E9: 309 ADL 23-12-2020

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

Housekeeping

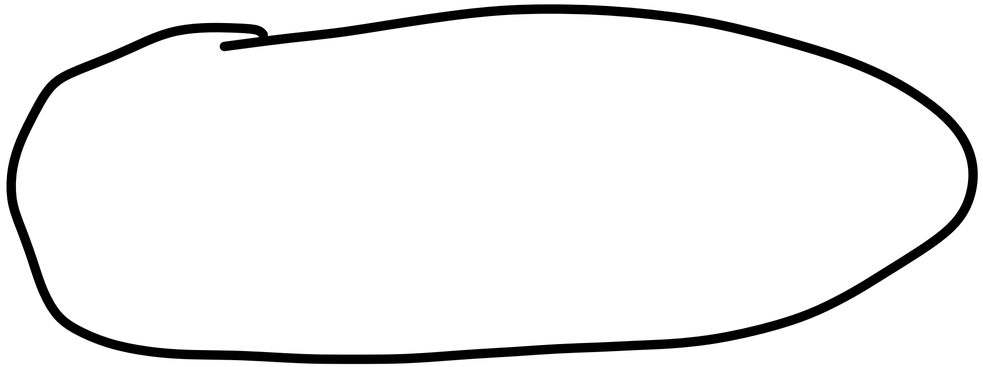
- Midterm project II presentations
  - Done during Dec. 29,31st (4-6pm)
  - Same format as previous evaluation
- Midterm project III
  - Abstract submission deadline (Jan 10th)
    - ✓ Evaluation after final exam (1st week of Feb)
- Final Exam (as per IISc schedule)
    - ✓ Jan 2nd afternoon!

All reports and slides by 11am 29th tomorrow) Teams folder

first name _ last name _ proj 2 _ (report/slides) (pdf/ppt)

3rd (Saturday)

