Analyzing Deep Neural Networks

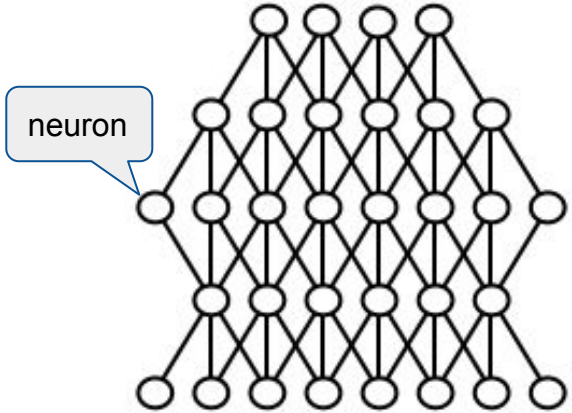
Ankur Taly, Google AI
ataly@google.com

Joint work with Mukund Sundararajan[1], Qiqi Yan[1], Kedar Dhamdhere[1], and Pramod Mudrakarta[2]
[1]Google, [2]U Chicago

Deep Neural Networks are widely applicable

Output

(Label, sentence, next word, next move, etc.)



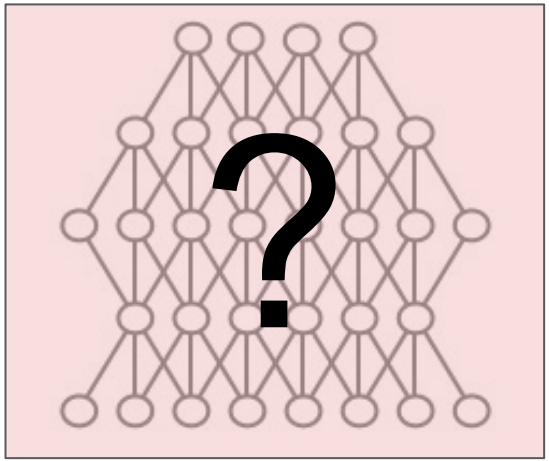
Input

(Image, sentence, game position, etc.)

But a trained network is largely a black box to humans

Output

(Label, sentence, next word, next move, etc.)



Input

(Image, sentence, game position, etc.)

Top label: **“fireboat”**

Why did the network label this image as **“fireboat”**?

Top label: **“clog”**

Why did the network label this image as **“clog”**?

Beautiful design and execution, both the space itself and the food. Many interesting options beyond the excellent traditional Italian meatball. Great for the Financial District.

Why did the network predict **positive sentiment** for this review?

The Attribution Problem

Given an input, attribute the network's prediction to features of the input, relative to a certain baseline input

- Examples:
  - Attribute an object recognition network's prediction to its pixels
  - Attribute a text sentiment network's prediction to individual words

A reductive formulation of "why this prediction" but surprisingly useful :-)

Applications of Attributions

- Debugging network predictions
- Generating an explanation for the end-user
- Analyzing network robustness
- Assessing prediction confidence

Need for a baseline

- Every explanation involves an implicit or explicit counterfactual
  - E.g., A man suffers from indigestion. Doctor blames it to a stomach ulcer. Wife blames it on eating turnips. Both are correct relative to their baselines (see [Kahneman-Miller 86])
- Ideally, the baseline is an informationless input for the network
  - E.g., Black image for image networks
- The baseline may also be an important analysis knob

We explain **F(input) - F(baseline)** in terms of input features

Plan

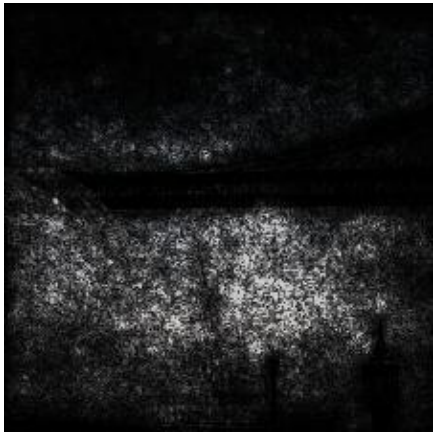
- Our attribution method: **Integrated Gradients**
- Applications:
  - Debugging network behavior
  - Generating explanations for end-users
  - Analyzing network robustness
- Caveats and Limitations
- Work in Progress: Preconditions for Deep Neural Networks

Naive approaches

- **Ablations**: Drop each feature and note the change in prediction
  - Computationally expensive, Unrealistic inputs, Misleading when features interact

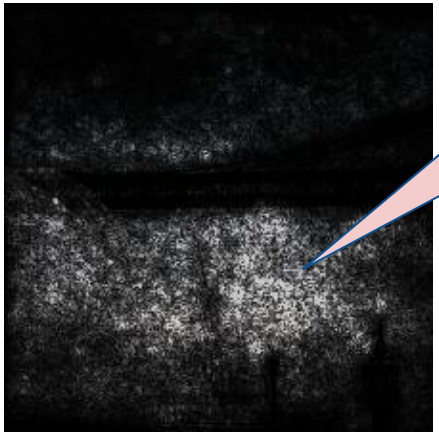
Naive approaches

- **Ablations**: Drop each feature and note the change in prediction
  - Computationally expensive, Unrealistic inputs, Misleading when features interact
- **Feature*Gradient**: Attribution for feature x_i is $x_i * \partial y/\partial x_i$

