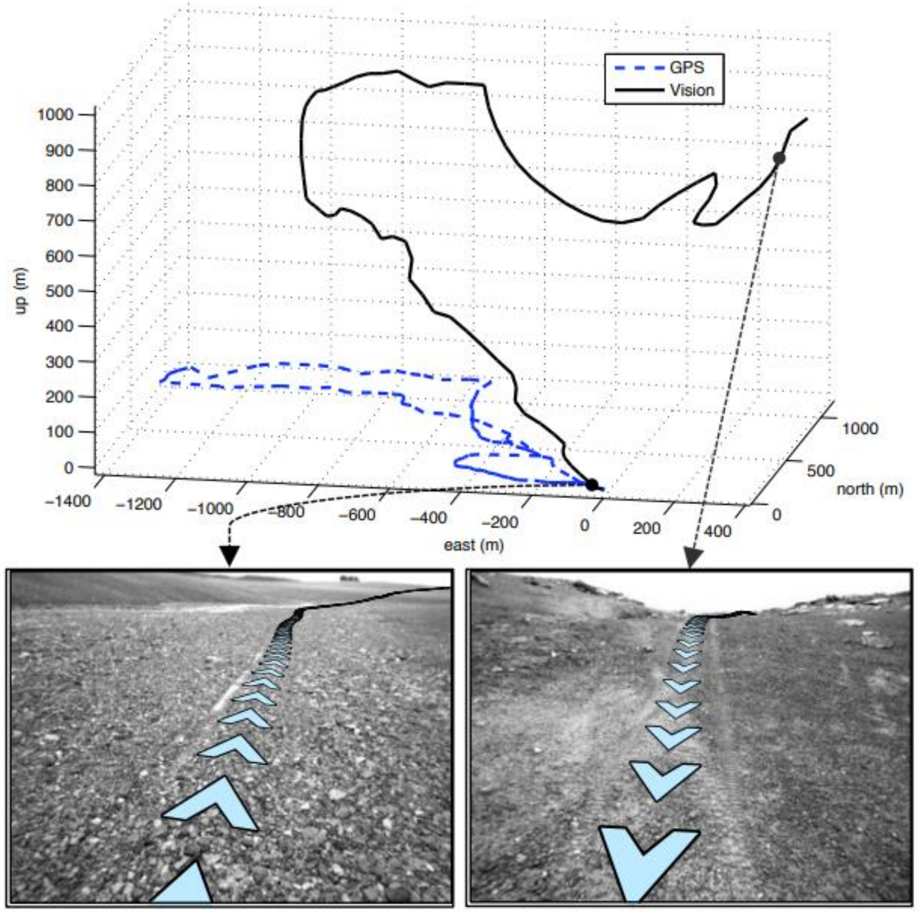


Fig. 6. The visual reconstruction of a five kilometer rover traverse plotted against GPS (Top). Although the reconstruction is wildly inaccurate at this scale, locally it is good enough to enable retracing of the route. The bottom images show views from either end of the path, with the reference path plotted as a series of chevrons. To the rover, the map is locally Euclidean.

Visual Path Following on a Manifold in Unstructured Three-Dimensional Terrain, Furgale & Barfoot, 2010

Visual Teach & Repeat

https://www.youtube.com/watch?v=_ZdBfU4xJnQ

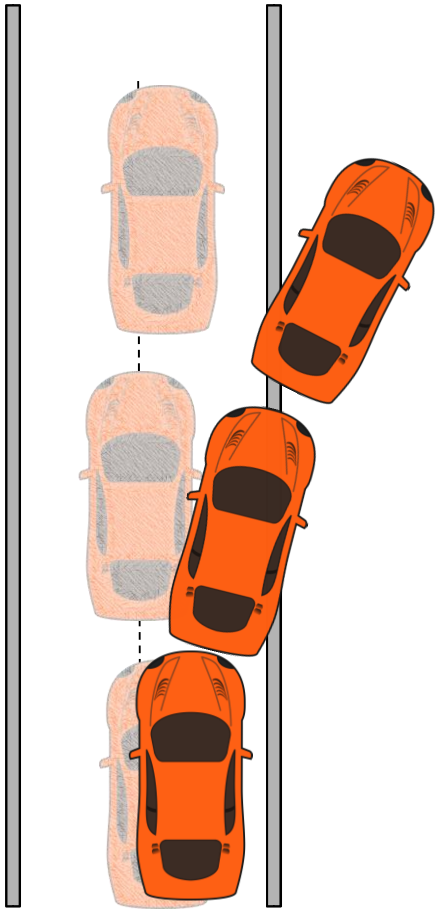
https://www.youtube.com/watch?v=9dN0wwXDuqo

Centimeter-level precision in tracking the demonstrated path over kilometers-long trails.