Topics Discussed thus far

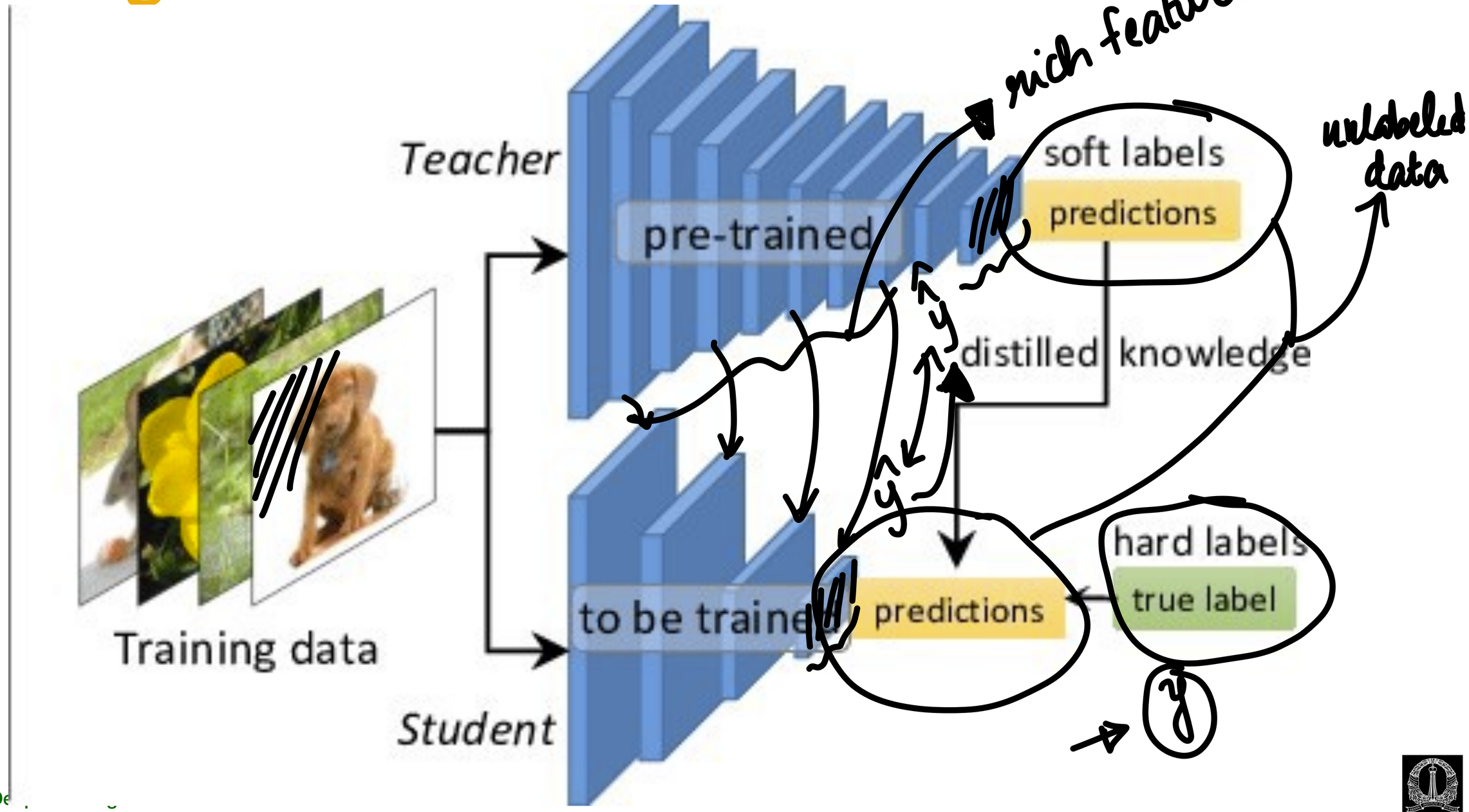


Adversarial attacks

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

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

LIME

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

local approx of a complex model with a simpler one.

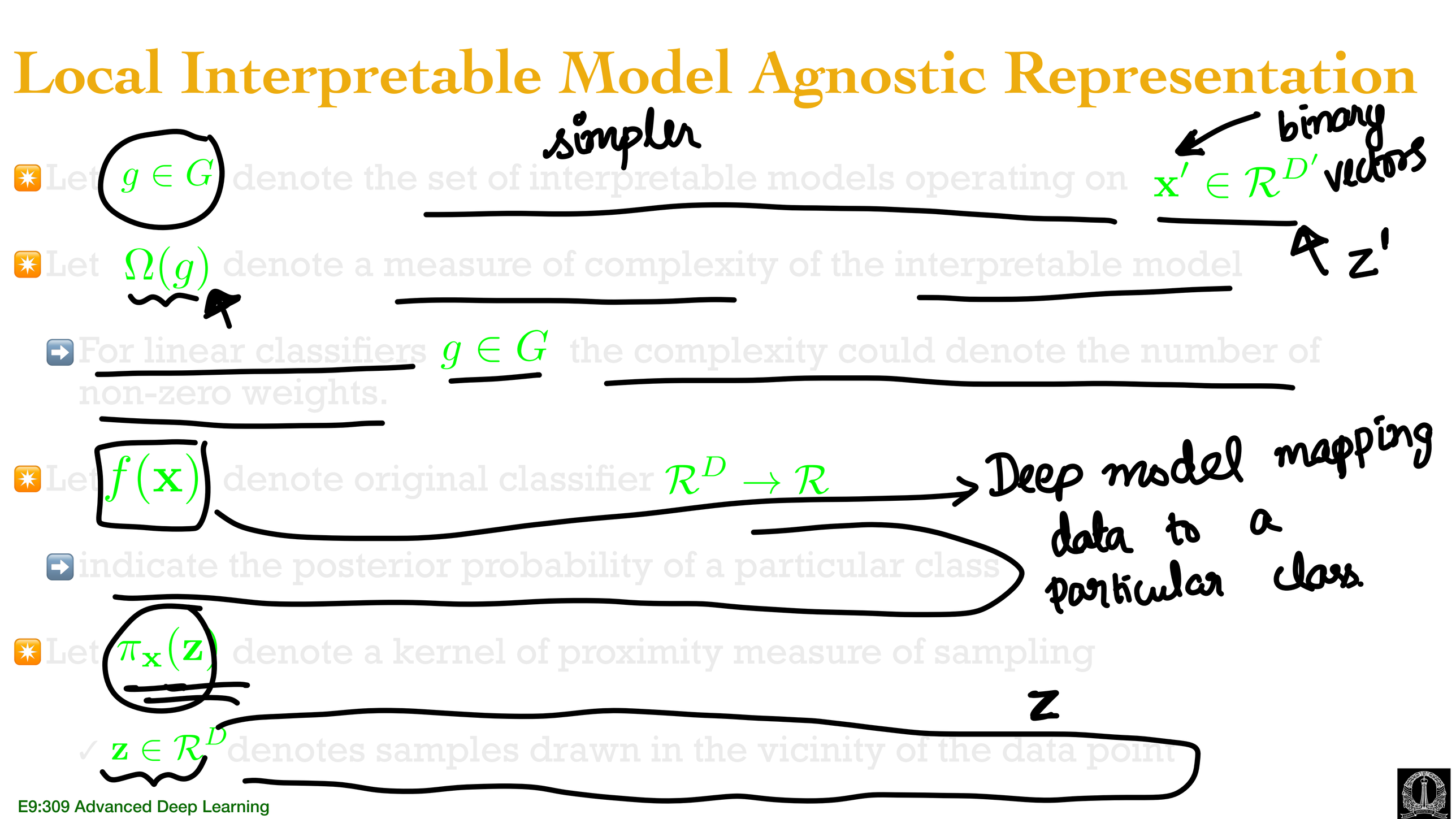
Local Interpretable Model Agnostic Representation

