E9: 309 ADL 23-12-2020

http://leap.ee.iisc.ac.in/sriram/teaching/ADL2020/

Housekeeping

- ✴️ Midterm project II presentations
  - ➡️ Done during Dec. 29,30th
  - ➡️ Same format as previous evaluation
- ✴️ Midterm project III
  - ➡️ Abstract submission deadline (Jan 10th)
    - ✓ Evaluation after final exam (1st week of Feb)
- ✴️ Final Exam (as per IISc schedule)
  - ✓ Jan ~~22nd~~ afternoon! (Jan 23rd final exam)

Topics Discussed thus far

Explainable/Interpretable Deep Learning

Visualizing deep layer activations using tSNE

Interepoch evolution of acvitations

Transferability

Topics Discussed thus far

Explainable/Interpretable Deep Learning

- Visualizing deep layer activations using tSNE
- Backpropagation based approach - Deconv net (Visualization in input)
- Interepoch evolution of acvitations
- Establishing hierarchal representation learning
- Transferability

Topics Discussed thus far

Explainable/Interpretable Deep Learning

saliency

- Visualizing deep layer activations using tSNE
- Backpropagation based approach - Deconv net
- Using Global average pooling and interpolation — CAM
- Interepoch evolution of acvitations
- Establishing hierarchal representation learning
- Using gradients from last layer — Grad-CAM
- Transferability
- Attention

Topics Discussed thus far

Explainable/Interpretable Deep Learning

- Visualizing deep layer activations using tSNE
- Backpropagation based approach - Deconv net
- Using Global average pooling and interpolation — CAM
- Interepoch evolution of acvitations
- Establishing hierarchal representation learning
- Using gradients from last layer — Grad-CAM
- Transferability
- Attention
- Causal Inference

Causal inference

- Causal inference
  - Deriving the causal connection between conditions that cause an effect.
- Three levels of causation
  - association Seeing and observing the environment. Is the incidence of lung cancer higher among smokers?
  - intervention Doing and intervening in the environment. How do we reduce lung cancer? What is the effect if we ban cigarettes?
  - counterfactuals Imagining, restrospection, understanding the environment. What if I had not smoked for the last two years?

Pruning based approach to analyzing/compressing

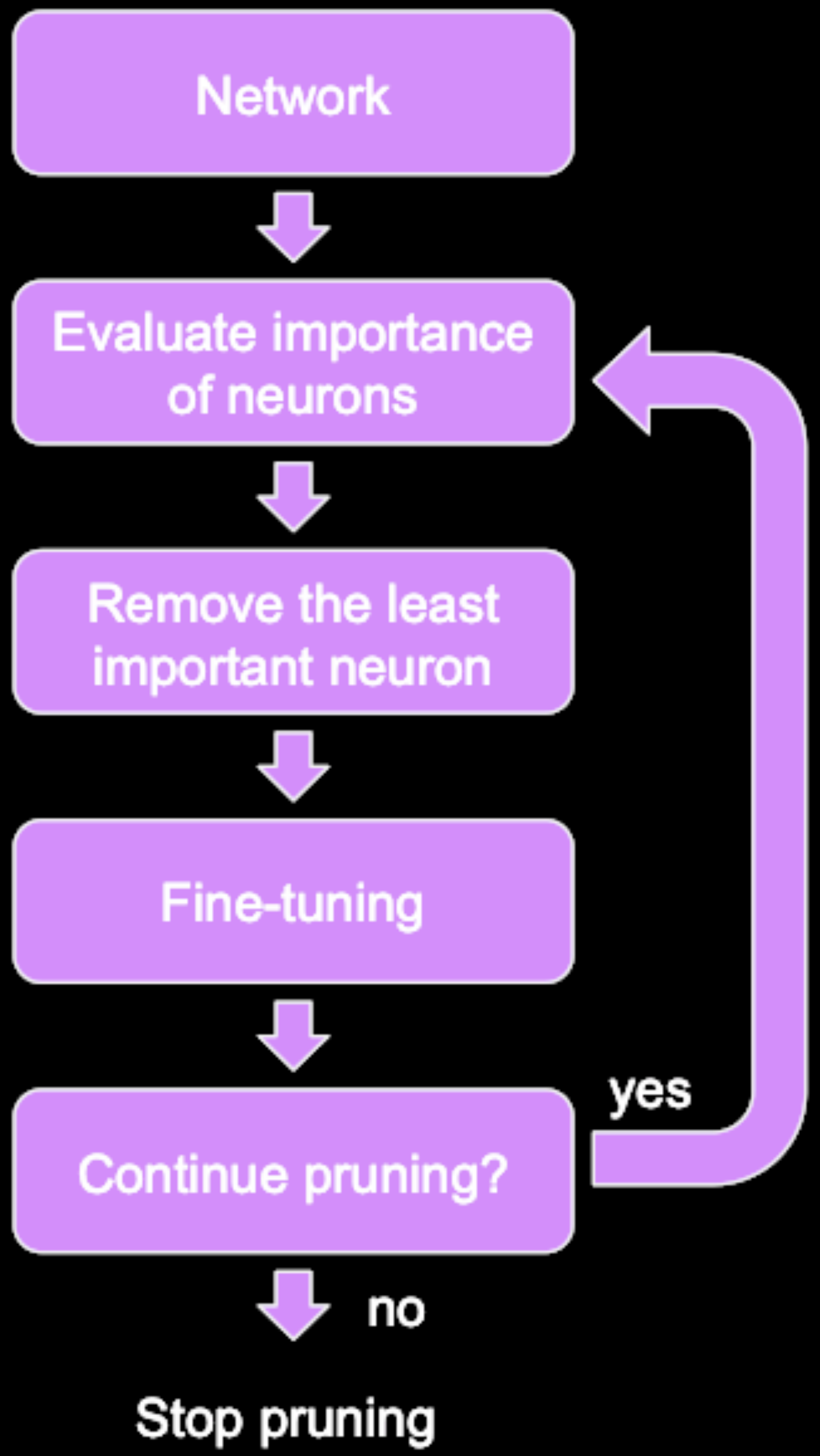
Published as a conference paper at ICLR 2017

PRUNING CONVOLUTIONAL NEURAL NETWORKS FOR RESOURCE EFFICIENT INFERENCE

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, Jan Kautz
NVIDIA
{pmolchanov, styree, tkarras, taila, jkautz}@nvidia.com