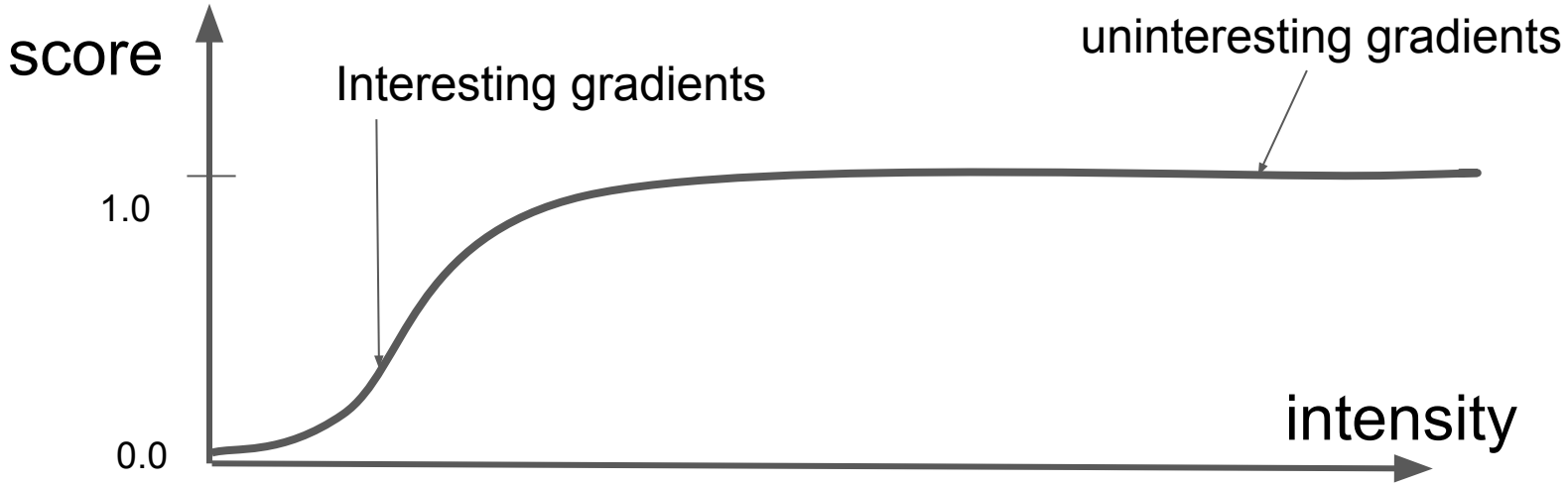


Naive approaches

- **Ablations**: Drop each feature and note the change in prediction
  - Computationally expensive, Unrealistic inputs, Misleading when features interact
- **Feature*Gradient**: Attribution for feature x_i is $x_i * \partial y/\partial x_i$



Gradients in the vicinity of the input seem like noise



Baseline

… scaled inputs ...

Input

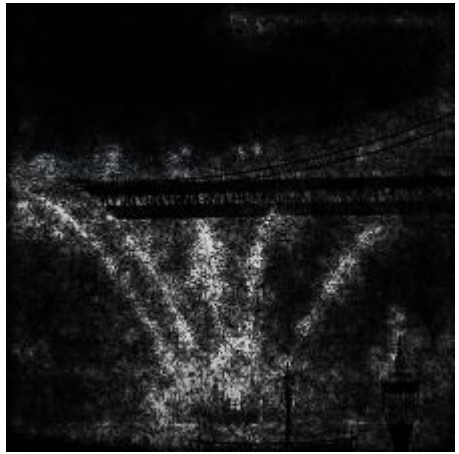
… gradients of scaled inputs ….

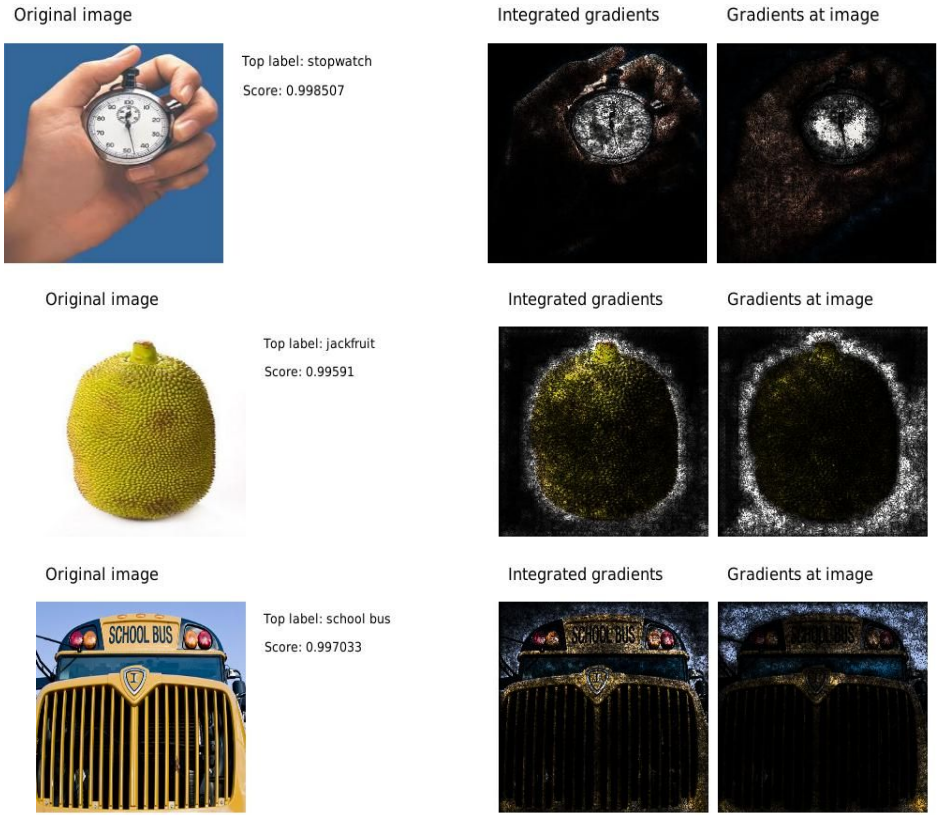
The Method: Integrated Gradients

$$IG(input, base) ::= (input - base) * \int_{0}^{1} \nabla F(\alpha * input + (1-\alpha) * base)\, d\alpha$$

Original image

Integrated Gradients





Many more Inception+ImageNet examples here

Evaluating an Attribution Method

- Ablate top attributed features and examine the change in prediction
  - Issue: May introduce artifacts in the input (e.g., the square below)

- Compare attributions to (human provided) groundtruth on “feature importance”
  - Issue 1: Attributions may appear incorrect because the network reasons differently
  - Issue 2 : **Confirmation bias**

Evaluating an Attribution Method

- Ablate top attributed features and examine the change in prediction
  - Issue: May introduce artifacts in the input (e.g., the square below)

- Compare attributions to (human provided) groundtruth on “feature importance”
  - Issue 1: Attributions may appear incorrect because the network reasons differently
  - Issue 2 : **Confirmation bias**

The mandate for attributions is to be faithful to the network’s reasoning

Our Approach: Axiomatic Justification

- List **desirable criteria (axioms)** for an attribution method
- Establish a uniqueness result: X is the **only** method that satisfies these criteria

Axioms

- **Insensitivity:** A variable that has no effect on the output gets no attribution
- **Sensitivity**: If baseline and input differ in a single variable, and have different outputs, then that variable should receive some attribution
- **Linearity preservation**: Attributions(α*F1 + ß*F2) = α*Attributions(F1) + ß*Attributions(F2)
- **Implementation invariance**: Two networks that compute identical functions for all inputs get identical attributions
- **Completeness**: Sum(attributions) = F(input) - F(baseline)
- **Symmetry**: Symmetric variables with identical values get equal attributions

Result

Theorem [ICML 2017]: Integrated Gradients is the **unique** path-integral method satisfying: Sensitivity, Insensitivity, Linearity preservation, Implementation invariance, Completeness, and Symmetry

Historical note:

- Integrated Gradients is the **Aumann-Shapley method** from cooperative game theory, which has a similar characterization; see [Friedman 2004]

Highlights of Integrated Gradients

- Easy to implement
  - Gradient calls on a batch, no instrumentation of the network, no new training
- Widely applicable
  - Used by 20+ product teams, and 3 ML frameworks at Google
- Backed by an axiomatic guarantee

References

- Paper: Axiomatic Attribution for Deep Networks **[ICML 2017]**
- Blog post: Attributing a deep network's prediction to its input
- Code: https://github.com/ankurtaly/Integrated-Gradients

Debugging network behavior

Why is this image labeled as “clog”?

