Right for the Right Reasons:

Training Differentiable Models by Constraining their Explanations

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez
August 24, 2017, IJCAI, Melbourne

doi.org/10.24963/ijcai.2017/371
Code & data: github.com/dtak/rrr
These slides: goo.gl/fMZiRu

IACS
VE RI TAS
HARVARD
School of Engineering and Applied Sciences

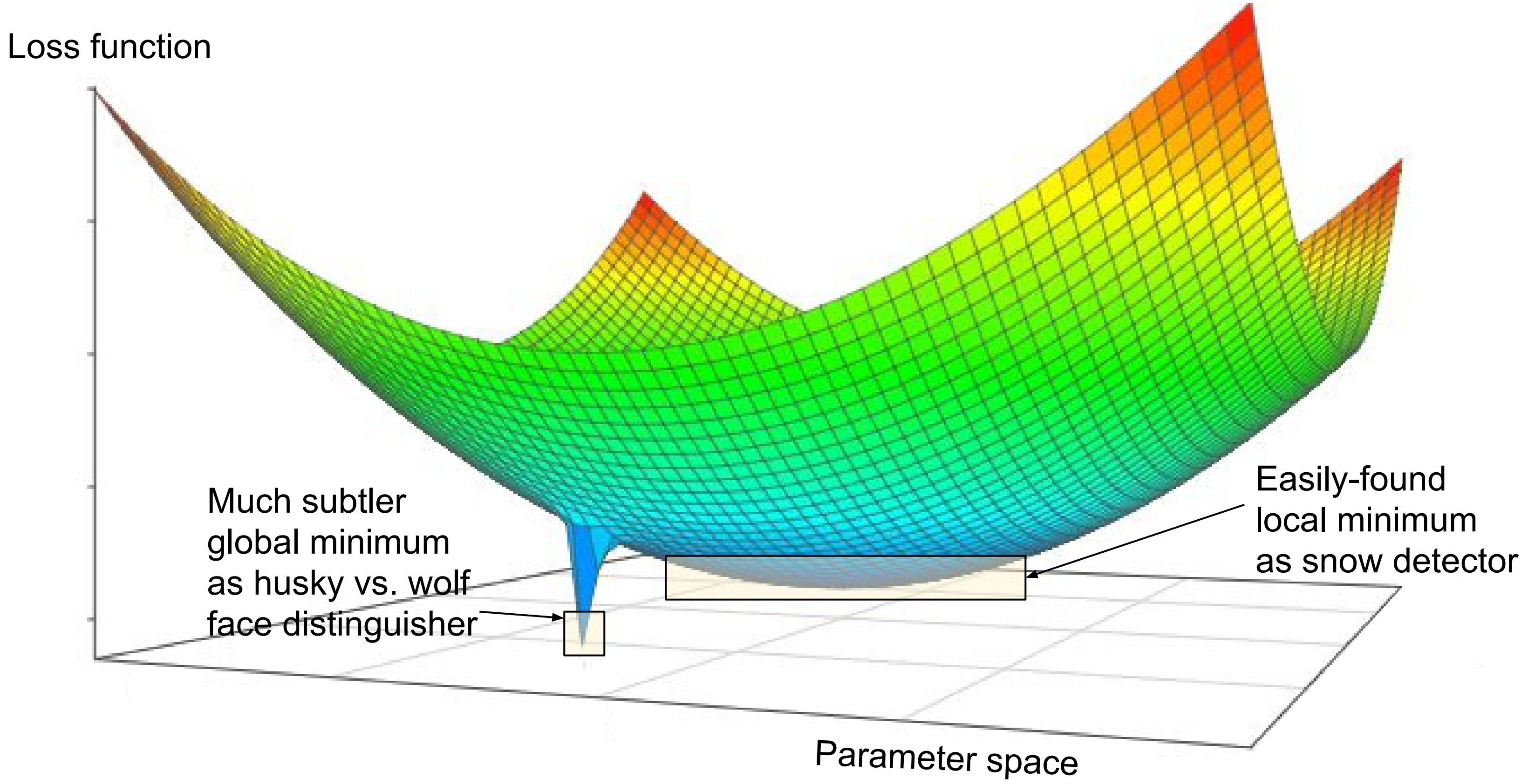
Models don’t always learn what you think they learn

[Ribiero et al., ACM 2016]

Models don't always learn what you think they learn

[Ribiero et al., ACM 2016]

The picture in my head



What are explanations?

What are explanations? (let me try to explain...)

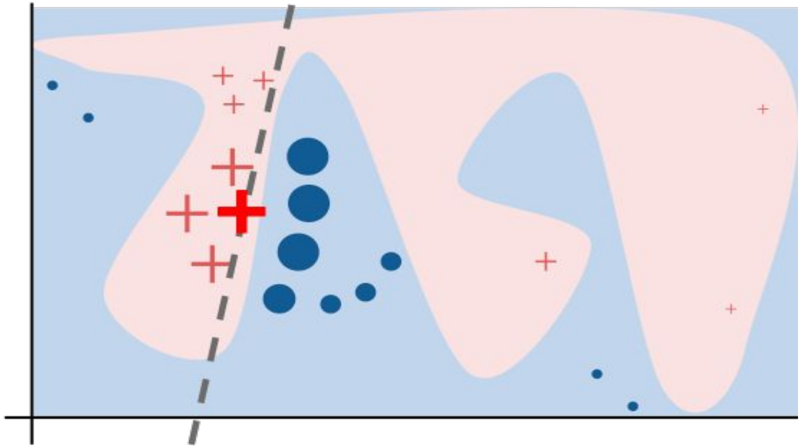
What are explanations? (let me try to explain...)

[Keil, 2006] [Miller, 2017]
[Doshi-Velez and Kim, 2017]
[Biran and Cotton, 2017]

What are explanations? (let me try to explain...)

[Keil, 2006] [Miller, 2017]
[Doshi-Velez and Kim, 2017]
[Biran and Cotton, 2017]

One approach: interpretable surrogates



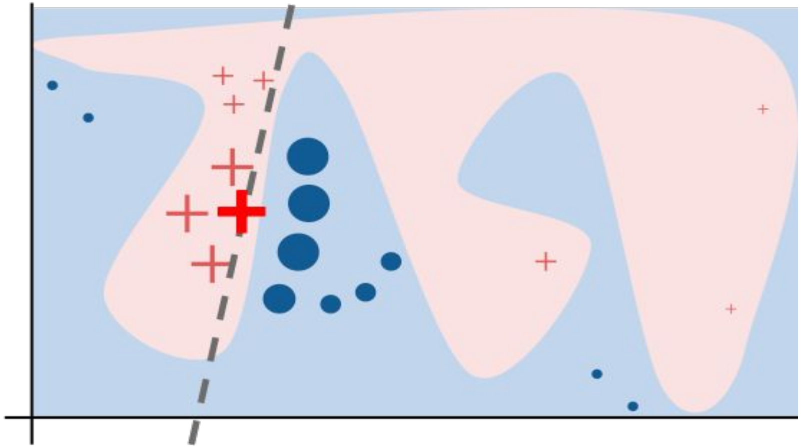
(a) Husky classified as wolf (b) Explanation

[Ribiero et al., ACM 2016, again]

What are explanations? (let me try to explain...)

[Keil, 2006] [Miller, 2017]
[Doshi-Velez and Kim, 2017]
[Biran and Cotton, 2017]

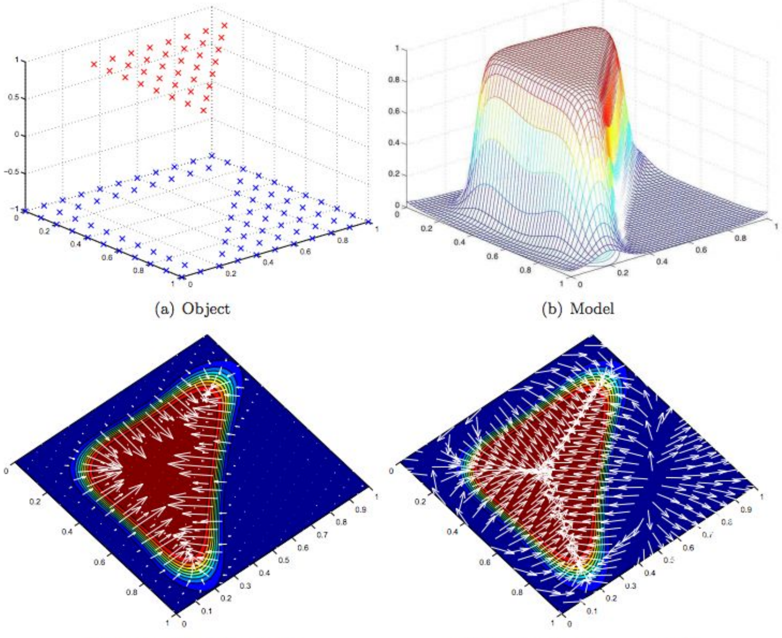
One approach: interpretable surrogates



(a) Husky classified as wolf (b) Explanation

[Ribiero et al., ACM 2016, again]

Another: gradients of output probabilities with respect to input features



(a) Object (b) Model

(c) Local explanation vectors (d) Direction of explanation vectors

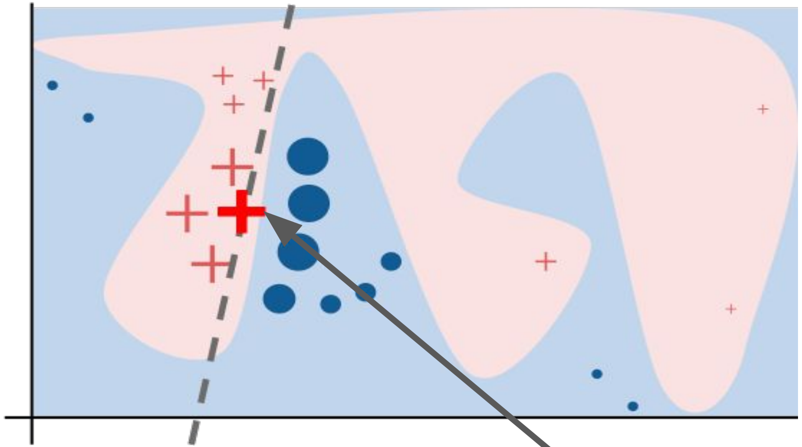
[Baehrens et al., JMLR 2010]

What are explanations?

(let me try to explain...)