Pruning based approach to analyzing/compressing

- Removing connections of a learned neural network
  - Analyzing the effect of this intervention on the output of the model.
    - ✓ Example of intervention based causal model analysis.
- **Pruning is interleaved with fine-tuning**



Pruning based analysis of neural networks

$$\mathbf{h} = \{\mathbf{z}_1^{(1)}, \mathbf{z}_2^{(1)}, \ldots, \mathbf{z}_{C_L}^{(L)}\}$$

- Taylor series expansion based

$$E(\mathcal{D}|\mathbf{h}_i) \approx E(\mathcal{D}|\mathbf{h}_i = 0) + \frac{\partial E}{\partial \mathbf{h}_i}\mathbf{h}_i$$

- Criterion for pruning feature maps

$$\left| \frac{\partial E}{\partial \mathbf{h}_i}^T \mathbf{h}_i \right|$$

Criterion in involved in identifying importance

- With non-linearities like ReLU (deriving gradients can have effects of saturation)

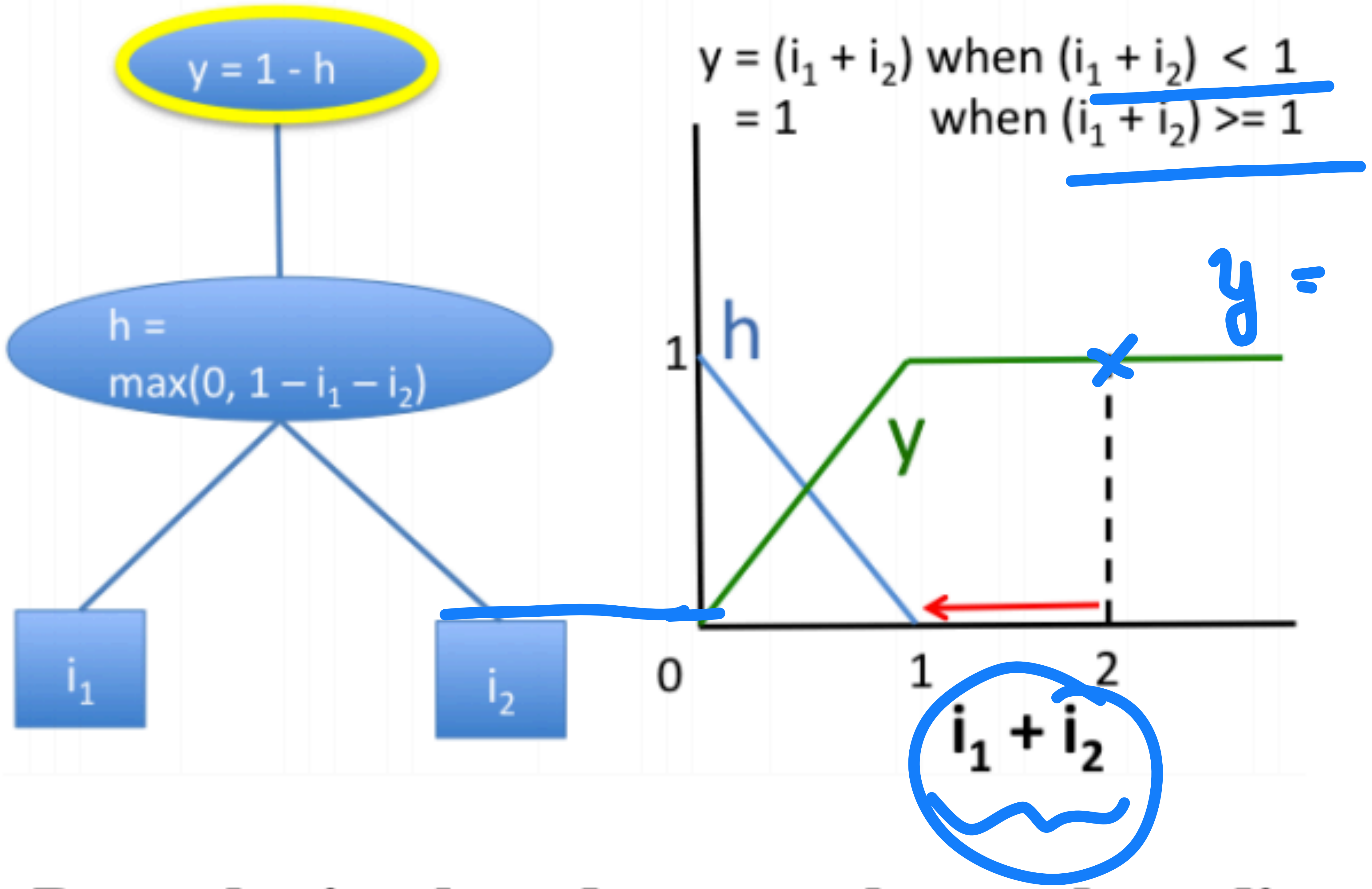


Figure 1. **Perturbation-based approaches and gradient-based approaches fail to model saturation**. Illustrated is a simple network exhibiting saturation in the signal from its inputs. At the point where $i_1 = 1$ and $i_2 = 1$, perturbing either i_1 or i_2 to 0 will not produce a change in the output. Note that the gradient of the output w.r.t the inputs is also zero when $i_1 + i_2 > 1$.

Criterion in involved in identifying importance

- With non-linearities like ReLU (deriving gradients can have effects of saturation)