Original image

“Clog”

Why is this image labeled as “clog”?

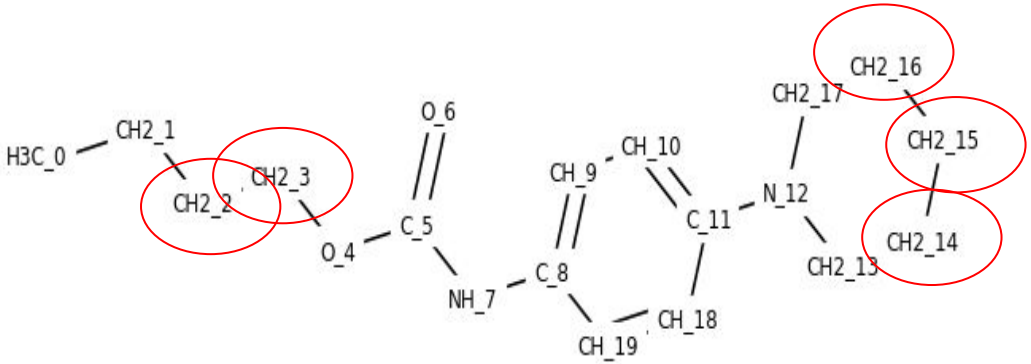
Original image

Integrated Gradients
(for label “clog”)

“Clog”

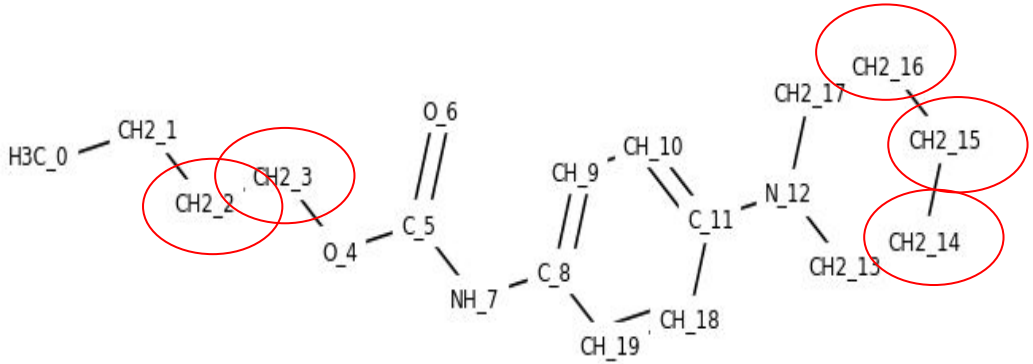
Detecting an architecture bug

- Deep network [Kearns, 2016] predicts if a molecule binds to certain DNA site
- **Finding**: Some atoms had identical attributions despite different connectivity



Detecting an architecture bug

- Deep network [Kearns, 2016] predicts if a molecule binds to certain DNA site
- **Finding**: Some atoms had identical attributions despite different connectivity

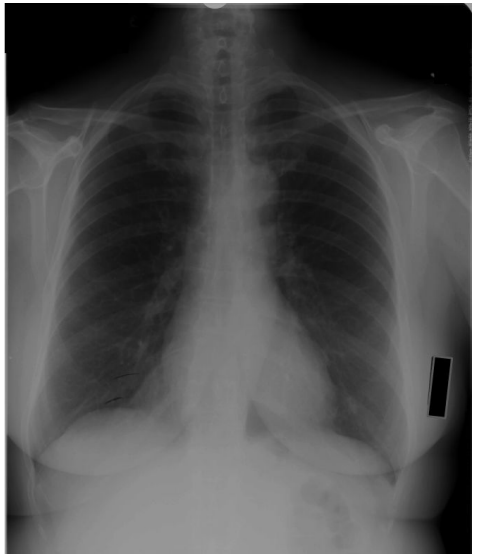


- **Bug**: The architecture had a bug due to which the convolved bond features did not affect the prediction!