[Keil, 2006] [Miller, 2017]
[Doshi-Velez and Kim, 2017]
[Biran and Cotton, 2017]

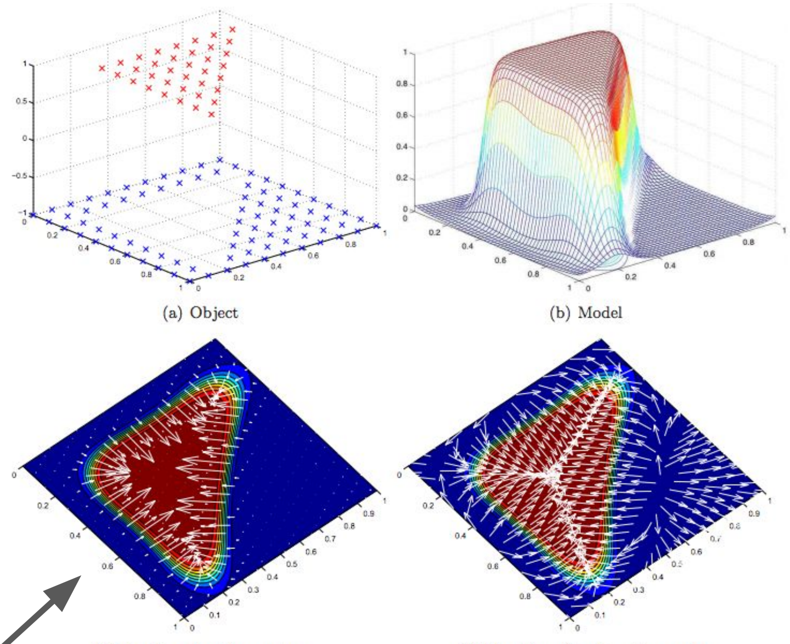
One approach: interpretable surrogates



(a) Husky classified as wolf (b) Explanation

[Ribiero et al., ACM 2016, again]

Another: gradients of output probabilities with respect to input features



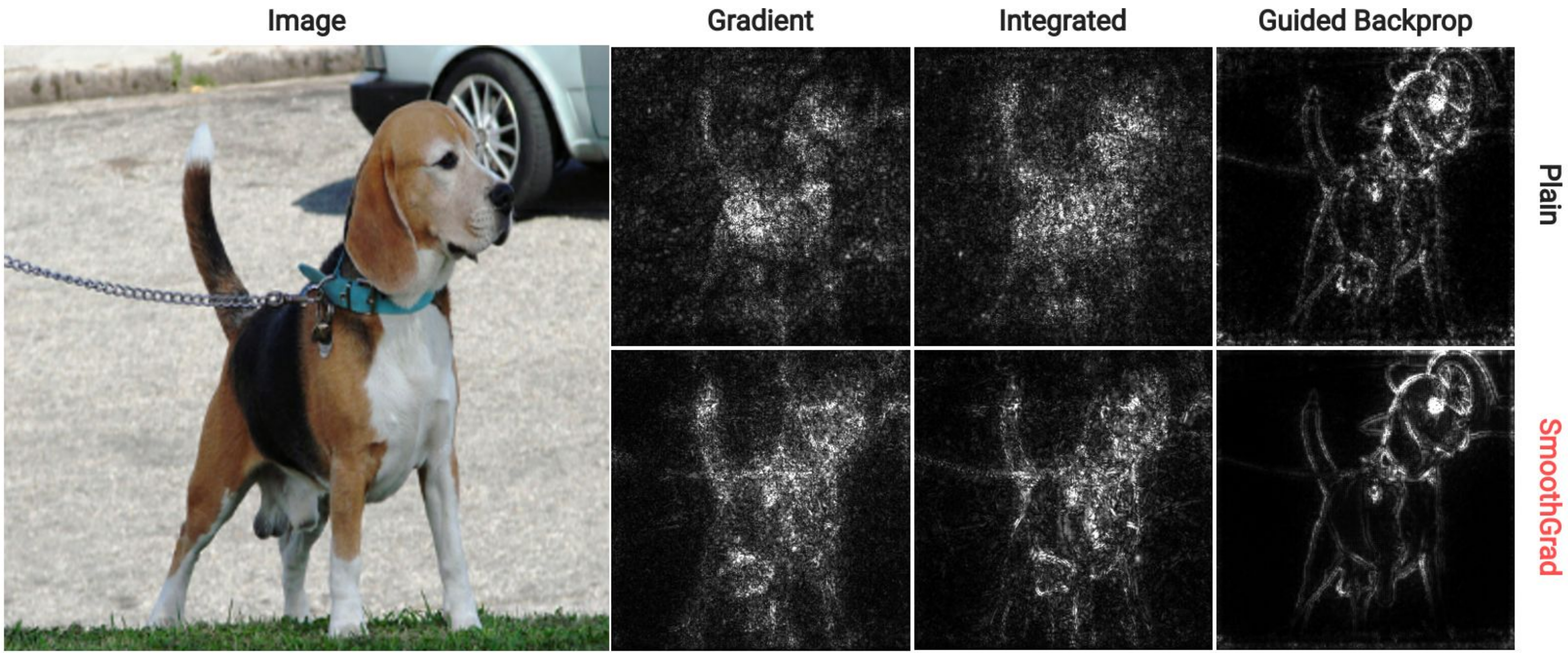
(a) Object (b) Model

(c) Local explanation vectors (d) Direction of explanation vectors

Actually quite similar!

[Baehrens et al., JMLR 2010]

Input gradients for image classifications



[Smilkov et al. 2017]

This kind of works!

So, we’re done, right?

This kind of works!

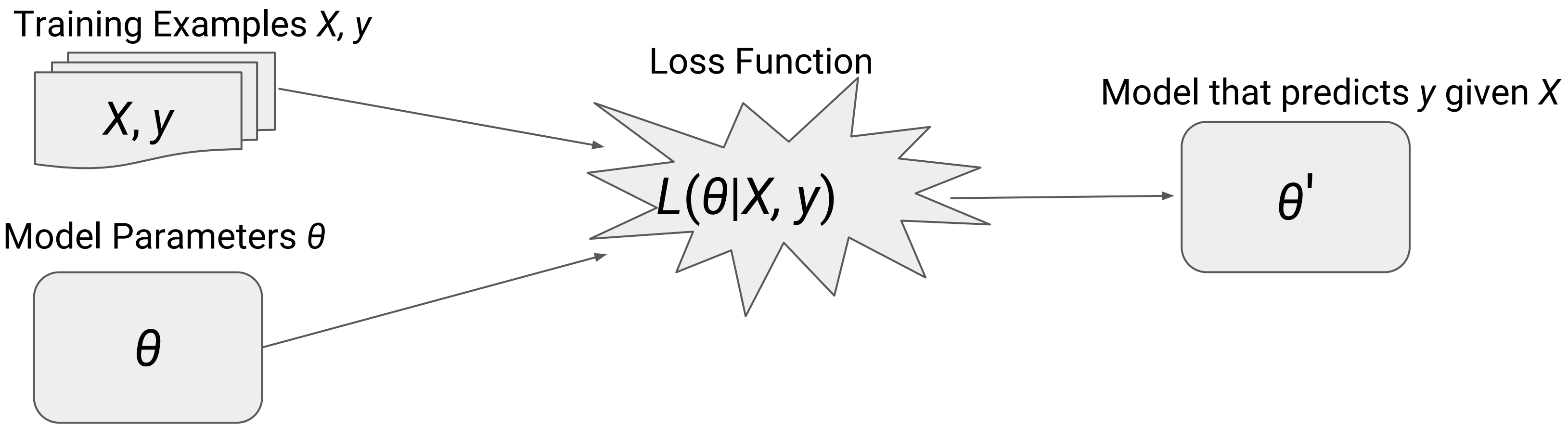
So, we’re done, right?

...what do we do if the explanations are wrong?

Optimizing for the right *reason*

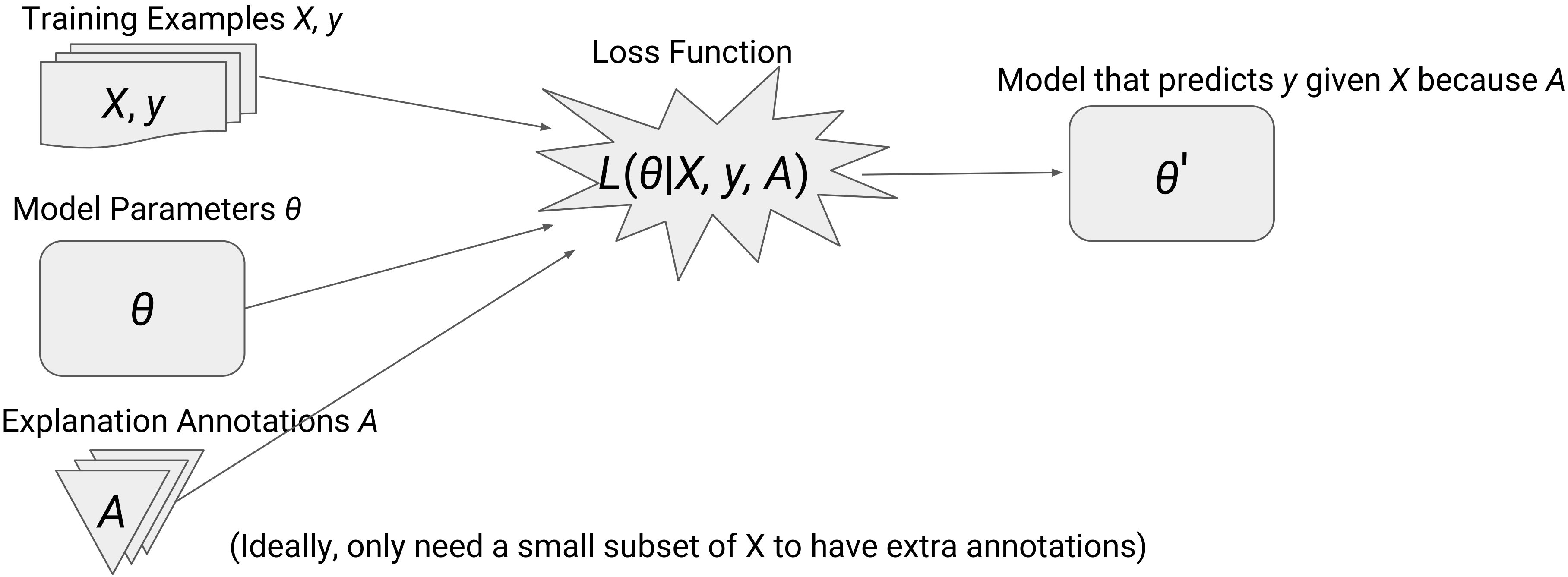
Optimizing for the right *reason*

Traditional ML



Optimizing for the right *reason*

Traditional ML + explanation regularization



(Ideally, only need a small subset of X to have extra annotations)