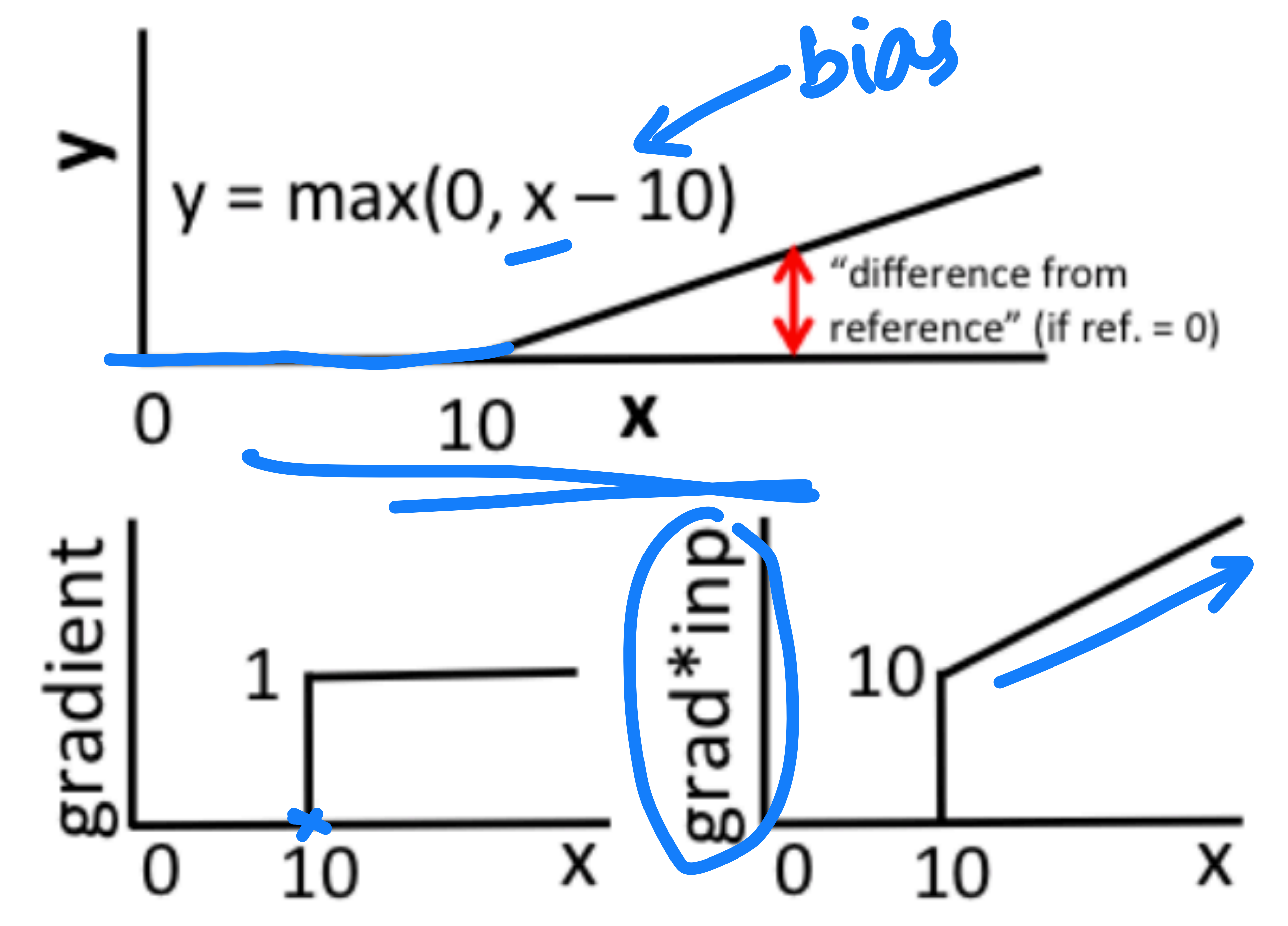


Figure 2. **Discontinuous gradients can produce misleading importance scores**. Response of a single rectified linear unit with a bias of -10. Both gradient and gradient$\times$input have a discontinuity at $x = 10$; at $x = 10 + \epsilon$, gradient$\times$input assigns a contribution of $10 + \epsilon$ to x and -10 to the bias term (ϵ is a small positive number). When $x < 10$, contributions on x and the bias term are both 0. By contrast, the difference-from-reference (red arrow, top figure) gives a continuous increase in the contribution score.

Approximating gradients

- Using finite differences w.r.t. reference.

Learning Important Features Through Propagating Activation Differences

Avanti Shrikumar [1] Peyton Greenside [1] Anshul Kundaje [1]

Approximating gradients

- Let input neurons be defined as (for a given input)

images like MNIST

$$\{x_1, x_2, \ldots x_D\}$$

- let reference input be defined as

$$\{x_1^0, x_2^0, \ldots x_D^0\}$$

References background of 0's for all pixels.

- Let y_c, y_c^0 denote the output of the model (at some dimension) for the given input and the reference. input

Approximating gradients

- Use finite differences for gradients

$$m_{\Delta y_c, \Delta x_j} = \frac{\Delta y_c}{\Delta x_j}$$

$\Delta y_c = y_c - y_c^0$

$\Delta x_j = x_j - x_j^0$

✓ Called as multipliers

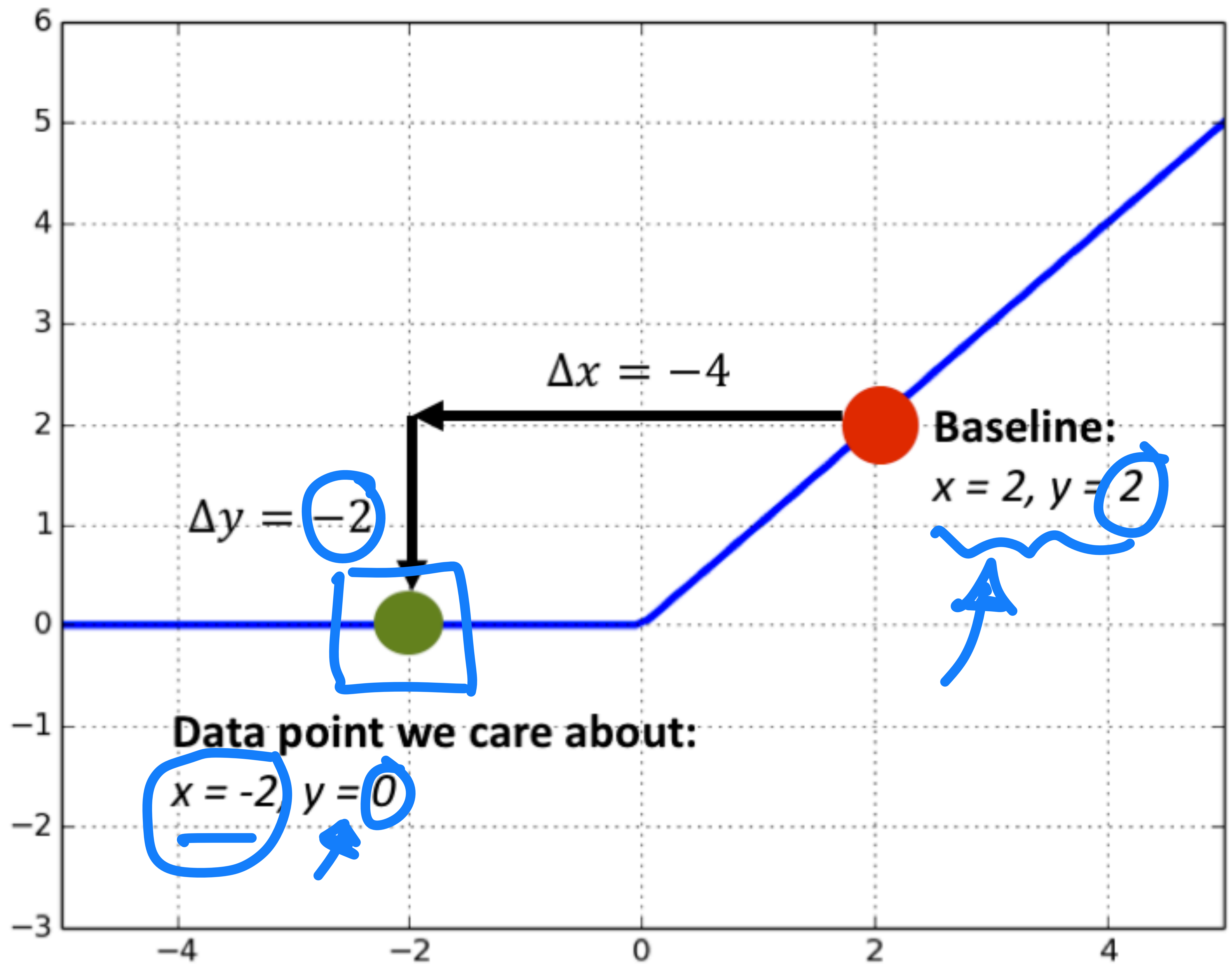
✓ Chain rule of partial deviatives can also be extended to multipliers.