- A possible interpretable representation for text classification is a binary vector indicating the presence or absence of a word, even though the classifier may use more complex (and incomprehensible) features such as word embeddings.
- Likewise for image classification, an interpretable representation may be a binary vector indicating the "presence" or "absence" of a contiguous patch of similar pixels (a super-pixel), while the classifier may represent the image as a tensor with three color channels per pixel.
- Let $\mathbf{x} \in \mathcal{R}^D$ denote the original data $\mathbf{x}' \in \mathcal{R}^{D'}$ be the interpretable representation

interpretable version

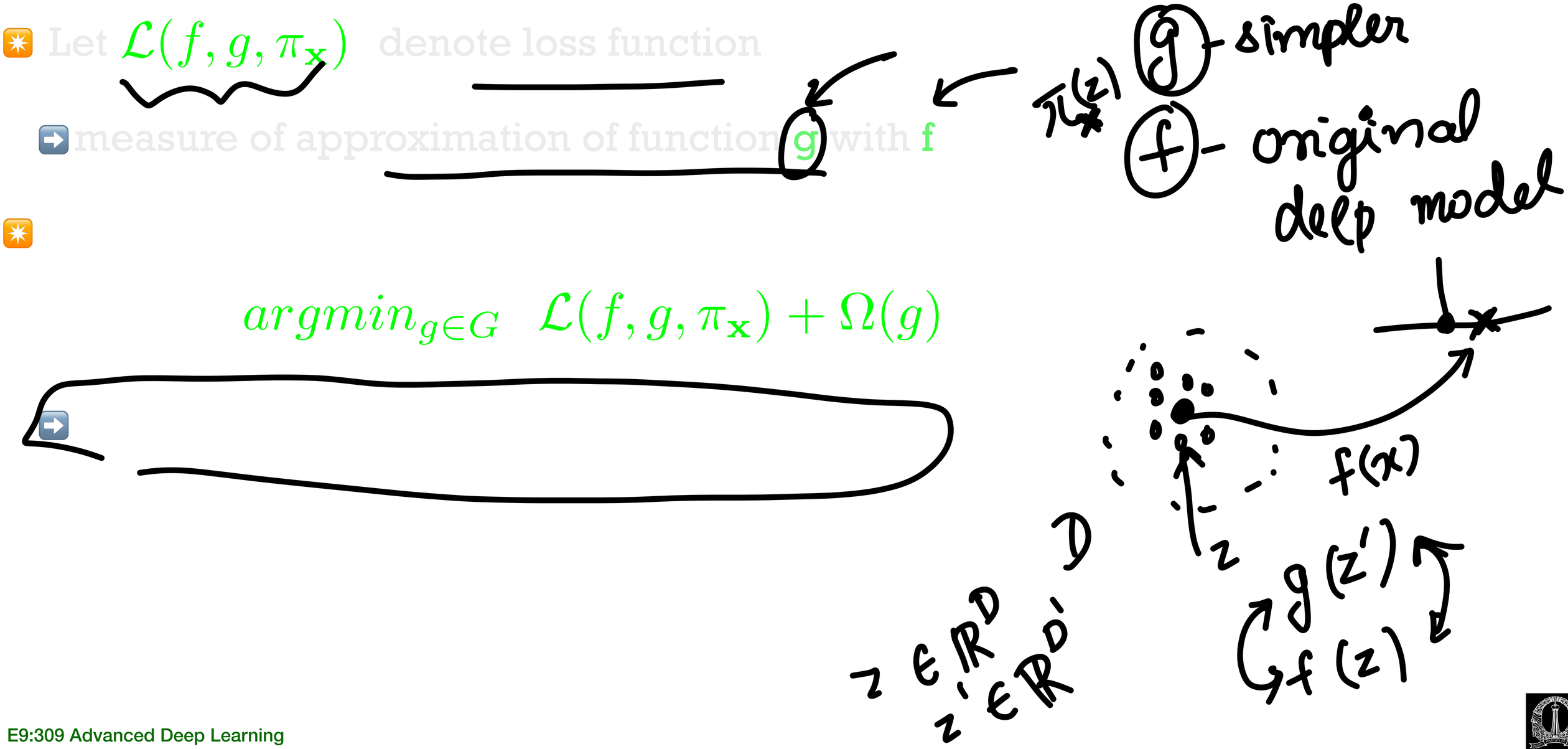
Local Interpretable Model Agnostic Representation

- Let $g \in G$ denote the set of interpretable models operating on $\mathbf{x}' \in \mathcal{R}^{D'}$
- Let $\Omega(g)$ denote a measure of complexity of the interpretable model $g \in G$
  - For linear classifiers the complexity could denote the number of non-zero weights.
- Let $f(\mathbf{x})$ denote original classifier $\mathcal{R}^D \rightarrow \mathcal{R}$
  - indicate the posterior probability of a particular class
- Let $\pi_{\mathbf{x}}(\mathbf{z})$ denote a kernel of proximity measure of sampling
  - $\mathbf{z} \in \mathcal{R}^D$ denotes samples drawn in the vicinity of the data point