Case 1: Annotations are given

How we encode domain knowledge

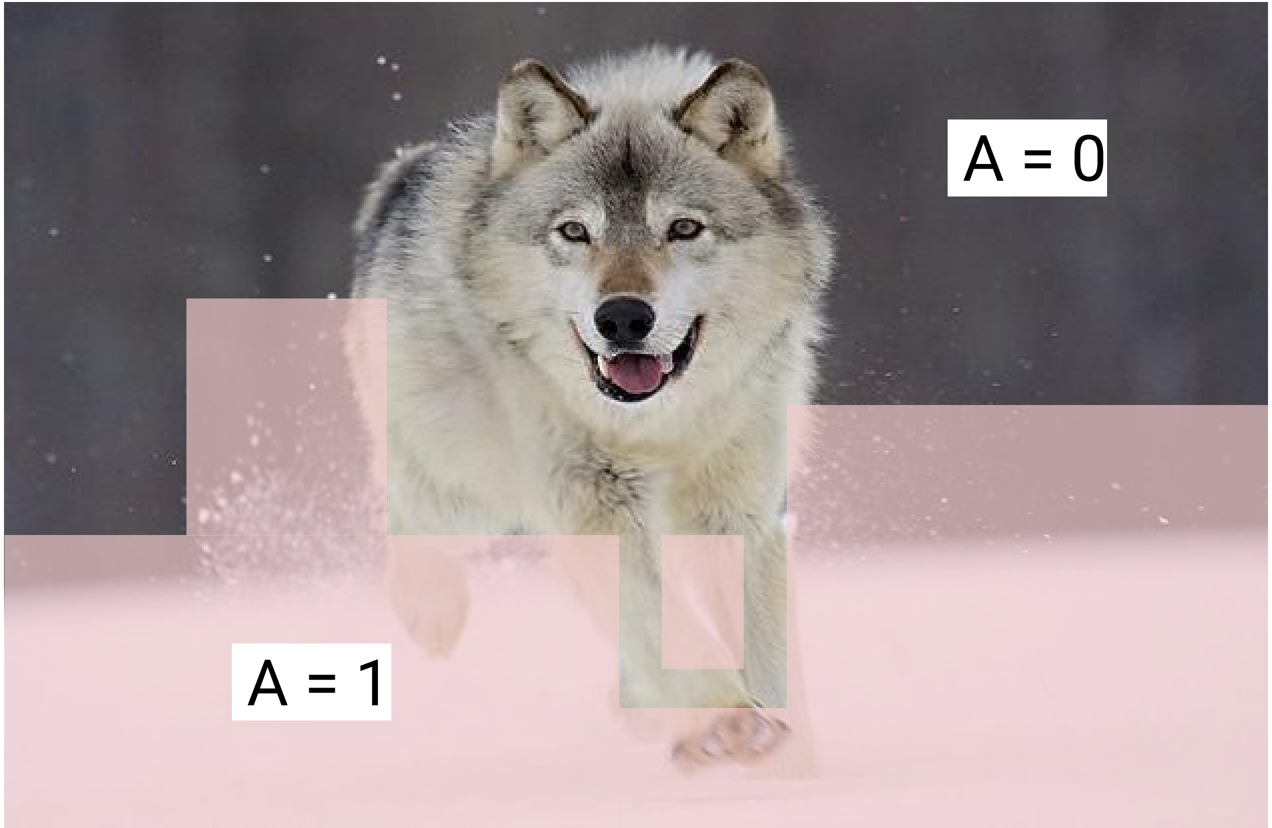
Features

$$A = \begin{bmatrix} 0 & \boxed{1} & 1 & 0 & 0 & 0 & 1 & \dots \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & \dots \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & \dots \\ \vdots & & & & & & & \end{bmatrix}$$

Examples

Signifies that second feature of first example should be *irrelevant* to model's prediction

Annotation Example



Our loss function

$$L(\theta, X, y, A) = \underbrace{\sum_{n=1}^{N} \sum_{k=1}^{K} -y_{nk} \log(\hat{y}_{nk})}_{\text{Right answers}} + \underbrace{\lambda_1 \sum_{n=1}^{N} \sum_{d=1}^{D} \left(A_{nd} \frac{\partial}{\partial x_d} \sum_{k=1}^{K} \log(\hat{y}_{nk}) \right)^2}_{\text{Right reasons}}$$

Our loss function

$$L(\theta, X, y, A) = \underbrace{\sum_{n=1}^{N} \sum_{k=1}^{K} -y_{nk} \log(\hat{y}_{nk})}_{\text{Right answers}} + \underbrace{\boxed{\lambda_1} \boxed{\sum_{n=1}^{N}} \boxed{\sum_{d=1}^{D}} \left(\boxed{A_{nd}} \boxed{\frac{\partial}{\partial x_d}} \boxed{\sum_{k=1}^{K} \log(\hat{y}_{nk})} \right)^2}_{\text{Right reasons}}$$

Our loss function

With some overall strength,

$$L(\theta, X, y, A) = \underbrace{\sum_{n=1}^{N}\sum_{k=1}^{K} -y_{nk}\log(\hat{y}_{nk})}_{\text{Right answers}} + \underbrace{\boxed{\lambda_1} \boxed{\sum_{n=1}^{N}}\boxed{\sum_{d=1}^{D}}\left(\boxed{A_{nd}}\boxed{\frac{\partial}{\partial x_d}}\boxed{\sum_{k=1}^{K}\log(\hat{y}_{nk})}\right)^2}_{\text{Right reasons}}$$

Our loss function

With some overall strength,

if a particular example

$$L(\theta, X, y, A) = \underbrace{\sum_{n=1}^{N}\sum_{k=1}^{K} -y_{nk}\log(\hat{y}_{nk})}_{\text{Right answers}} + \underbrace{\boxed{\lambda_1}\boxed{\sum_{n=1}^{N}}\boxed{\sum_{d=1}^{D}}\left(\boxed{A_{nd}}\boxed{\frac{\partial}{\partial x_d}}\boxed{\sum_{k=1}^{K}\log(\hat{y}_{nk})}\right)^2}_{\text{Right reasons}}$$

Our loss function

With some overall strength,

if a particular example's feature

$$L(\theta, X, y, A) = \underbrace{\sum_{n=1}^{N}\sum_{k=1}^{K} -y_{nk}\log(\hat{y}_{nk})}_{\text{Right answers}} + \underbrace{\boxed{\lambda_1}\boxed{\sum_{n=1}^{N}}\boxed{\sum_{d=1}^{D}}\left(\boxed{A_{nd}}\boxed{\frac{\partial}{\partial x_d}}\boxed{\sum_{k=1}^{K}\log(\hat{y}_{nk})}\right)^2}_{\text{Right reasons}}$$

Our loss function

With some overall strength,

if a particular example's feature

is marked irrelevant,

$$L(\theta, X, y, A) = \underbrace{\sum_{n=1}^{N}\sum_{k=1}^{K} -y_{nk}\log(\hat{y}_{nk})}_{\text{Right answers}} + \underbrace{\boxed{\lambda_1}\boxed{\sum_{n=1}^{N}}\boxed{\sum_{d=1}^{D}}\left(\boxed{A_{nd}}\boxed{\frac{\partial}{\partial x_d}}\boxed{\sum_{k=1}^{K}\log(\hat{y}_{nk})}\right)^2}_{\text{Right reasons}}$$

Our loss function

With some overall strength,

if a particular example's feature

is marked irrelevant,

$$L(\theta, X, y, A) = \underbrace{\sum_{n=1}^{N}\sum_{k=1}^{K} -y_{nk}\log(\hat{y}_{nk})}_{\text{Right answers}} + \underbrace{\boxed{\lambda_1}\boxed{\sum_{n=1}^{N}}\boxed{\sum_{d=1}^{D}}\left(\boxed{A_{nd}}\boxed{\frac{\partial}{\partial x_d}}\boxed{\sum_{k=1}^{K}\log(\hat{y}_{nk})}\right)^2}_{\text{Right reasons}}$$

then penalize,

Our loss function

With some overall strength,

if a particular example's feature

is marked irrelevant,

$$L(\theta, X, y, A) = \underbrace{\sum_{n=1}^{N}\sum_{k=1}^{K} -y_{nk}\log(\hat{y}_{nk})}_{\text{Right answers}} + \underbrace{\boxed{\lambda_1}\boxed{\sum_{n=1}^{N}}\boxed{\sum_{d=1}^{D}}\left(\boxed{A_{nd}}\boxed{\frac{\partial}{\partial x_d}}\boxed{\sum_{k=1}^{K}\log(\hat{y}_{nk})}\right)^2}_{\text{Right reasons}}$$

then penalize,

when the feature changes,

Our loss function

With some overall strength,

if a particular example's feature

is marked irrelevant,

$$L(\theta, X, y, A) = \underbrace{\sum_{n=1}^{N}\sum_{k=1}^{K} -y_{nk}\log(\hat{y}_{nk})}_{\text{Right answers}} + \underbrace{\boxed{\lambda_1}\boxed{\sum_{n=1}^{N}}\boxed{\sum_{d=1}^{D}}\left(\boxed{A_{nd}}\boxed{\frac{\partial}{\partial x_d}}\boxed{\sum_{k=1}^{K}\log(\hat{y}_{nk})}\right)^2}_{\text{Right reasons}}$$

then penalize,

when the feature changes,

how much the prediction changes.

Experiments

Basic philosophy:

- use or create datasets that we know can by classified with *qualitatively different rules*
- see if we can use explanations to “select” which implicit rule the model learns

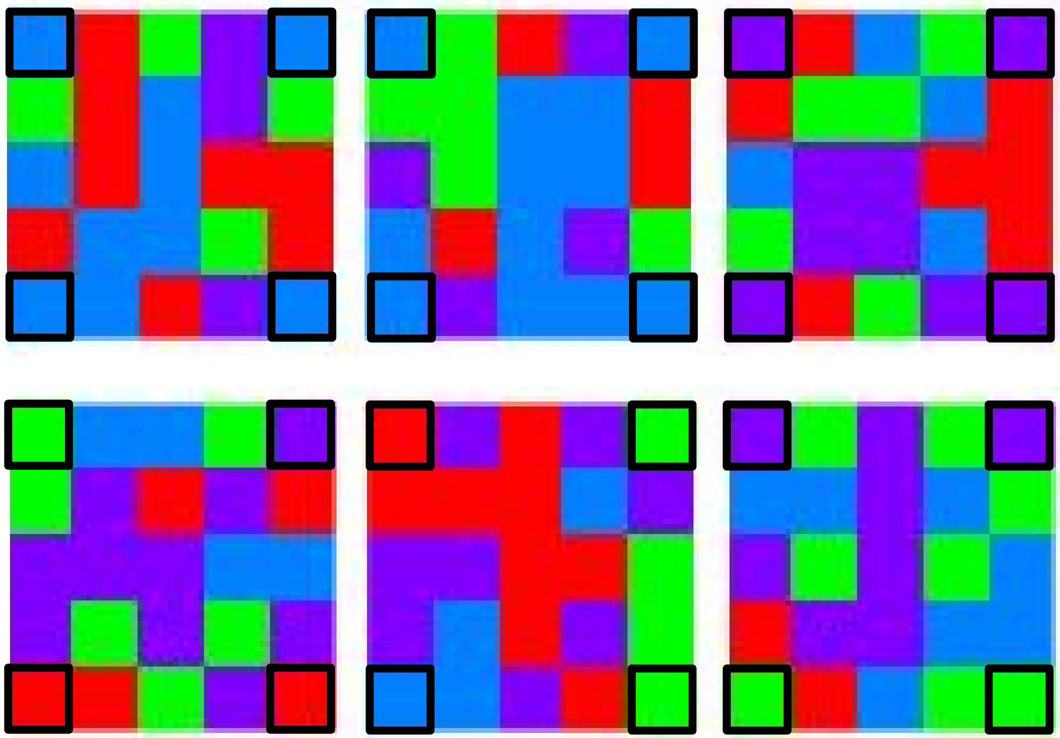
Models we used:

