ILLINOIS

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Artificial Intelligence Qualifying Exam

Alli Nilles

October 9, 2017

University of Illinois at Urbana-Champaign

Outline

- Brief overview of my research projects

Outline

- Brief overview of my research projects
- *Understanding Black Box Predictions via Influence Functions*

Outline

- Brief overview of my research projects
- *Understanding Black Box Predictions via Influence Functions*
- *Generating Plans that Predict Themselves*

My Research

Simple Mobile Robots

- Mobile robots can vacuum floors, transport goods in warehouses, act as security, etc

NASA's Mars Roomba Begins Mission To Clean Dust From Planet's Surface

According to NASA, the Mars Roomba's edge-cleaning mode will allow the vehicle to scour even the crevices where mountains meet the planet's surface.

Simple Mobile Robots

- Mobile robots can vacuum floors, transport goods in warehouses, act as security, etc

Simple Mobile Robots

- Mobile robots can vacuum floors, transport goods in warehouses, act as security, etc
- We want to **minimize** sensing, computation, actuation

Simple Mobile Robots

- Mobile robots can vacuum floors, transport goods in warehouses, act as security, etc
- We want to **minimize** sensing, computation, actuation
    - make robots less expensive, more energy efficient

Simple Mobile Robots

- Mobile robots can vacuum floors, transport goods in warehouses, act as security, etc
- We want to **minimize** sensing, computation, actuation
    - make robots less expensive, more energy efficient
- Often, robots can bump into things and be ok!

Simple Mobile Robots

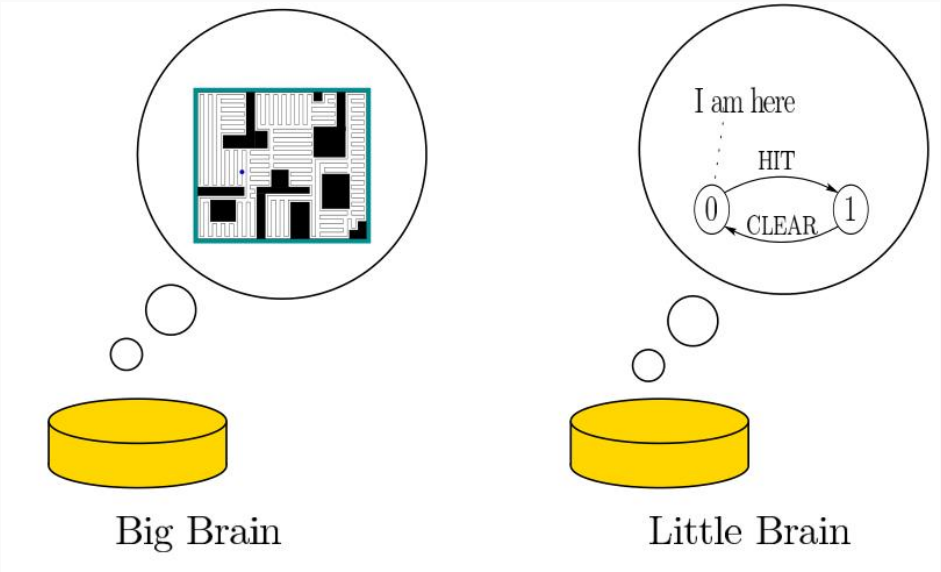
- Mobile robots can vacuum floors, transport goods in warehouses, act as security, etc
- We want to **minimize** sensing, computation, actuation
    - make robots less expensive, more energy efficient
- Often, robots can bump into things and be ok!
- How can we use **contact with the environment** as a strategy or source of information?

Simple Mobile Robots

- Mobile robots can vacuum floors, transport goods in warehouses, act as security, etc
- We want to **minimize** sensing, computation, actuation
    - make robots less expensive, more energy efficient
- Often, robots can bump into things and be ok!
- How can we use **contact with the environment** as a strategy or source of information?

Simple Mobile Robots

- Mobile robots can vacuum floors, transport goods in warehouses, act as security, etc
- We want to **minimize** sensing, computation, actuation
    - make robots less expensive, more energy efficient
- Often, robots can bump into things and be ok!
- How can we use **contact with the environment** as a strategy or source of information?



Blind, Bouncing Robots

Model the robot a point moving in a planar environment, which:

- moves forward in straight lines until collision

[2][1], ICRA 13

Blind, Bouncing Robots

Model the robot a point moving in a planar environment, which:

- moves forward in straight lines until collision
- when in contact with boundary, rotates in place to some angle θ, then moves forward again

[2][1], ICRA 13

Blind, Bouncing Robots

Model the robot a point moving in a planar environment, which:

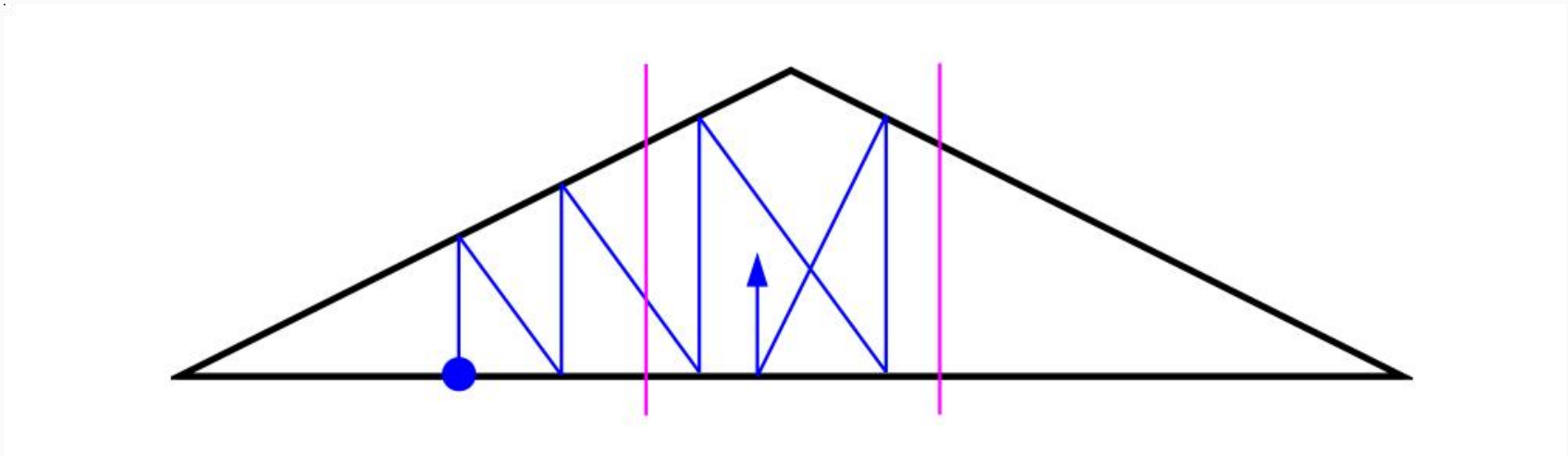
- moves forward in straight lines until collision
- when in contact with boundary, rotates in place to some angle θ, then moves forward again

[2][1], ICRA 13

Blind, Bouncing Robots

Model the robot a point moving in a planar environment, which:

- moves forward in straight lines until collision
- when in contact with boundary, rotates in place to some angle θ, then moves forward again



In this environment, bouncing at the normal, the robot will become trapped in the area between the purple lines.[2]

[2] [1], ICRA 13

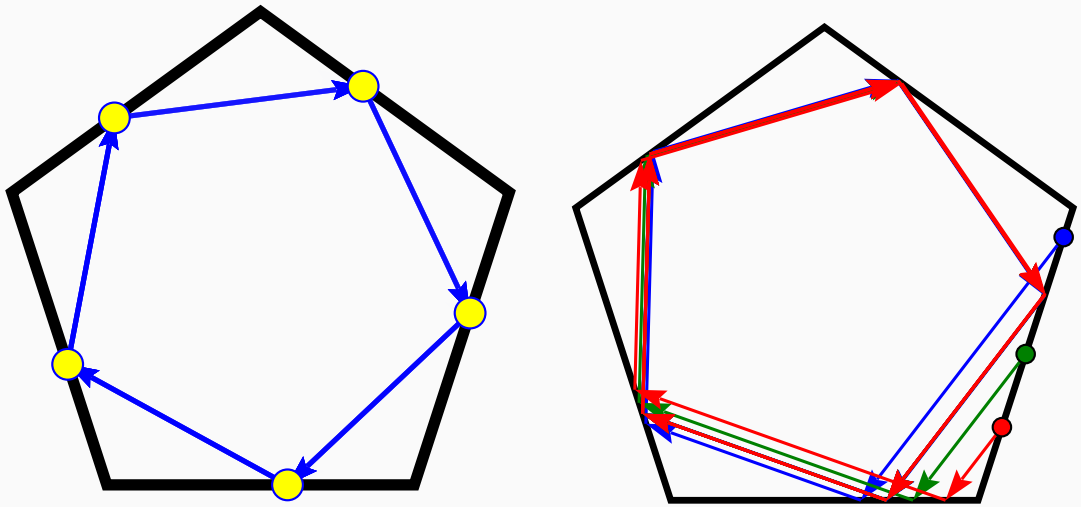
Research Questions

Given a constant control strategy, will the robot become “trapped” in a certain motion pattern (attractor)?