Today's agenda

- Administrivia
- Topics covered by the course
- Behavioral cloning
- Imitation learning
- Quiz about background and interests
- Identify first group of presenters for week 3

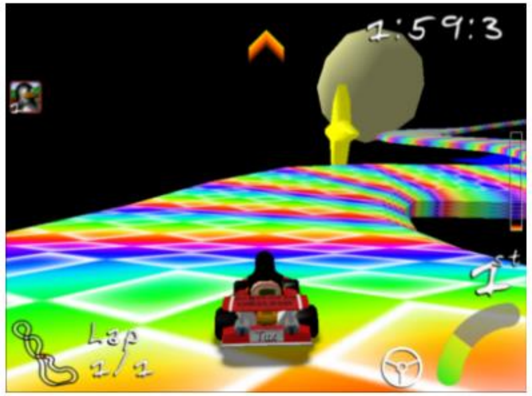
Back to Pomerlau



Test distribution is different from training distribution (covariate shift)

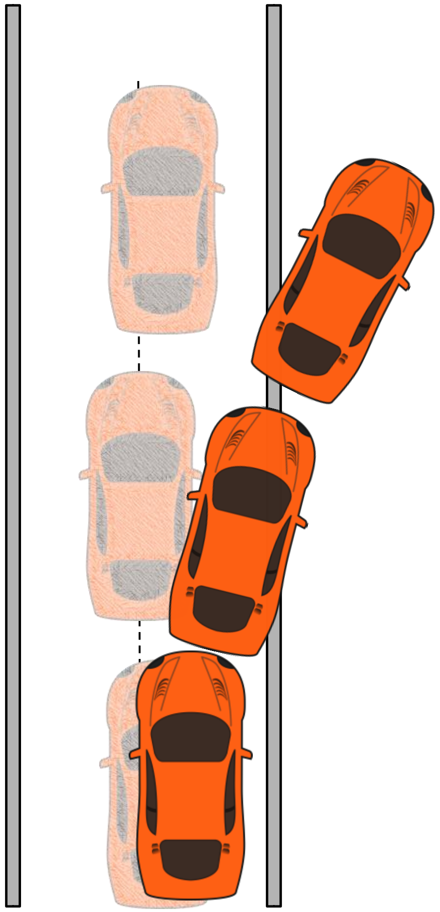
(Ross & Bagnell, 2010): How are we sure these errors are not due to overfitting or underfitting?

1. Maybe the network was too small (underfitting)
2. Maybe the dataset was too small and the network overfit it



Steering commands $\pi_\theta(s) = \theta^\top s$
where s are image features

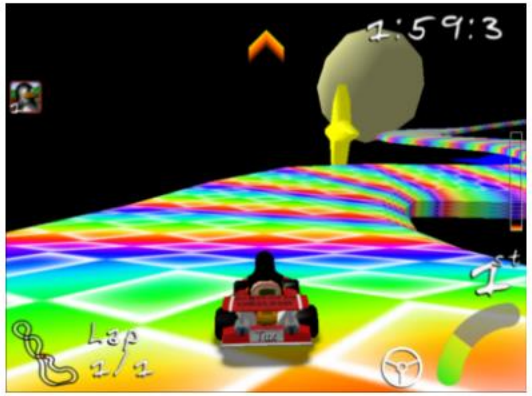
Back to Pomerlau



Test distribution is different from training distribution (covariate shift)

(Ross & Bagnell, 2010): How are we sure these errors are not due to overfitting or underfitting?