✓ Can be used instead of actual gradients to compute the importance of a feature/hidden layer output.

Interesting math

Illustration of finite gradients for ReLU

$$y = ReLU(x) = max(0, x)$$



x_i — internal neuron

x_i^0 — neuron (ref)

1. Calculating the slope

$$\frac{\Delta y}{\Delta x} = \frac{-2}{-4} = 0.5$$ [approx]

2. Finding the feature importance

$$\Delta x \times \frac{\Delta y}{\Delta x} = -4 \times 0.5 = -2$$

$$x_i \frac{\partial y_i}{\partial x_i}$$ ← approx.

Adversarial Examples and Explainability

Adversarial examples and learning

Adversarial Examples: Attacks and Defenses for Deep Learning

Xiaoyong Yuan, Pan He, Qile Zhu, Xiaolin Li*
National Science Foundation Center for Big Learning, University of Florida
{chbrian, pan.he, valder}@ufl.edu, andyli@ece.ufl.edu

Adversarial attacks and model defenses

- Examples that fool the model
  - Using a trained image classifier published by a third party, a user inputs one image to get the prediction of class label. Adversarial images are original clean images with small perturbations, often barely recognizable by humans. However, such perturbations misguide the image classifier

$$
\begin{aligned}
\min_{x'} \quad & \|x' - x\| \\
s.t. \quad & f(x') = l', \\
& f(x) = l, \\
& l \neq l',
\end{aligned}
$$

f: classifier

Types of Adversarial attacks

- False positive attacks generate a negative sample which is misclassified as a positive one (Type I Error). In a malware detection task, a benign software being classified as malware is a false positive. In an image classification task, a false positive can be an adversarial image unrecognizable to human, while deep neural networks predict it to a class with a high confidence score.

- False negative attacks generate a positive sample which is misclassified as a negative one (Type II Error). In a malware detection task, a false negative can be the condition that a malware (usually considered as positive) cannot be identified by the trained model. False negative attack is also called machine learning evasion. This error is shown in most adversarial images, where human can recognize the image, but the neural networks cannot identify it.