Research Questions

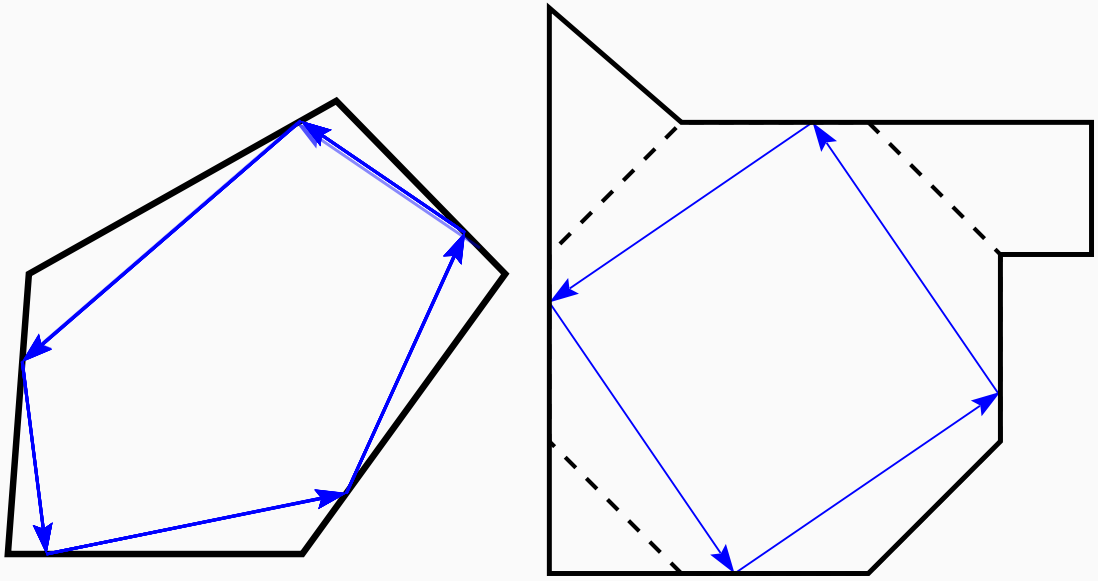
Given a constant control strategy, will the robot become “trapped” in a certain motion pattern (attractor)?

We show that such a robot can perform the task of **patrolling**: periodically following the same path.



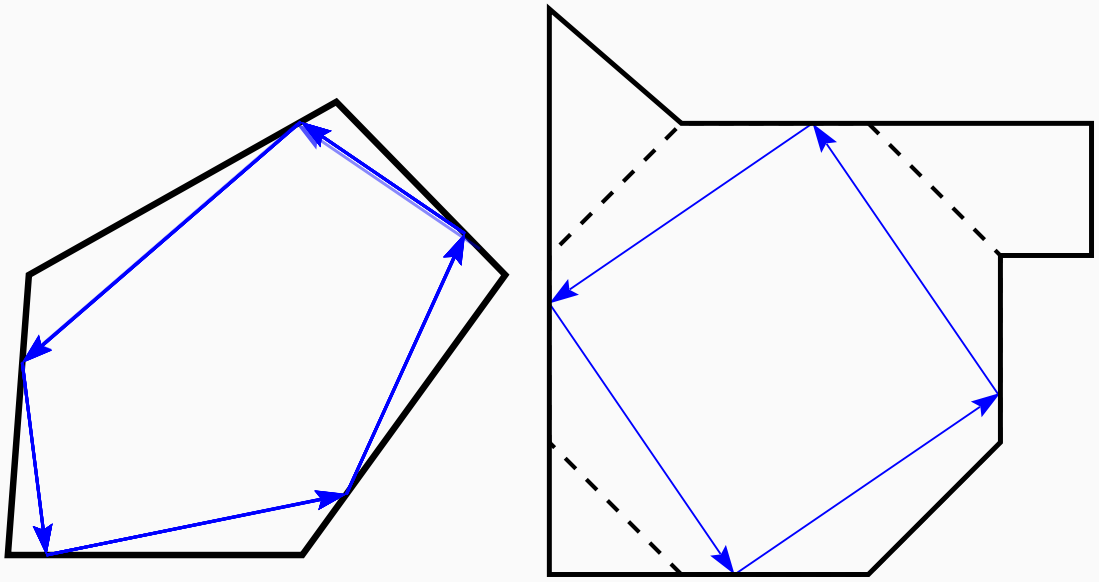
Results

- Periodic trajectories in regular polygons [2], IROS 17



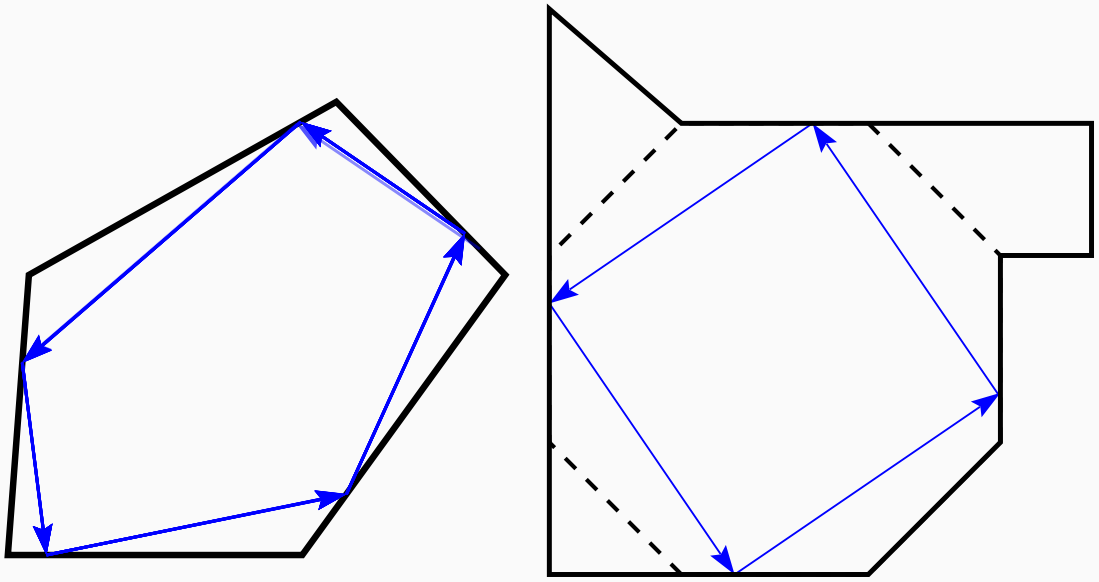
Results

- Periodic trajectories in regular polygons [2], IROS 17
- Periodic trajectories in convex polygons (upcoming, with Israel Becerra, postdoc)



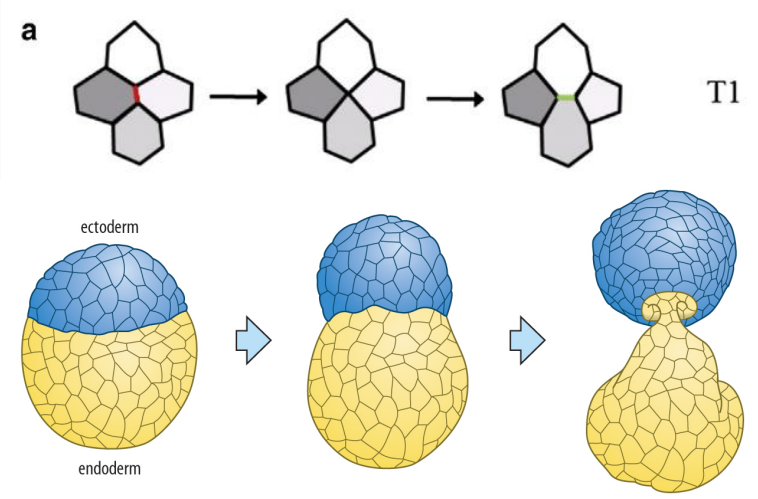
Results

- Periodic trajectories in regular polygons [2], IROS 17
- Periodic trajectories in convex polygons (upcoming, with Israel Becerra, postdoc)
- Next steps: incorporate feedback control, and explore design space (other sensors, actuation strategies, etc), multiple robots, etc



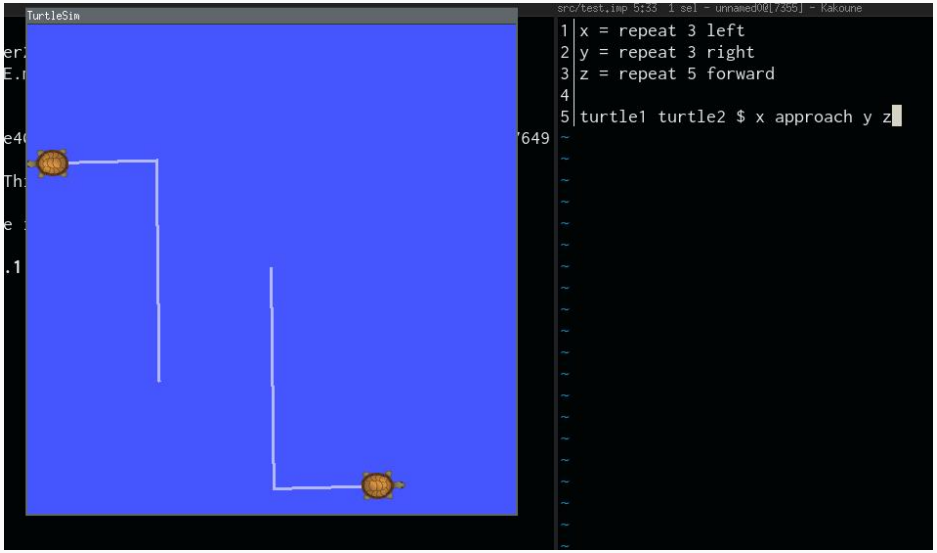
Other Projects

With Dr. Yuliy Baryshnikov: morphogenesis from local cell reconfigurations. Figures from [3] [4]

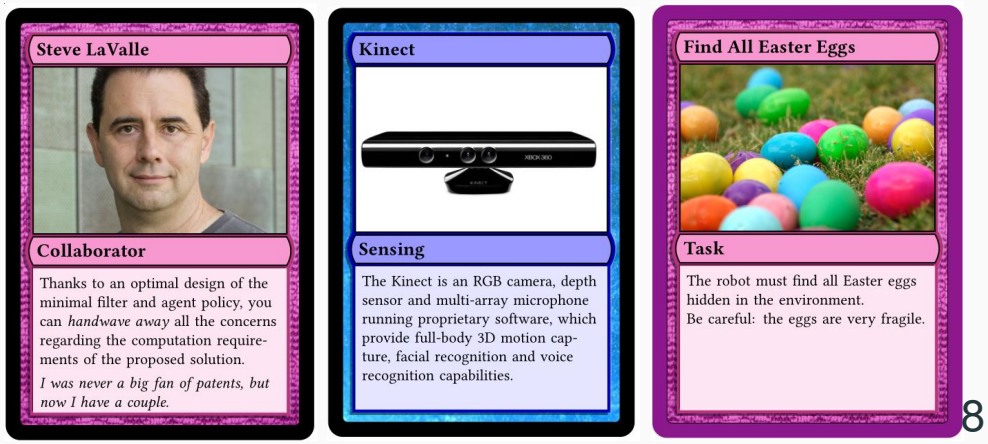


Weaselball assemblies (undergradates run this project!)