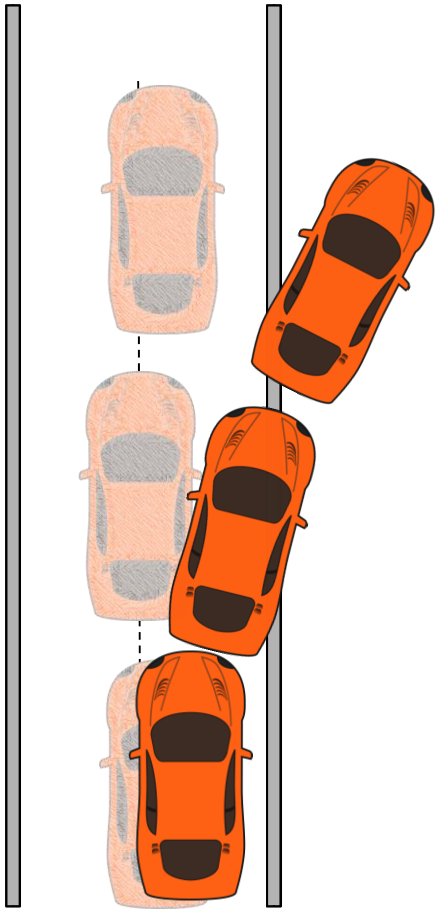
1. Maybe the network was too small (underfitting)
2. Maybe the dataset was too small and the network overfit it



Steering commands $\pi_\theta(s) = \theta^\top s$
where s are image features

It was not 1: they showed that even a linear policy can work well.
It was not 2: their error on held-out data was close to training error.

Imitation learning $\neq$ Supervised learning



Test distribution is different from training distribution (covariate shift)

(Ross & Bagnell, 2010): IL is a sequential decision-making problem.

- Your actions affect future observations/data.
- This is not the case in supervised learning

Supervised Learning

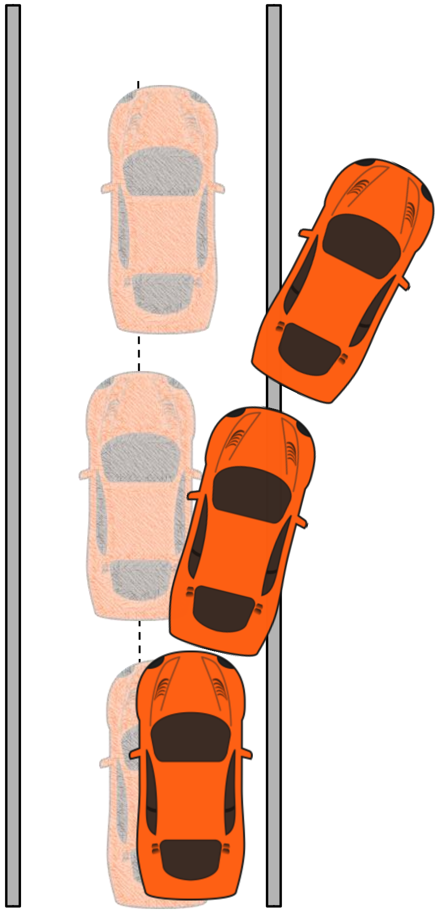
Assumes train/test data are i.i.d.

If expected training error is ϵ
Expected test error after T decisions

$$T\epsilon$$

Errors are independent

Imitation learning $\neq$ Supervised learning



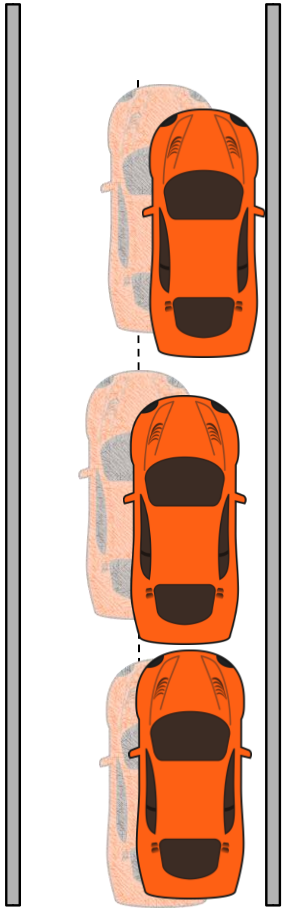
Test distribution is different from training distribution (covariate shift)

(Ross & Bagnell, 2010): IL is a sequential decision-making problem.