Types of Adversarial attacks

for more consequential

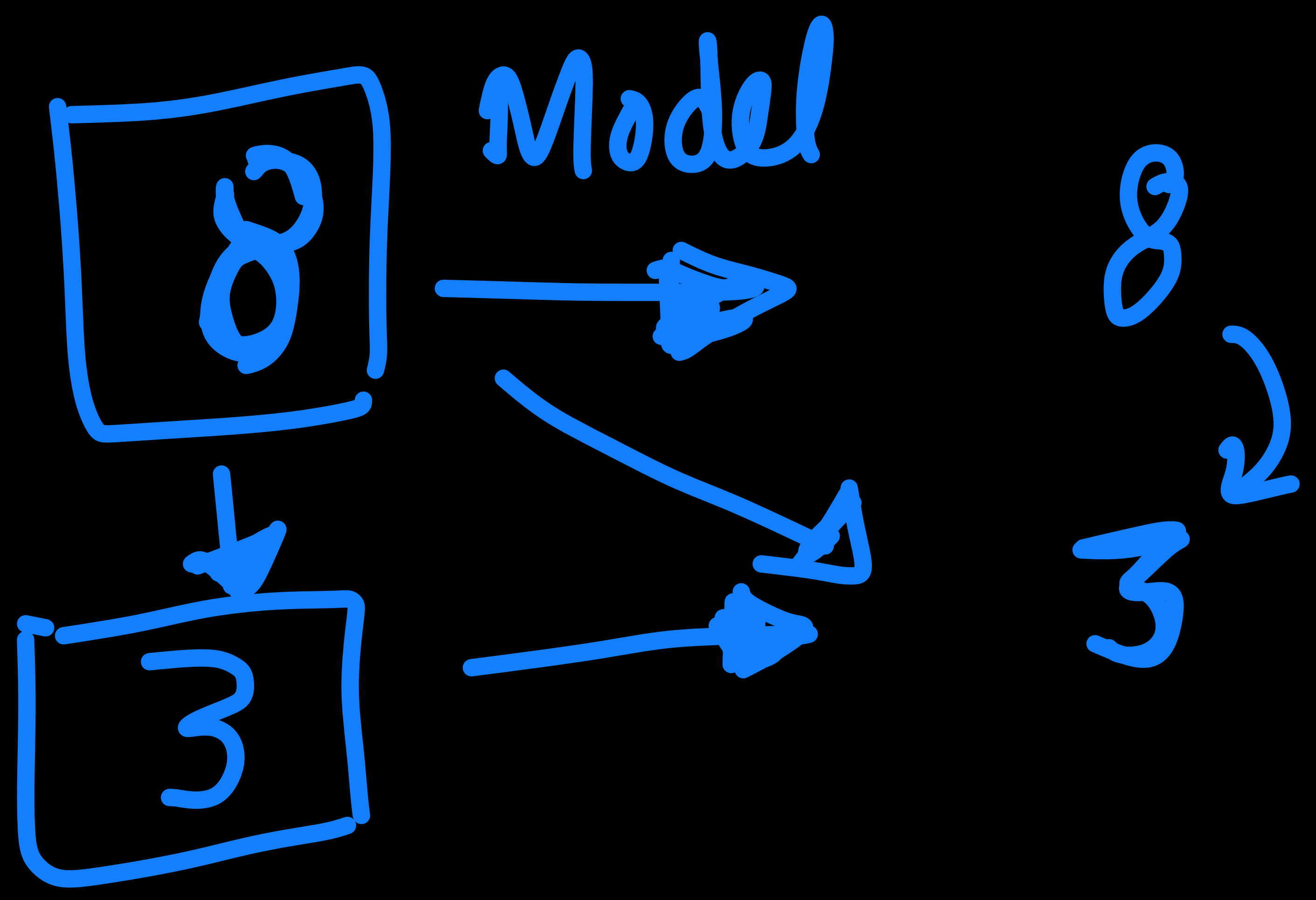
- White box versus Black box

↑ easier

change of label

- Targeted versus Non-targeted

↑ make it all to one class



- One-time versus many time

do you one-shot at fooling the system

Simple adversarial attack

- Fast Gradient Sign Method

$$\mathbf{x}_i = \mathbf{x}_i + \epsilon sign(\frac{\partial E(\mathbf{x}_i, l)}{\partial \mathbf{x}_i})$$

non-targeted adversarial

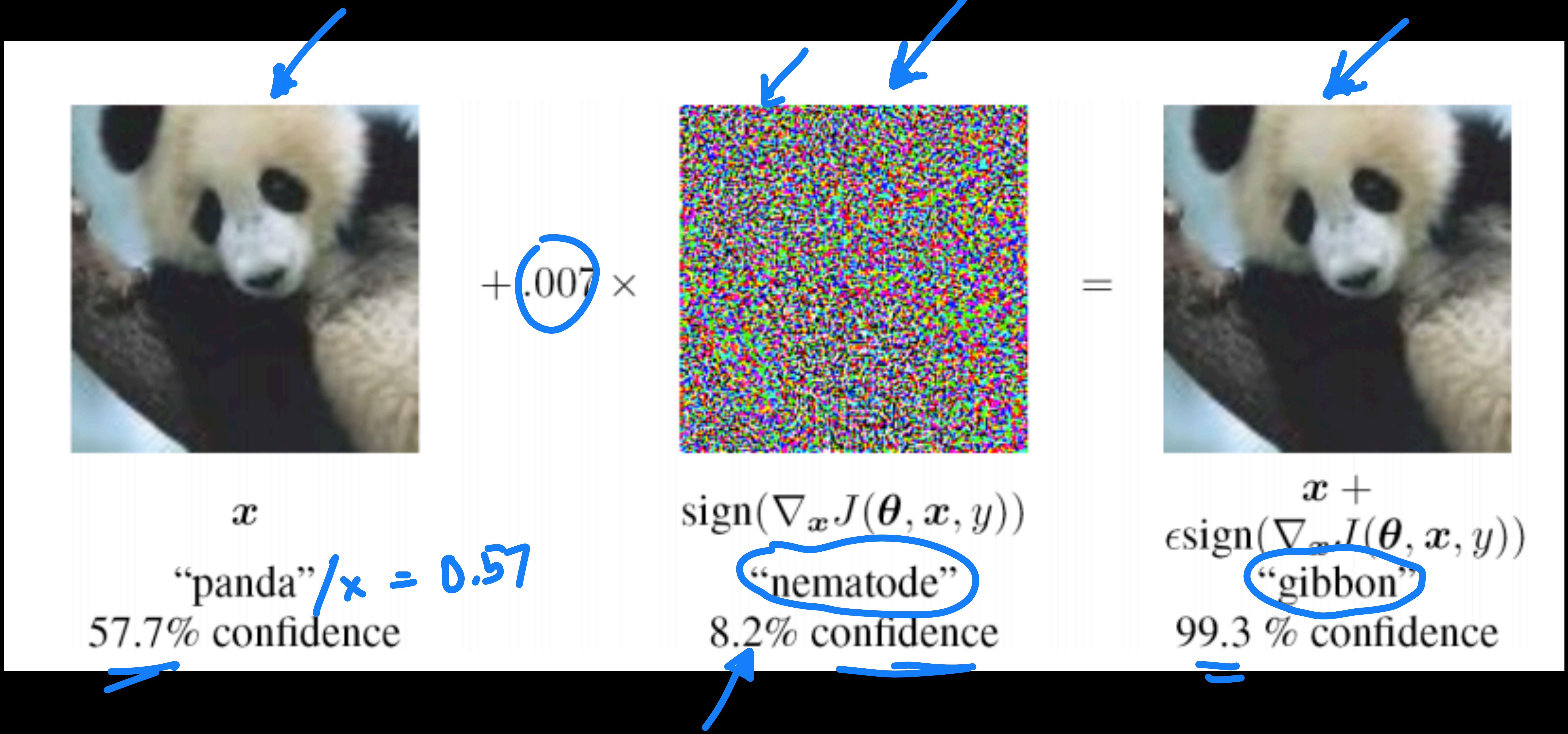
- Move in the direction of the gradient ascent on inputs

- Other similar rules - gradient value based adverasarial learning

$$\mathbf{x}_i = \mathbf{x}_i + \epsilon(\frac{\partial E(\mathbf{x}_i, l)}{\partial \mathbf{x}_i})$$