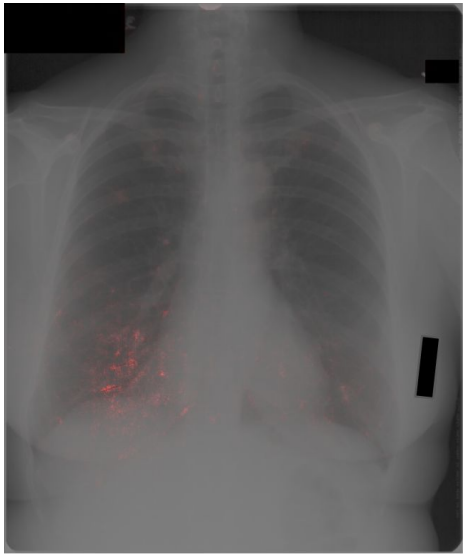
Detecting a data issue

- Deep network predicts various diseases from chest x-rays

Original image

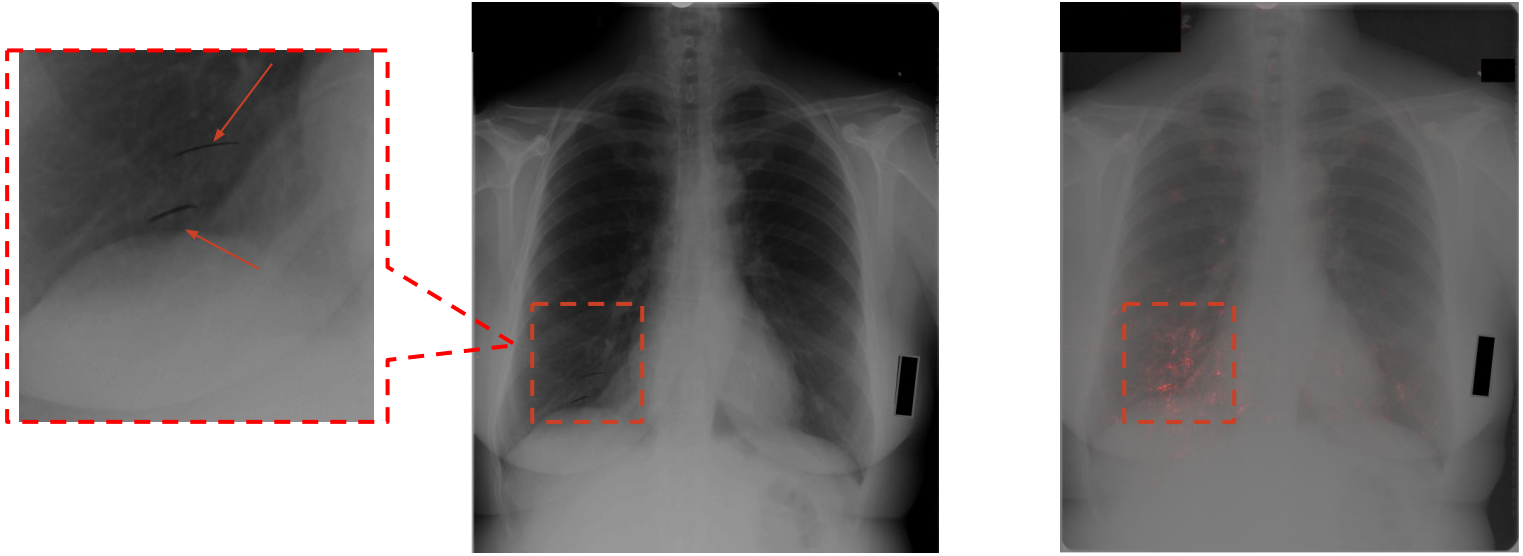


Integrated gradients
(for top label)



Detecting a data issue

- Deep network predicts various diseases from chest x-rays
- **Finding**: Attributions fell on radiologist's markings (rather than the pathology)



Generating explanations for Diabetic Retinopathy[1] predictions

[1]**Diabetic Retinopathy (DR)** is a diabetes complication that affects the eye. Deep networks can predict DR grade from retinal fundus images with high accuracy (AUC ~0.97) [JAMA, 2016].

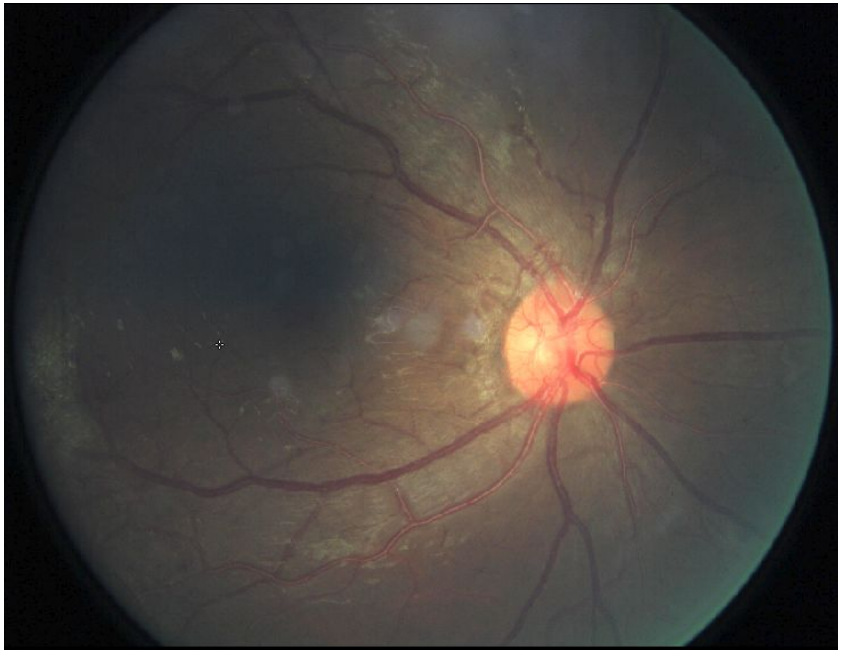
Retinal Fundus Image

⇨ Prediction: **“proliferative”** DR

- Proliferative implies vision-threatening

Can we provide an explanation to the doctor with supporting evidence for **“proliferative”** DR?

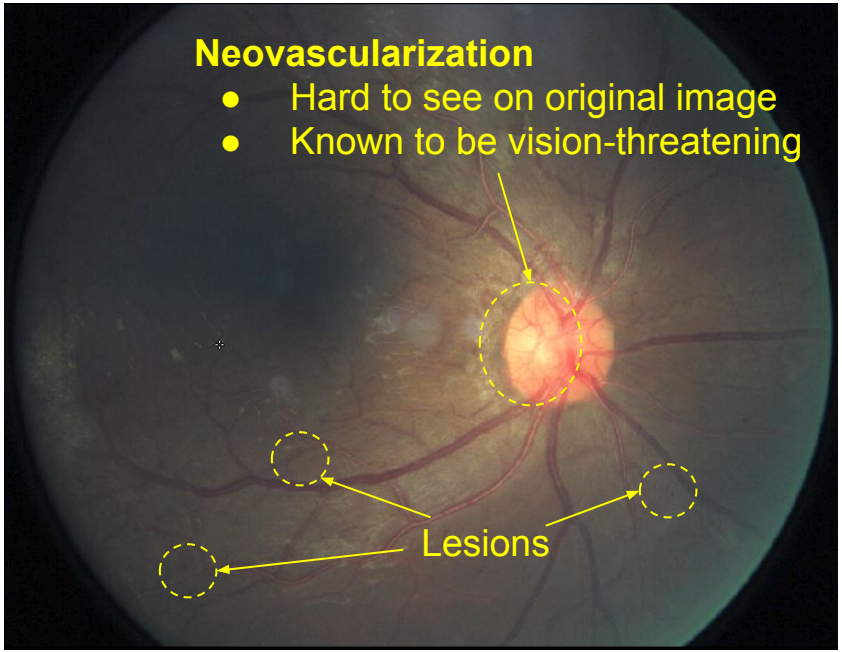
Retinal Fundus Image



Integrated Gradients for label: “proliferative”

Visualization: Overlay heatmap on green channel

Retinal Fundus Image



Integrated Gradients for label: “proliferative”

Visualization: Overlay heatmap on green channel

Efficacy of Explanations

Explanations help when:

- Model is right, and explanation convinces the doctor
- Model is wrong, and explanation reveals the flaw in the model's reasoning

But, Explanations can also hurt when:

- Model is right, but explanation is unintelligible
- Model is wrong, but the explanation convinces the doctor

Be careful about long-term effects too!