Robot “live coding” language for ROS - with Chase Gladish, Drs. Amy LaViers, Mattox Beckman

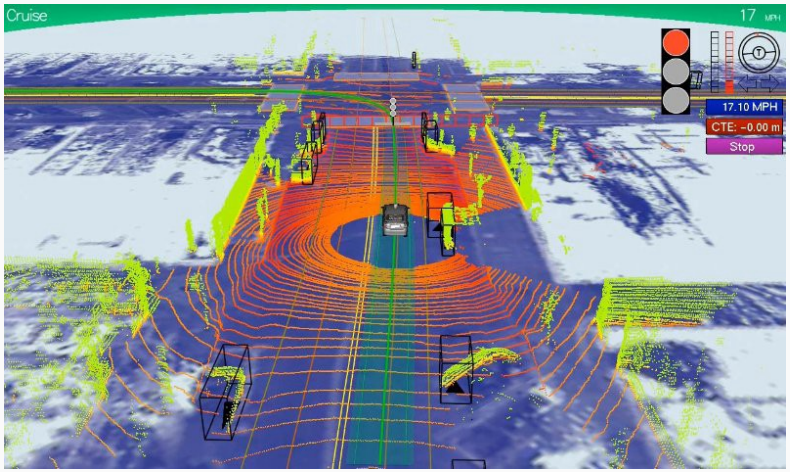


Robot Design Game (RSS 17 Workshop)



Understanding Black Box Predictions via Influence Functions

Motivation

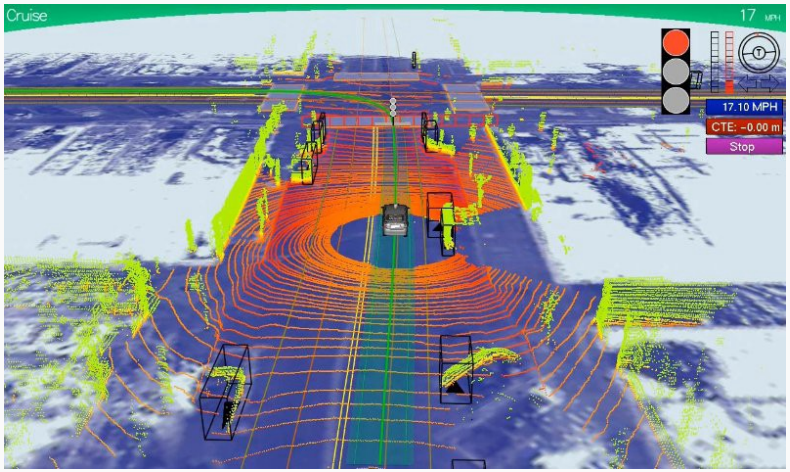


3,4

[3] From Voyage Auto, “An Introduction to Lidar”

[4] From Google Research

Motivation

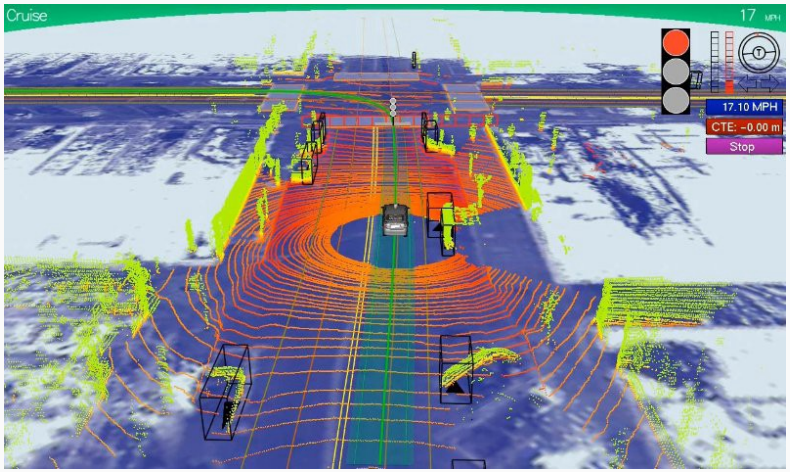


3,4

- How can we interpret trained models, and perform sanity checks?

[3] From Voyage Auto, “An Introduction to Lidar”
[4] From Google Research

Motivation

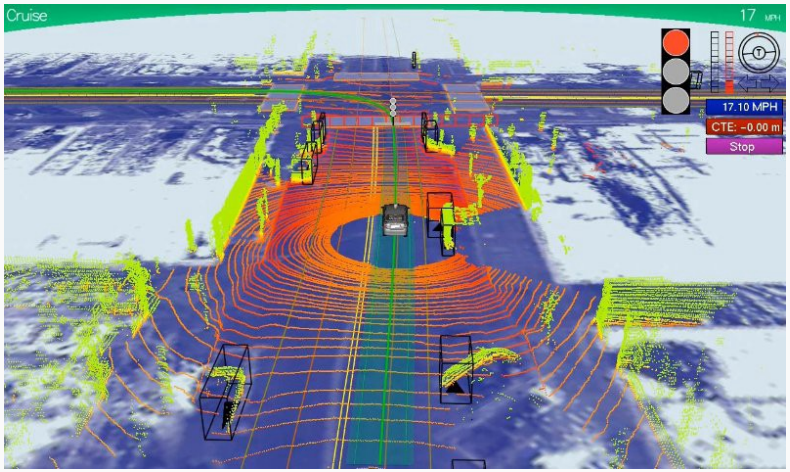


3,4

- How can we interpret trained models, and perform sanity checks?
- How can we avoid possible training-set and test-set attacks?

[3] From Voyage Auto, “An Introduction to Lidar”
[4] From Google Research

Motivation



3,4

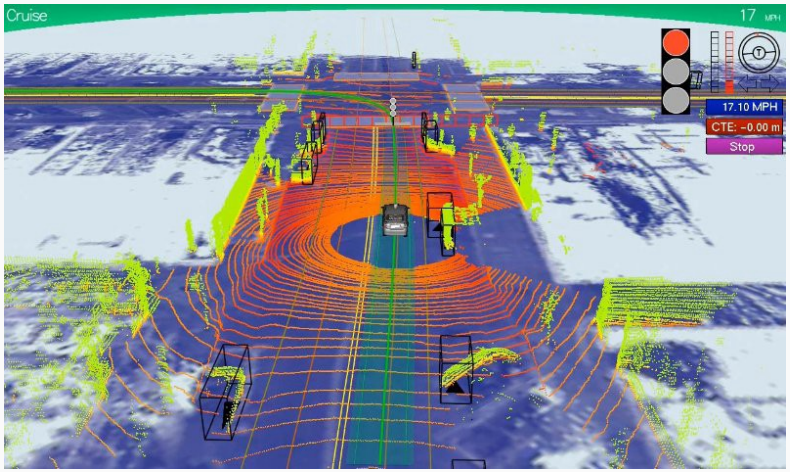
- How can we interpret trained models, and perform sanity checks?
- How can we avoid possible training-set and test-set attacks?
- How robust are our models to noise?

---

[3] From Voyage Auto, “An Introduction to Lidar”

[4] From Google Research

Motivation



3,4

- How can we interpret trained models, and perform sanity checks?
- How can we avoid possible training-set and test-set attacks?
- How robust are our models to noise?
- Strong need for quantitative analysis tools

[3] From Voyage Auto, “An Introduction to Lidar”
[4] From Google Research

Paper Contributions

- A scalable implementation of influence functions, parameterized over loss function

Paper Contributions

- A scalable implementation of influence functions, parameterized over loss function
- Evidence of usefulness for

Paper Contributions

- A scalable implementation of influence functions, parameterized over loss function
- Evidence of usefulness for
    - model understanding

Paper Contributions

- A scalable implementation of influence functions, parameterized over loss function
- Evidence of usefulness for
  - model understanding
  - generating adversarial training examples

Paper Contributions

- A scalable implementation of influence functions, parameterized over loss function
- Evidence of usefulness for
  - model understanding
  - generating adversarial training examples
  - debugging domain mismatch

Paper Contributions

- A scalable implementation of influence functions, parameterized over loss function
- Evidence of usefulness for
  - model understanding
  - generating adversarial training examples
  - debugging domain mismatch
  - fixing mislabeled examples

Paper Contributions

- A scalable implementation of influence functions, parameterized over loss function
- Evidence of usefulness for
  - model understanding
  - generating adversarial training examples
  - debugging domain mismatch
  - fixing mislabeled examples

Paper Contributions

- A scalable implementation of influence functions, parameterized over loss function
- Evidence of usefulness for
  - model understanding
  - generating adversarial training examples
  - debugging domain mismatch
  - fixing mislabeled examples

"otherwise high-performing models are still difficult to debug and fail catastrophically in the presence of changing data distributions and adversaries. . . it is critical to build tools to help us make machine learning more reliable 'in the wild.'" – Percy Liang

Much more work remains to be done! This is an analysis tool - what to do with results of analysis?