- 2 hidden layer fully connected network, but method works for CNNs and larger models

Experiments: Toy Colors

Class 1

Class 2

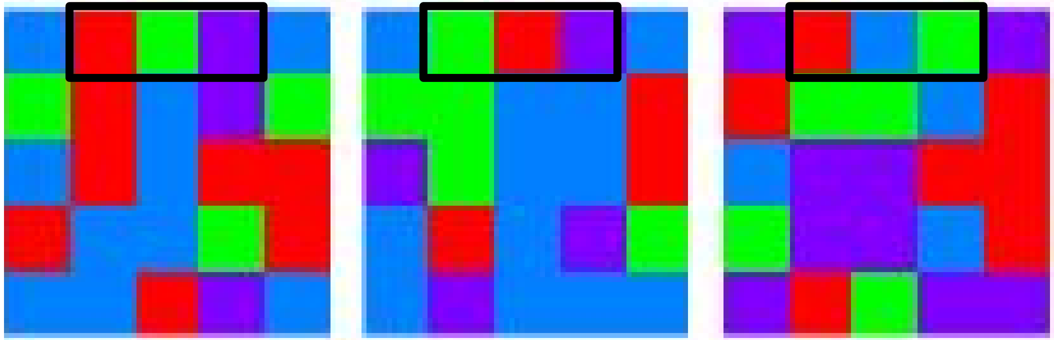


All colors shared

At least two different

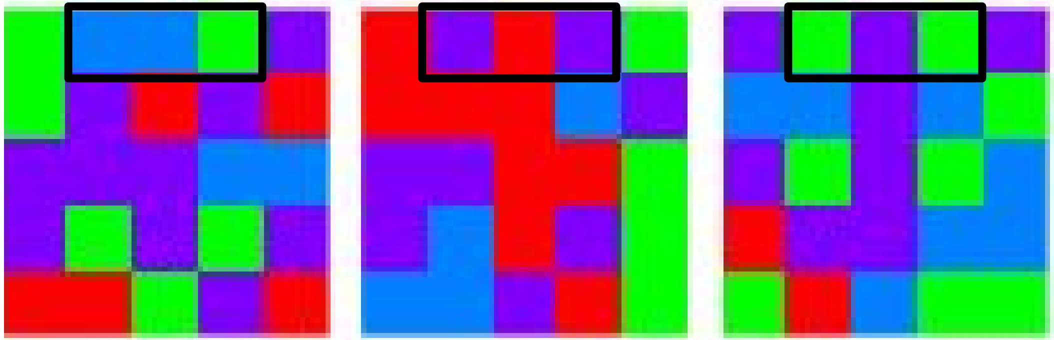
Experiments: Toy Colors

Class 1



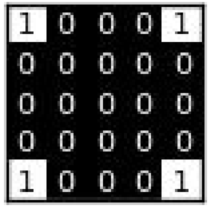
All colors different

Class 2

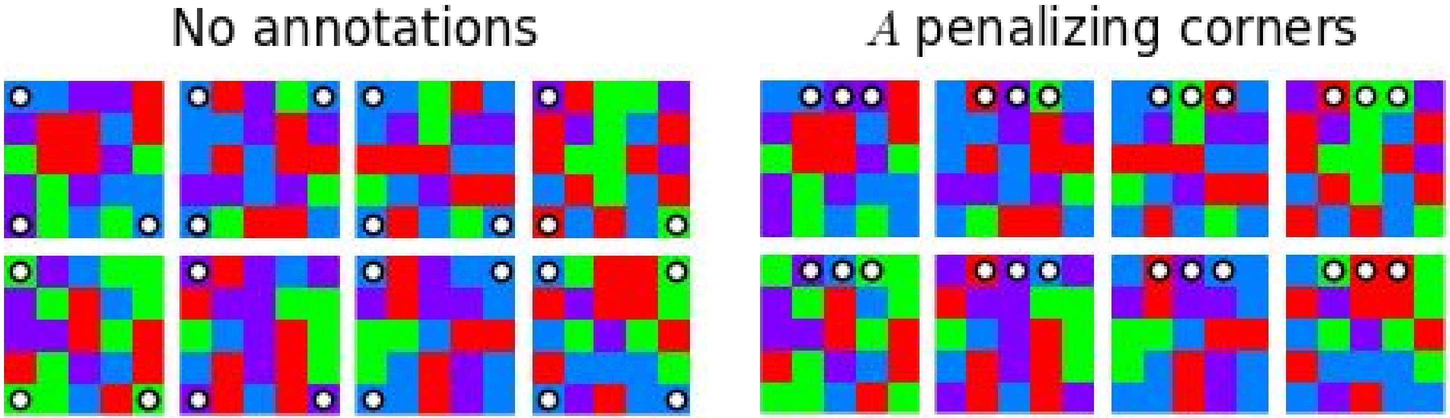


At least two shared

Learning an otherwise-unreachable rule



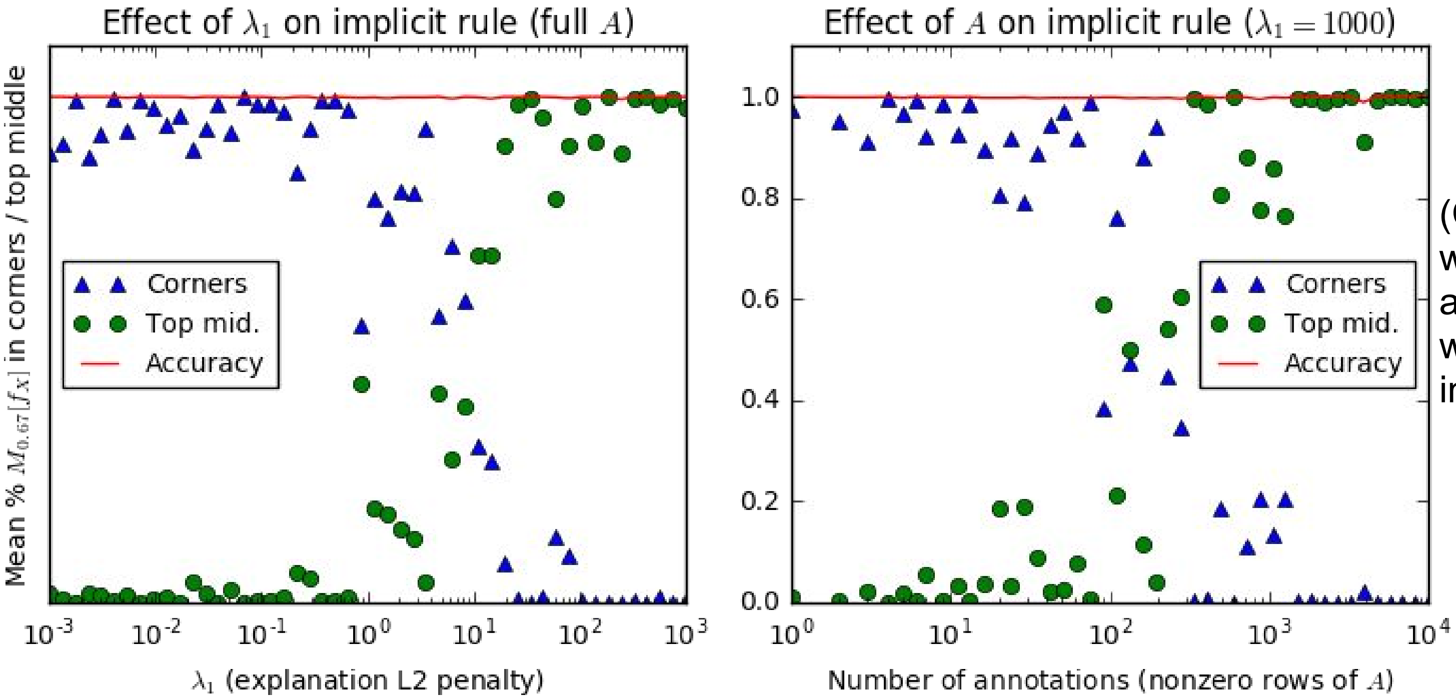
Pixels w/ largest magnitude gradients



By default, model appears to learn corner rule.

If we penalize corners, model discovers top-mid rule!

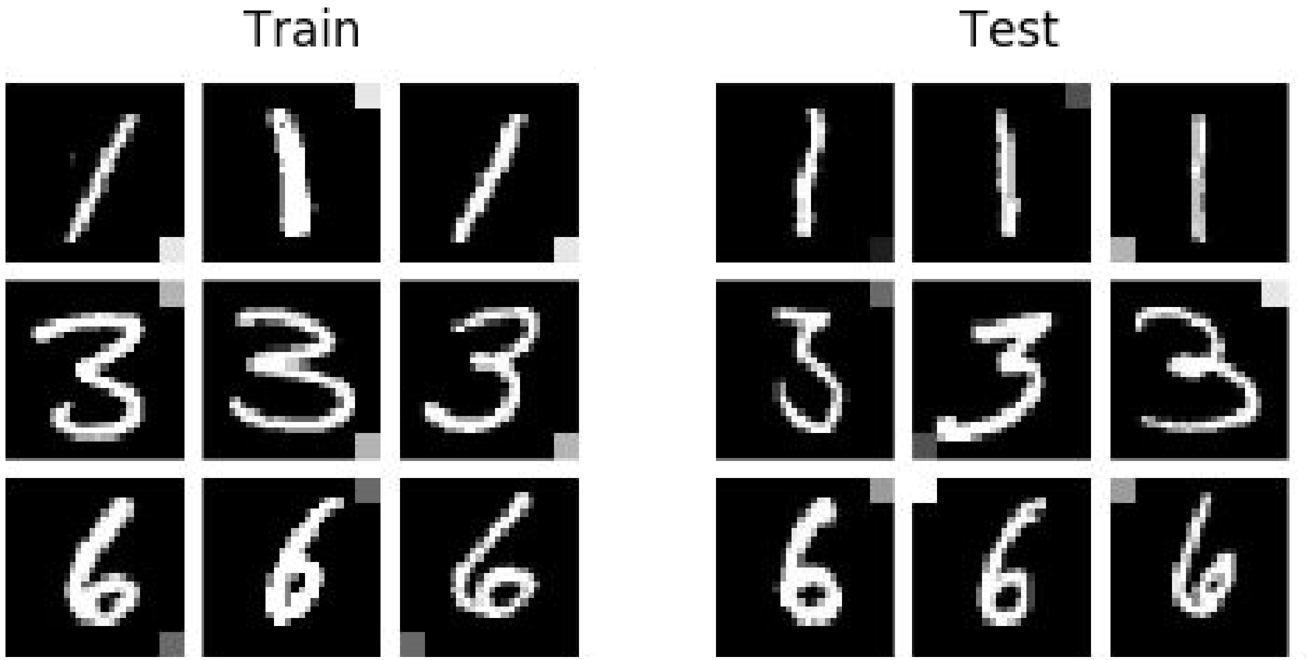
How regularization strength affects what we learn



(Can transition with 10s of annotations if we oversample in minibatches)

Smooth transition between model learning each rule!