Humans and Automation: Use, Misuse, Disuse, Abuse - Parsuraman and Riley, 1997

Assisted-read study

9 doctors grade 2000 images under three different conditions

A. Image only
B. Image + Model's prediction scores
C. Image + Model's prediction scores + Explanation (Integrated Gradients)

Some findings:

- Seeing prediction scores (B) significantly increases accuracy vs. image only (A)
- Showing explanations (C) only provides slight additional improvement
  - Masks help more when model certainty is low
- Both B and C increase doctor ↔ model agreement

Paper: Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy --- Journal of Ophthalmology [2018]

Analyzing robustness of Question-Answering (QA) networks

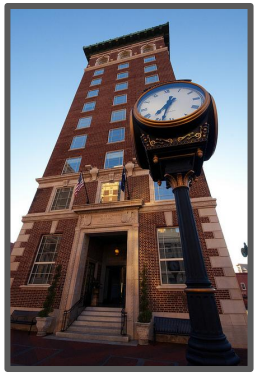
Tabular QA

Rank	Nation	Gold	Silver	Bronze	Total
1	India	102	58	37	197
2	Nepal	32	10	24	65
3	Sri Lanka	16	42	62	120
4	Pakistan	10	36	30	76
5	Bangladesh	2	10	35	47
6	Bhutan	1	6	7	14
7	Maldives	0	0	4	4

Q: How many medals did India win?
A: 197

Neural Programmer (2017) model
33.5% accuracy on WikiTableQuestions

Visual QA



Q: How symmetrical are the white bricks on either side of the building?
A: very

Kazemi and Elqursh (2017) model.
61.1% on VQA 1.0 dataset
(state of the art = 66.7%)

Reading Comprehension

Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager

Q: Name of the quarterback who was 38 in Super Bowl XXXIII?
A: John Elway

Yu et al (2018) model.
84.6 F-1 score on SQuAD (state of the art)

First rule of Question-Answering: **Read the question carefully!**

MATHEMATICS

Time: 2 hours

PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY

1. This question paper consists of 15 pages and an Information Sheet. Please check that your paper is complete.

2. Read the questions carefully.

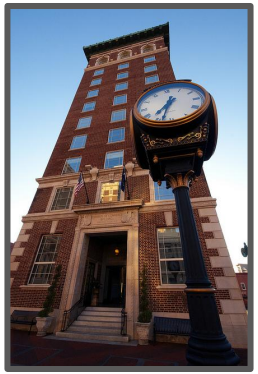
Tabular QA

Rank	Nation	Gold	Silver	Bronze	Total
1	India	102	58	37	197
2	Nepal	32	10	24	65
3	Sri Lanka	16	42	62	120
4	Pakistan	10	36	30	76
5	Bangladesh	2	10	35	47
6	Bhutan	1	6	7	14
7	Maldives	0	0	4	4

Q: How many medals did India win?
A: 197

Neural Programmer (2017) model
33.5% accuracy on WikiTableQuestions

Visual QA



Q: How symmetrical are the white bricks on either side of the building?
A: very

Kazemi and Elqursh (2017) model.
61.1% on VQA 1.0 dataset
(state of the art = 66.7%)

Reading Comprehension

Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager

Q: Name of the quarterback who was 38 in Super Bowl XXXIII?
A: John Elway

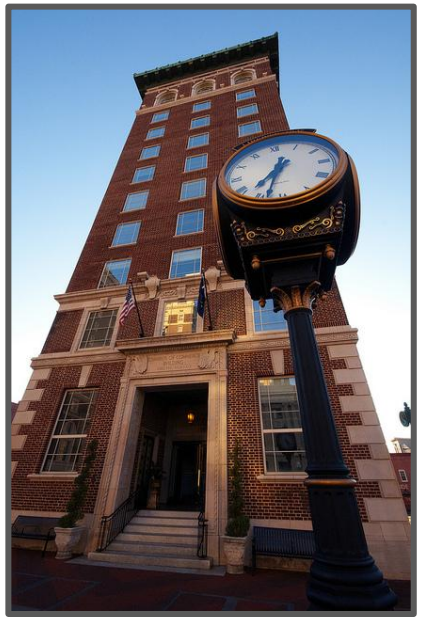
Yu et al (2018) model.
84.6 F-1 score on SQuAD (state of the art)

Robustness question: Do these network read the question carefully? :-)

Visual QA

Kazemi and Elqursh (2017) model.

Accuracy: **61.1%** (state of the art: 66.7%)

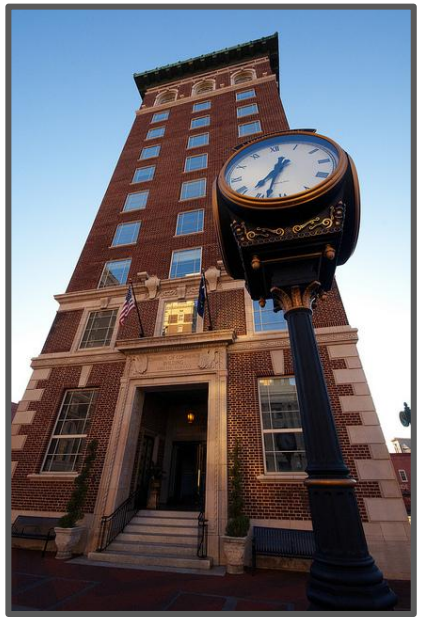


Q: How symmetrical are the white bricks on either side of the building?
A: very

Visual QA

Kazemi and Elqursh (2017) model.

Accuracy: **61.1%** (state of the art: 66.7%)



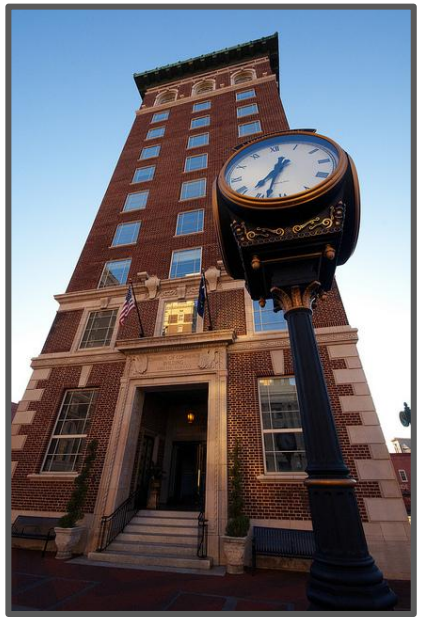
Q: How symmetrical are the white bricks on either side of the building?
A: very

Q: How asymmetrical are the white bricks on either side of the building?
A: very

Visual QA

Kazemi and Elqursh (2017) model.