Simple adversarial attack

- Fast Gradient Sign Method



Adversarial examples

Adversarial Examples for Evaluating Reading Comprehension Systems

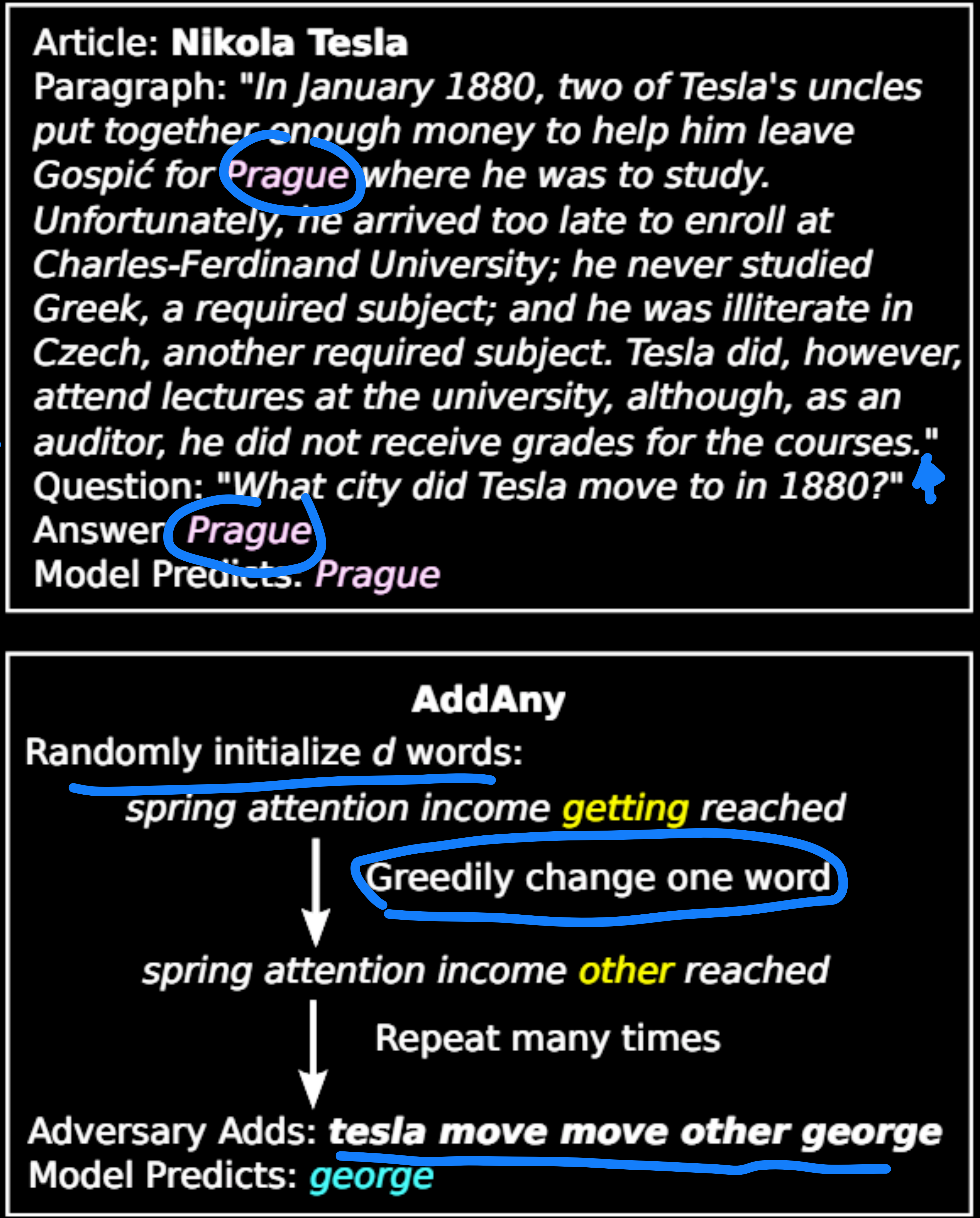
Robin Jia
Computer Science Department
Stanford University
robinjia@cs.stanford.edu

Percy Liang
Computer Science Department
Stanford University
pliang@cs.stanford.edu

Adversarial examples in text

- Using similar methods to gradient based update.
- Adding sentences confuses models which will typically not confuse humans

≠ Locally very smooth



Adversarial attacks

- ✴ Understanding adversarial attacks
    - ✓ Allows explainability
    - ✓ Build defenses to these attacks