Experiments: Decoy MNIST

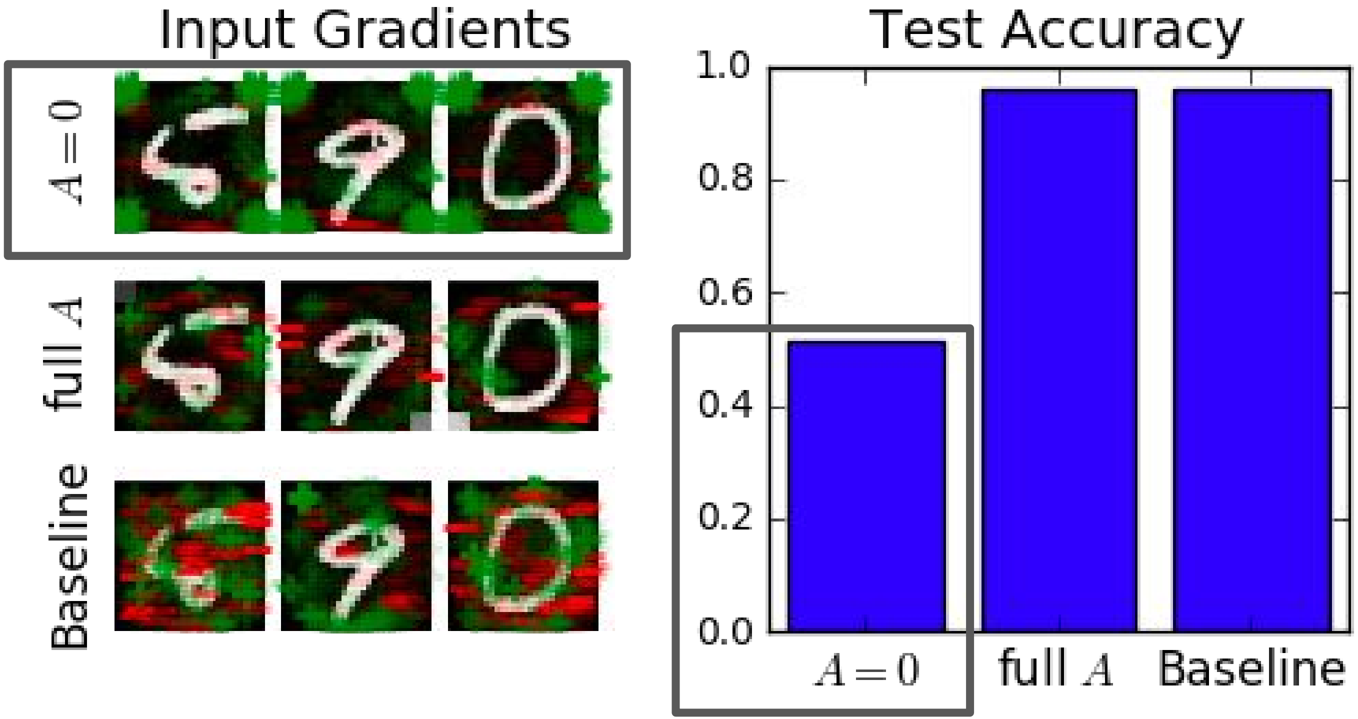


Swatch shades a simple function of y in train, but not in test.

Experiments: Decoy MNIST

= increasing pixel increases predicted label prob

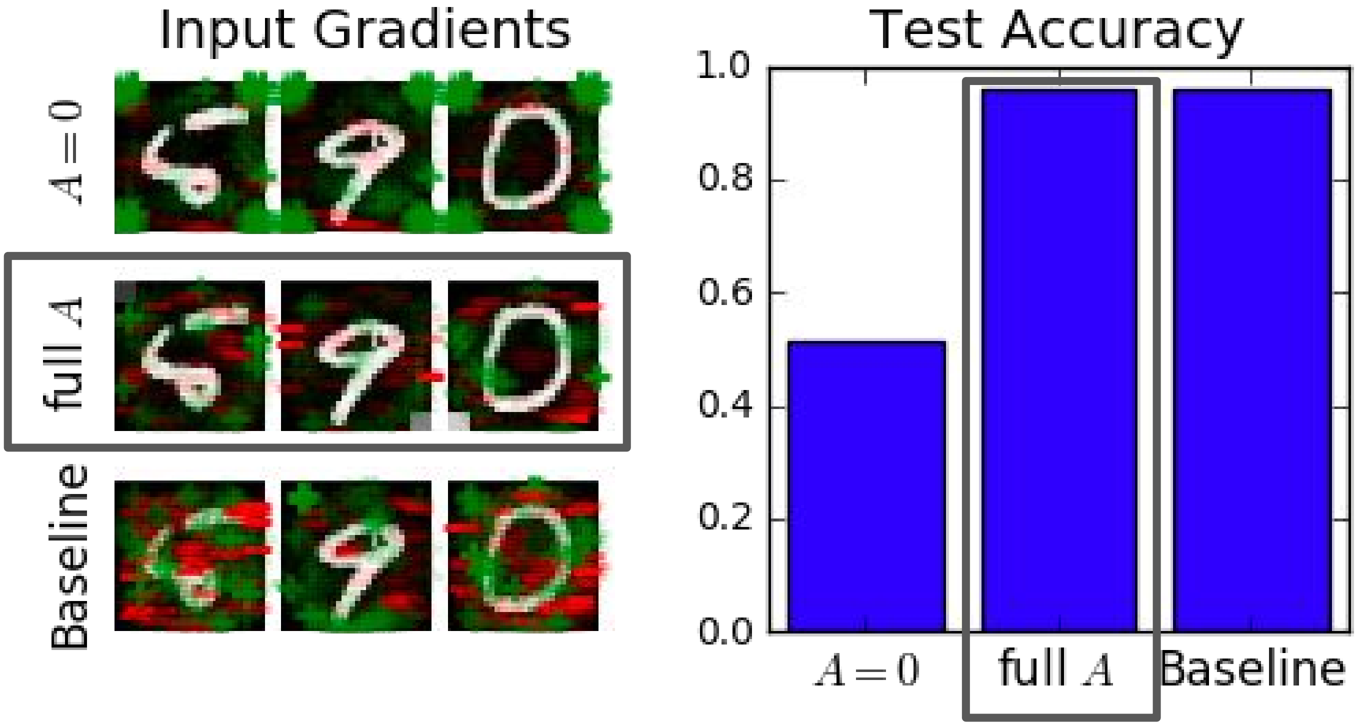
= increasing pixel decreases predicted label prob



Normal model has low accuracy; gradients focus on swatches

Experiments: Decoy MNIST

+ = increasing pixel increases predicted label prob
- = increasing pixel decreases predicted label prob



Model with gradient regularization recovers baseline accuracy!

Case 2: What if we don’t have annotations?

Find-another-explanation

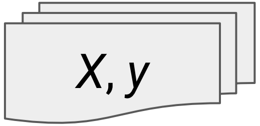
Find-another-explanation

Overall goal: obtain an ensemble of models that are all accurate but for different reasons.

Find-another-explanation

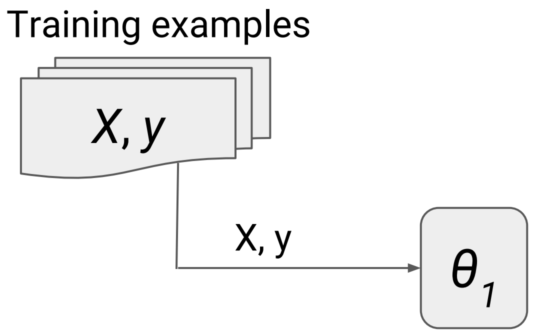
Overall goal: obtain an ensemble of models that are all accurate but for different reasons.

Training examples



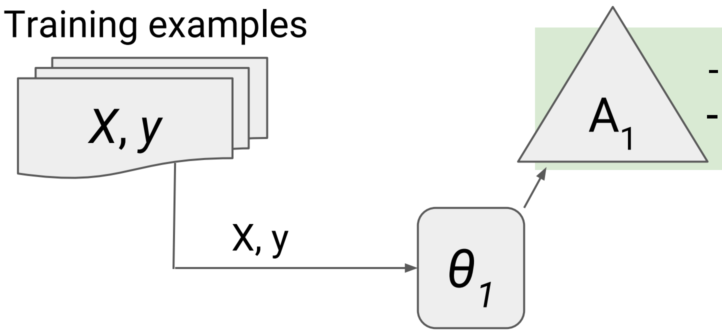
Find-another-explanation

Overall goal: obtain an ensemble of models that are all accurate but for different reasons.



Find-another-explanation

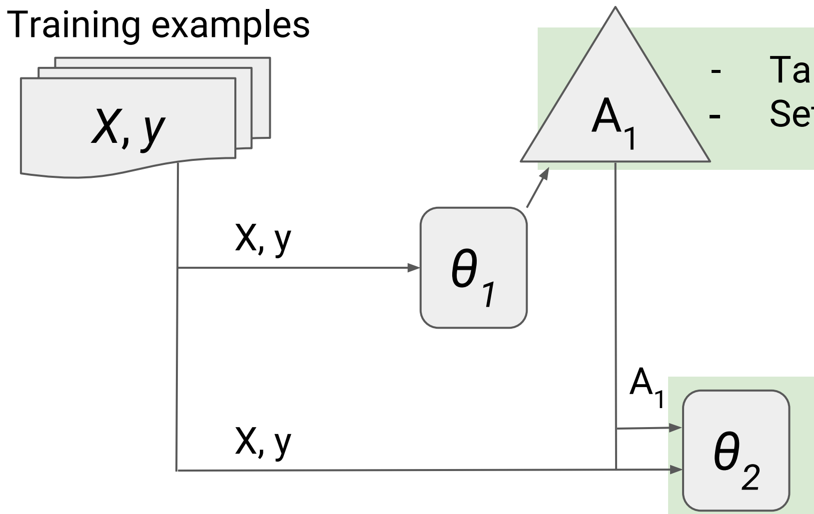
Overall goal: obtain an ensemble of models that are all accurate but for different reasons.



- Take 1st model gradients of full training set
- Set $A_{n,d} = 1$ where $|\partial f/\partial X_{nd}|$ is relatively large

Find-another-explanation

Overall goal: obtain an ensemble of models that are all accurate but for different reasons.