Context: Influence Functions

- 1980s: robust statistics: Cook, Weisberg *Residuals and influence in regression*

Context: Influence Functions

- 1980s: robust statistics: Cook, Weisberg *Residuals and influence in regression*
    - focused on linear models, exact solutions

Context: Influence Functions

- 1980s: robust statistics: Cook, Weisberg *Residuals and influence in regression*
    - focused on linear models, exact solutions
- 2004: *Robustness of Convex Risk Minimization Models* [5]

Context: Influence Functions

- 1980s: robust statistics: Cook, Weisberg *Residuals and influence in regression*
  - focused on linear models, exact solutions
- 2004: *Robustness of Convex Risk Minimization Models* [5]
  - $n = 500$, SVM with different kernels, focus on effect of adding a data point

Context: Influence Functions

- 1980s: robust statistics: Cook, Weisberg *Residuals and influence in regression*
  - focused on linear models, exact solutions
- 2004: *Robustness of Convex Risk Minimization Models* [5]
  - $n = 500$, SVM with different kernels, focus on effect of adding a data point
- 2016: *"Influence Sketching": Finding Influential Samples In Large-Scale Regressions* [6]

Context: Influence Functions

- 1980s: robust statistics: Cook, Weisberg *Residuals and influence in regression*
  - focused on linear models, exact solutions
- 2004: *Robustness of Convex Risk Minimization Models* [5]
  - $n = 500$, SVM with different kernels, focus on effect of adding a data point
- 2016: *"Influence Sketching": Finding Influential Samples In Large-Scale Regressions* [6]
  - randomized algorithm for approximating Cook's Distance (effect of deleting sample on linear model)

Context: Influence Functions

- 1980s: robust statistics: Cook, Weisberg *Residuals and influence in regression*
  - focused on linear models, exact solutions
- 2004: *Robustness of Convex Risk Minimization Models* [5]
  - $n = 500$, SVM with different kernels, focus on effect of adding a data point
- 2016: *"Influence Sketching": Finding Influential Samples In Large-Scale Regressions* [6]
  - randomized algorithm for approximating Cook's Distance (effect of deleting sample on linear model)
  - $n = 2$ million

Context: Interpretability and Verification

- Examine activation patterns for different inputs

[5] [7], Liu et. al. 2016

Context: Interpretability and Verification

- Examine activation patterns for different inputs

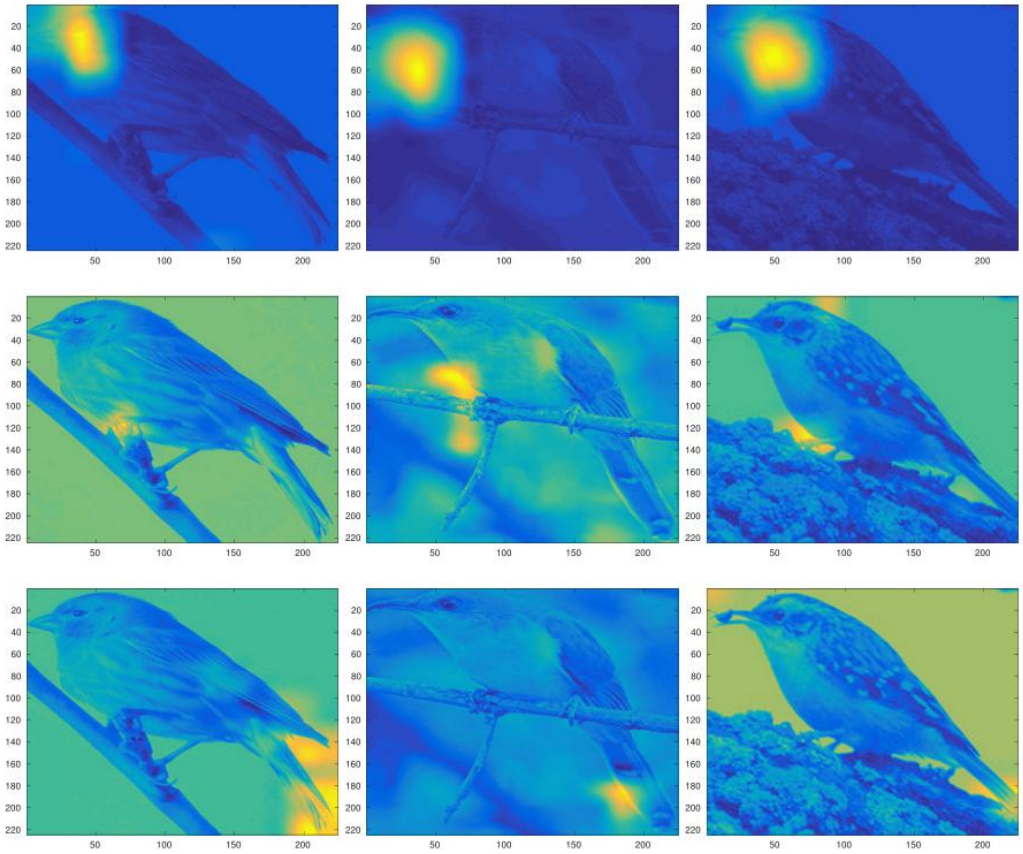


Fig. 3. Visualization of feature maps extracted from the conv5-4 layer of the VGG Net. Three feature maps and their activations on three different images are shown. Each row represents the feature map corresponding to the same filter. Warmer color indicates higher activation values.

5

[5][7], Liu et. al. 2016

Context: Interpretability and Verification

- Examine activation patterns for different inputs

Context: Interpretability and Verification

- Examine activation patterns for different inputs
- Qualitative explanations (*"Why Should I Trust You?"* [8])

Context: Interpretability and Verification

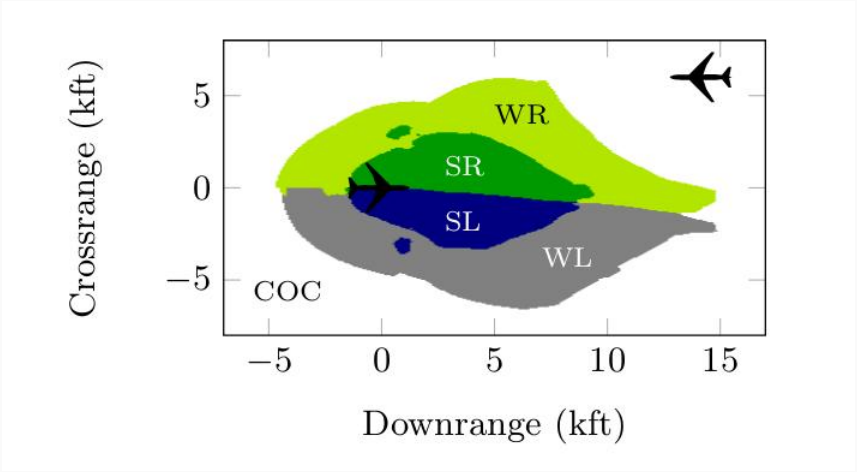
- Examine activation patterns for different inputs
- Qualitative explanations (*"Why Should I Trust You?"* [8])
- Prove invariants about predictions on subsets of input space (*"Reluplex: An Efficent SMT Solver for Verifying Deep Neural Networks"* [9])

Context: Interpretability and Verification

- Examine activation patterns for different inputs
- Qualitative explanations (*"Why Should I Trust You?"* [8])
- Prove invariants about predictions on subsets of input space (*"Reluplex: An Efficent SMT Solver for Verifying Deep Neural Networks"* [9])

Context: Interpretability and Verification

- Examine activation patterns for different inputs
- Qualitative explanations (*"Why Should I Trust You?"* [8])
- Prove invariants about predictions on subsets of input space (*"Reluplex: An Efficent SMT Solver for Verifying Deep Neural Networks"* [9])



A Different Approach

Instead of treating the model as **fixed**, treat the model as a **function of the training data**.

A Different Approach

Instead of treating the model as **fixed**, treat the model as a **function of the training data**.

Explore the **marginal effect** of each data point.

A Different Approach

Instead of treating the model as **fixed**, treat the model as a **function of the training data**.

Explore the **marginal effect** of each data point.

Get **quantitative** measure of *influence.*

Problem Formulation

What does it mean for a training point to be *influential*?

Problem Formulation

What does it mean for a training point to be *influential*?

For a given learned model (with known loss function):

- How would the model's predictions change if we **omit** a specific training point?

Problem Formulation

What does it mean for a training point to be *influential*?

For a given learned model (with known loss function):

- How would the model's predictions change if we **omit** a specific training point?
- How would the model's predictions change if we **perturb** a specific training point?

Problem Formulation

What does it mean for a training point to be *influential*?

For a given learned model (with known loss function):

- How would the model's predictions change if we **omit** a specific training point?
- How would the model's predictions change if we **perturb** a specific training point?

Problem Formulation

What does it mean for a training point to be *influential*?

For a given learned model (with known loss function):

- How would the model's predictions change if we **omit** a specific training point?
- How would the model's predictions change if we **perturb** a specific training point?

To approach these questions, study the *derivative* of the *loss* with respect to perturbation of a single training point.

Problem Formulation

What does it mean for a training point to be *influential*?