Local Interpretable Model Agnostic Representation

- Let $\mathcal{L}(f, g, \pi_{\mathbf{x}})$ denote loss function
  - measure of approximation of function g with f
- $argmin_{g \in G} \quad \mathcal{L}(f, g, \pi_{\mathbf{x}}) + \Omega(g)$
  -



- For example

$$\mathcal{L}(f, g, \pi_{\mathbf{x}}) = \sum_{\mathbf{z}, \mathbf{z}'} \pi_{\mathbf{x}}(\mathbf{z})(f(\mathbf{z}) - g(\mathbf{z}'))^2$$

proximity

D - distance

- With $\pi_{\mathbf{x}}(\mathbf{z}) = e^{-D(\mathbf{x}, \mathbf{z})^2}$ and $g = \mathbf{w}_g^T \mathbf{z}'$ $\Omega(g) = ||\mathbf{w}_g||_0$

↑ linear classifier

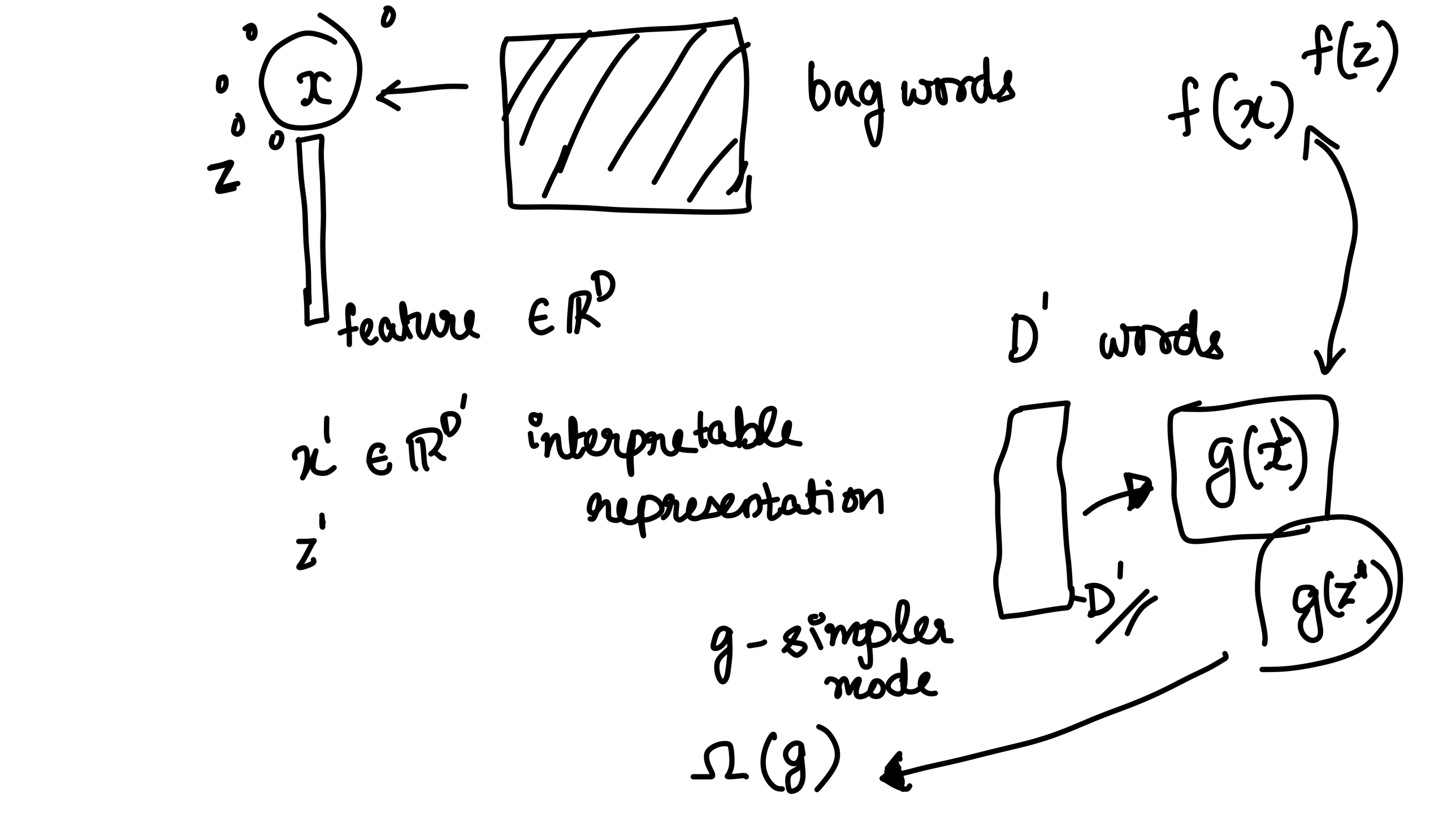
← non-zero weights

- Solve the sparse optimization problem

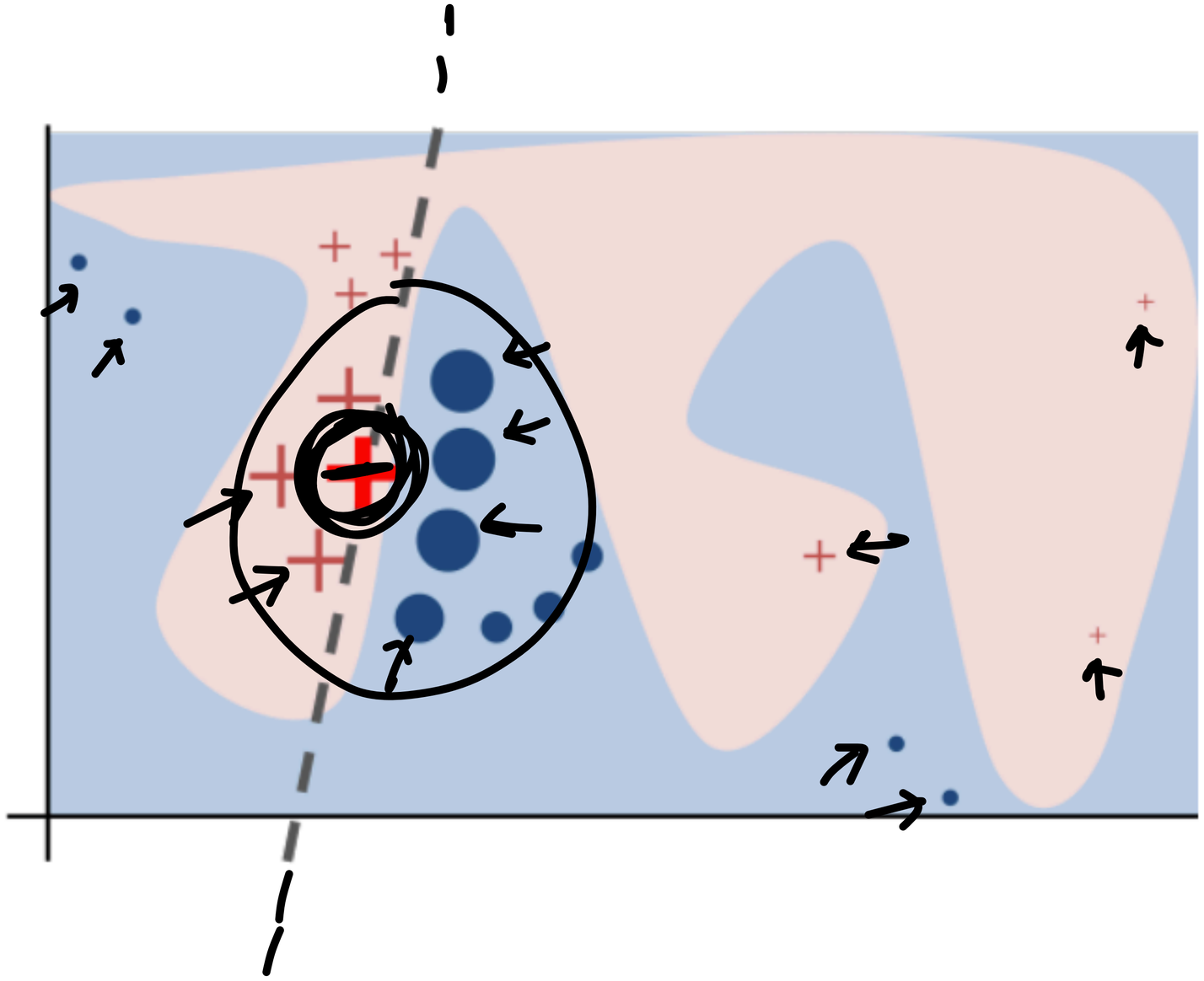
✓ $argmin_{g \in G} \; \mathcal{L}(f, g, \pi_{\mathbf{x}}) + \Omega(g)$