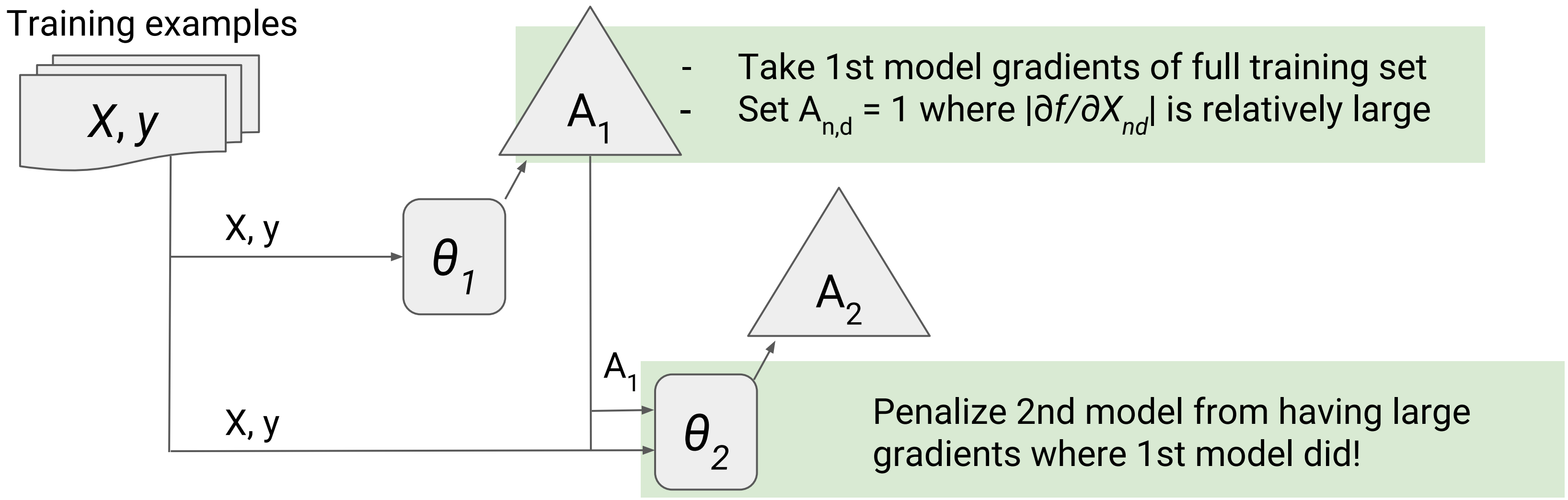


- Take 1st model gradients of full training set
- Set $A_{n,d}$ = 1 where $|\partial f/\partial X_{nd}|$ is relatively large

Penalize 2nd model from having large gradients where 1st model did!

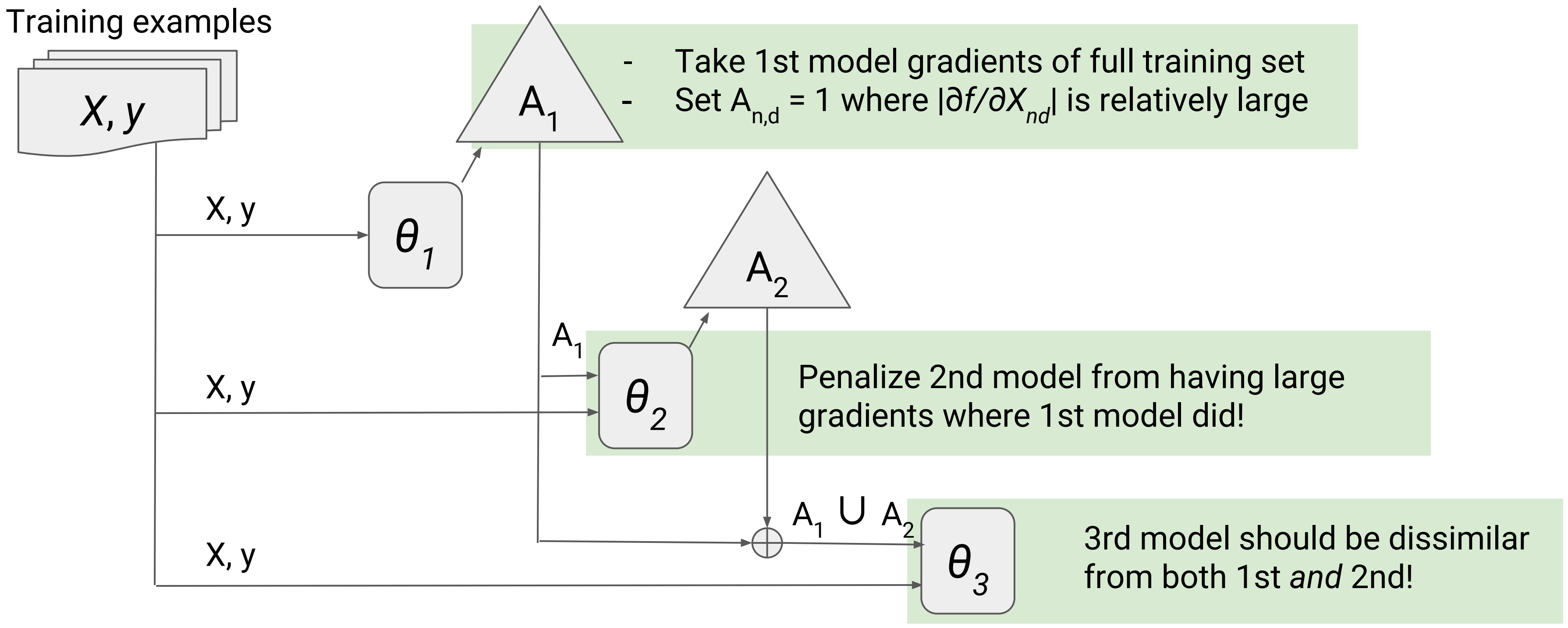
Find-another-explanation

Overall goal: obtain an ensemble of models that are all accurate but for different reasons.



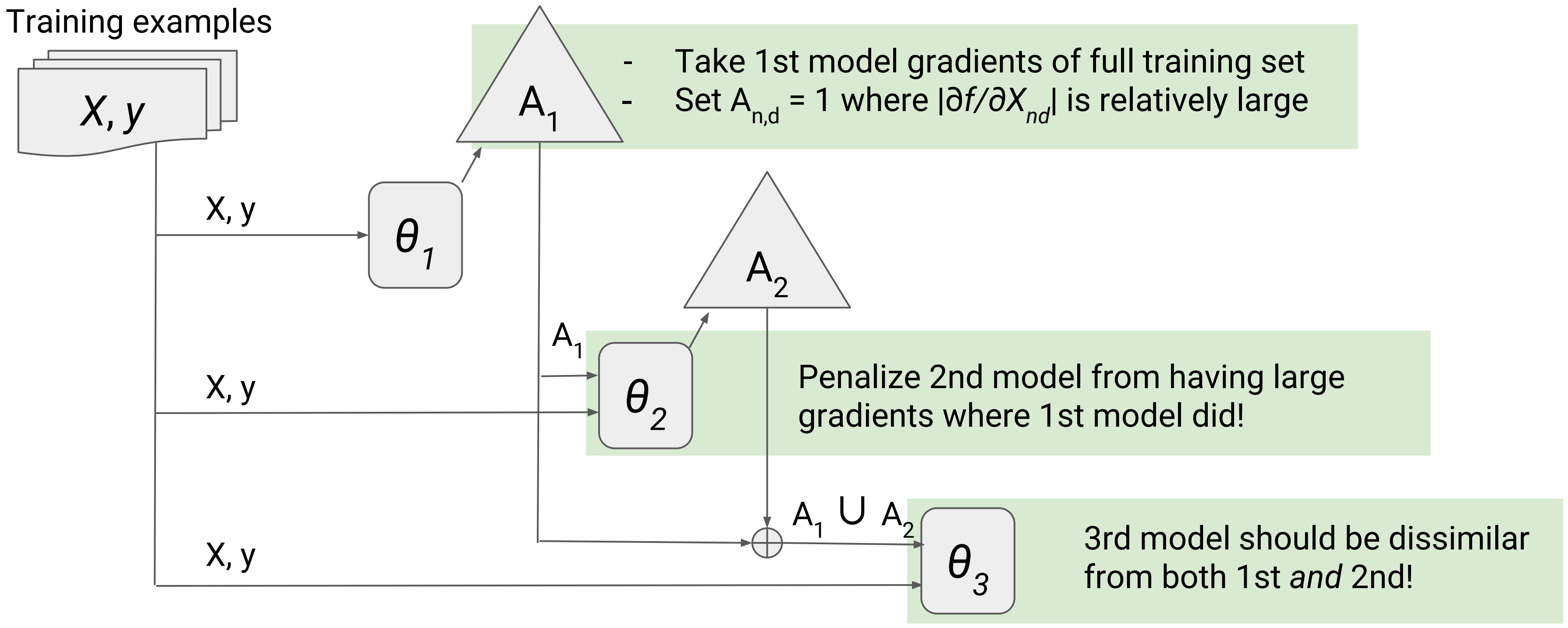
Find-another-explanation

Overall goal: obtain an ensemble of models that are all accurate but for different reasons.



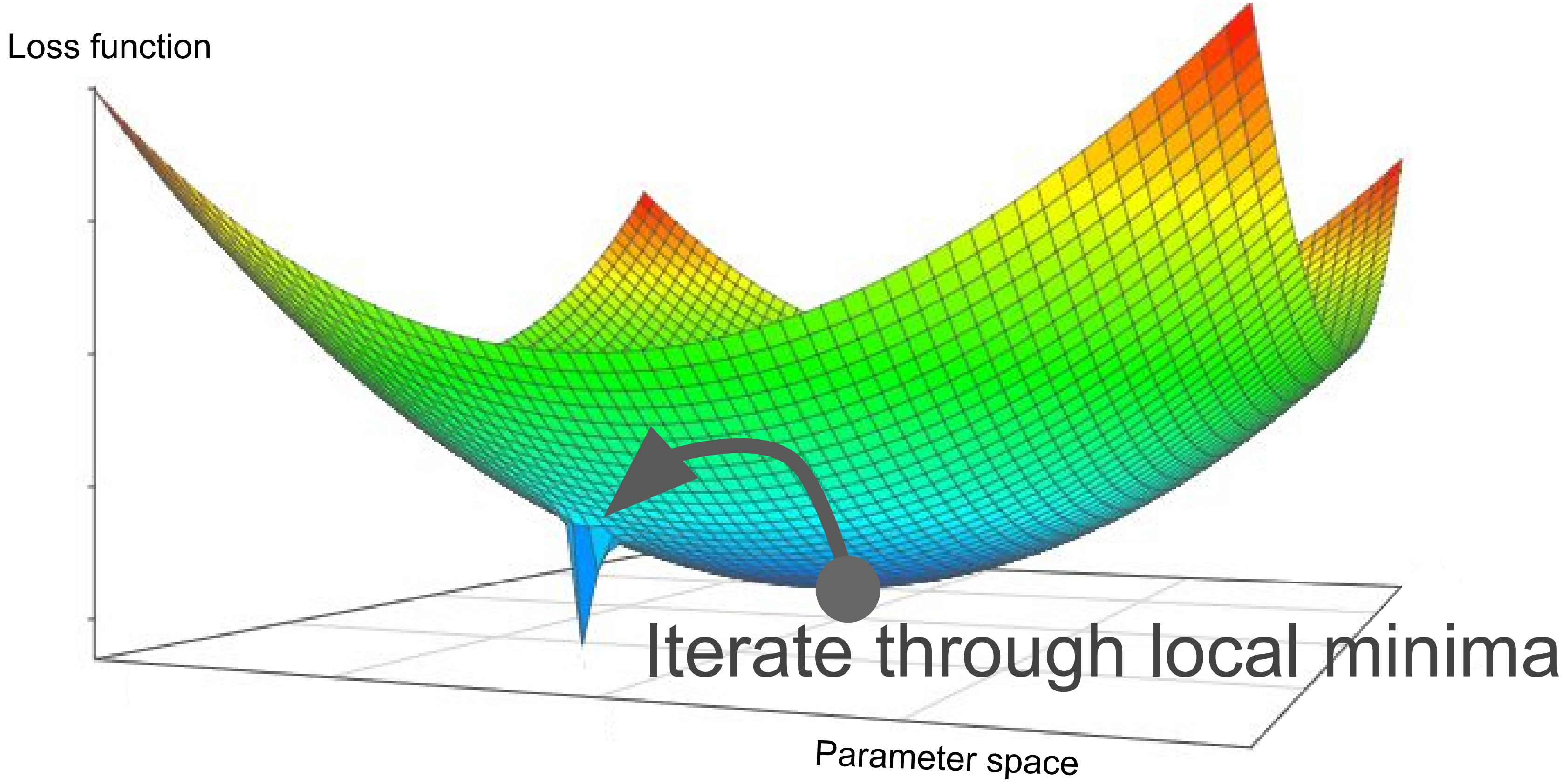
Find-another-explanation

Overall goal: obtain an ensemble of models that are all accurate but for different reasons.