For a given learned model (with known loss function):

- How would the model's predictions change if we **omit** a specific training point?
- How would the model's predictions change if we **perturb** a specific training point?

To approach these questions, study the *derivative* of the *loss* with respect to perturbation of a single training point.

When this value is larger, that training point is more *influential*.

Definitions

predictor: $\mathcal{X} \to \mathcal{Y}$

Definitions

predictor: $\mathcal{X} \to \mathcal{Y}$

given training points $z_1, \ldots, z_n$, where $z_i \in \mathcal{X} \times \mathcal{Y}$

Definitions

predictor: $\mathcal{X} \to \mathcal{Y}$

given training points $z_1, \ldots, z_n$, where $z_i \in \mathcal{X} \times \mathcal{Y}$

trained parameters $\theta \in \Theta$

Definitions

predictor: $\mathcal{X} \to \mathcal{Y}$

given training points $z_1, \ldots, z_n$, where $z_i \in \mathcal{X} \times \mathcal{Y}$

trained parameters $\theta \in \Theta$

loss $L(z, \theta)$ and empirical risk $R(\theta) = \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta)$

- approach is agnostic to loss (but assumes convex, twice-differentiable wrt θ)
- Hessian $H_{\hat{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta}^2 L(z_i, \hat{\theta})$

Definitions

predictor: $\mathcal{X} \to \mathcal{Y}$

given training points $z_1, \ldots, z_n$, where $z_i \in \mathcal{X} \times \mathcal{Y}$

trained parameters $\theta \in \Theta$

loss $L(z, \theta)$ and empirical risk $R(\theta) = \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta)$

- approach is agnostic to loss (but assumes convex, twice-differentiable wrt θ)
- Hessian $H_{\hat{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 L(z_i, \hat{\theta})$

empirical risk minimizer $\hat{\theta} = \arg\min_{\theta \in \Theta} R(\theta)$

Influence

We want to find change in model parameters if training point z is removed, but we don't want to retrain!

Influence

We want to find change in model parameters if training point z is removed, but we don't want to retrain!

Instead, weight z by ϵ:

$$\hat{\theta}_{\epsilon,z} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$$

Influence

We want to find change in model parameters if training point z is removed, but we don't want to retrain!

Instead, weight z by ϵ:

$$\hat{\theta}_{\epsilon,z} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$$

Define influence as:

$$\mathcal{I}_{\hat{\theta}}(z) = \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}$$

$$\left.\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla L(z, \hat{\theta})$$

Removing and Perturbing Training Points

Propagate to find influence on loss:

$$\begin{aligned}\mathcal{I}_L(z, z_{test}) &= \frac{dL(z_{test}, \hat{\theta}_{\epsilon,z})}{d\epsilon}\Big|_{\epsilon=0} \\ &= -\nabla_\theta L(z_{\text{test}}, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta})\end{aligned}$$

this will be our working definition

Removing and Perturbing Training Points

Propagate to find influence on loss:

$$\begin{aligned}\mathcal{I}_L(z, z_{test}) &= \frac{dL(z_{test}, \hat{\theta}_{\epsilon,z})}{d\epsilon}\Big|_{\epsilon=0} \\ &= -\nabla_\theta L(z_{\text{test}}, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta})\end{aligned}$$

this will be our working definition

We can also measure the influence of perturbing the **value** of a training input, $z_\delta = (x + \delta, y)$:

Removing and Perturbing Training Points

Propagate to find influence on loss:

$$\begin{aligned}\mathcal{I}_L(z, z_{test}) &= \frac{dL(z_{test}, \hat{\theta}_{\epsilon,z})}{d\epsilon}\Big|_{\epsilon=0} \\ &= -\nabla_\theta L(z_{\text{test}}, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta})\end{aligned}$$

this will be our working definition

We can also measure the influence of perturbing the **value** of a training input, $z_\delta = (x + \delta, y)$:

$$\begin{aligned}\frac{d\hat{\theta}_{\epsilon,z_\delta,-z}}{d\epsilon}\Big|_{\epsilon=0} &= \mathcal{I}_{\hat{\theta}}(z_\delta) - \mathcal{I}_{\hat{\theta}}(z) \\ &= -H_{\hat{\theta}}^{-1}\left(\nabla_\theta L(z_\delta, \hat{\theta}) - \nabla_\theta L(z, \hat{\theta})\right). \qquad (1)\end{aligned}$$

Analysis - Remove Terms from Influence

Let $p(y \mid x) = \sigma(y\theta^\top x)$, with $y \in \{-1, 1\}$ and $\sigma(t) = \frac{1}{1+\exp(-t)}$.

Analysis - Remove Terms from Influence

Let $p(y \mid x) = \sigma(y\theta^\top x)$, with $y \in \{-1, 1\}$ and $\sigma(t) = \frac{1}{1+\exp(-t)}$.

For a training point $z = (x, y)$,

$$L(z, \theta) = \log(1 + \exp(-y\theta^\top x))$$

$$\nabla_\theta L(z, \theta) = -\sigma(-y\theta^\top x)yx$$

$$H_\theta = \frac{1}{n}\sum_{i=1}^{n} \sigma(\theta^\top x_i)\sigma(-\theta^\top x_i)x_i x_i^\top$$

Analysis - Remove Terms from Influence

Let $p(y \mid x) = \sigma(y\theta^\top x)$, with $y \in \{-1, 1\}$ and $\sigma(t) = \frac{1}{1+\exp(-t)}$.

For a training point $z = (x, y)$,

$$L(z, \theta) = \log(1 + \exp(-y\theta^\top x))$$
$$\nabla_\theta L(z, \theta) = -\sigma(-y\theta^\top x) yx$$
$$H_\theta = \frac{1}{n} \sum_{i=1}^{n} \sigma(\theta^\top x_i)\sigma(-\theta^\top x_i) x_i x_i^\top$$

and $\mathcal{I}_L(z, z_{\text{test}})$ is

$$-y_{\text{test}} y \cdot \sigma(-y_{\text{test}}\theta^\top x_{\text{test}}) \cdot \sigma(-y\theta^\top x) \cdot x_{\text{test}}^\top H_{\hat{\theta}}^{-1} x.$$

Analysis - Remove Terms from Influence

Let $p(y \mid x) = \sigma(y\theta^\top x)$, with $y \in \{-1, 1\}$ and $\sigma(t) = \frac{1}{1+\exp(-t)}$.

For a training point $z = (x, y)$,

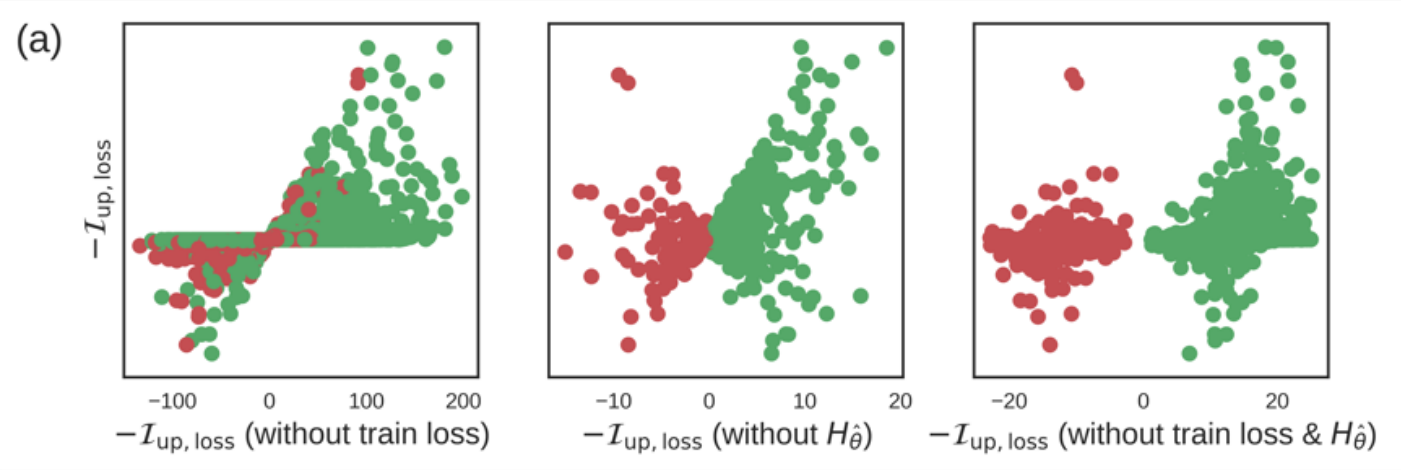
$$L(z, \theta) = \log(1 + \exp(-y\theta^\top x))$$
$$\nabla_\theta L(z, \theta) = -\sigma(-y\theta^\top x)yx$$
$$H_\theta = \frac{1}{n}\sum_{i=1}^{n} \sigma(\theta^\top x_i)\sigma(-\theta^\top x_i)x_i x_i^\top$$