✓ Using LASSO style algorithm

$\underline{w}_g$

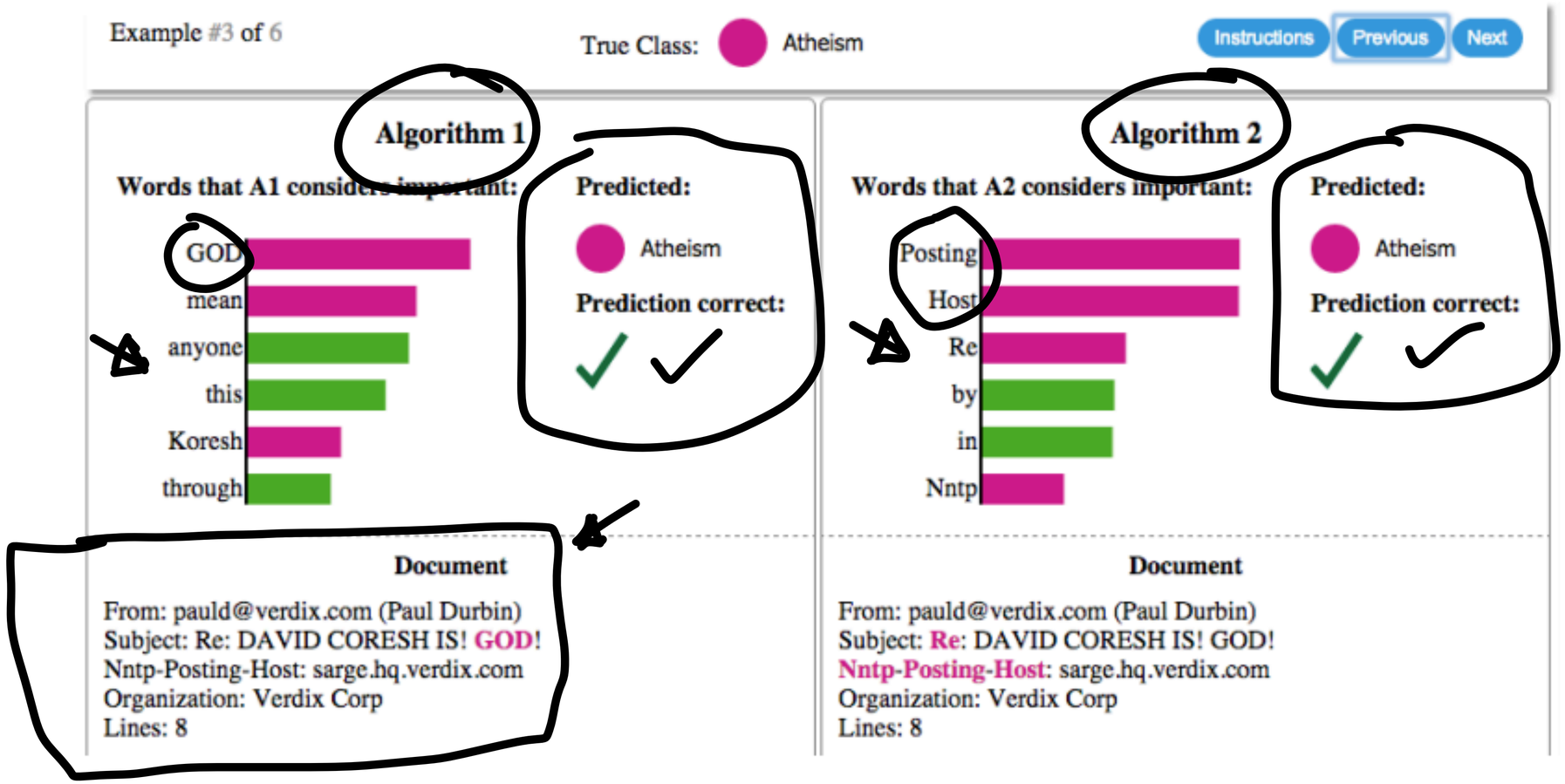

x
z
bag words
feature $\in \mathbb{R}^D$
$x' \in \mathbb{R}^{D'}$
interpretable representation
z'
f(z)
f(x)
D' words
g(z)
g(z')
D'
g - simpler model
$\Omega(g)$