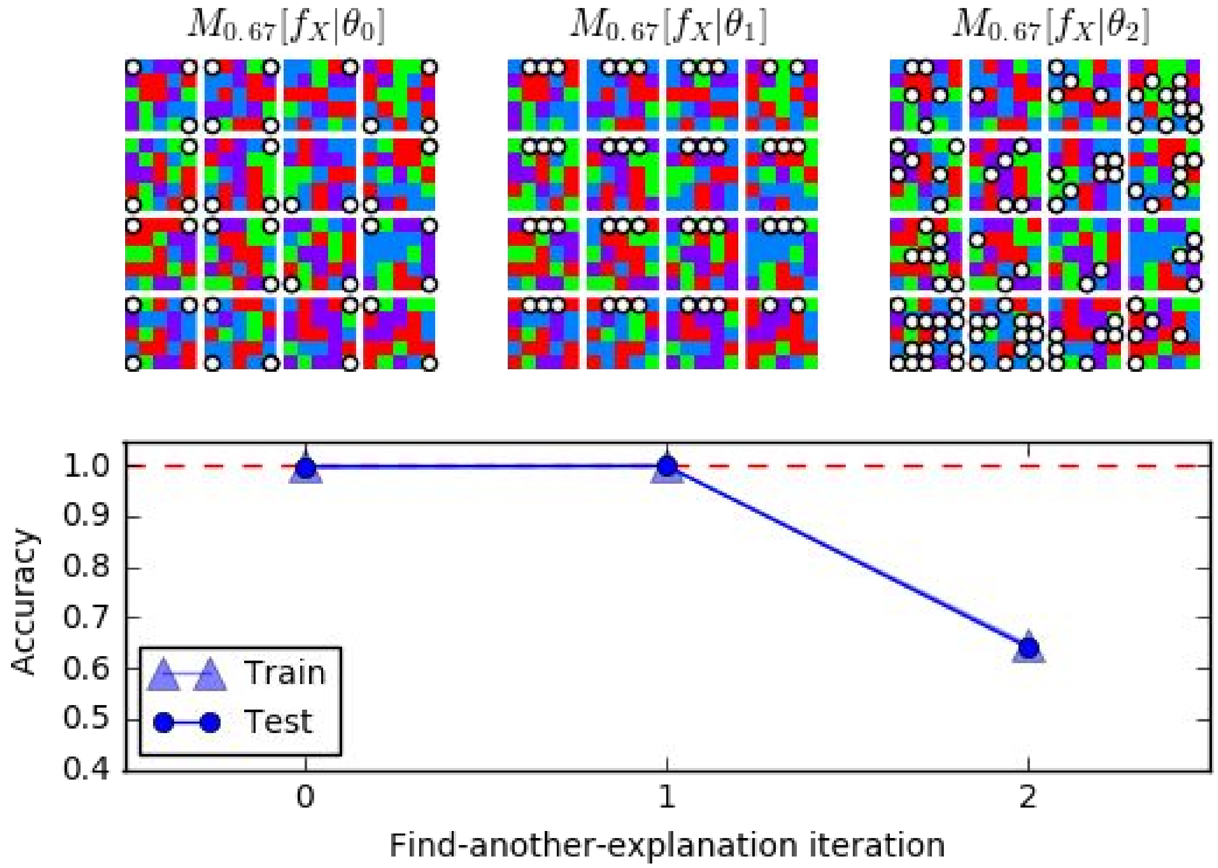


And so on...

Back to the picture in my head

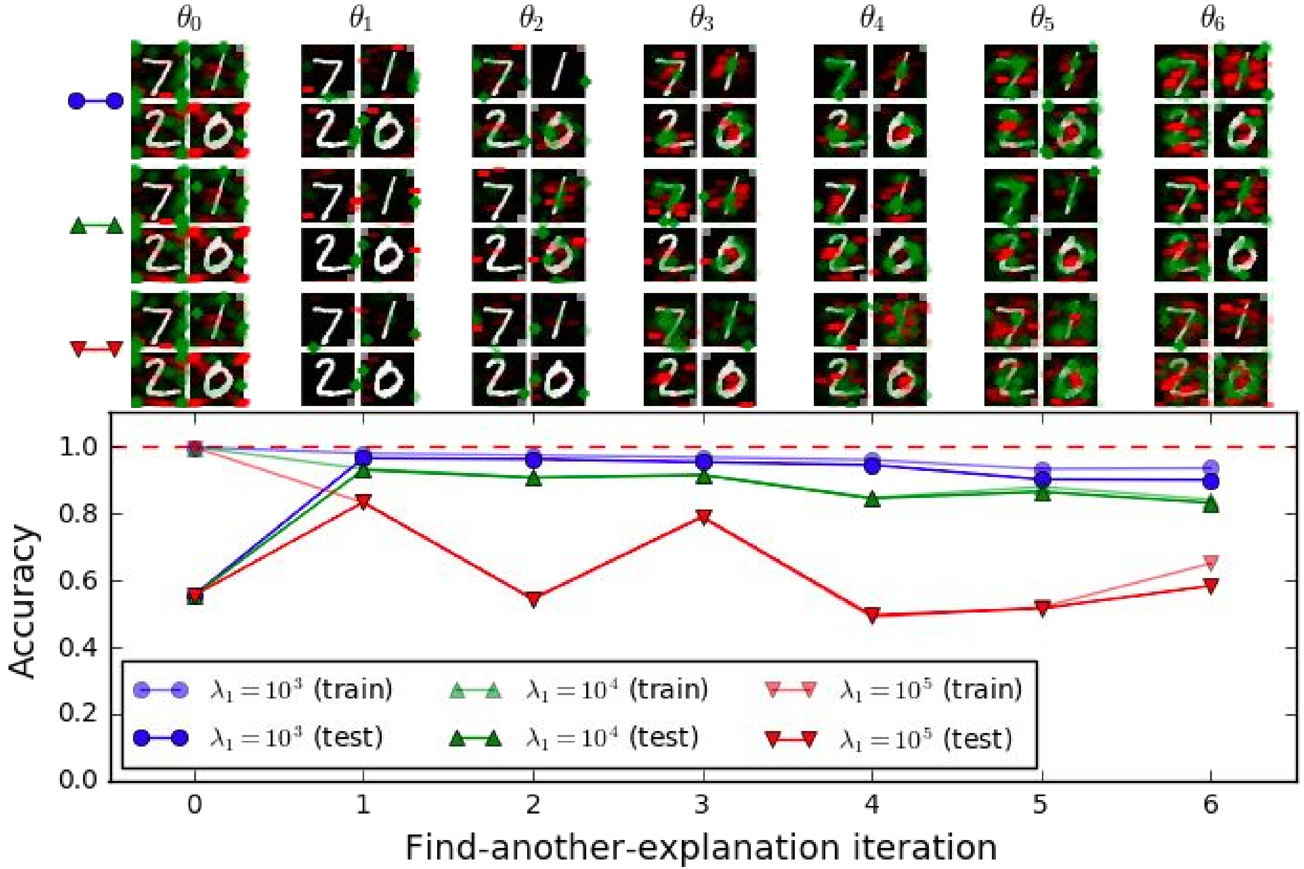


Find-another-explanation: Toy Colors



Model initially learns corner rule, falls back to top-three rule, then fails to learn anything.

Find-another-explanation: Decoy MNIST



Models initially learn decoy rule, then use other features.

Accuracy falls, but very slowly (MNIST is redundant)

Summary / Contributions

For when learning from X, y alone is insufficient:

- Introduced a novel method of injecting domain knowledge into NN training
    - Works for any differentiable model, no need to modify architecture
    - Can start using it with a small number of annotated examples
- Demonstrated how it can be used to obtain otherwise unreachable models
    - If we have domain knowledge, we can use it to avoid fitting to spurious correlations
    - If we don't, we can obtain a diverse ensemble of models

Summary / Contributions

For when learning from X, y alone is insufficient:

- Introduced a novel method of injecting domain knowledge into NN training
  - Works for any differentiable model, no need to modify architecture
  - Can start using it with a small number of annotated examples
- Demonstrated how it can be used to obtain otherwise unreachable models
  - If we have domain knowledge, we can use it to avoid fitting to spurious correlations
  - If we don’t, we can obtain a diverse ensemble of models

May be more common than we think!