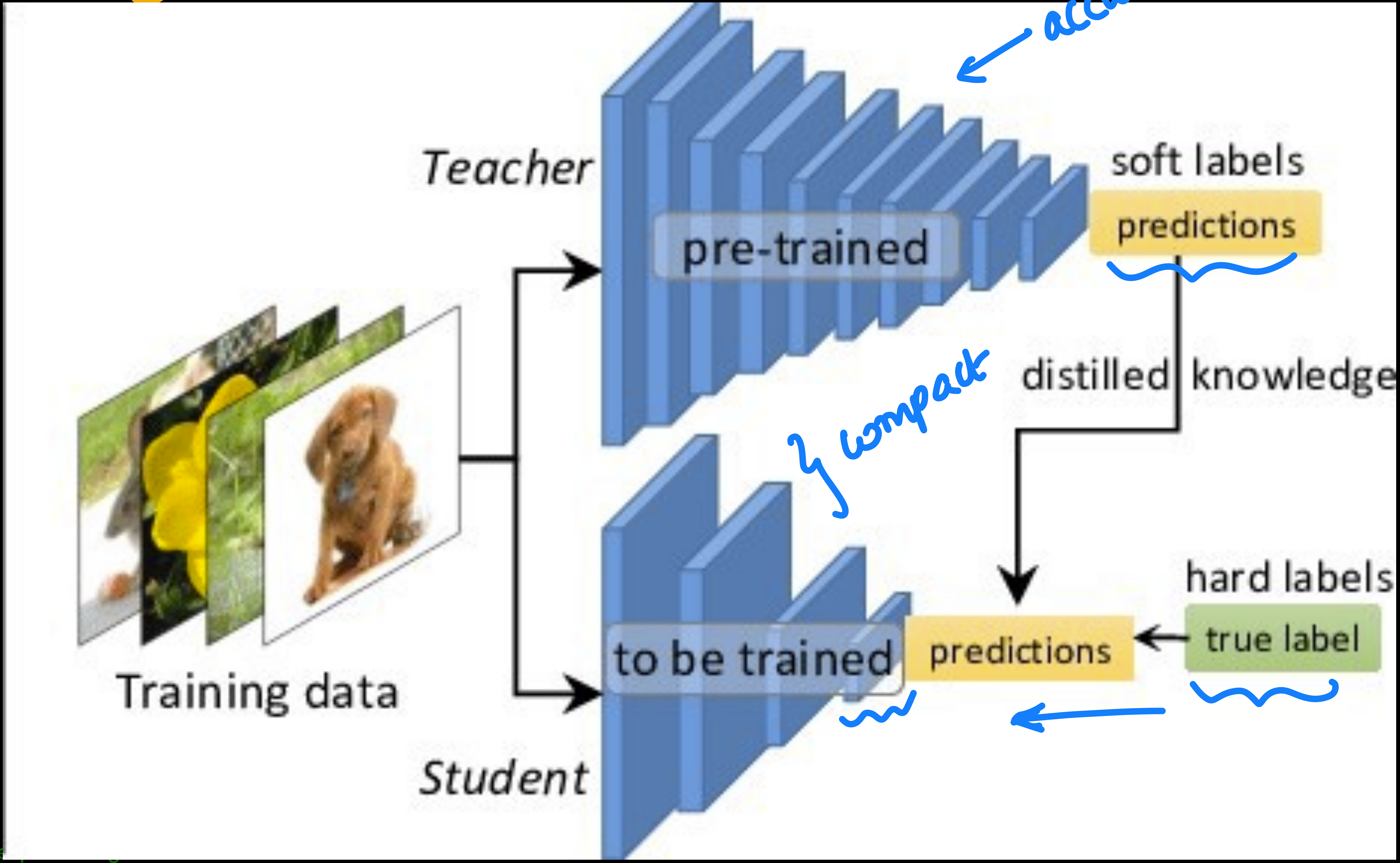
Adversarial example

Defense (data augmentation)

regularization (locally smooth)

Explainability with distillation

Knowledge distillation



Knowledge distillation

- ✴️ Teacher models are complex large neural networks
  - ➡️ Student models are typically lighter models.
- ✴️ Useful in semi-supervised learning
  - ➡️ Student model has to approximate outputs from a teacher model.
    - ✓ Also needs to learn from small amounts of labelled data.

Knowledge distillation for explainability

- Use a simpler explainable model for student model to approximate the deeper model

"Why Should I Trust You?" Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

Knowledge distillation for explainability

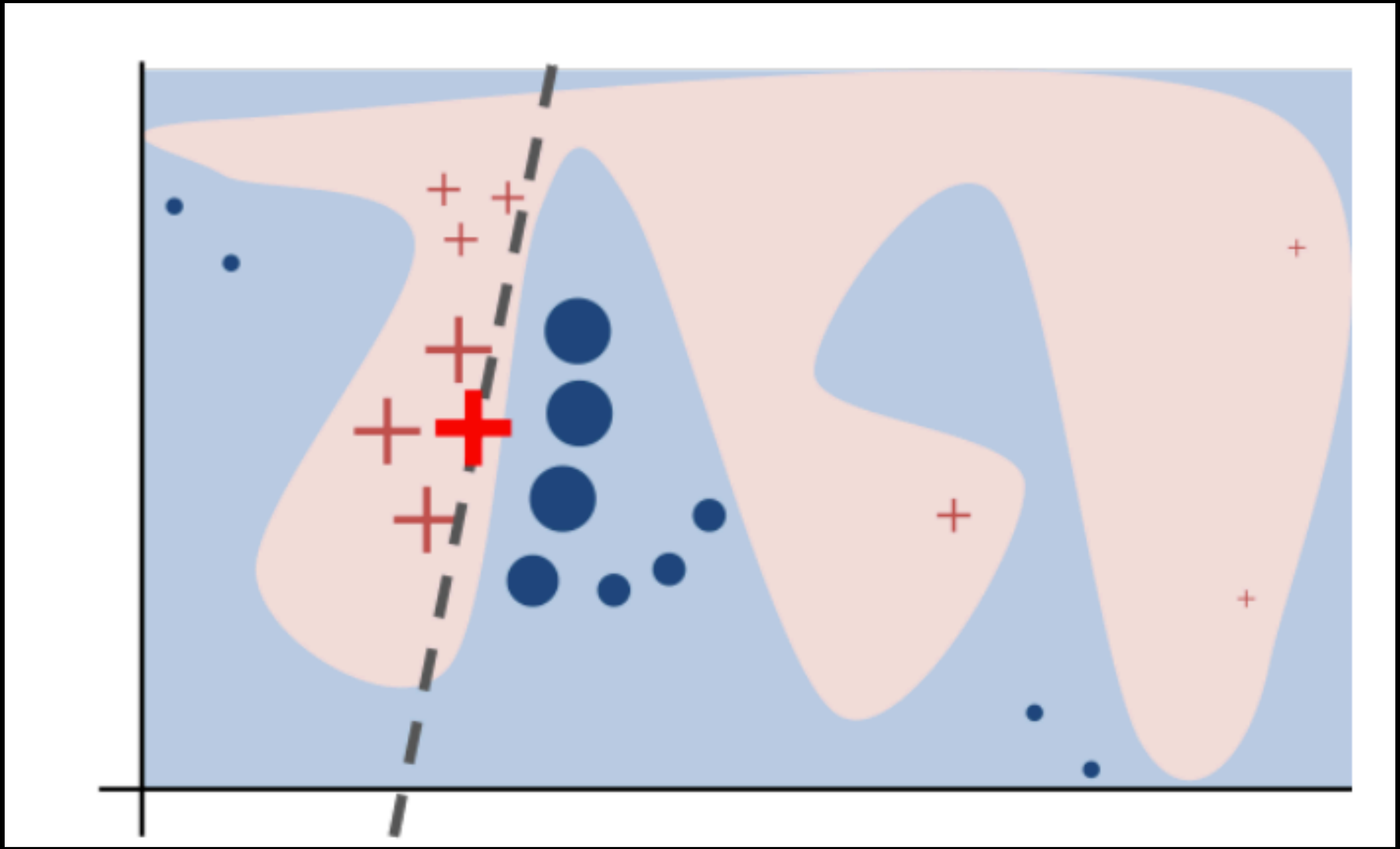
- ✴️ Use a simpler explainable model for student model to approximate the deeper model.