- Your actions affect future observations/data.
- This is not the case in supervised learning

Imitation Learning	←	**Supervised Learning**
Train/test data are not i.i.d.		Assumes train/test data are i.i.d.
If expected training error is ϵ Expected test error after T decisions is up to $T^2\epsilon$		If expected training error is ϵ Expected test error after T decisions $T\epsilon$
Errors compound		Errors are independent

DAgger



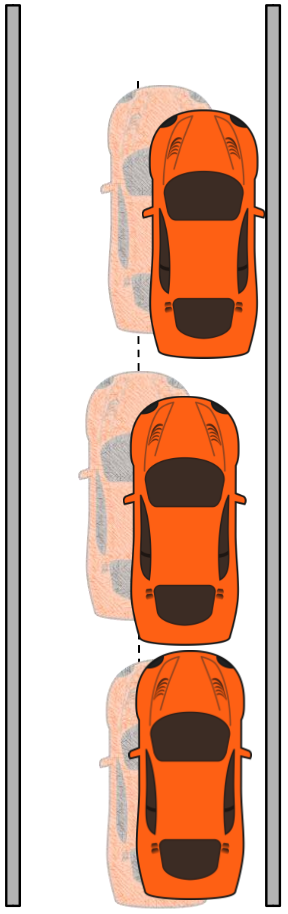
(Ross & Gordon & Bagnell, 2011): DAgger, or Dataset Aggregation

- Imitation learning as interactive supervision
- Aggregate training data from expert with test data from execution

Algorithm 1 DAgger

1: $D = \{(s, a)\}$ initial expert demonstrations
2: $\theta_1 \leftarrow$ train learner's policy parameters on D
3: **for** $i = 1 \ldots N$ **do**
4: Execute learner's policy π_{θ_i}, get visited states $S_{\theta_i} = \{s_0, \ldots, s_T\}$
5: Query the expert at those states to get actions $A = \{a_0, \ldots, a_T\}$
6: Aggregate dataset $D = D \cup \{(s, a) \mid s \in S_{\theta_i},\ a \in A\}$
7: Train learner's policy $\pi_{\theta_{i+1}}$ on dataset D
8: Return one of the policies π_{θ_i} that performs best on validation set

DAgger



(Ross & Gordon & Bagnell, 2011): DAgger, or Dataset Aggregation

- Imitation learning as interactive supervision
- Aggregate training data from expert with test data from execution

Imitation Learning via DAgger

Train/test data are not i.i.d.

If expected training error on aggr. dataset is ϵ
Expected test error after T decisions is

$$O(T\epsilon)$$

Errors do not compound

Supervised Learning

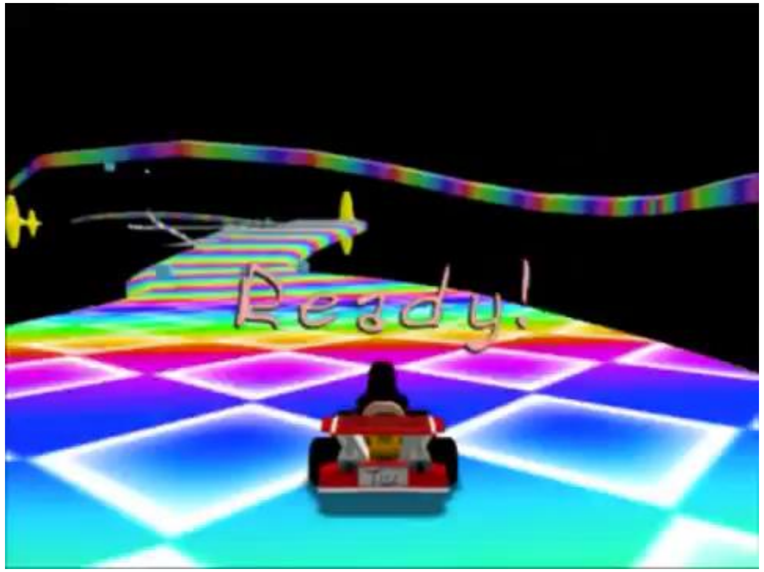
Assumes train/test data are i.i.d.