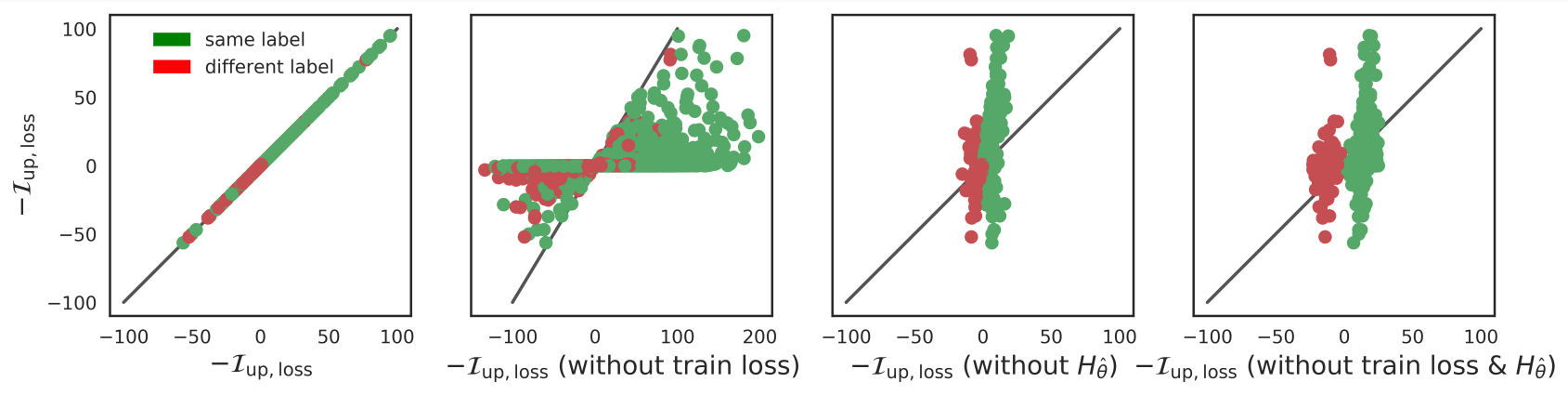
and $\mathcal{I}_L(z, z_{\text{test}})$ is

$$-y_{\text{test}} y \cdot \sigma(-y_{\text{test}}\theta^\top x_{\text{test}}) \cdot \sigma(-y\theta^\top x) \cdot \mathbf{x}_{\mathbf{test}}^\top H_{\hat{\theta}}^{-1}\mathbf{x}.$$

Analysis - Remove Terms from Influence (Fig 1)

$$-y_{\text{test}} y \cdot \sigma(-y_{\text{test}} \theta^\top x_{\text{test}}) \cdot \sigma(-y\theta^\top x) \cdot x_{\text{test}}^\top H_{\hat{\theta}}^{-1} x.$$





Efficiency

Two challenges:

1. Forming and inverting $H_{\hat{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta}^{2} L(z_i, \hat{\theta})$

Efficiency

Two challenges:

1. Forming and inverting $H_{\hat{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta}^2 L(z_i, \hat{\theta})$
    - n training points, $\theta \in \mathbb{R}^p$ requires $\mathcal{O}(np^2 + p^3)$ ops

Efficiency

Two challenges:

1. Forming and inverting $H_{\hat{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta}^2 L(z_i, \hat{\theta})$
    - n training points, $\theta \in \mathbb{R}^p$ requires $\mathcal{O}(np^2 + p^3)$ ops
2. Often want to calculate influence across all training points for a specific test point

How to Make Faster?

Overall approach:

- Efficiently approximate $s_{test} = H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z_{test}, \hat{\theta})$

How to Make Faster?

Overall approach:

- Efficiently approximate $s_{test} = H_{\hat{\theta}}^{-1} \nabla_\theta L(z_{test}, \hat{\theta})$
- Use this to efficiently compute $\mathcal{I}_L(z, z_{test})$ by just multiplying s_{test} by $\nabla_\theta L(z, \theta)$ as needed!

How to Make Faster?

Overall approach:

- Efficiently approximate $s_{test} = H_{\hat{\theta}}^{-1} \nabla_\theta L(z_{test}, \hat{\theta})$
- Use this to efficiently compute $\mathcal{I}_L(z, z_{test})$ by just multiplying s_{test} by $\nabla_\theta L(z, \theta)$ as needed!

How to Make Faster?

Overall approach:

- Efficiently approximate $s_{test} = H_{\hat{\theta}}^{-1} \nabla_\theta L(z_{test}, \hat{\theta})$
- Use this to efficiently compute $\mathcal{I}_L(z, z_{test})$ by just multiplying s_{test} by $\nabla_\theta L(z, \theta)$ as needed!

Conjugate Gradients + Stochastic Estimation ([10], Agarwal 2016)

Both automatically handled in systems like TensorFlow, Theano - users just specify L.

Speeds up calculating influence for all training points on a given test point to $\mathcal{O}(np)$.

When Does it Break?

Influence matches leave-one-out retraining

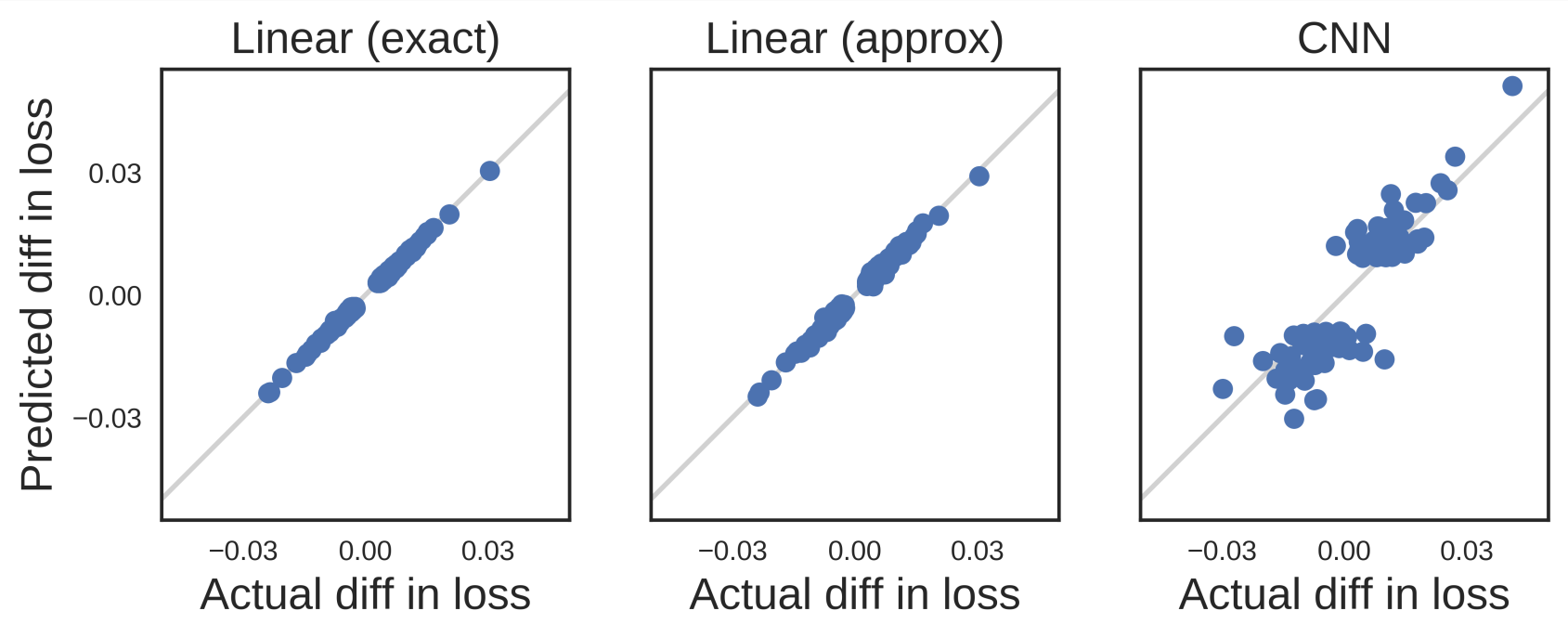


Fig 2 from paper. Left two: MNIST, $n = 55,000$. Right: CNN, tanh activations, not fully converged, $n = 500$, R=0.86

Non-differentiable losses

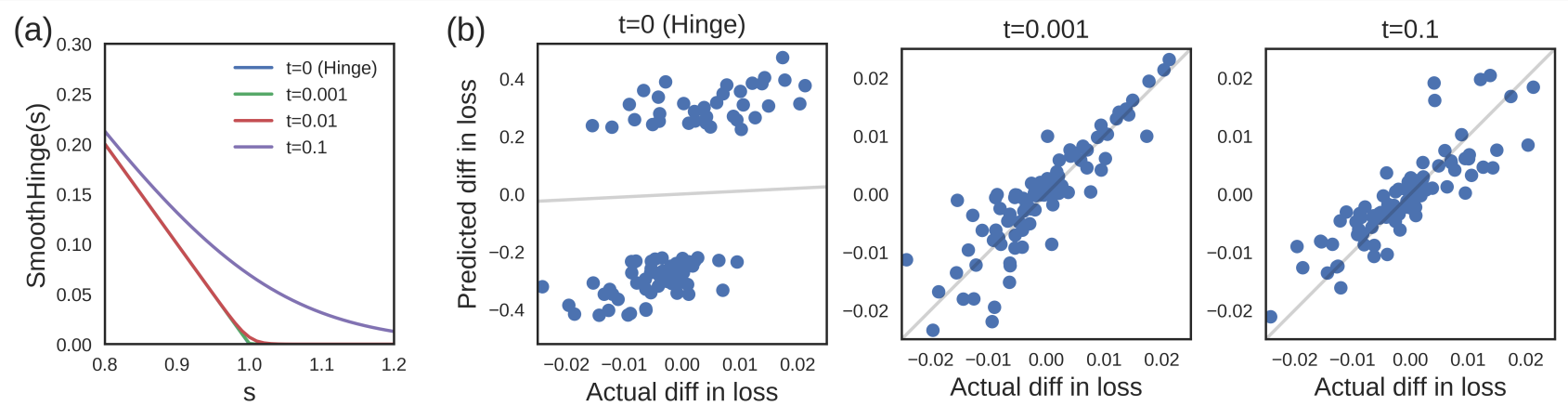


Fig 3 from paper. $smoothHinge(s, t) = t \log(1 + \exp(\frac{1-s}{t}))$

What to Use it For?