Future Work

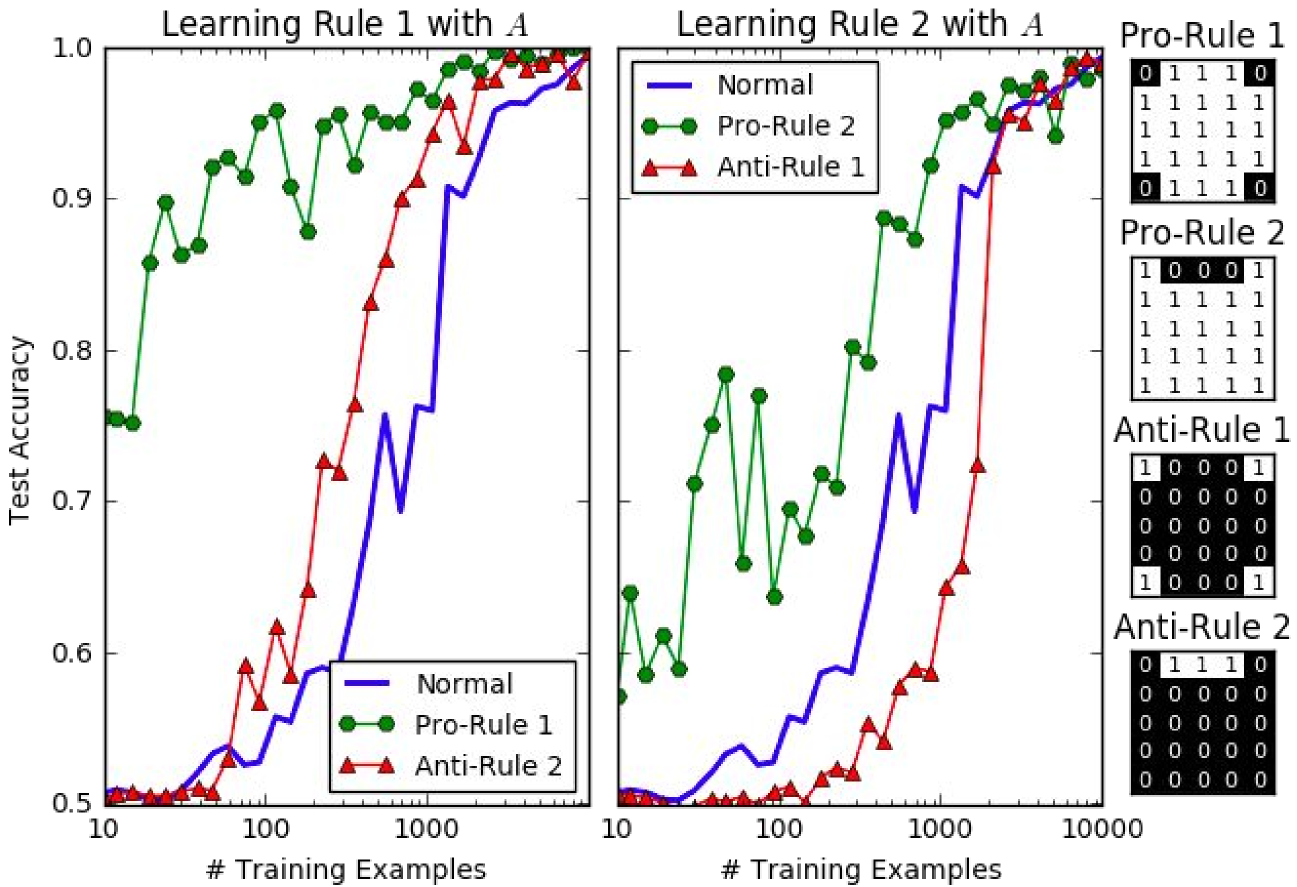
- Human-in-the-loop
  - Interactively select the best explanations, train new models
- Bridging features and concepts
  - E.g. for images, “concepts” are only emergent at upper layers
  - If we can identify concepts like in [Bau et al. CVPR 2017], regularize wrt concepts?
- Explore more options for loss functions and annotations
  - Use non-binary *A*, L1 regularization, class specific positive/negative penalties rather than sum
- Much bigger networks
  - Have already validated the approach for mid-size CNNs, but I’m a newbie
- Defending against adversarial perturbations
  - Have results that setting A=1 universally builds robustness to FGSM and JSMA attacks
- Applications to medical domain
  - Many types of medical knowledge are easily encodable as annotations

These slides again: goo.gl/fMZiRu

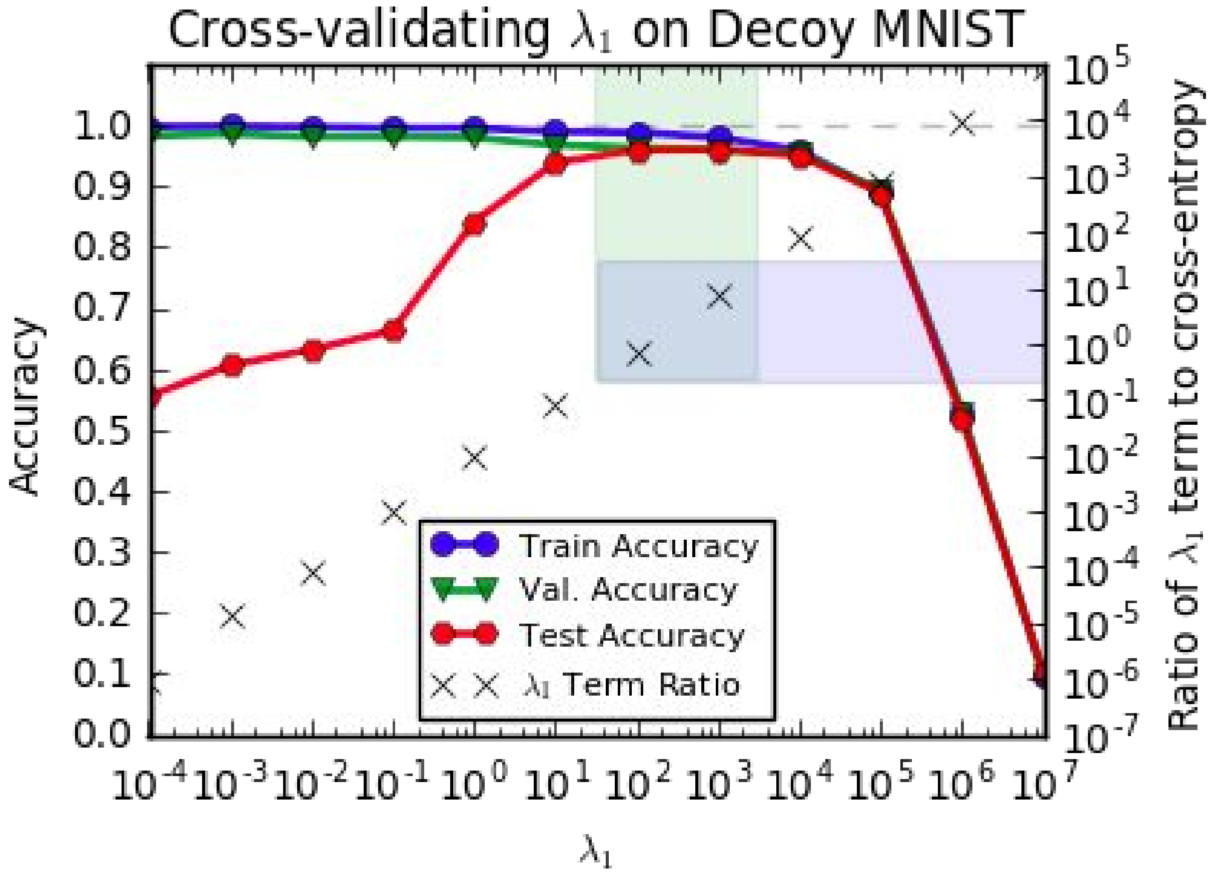
Future Work

- Human-in-the-loop
  - Interactively select the best explanations, train new models
- Bridging features and concepts
  - E.g. for images, “concepts” are only emergent at upper layers
  - If we can identify concepts like in [Bau et al. CVPR 2017], regularize wrt concepts?
- Explore more options for loss functions and annotations
  - Use non-binary *A*, L1 regularization, class specific positive/negative penalties rather than sum
- Much bigger networks
  - Have already validated the approach for mid-size CNNs, but I’m a newbie
- Defending against adversarial perturbations
  - Have results that setting A=1 universally builds robustness to FGSM and JSMA attacks
- Applications to medical domain
  - Many types of medical knowledge are easily encodable as annotations

Learning with less data?



Best if “right answers” term ≈ “right reasons” term



Gradients are consistent with LIME but less sparse

Input gradients +soc.religion.christian +alt.atheism

From: USTS012@uabdpo.dpo.uab.edu
Subject: Should teenagers pick a church parents don't attend?
Organization: UTexas Mail-to-News Gateway
Lines: 13

Q. Should teenagers have the freedom to choose what church they go to?

My friends teenage kids do not like to go to church.
If left up to them they would sleep, but that's not an option.
They complain that they have no friends that go there, yet don't
attempt to make friends. They mention not respecting their Sunday
school teacher, and usually find a way to miss Sunday school but
do make it to the church service, (after their parents are thoroughly
disgusted) I might add. A never ending battle? It can just ruin your
whole day if you let it.

Has anyone had this problem and how did it get resolved?
f.

LIME +soc.religion.christian +alt.atheism

From: USTS012@uabdpo.dpo.uab.edu
Subject: Should teenagers pick a church parents don't attend?
Organization: UTexas Mail-to-News Gateway
Lines: 13

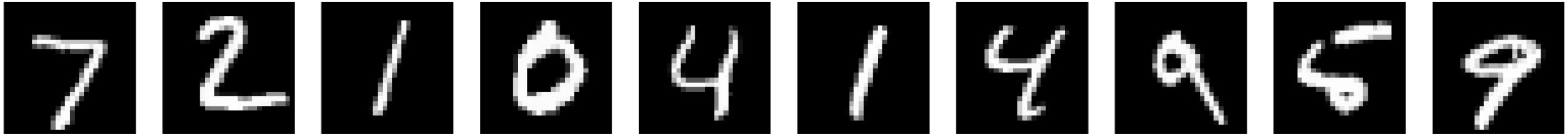
Q. Should teenagers have the freedom to choose what church they go to?

My friends teenage kids do not like to go to church.
If left up to them they would sleep, but that's not an option.
They complain that they have no friends that go there, yet don't
attempt to make friends. They mention not respecting their Sunday
school teacher, and usually find a way to miss Sunday school but
do make it to the church service, (after their parents are thoroughly
disgusted) I might add. A never ending battle? It can just ruin your
whole day if you let it.

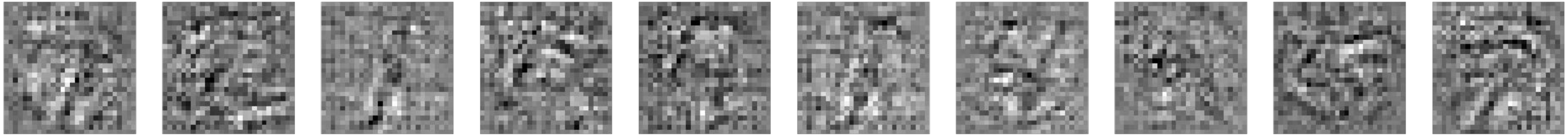
Has anyone had this problem and how did it get resolved?
f.

Setting A = 1 for all features (“certainty regularization”)

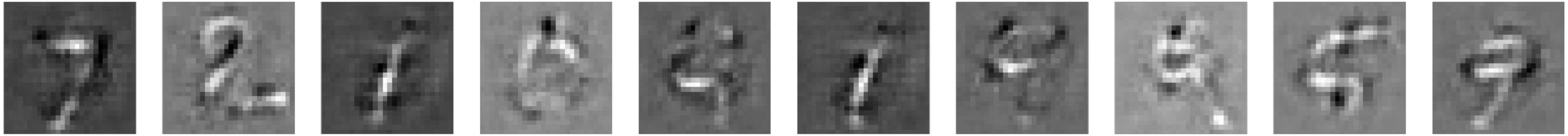
Input images:



Sum-of-log-prob input gradients for normal CNN:



...for certainty-regularized CNN:



Certainty-regularized CNN also much more resistant to adversarial perturbations

Information-Theoretic Interpretation of Loss Function

$$\underbrace{\lambda_1 \sum_{n=1}^{N} \sum_{d=1}^{D} \left(A_{nd} \frac{\partial}{\partial x_d} \boxed{\sum_{k=1}^{K} \log(\hat{y}_{nk})} \right)^2}_{\text{Right reasons}}$$

= K * Cross-entropy of prediction w/ uniformly random guess

"distance from total uncertainty"