Accuracy: **61.1%** (state of the art: 66.7%)



Q: How symmetrical are the white bricks on either side of the building?
A: very

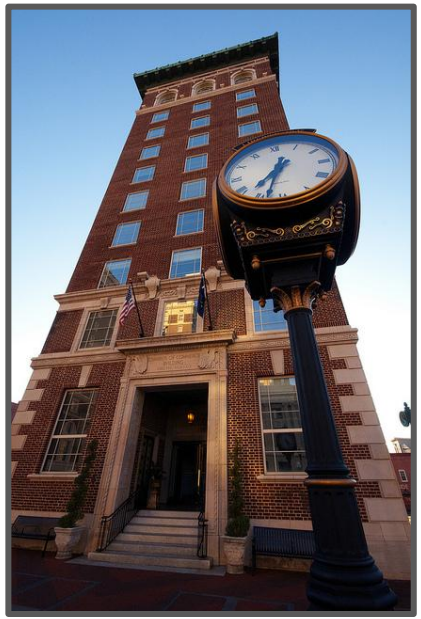
Q: How asymmetrical are the white bricks on either side of the building?
A: very

Q: How big are the white bricks on either side of the building?
A: very

Visual QA

Kazemi and Elqursh (2017) model.

Accuracy: **61.1%** (state of the art: 66.7%)



Q: How symmetrical are the white bricks on either side of the building?
A: very

Q: How asymmetrical are the white bricks on either side of the building?
A: very

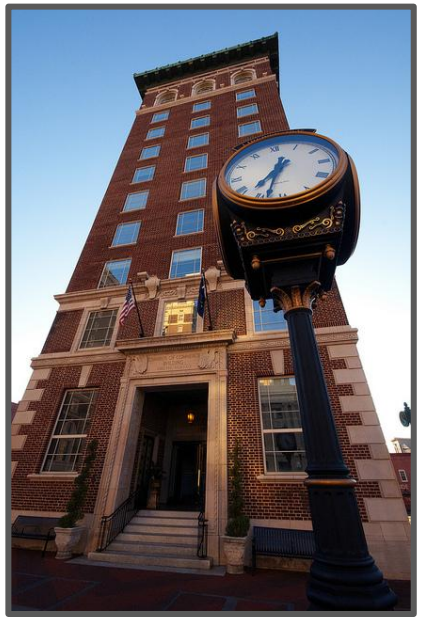
Q: How big are the white bricks on either side of the building?
A: very

Q: How fast are the bricks speaking on either side of the building?
A: very

Visual QA

Kazemi and Elqursh (2017) model.

Accuracy: **61.1%** (state of the art: 66.7%)



Q: How symmetrical are the white bricks on either side of the building?
A: very

Q: How asymmetrical are the white bricks on either side of the building?
A: very

Q: How big are the white bricks on either side of the building?
A: very

Q: How fast are the bricks speaking on either side of the building?
A: very

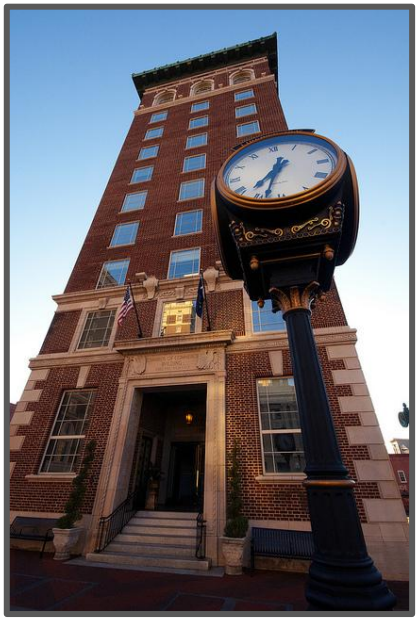
Test/dev accuracy does not show us the entire picture. Need to look inside!

Analysis procedure

- Attribute the answer (or answer selection logic) to question words
  - **Baseline**: Empty question, but full context (image, text, paragraph)
    - By design, attribution will **not** fall on the context

- Visualize attributions per example
- Aggregate attributions across examples

Visual QA attributions



Q: How symmetrical are the white bricks on either side of the building?
A: very

How symmetrical are the white bricks on either side of the building?

red: high attribution
blue: negative attribution
gray: near-zero attribution

Over-stability [Jia and Liang, EMNLP 2017 (outstanding paper)]

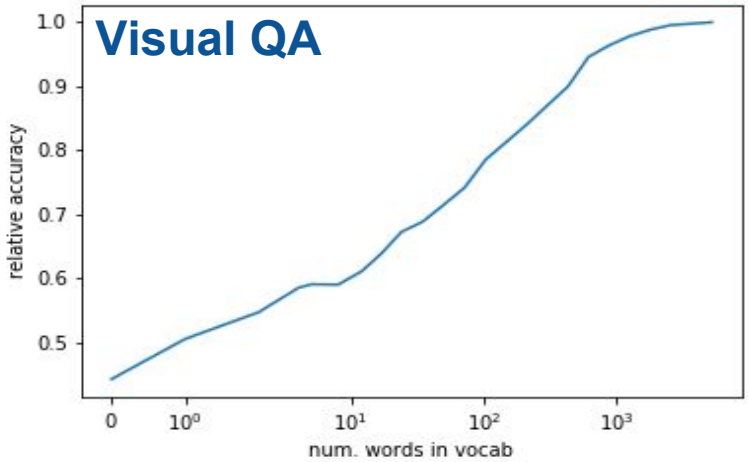
- Image networks suffer from **“over-sensitivity”** to pixel perturbations
- Paragraph QA models suffer from **“over-stability”** to semantics-altering edits

Attributions show how such over-stability manifests in Visual QA, Tabular QA and Paragraph QA networks

Over-stability

During inference, drop all words from the dataset except ones which are frequently top attributions