Understanding Model Behavior

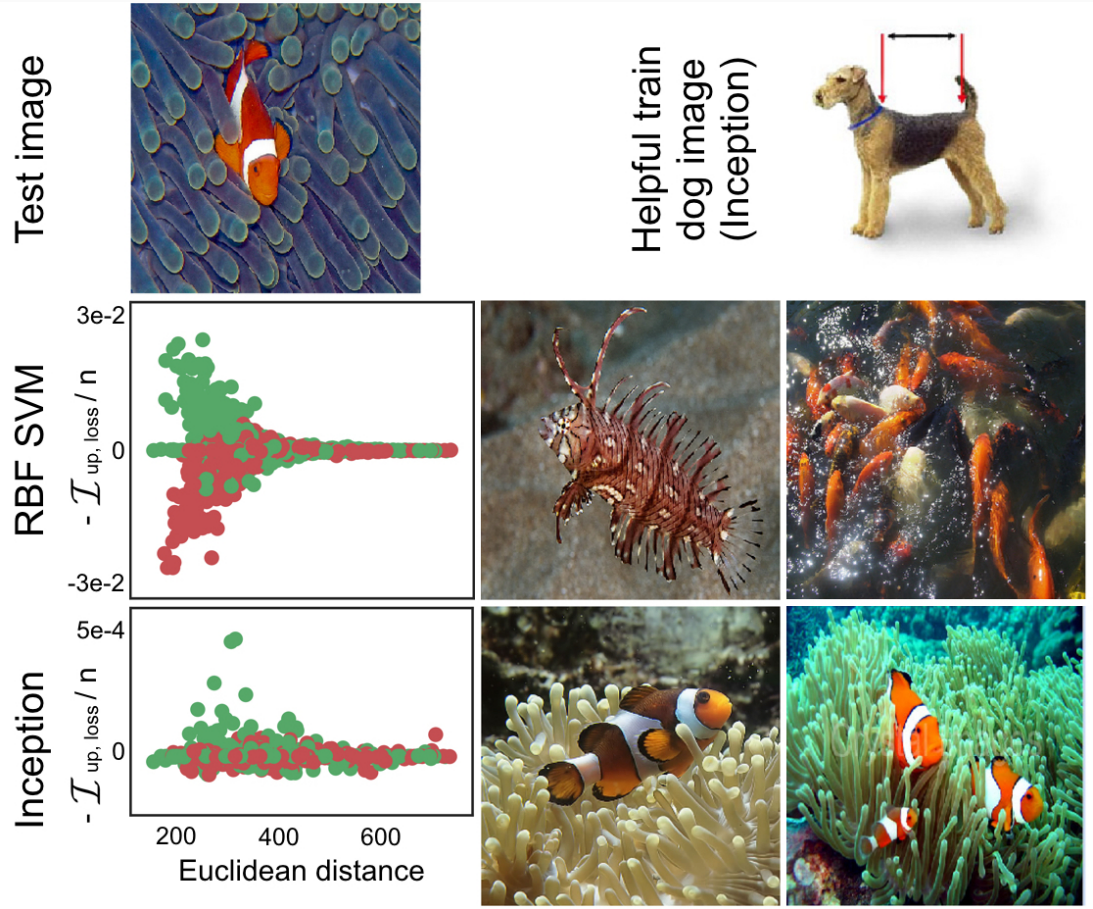


Figure 4 from paper. Examining most influential training points can provide insight.

Adversarial Training Examples

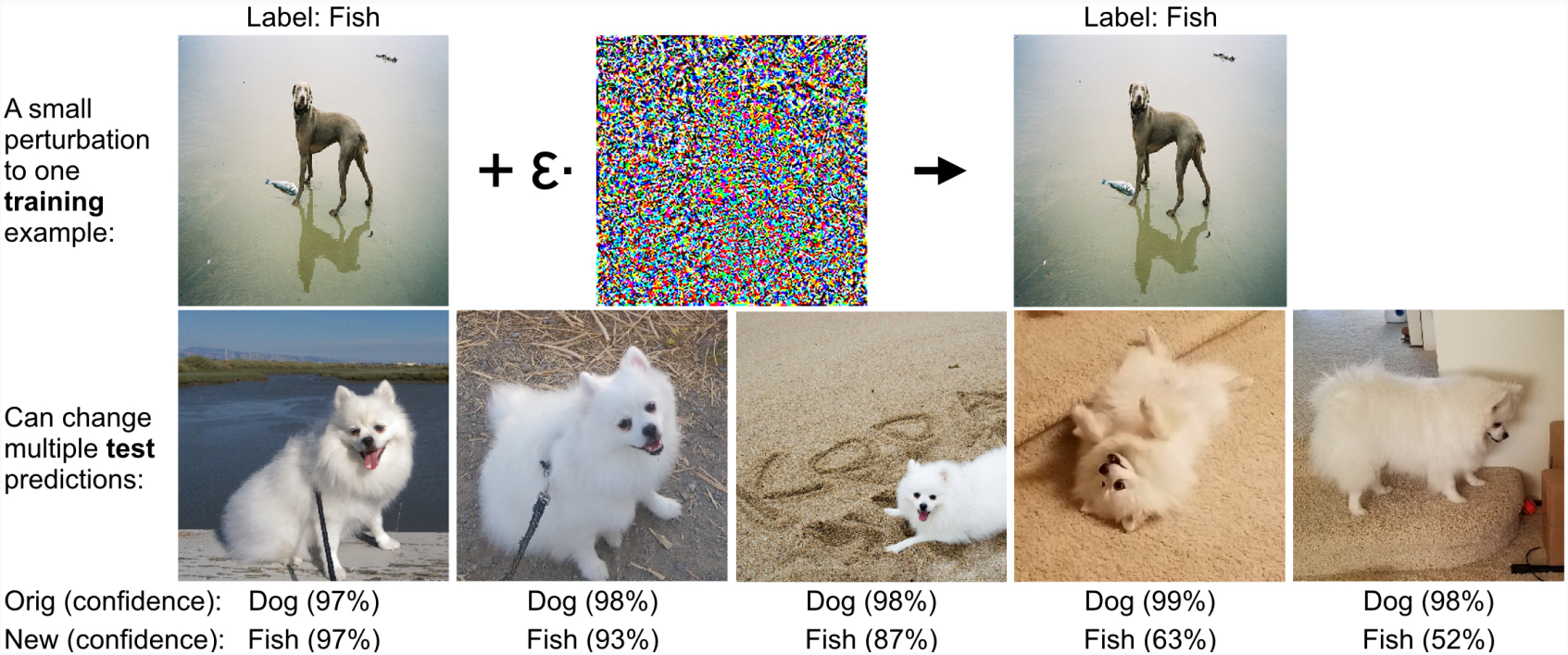


Fig 5 from paper. By maximizing the average loss over the test images, a visually-imperceptible change to a particular training image flips predictions on 16 test images.

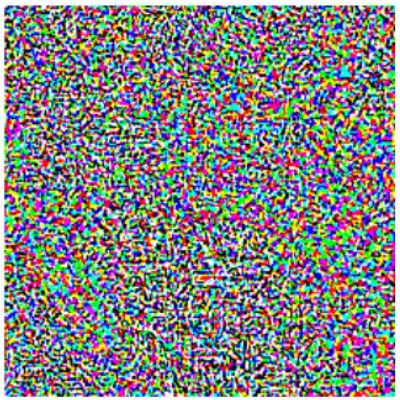
$$\mathcal{I}_{pert,loss}(z, z_{test}) = -\nabla_\theta L(z_{test}, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_x \nabla_\theta L(z, \hat{\theta})$$

Background: Adversarial Test Examples

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$=$

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

6

[6] [11] Goodfellow et. al. 2014

Background: Adversarial Test Examples

[7]

[7] [12] Evtimov et. al. 2017

Domain Mismatch

- 20,000 patients, 3 out of 24 children under 10 were re-admitted

Domain Mismatch

- 20,000 patients, 3 out of 24 children under 10 were re-admitted
- filter out 20 of these children who were not re-admitted, train

Domain Mismatch

- 20,000 patients, 3 out of 24 children under 10 were re-admitted
- filter out 20 of these children who were not re-admitted, train
- “child” feature coefficient was 15/127 in magnitude

Domain Mismatch

- 20,000 patients, 3 out of 24 children under 10 were re-admitted
- filter out 20 of these children who were not re-admitted, train
- “child” feature coefficient was 15/127 in magnitude
- calculate influence on all training points for mis-labeled child

Domain Mismatch

- 20,000 patients, 3 out of 24 children under 10 were re-admitted
- filter out 20 of these children who were not re-admitted, train
- “child” feature coefficient was 15/127 in magnitude
- calculate influence on all training points for mis-labeled child
    - four children were 30-40x more influential than other training examples