If expected training error is ϵ
Expected test error after T decisions

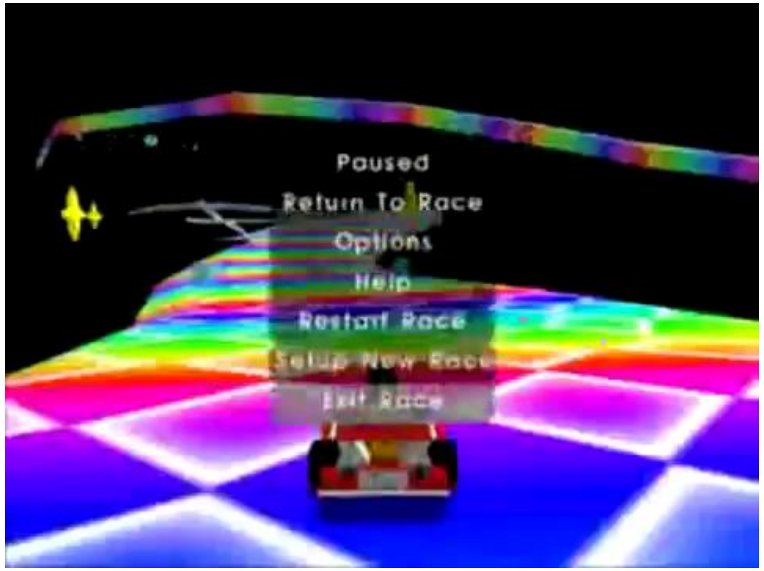
$$T\epsilon$$

Errors are independent

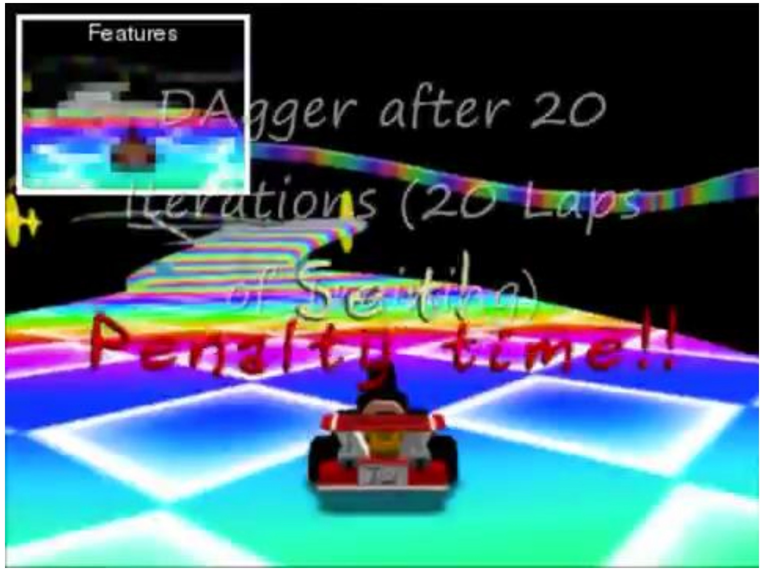
DAgger



Initial expert trajectories



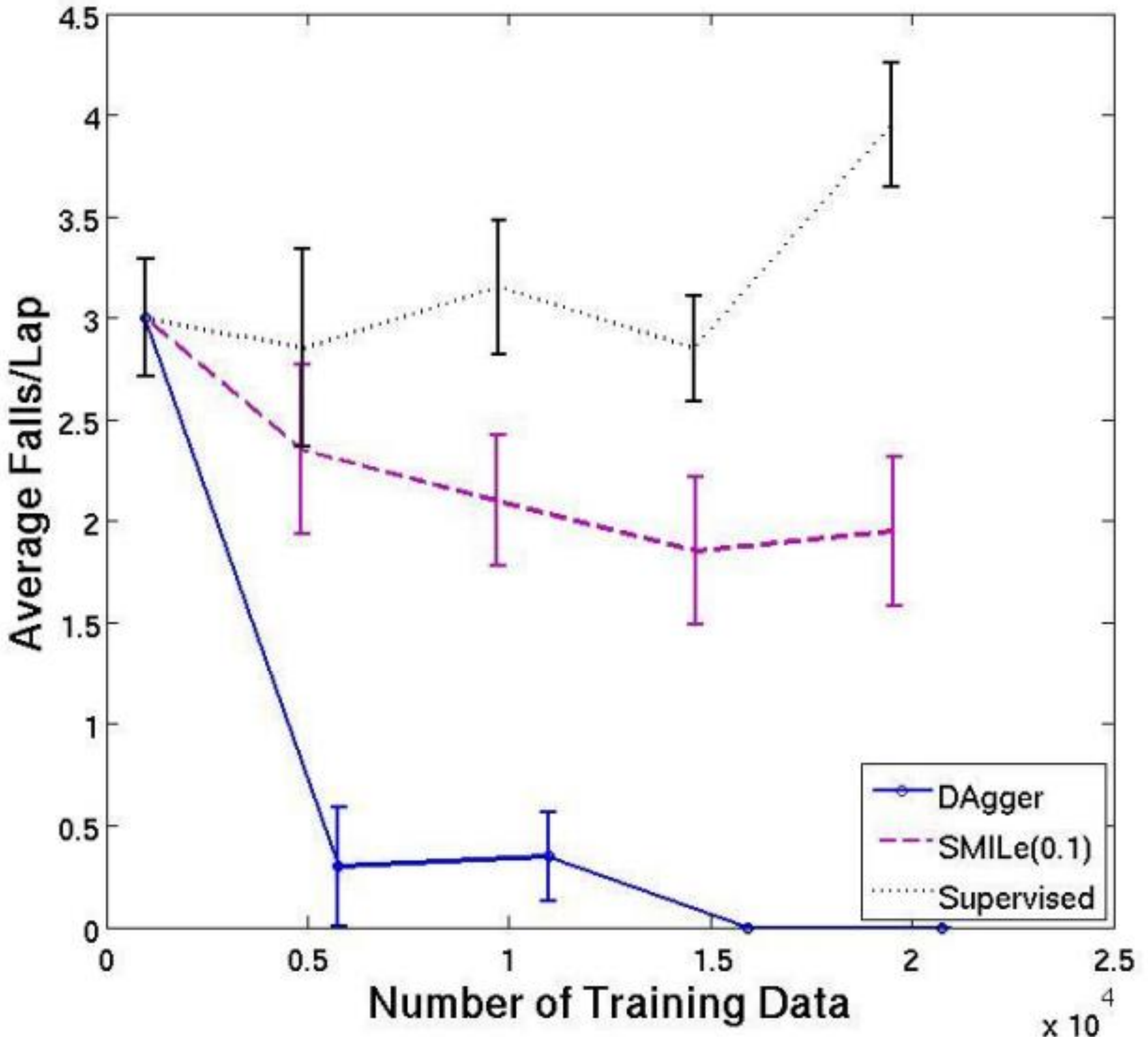
Supervised learning



DAgger

https://www.youtube.com/watch?v=V00npNnWzSU

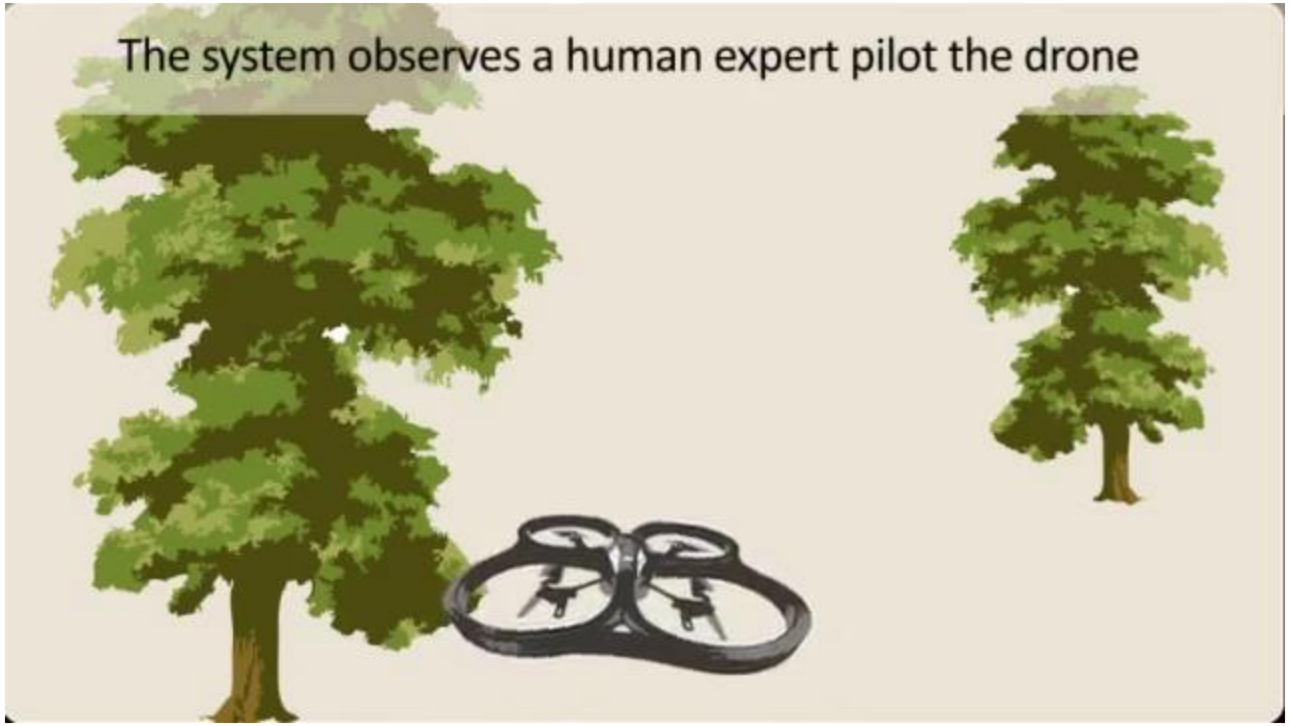
DAgger


Average Falls/Lap
Number of Training Data
x 10^4
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
DAgger
SMILe(0.1)
Supervised

DAgger

Q: Any drawbacks of using it in a robotics setting?

DAgger



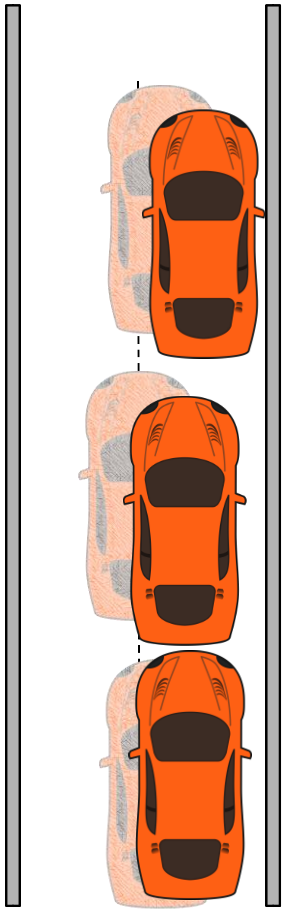
https://www.youtube.com/watch?v=hNsP6-K3Hn4