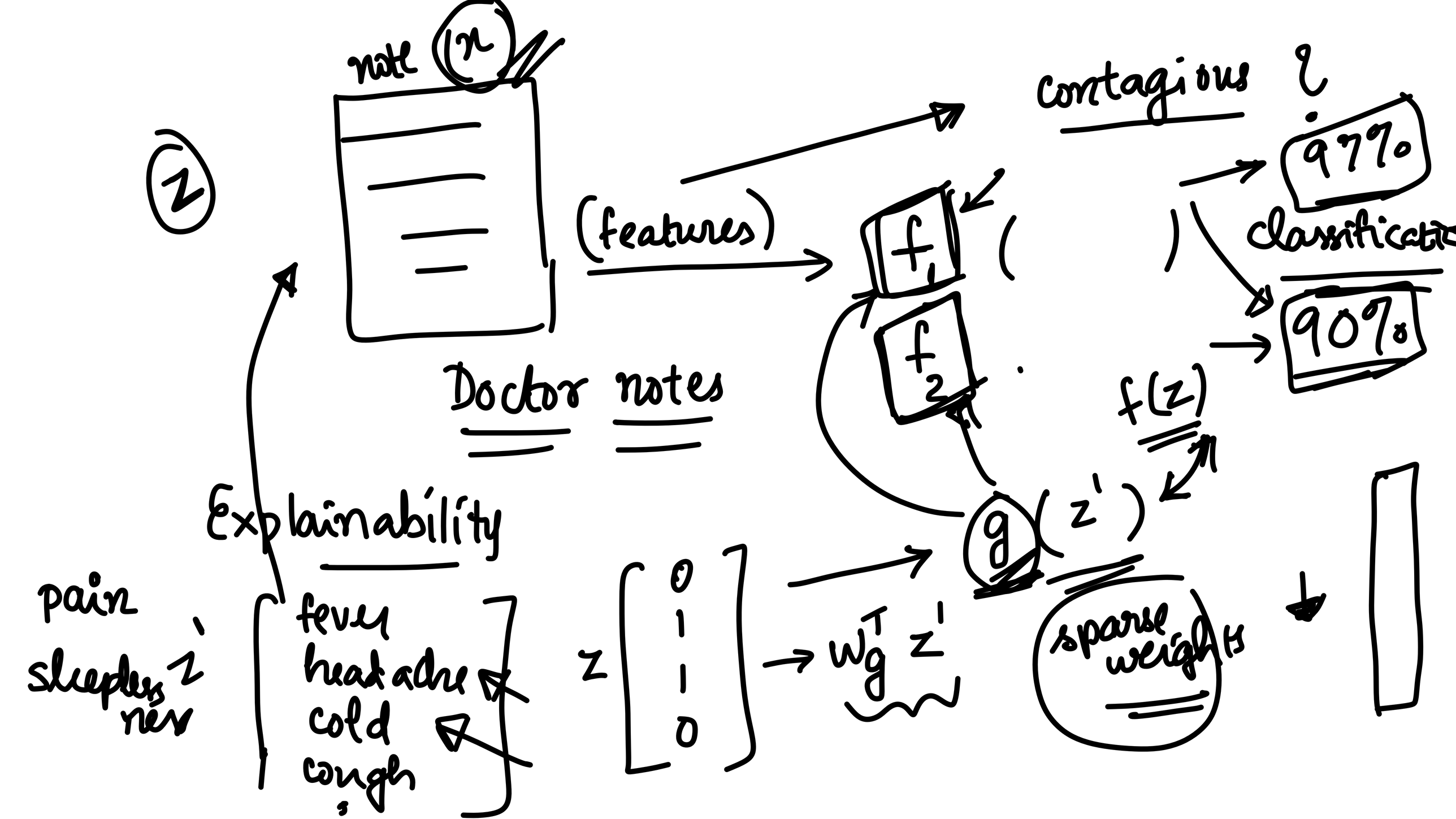
Knowledge distillation for explainability



LIME model - text example

- Building sparse linear regression for each output class



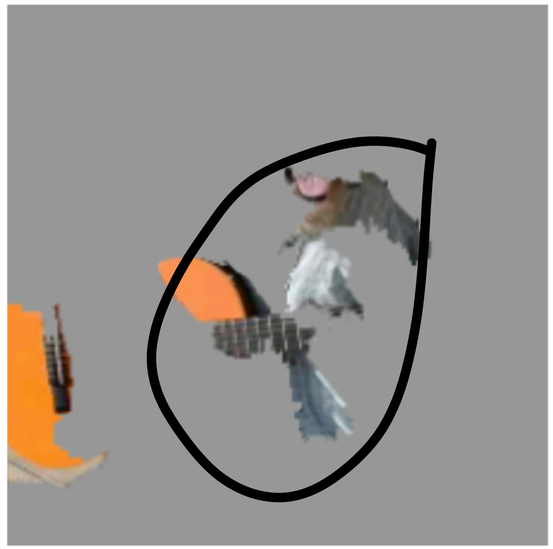

note
x
z
(features)
contagious ?
97%
90%
Doctor notes
Explainability
pain
sleepless nes
fever
headache
cold
cough
z
0
1
1
0
f(z)
g
z'
sparse weights

LIME model - Image example

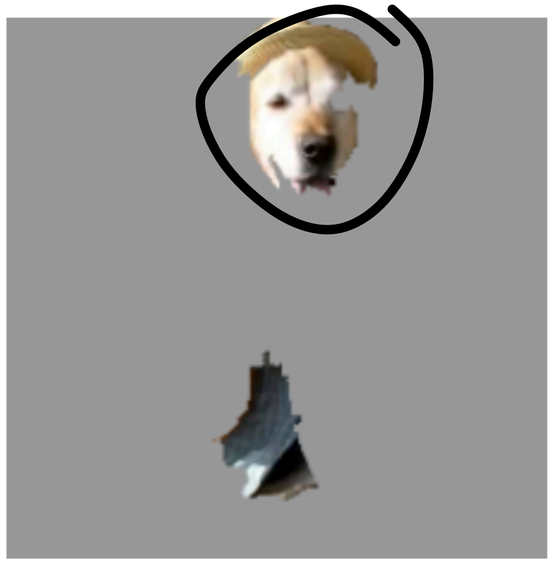
- Building sparse linear regression for each output class