Fixing Mislabeled Examples

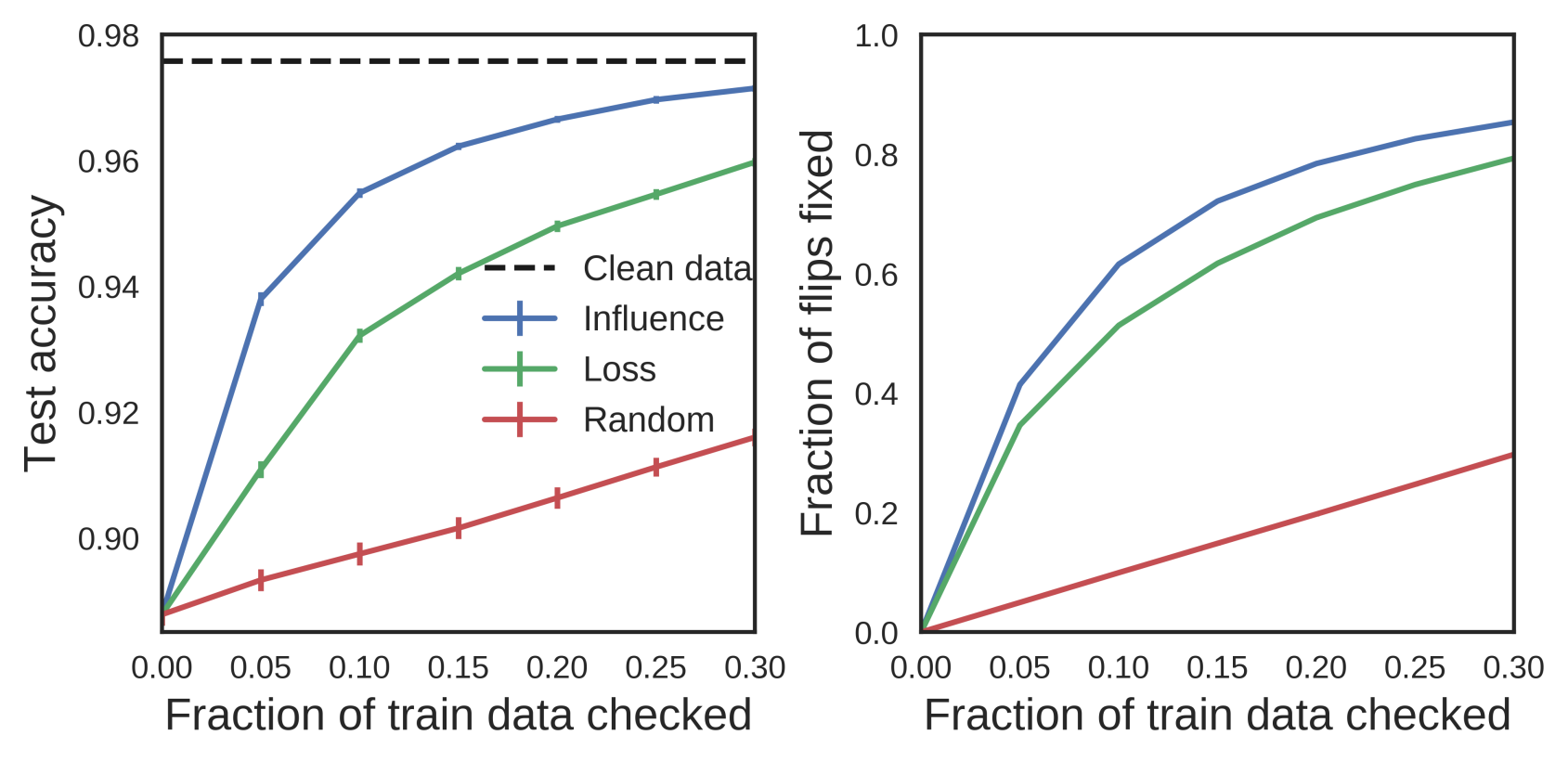


Fig 6 in paper. Prioritizing which training examples to check speeds up improvements.

Conclusion

Why Best Paper?

- Quantitative analysis tool for evaluating training sets
- Connects statistical technique with large-scale applications
- Usable “out of the box”: code and datasets available, parameterized over loss

Conclusion

What could be better / remaining questions?

- Still not applicable to every “black box”: many nonconvex, non-differentiable architectures

Conclusion

What could be better / remaining questions?

- Still not applicable to every “black box”: many nonconvex, non-differentiable architectures
- Comparisons between pixel space and feature space

Conclusion

What could be better / remaining questions?

- Still not applicable to every “black box”: many nonconvex, non-differentiable architectures
- Comparisons between pixel space and feature space
- Should analyze nonconvexity and nonconvergence separately, not together.

Conclusion

What could be better / remaining questions?

- Still not applicable to every “black box”: many nonconvex, non-differentiable architectures
- Comparisons between pixel space and feature space
- Should analyze nonconvexity and nonconvergence separately, not together.
- How to make datasets uniformly influential?

Conclusion

What could be better / remaining questions?

- Still not applicable to every “black box”: many nonconvex, non-differentiable architectures
- Comparisons between pixel space and feature space
- Should analyze nonconvexity and nonconvergence separately, not together.
- How to make datasets uniformly influential?
- Multiclass classification?

Conclusion

What could be better / remaining questions?

- Still not applicable to every “black box”: many nonconvex, non-differentiable architectures
- Comparisons between pixel space and feature space
- Should analyze nonconvexity and nonconvergence separately, not together.
- How to make datasets uniformly influential?
- Multiclass classification?
- Does not account for relationships between training points: removing a subset of the training set vs. just one point

Thank you!

Appendix

Sketch of Derivation

We want to find change in model parameters if training point z is removed, but we don't want to retrain

Instead, weight z by ϵ:

$$\hat{\theta}_{\epsilon,z} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$$

Sketch of Derivation

We want to find change in model parameters if training point z is removed, but we don't want to retrain

Instead, weight z by ϵ:

$$\hat{\theta}_{\epsilon,z} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$$

With $\Delta_\epsilon = \hat{\theta}_{\epsilon,z} - \hat{\theta}$, we can calculate influence as:

$$\mathcal{I}_{\hat{\theta}}(z) = \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon} = \frac{d\Delta_{\epsilon,z}}{d\epsilon}$$

Sketch of Derivation

$\hat{\theta}_{\epsilon,z}$ minimizes $R(\theta) + \epsilon L(z, \theta)$:

$$0 = \nabla R(\hat{\theta}_{\epsilon,z}) + \epsilon \nabla L(z, \hat{\theta}_{\epsilon,z})$$

Sketch of Derivation

$\hat{\theta}_{\epsilon,z}$ minimizes $R(\theta) + \epsilon L(z, \theta)$:

$$0 = \nabla R(\hat{\theta}_{\epsilon,z}) + \epsilon \nabla L(z, \hat{\theta}_{\epsilon,z})$$

Taylor expand the right hand side around $\hat{\theta}$

$$\begin{aligned} 0 \approx & \nabla R(\hat{\theta}) + \epsilon \nabla L(z, \hat{\theta}) + \\ & \nabla^2 R(\hat{\theta}) + \epsilon \nabla^2 L(z, \hat{\theta}) \Delta_\epsilon \end{aligned}$$

Sketch of Derivation

$\hat{\theta}_{\epsilon,z}$ minimizes $R(\theta) + \epsilon L(z, \theta)$:

$$0 = \nabla R(\hat{\theta}_{\epsilon,z}) + \epsilon \nabla L(z, \hat{\theta}_{\epsilon,z})$$

Taylor expand the right hand side around $\hat{\theta}$

$$\begin{aligned} 0 \approx & \nabla R(\hat{\theta}) + \epsilon \nabla L(z, \hat{\theta}) + \\ & \nabla^2 R(\hat{\theta}) + \epsilon \nabla^2 L(z, \hat{\theta}) \Delta_\epsilon \end{aligned}$$

and solve for Δ_ϵ

$$\begin{aligned} \Delta_\epsilon \approx & [-\nabla^2 R(\hat{\theta}) + \epsilon \nabla^2 L(z, \hat{\theta})]^{-1} \\ & \nabla R(\hat{\theta}) + \epsilon \nabla L(z, \hat{\theta}) \end{aligned}$$

Sketch of Derivation

But $\nabla R(\hat{\theta}) = 0$. Keeping only $O(\epsilon)$ terms, we have

$$\Delta_\epsilon \approx -\nabla^2 R(\hat{\theta})^{-1} \nabla L(z, \hat{\theta})\epsilon.$$

Sketch of Derivation

But $\nabla R(\hat{\theta}) = 0$. Keeping only $O(\epsilon)$ terms, we have

$$\Delta_\epsilon \approx -\nabla^2 R(\hat{\theta})^{-1} \nabla L(z, \hat{\theta})\epsilon.$$

We conclude that:

$$\begin{aligned} \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\Big|_{\epsilon=0} &= -H_{\hat{\theta}}^{-1} \nabla L(z, \hat{\theta}) \\ &= \mathcal{I}_{\hat{\theta}}(z). \end{aligned}$$

References

[1] L. H. Erickson and S. M. LaValle, “Toward the design and analysis of blind, bouncing robots,” in *IEEE international conference on robotics and automation*, 2013.

[2] A. Nilles, I. Becerra, and S. M. LaValle, “Periodic trajectories of mobile robots,” *IROS*, 2017.

[3] A. G. Fletcher, M. Osterfield, R. E. Baker, and S. Y. Shvartsman, “Vertex models of epithelial morphogenesis,” *Biophysical journal*, vol. 106, no. 11, pp. 2291–2304, 2014.

[4] B. E. Staveley, “Molecular and developmental biology (biol3530),” *Department of Biology, Memorial University of Newfoundland*.

[5] A. Christmann and I. Steinwart, “On robustness properties of convex risk minimization methods for pattern recognition,” *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 1007–1034, 2004.

[6] M. Wojnowicz, B. Cruz, X. Zhao, B. Wallace, M. Wolff, J. Luan, and C. Crable, “‘Influence sketching’: Finding influential samples in large-scale regressions,” in *Big data (big data), 2016 ieee international conference on*, 2016, pp. 3601–3612.

[7] L. Liu, C. Shen, and A. van den Hengel, “Cross-convolutional-layer pooling for image recognition,” *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should i trust you?: Explaining the predictions of any classifier,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[9] G. Katz, C. W. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, “Reluplex: An efficient SMT solver for verifying deep neural networks,” *CoRR*, vol. abs/1702.01135, 2017.

[10] N. Agarwal, B. Bullins, and E. Hazan, “Second order stochastic optimization in linear time,” *arXiv preprint arXiv:1602.03943*, 2016.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *arXiv preprint arXiv:1412.6572*, 2014.