Learning Monocular Reactive UAV Control in Cluttered Natural Environments, Ross et al, 2013

Today's agenda

- Administrivia
- Topics covered by the course
- Behavioral cloning
- Imitation learning
- Quiz about background and interests
- Identify first group of presenters for week 3

DAgger: Assumptions for theoretical guarantees



(Ross & Gordon & Bagnell, 2011): DAgger, or Dataset Aggregation

- Imitation learning as interactive supervision
- Aggregate training data from expert with test data from execution

Imitation Learning via DAgger

Train/test data are not i.i.d.

If expected training error on aggr. dataset is ϵ
Expected test error after T decisions is

$$O(T\epsilon)$$

Errors do not compound

Supervised Learning

Assumes train/test data are i.i.d.

If expected training error is ϵ
Expected test error after T decisions

$$T\epsilon$$

Errors are independent

Strongly convex loss
No-regret online learner

Appendix: No-Regret Online Learners

Intuition: No matter what the distribution of input data, your online policy/classifier will do asymptotically as well as the best-in-hindsight policy/classifier.

$$r_N = \frac{1}{N}\sum_{i=1}^{N} L_i(\theta_i) - \min_{\theta\in\Theta}\left[\frac{1}{N}\sum_{i=1}^{N} L_i(\theta)\right]$$

Policy has access to data up to round i

Policy has access to data up to round N

No-regret: $\lim_{N\to\infty} r_N = 0$