- ✴️ Use locality preservation as a criterion for sampling

    - ✓ Method - Local Interpretable Model Agnostic Representations

    - ✓ Explainability for each sample under consideration

Knowledge distillation for explainability



Knowledge distillation for explainability

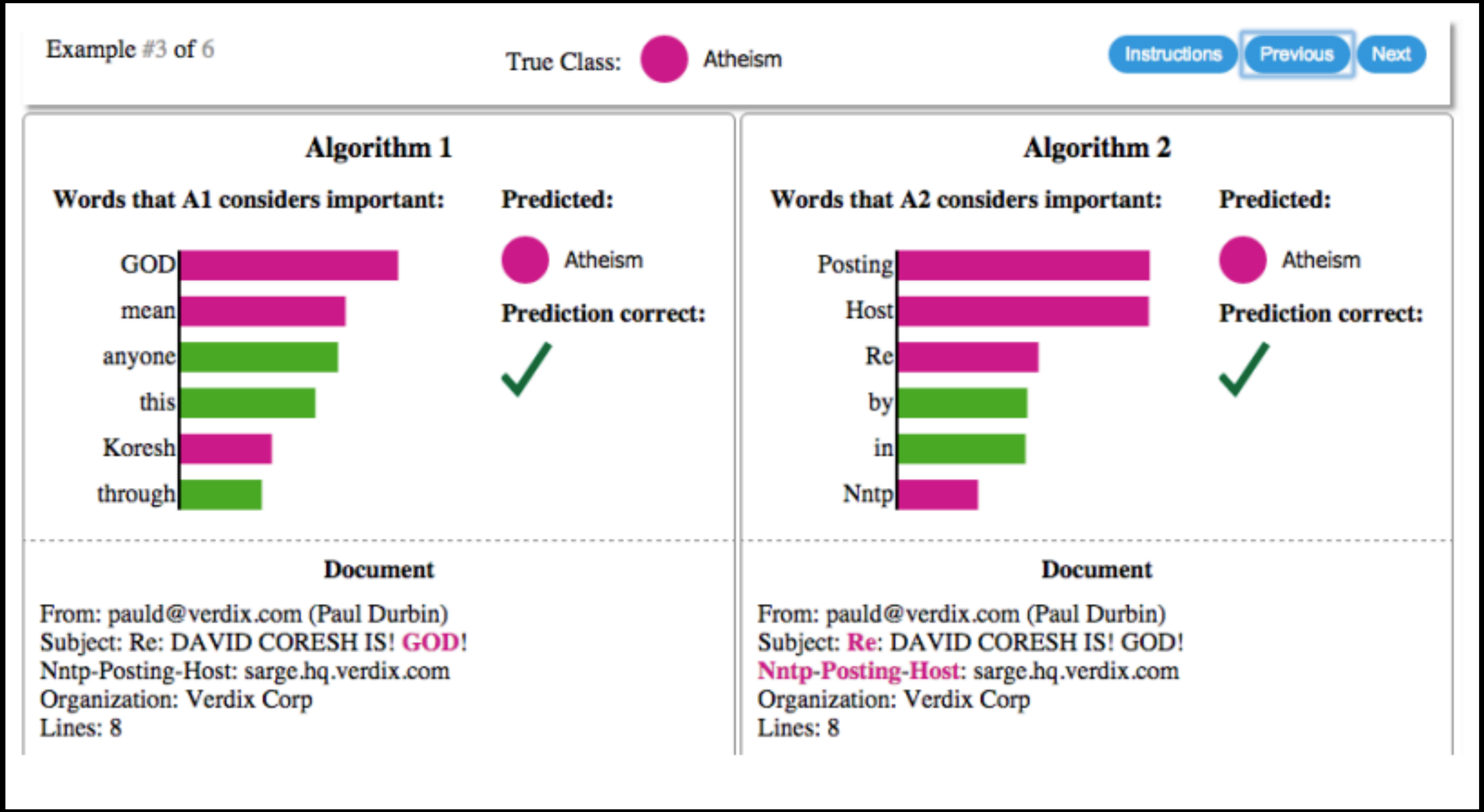
- Let $f(\mathbf{x})$ denote the original neural network
- Let $\mathbf{x}'$ denote the interpretable version of input
- Let $\mathbf{z}, \mathbf{z}'$ denote samples drawn around input and its interpretable version.

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) \left(f(z) - g(z')\right)^2$$

- The function g can be sparse linear regression

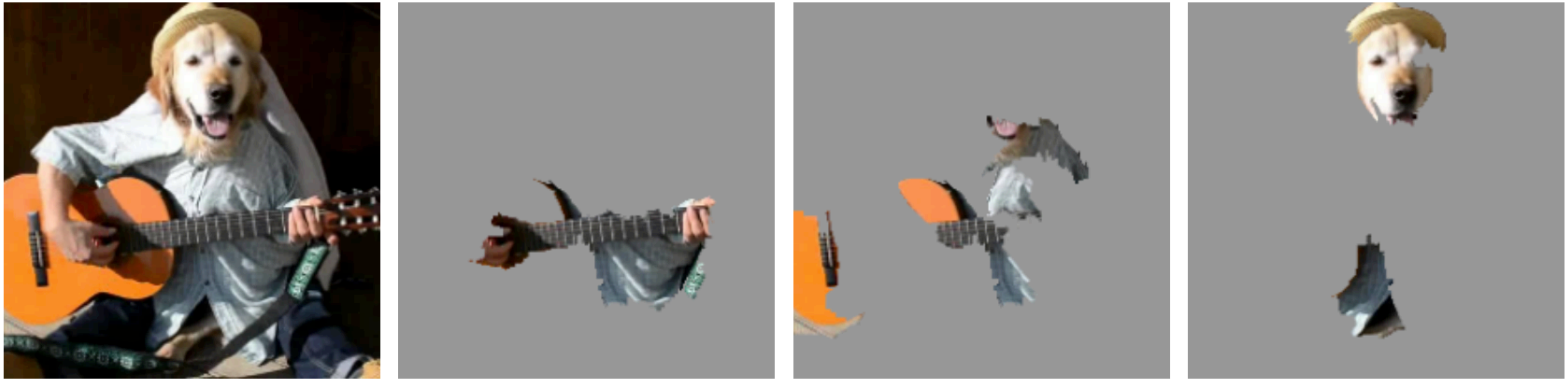
LIME model - text example

- Building sparse linear regression for each output class



LIME model - Image example

- Building sparse linear regression for each output class



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*