- E.g. How many ~~red~~ ~~buses~~ ~~are~~ ~~in~~ ~~the~~ ~~picture?~~

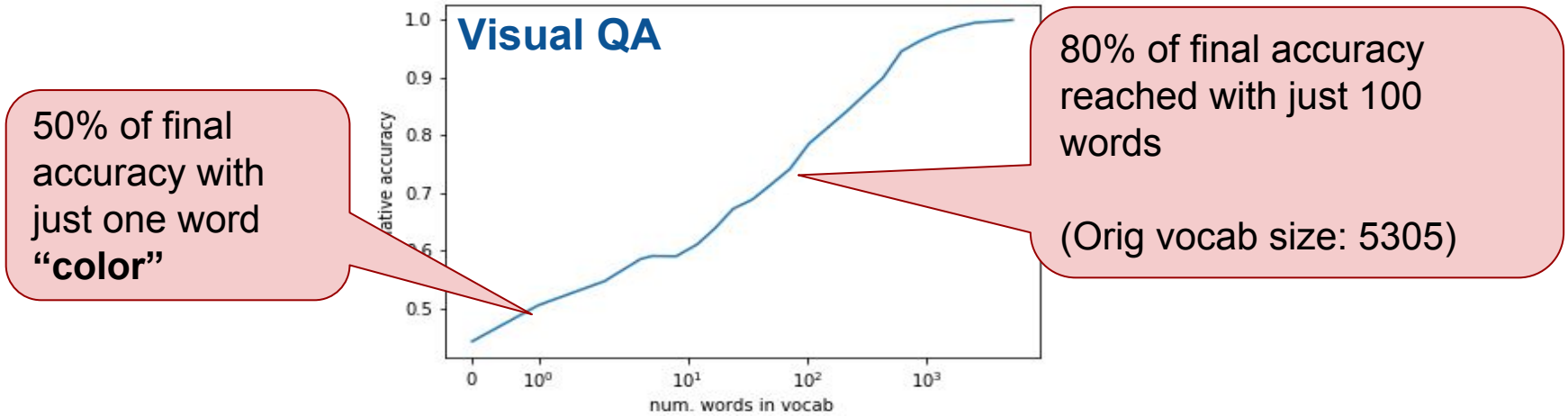


Top tokens: color, many, what, is, how, there, …

Over-stability

During inference, drop all words from the dataset except ones which are frequently top attributions

- E.g. How many ~~red~~ ~~buses~~ ~~are~~ ~~in~~ ~~the~~ ~~picture?~~



Top tokens: color, many, what, is, how, there, …

Attack: Subject ablation

Replace the subject of a question with a low-attribution noun from the vocabulary

- This **ought to change** the answer but often does not!

Low-attribution nouns

'tweet',
'childhood',
'copyrights',
'mornings',
'disorder',
'importance',
'topless',
'critter',
'jumper',
'fits'

What is the **man** doing? → What is the **tweet** doing?
How many **children** are there? → How many **tweet** are there?

VQA model's response remains the same 75.6% of the time on questions that it originally answered correctly

Many other attacks!

- Visual QA
  - Prefix concatenation attack (accuracy drop: **61.1% to 19%**)
  - Stop word deletion attack (accuracy drop: **61.1% to 52%**)
- Tabular QA
  - Prefix concatenation attack (accuracy drop: **33.5% to 11.4%**)
  - Stop word deletion attack (accuracy drop: **33.5% to 28.5%**)
  - Table row reordering attack (accuracy drop: **33.5 to 23%**)
- Paragraph QA
  - Improved paragraph concatenation attacks of Jia and Liang from [EMNLP 2017]

Paper: Did the model understand the question? **[ACL 2018]**

Some limitations and caveats

Attributions are pretty shallow

Attributions do not explain:

- How the network combines the features to produce the answer?
- What training data influenced the prediction
- Why gradient descent converged
- etc.

An instance where attributions are useless:

- A network that predicts TRUE when there are **even number** of black pixels and FALSE otherwise

Attributions are useful when the network behavior entails that a strict subset of input features are important

Attributions are for human consumption

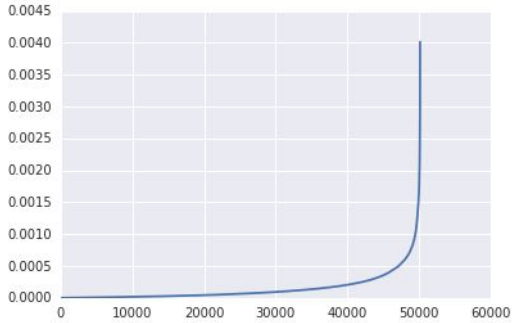
- **Humans** interpret attributions and generate insights
  - Doctor maps attributions for diabetic retinopathy to pathologies like microaneurysms, hemorrhages, etc.
- **Visualization** matters as much as the attribution technique

Attributions are for human consumption

- **Humans** interpret attributions and generate insights
  - Doctor maps attributions for diabetic retinopathy to pathologies like microaneurysms, hemorrhages, etc.
- **Visualization** matters as much as the attribution technique

Naive scaling of attributions from 0 to 255

Attributions have a **large range** and **long tail** across pixels



After clipping attributions at 99% to reduce range

Work in Progress: **Preconditions for DNNs**

Joint work with Corina Pasareanu[1], Divya Gopinath[1], Hayes Converse[2]

[1]CMU, [2]UT Austin

Preconditions

An **precondition** ϕ(X) is a predicate that implies a certain output property P

Formally, **∀ X:** ϕ(X) **=> P(F(X))**

```
def F(x1, x2):
  y1 = x1 + x2
  y2 = x2
  return y1, y2
```

x1 > 0 is a precondition for the output property **y1 > y2**

Preconditions

An **precondition** $\phi(X)$ is a predicate that implies a certain output property P

Formally, **$\forall$ X:** $\phi(X)$ **=> P(F(X))**

```
def F(x1, x2):
  y1 = x1 + x2
  y2 = x2
  return y1, y2
```

x1 > 0 is a precondition for the output property **y1 > y2**

Can we identify preconditions for properties of deep networks, e.g., the highest-scoring class is class K?

Related work: Anchors: High-Precision Model-Agnostic Explanations [AAAI, 2018]

Why is this useful?

- Explain and debug network predictions
  - Decompose the prediction logic as a disjunction of precondition rules
    - Different formulation of “why this prediction”
  - Examine preconditions satisfied by misclassified inputs
- Decompose proofs of the properties of the form: **A(input) => B(output)**
  - Use ϕ as an **interpolant**, and prove **A(X) => ϕ(X)** and **ϕ(X) => B(F(X))**
- Selectively distill the network
  - Directly return the prediction class for inputs that satisfy a precondition for the class

Identifying Preconditions

The set of all training inputs satisfying the property is a precondition, but this is uninteresting

Ideally, a precondition must:

- Have a mathematically concise representation
- Have high support
- Be based on the decision logic of the network

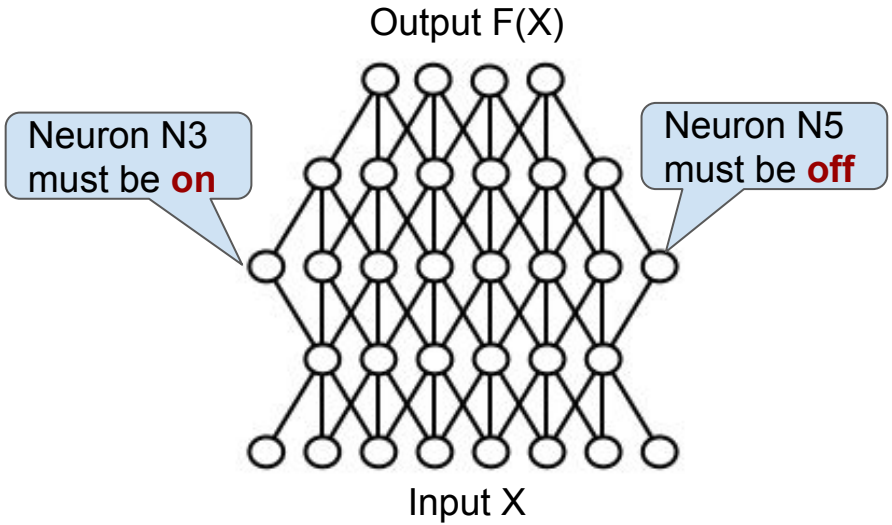
Our Idea: Define preconditions using **decision patterns** of neurons in a network.

For now, we focus on **Feed-forwards ReLU networks**

Decision Patterns

A **decision pattern σ** is a constraint stating whether some neurons are **on or off**; (all other neurons are *unconstrained)*

It defines a predicate **σ**(X) that holds if execution of input X satisfies the pattern



$$\sigma(X) \;::=\; (N3(X) > 0) \;\wedge\; (N5(X) = 0)$$

Decision Patterns

A pattern **σ** is $\lessdot$**-closed** if for each neuron constrained by the pattern, all neurons feeding into are also constrained.

Theorem: For all $\lessdot$-closed patterns **σ**, the predicate **σ**(X) is convex

Input Preconditions

Preconditions defined using $\lessdot$**-closed decision patterns**

Iterative relaxation technique for identifying input preconditions

1. Pick an input X satisfying the property, and consider its **activation signature** $\boldsymbol{\sigma}_X$
2. Check[1] if $\boldsymbol{\sigma}_X$ is a precondition for the property. If not, go to step (1)
3. Starting with the last layer, iteratively unconstrain neurons from $\boldsymbol{\sigma}_X$ till it is no longer a precondition for the property

[1]We use **ReluPlex [Katz et al., CAV 2017]** as our theorem prover

Layer Preconditions

Preconditions defined using patterns that constrain neurons in a single hidden layer

- Capture “semantic” patterns while input preconditions capture “syntactic” patterns

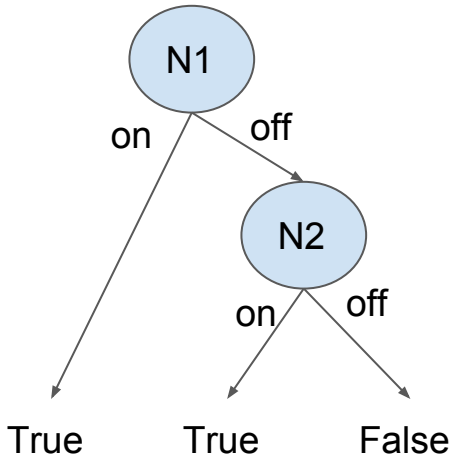
Layer Preconditions

Preconditions defined using patterns that constrain neurons in a single hidden layer

- Capture “semantic” patterns while input preconditions capture “syntactic” patterns

Decision tree learning for identifying layer preconditions from data

[N1, N2]	Property
[on, off]	True
[on, on]	True
[off, off]	False
[off, on]	True
[off, off]	False



Each path from root to a “True” leaf is a precondition candidate

Use a theorem prover to verify candidates OR **empirically validate** them on a held-out set

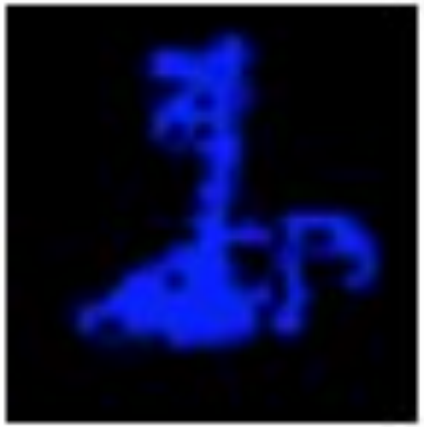
Experiment: Debugging Misclassifications

- Identify an input precondition for the prediction class that contains the misclassified input
- Visualize the (convex) precondition via an under-approximation (UA) box

Image of digit 1 misclassified as 2

Input precondition for prediction = digit 2

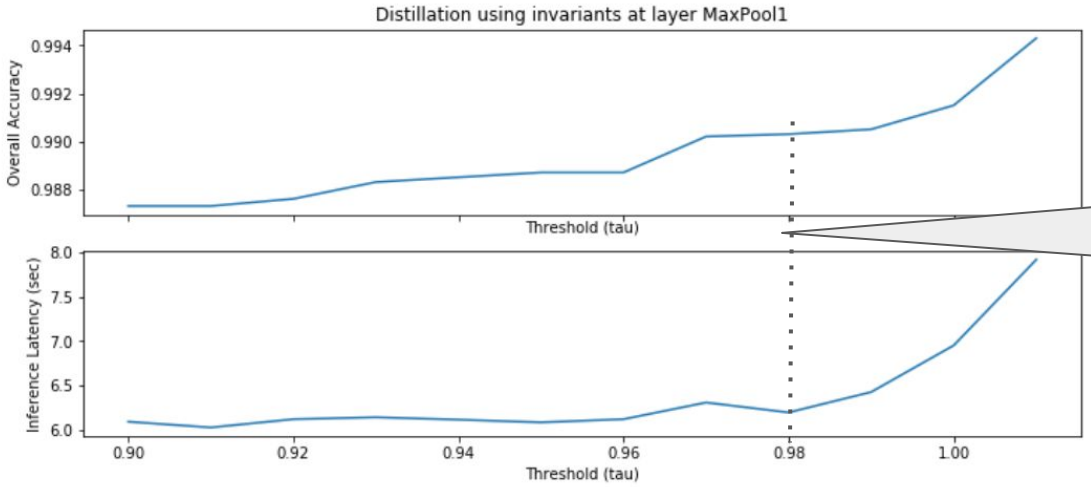
Max value of UA box



Min value of UA box

Experiment: Selective Distillation

- **Network**: 8 layer convolutional MNIST network with accuracy: **0.9943**
- Identify precondition candidates from a hidden layer, and empirically validate them
- Use predictions to directly return the prediction for inputs that satisfy it.



20% speedup in inference with accuracy degrading from 0.9943 to **0.9903**

Other Results

- Efficient proofs of properties for the ACAS Xu network
- Input clustering using layer preconditions for MNIST
- Adversarial examples from counter-examples of preconditions proofs

Thank you!

Summary: **Integrated Gradients** is a technique for attributing a deep network's prediction to its input features. It is **very easy to apply**, **widely applicable** and backed by an **axiomatic theory.**

My email: ataly@google.com

References:

- Axiomatic Attribution for Deep Networks [ICML 2017]
- Did the model understand the question? [ACL 2018]
- Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy [Journal of Ophthalmology, 2018]
- Exploring Principled Visualizations for Deep Network Attributions [EXSS Workshop, 2019]