(a) Original Image

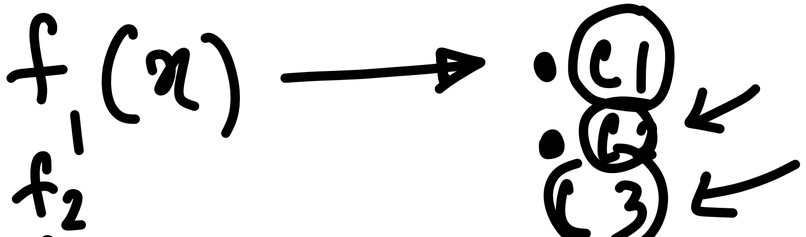
(b) Explaining *Electric guitar*



(c) Explaining *Acoustic guitar*



(d) Explaining *Labrador*



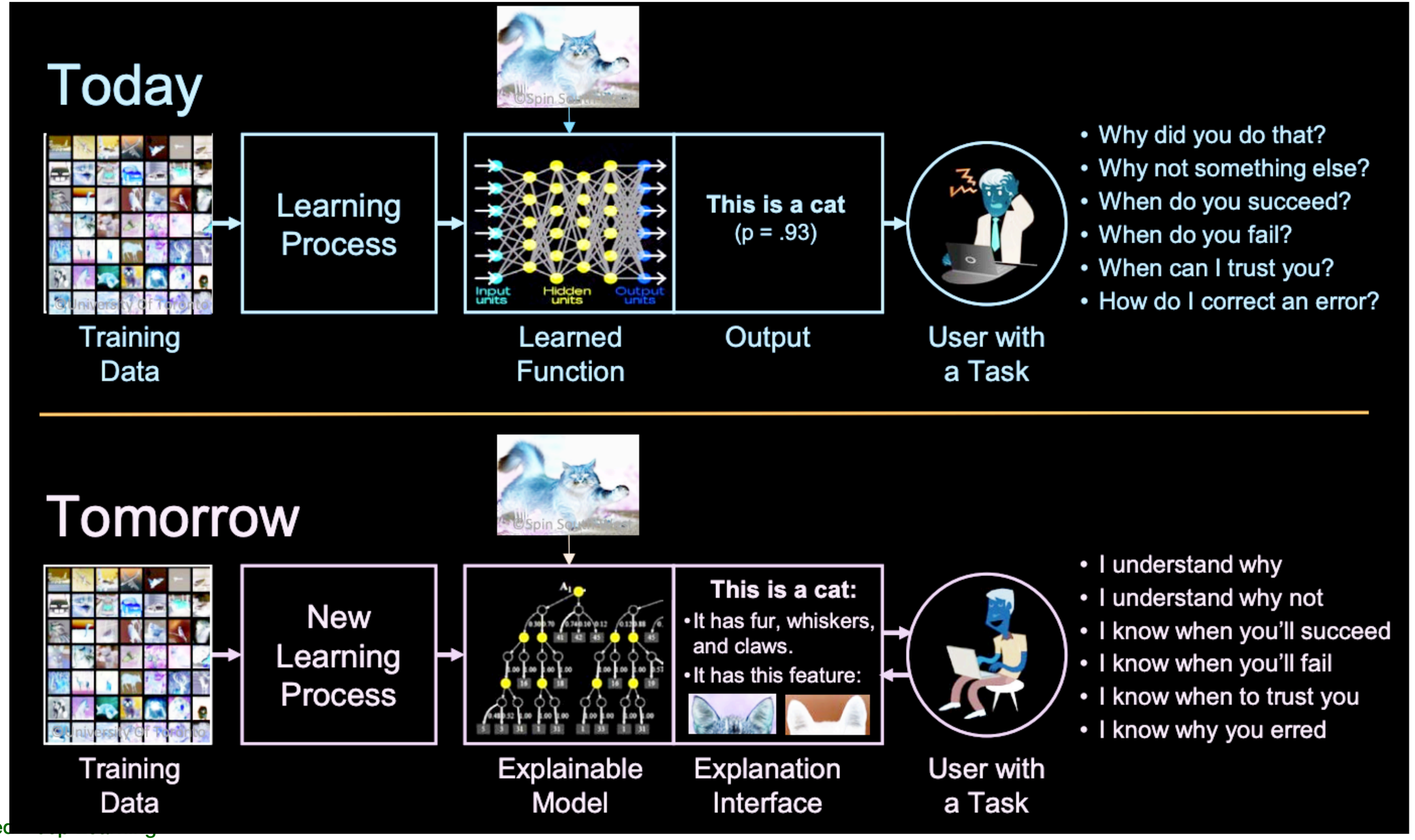
Identifying classification errors

(a) Husky classified as wolf

(b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

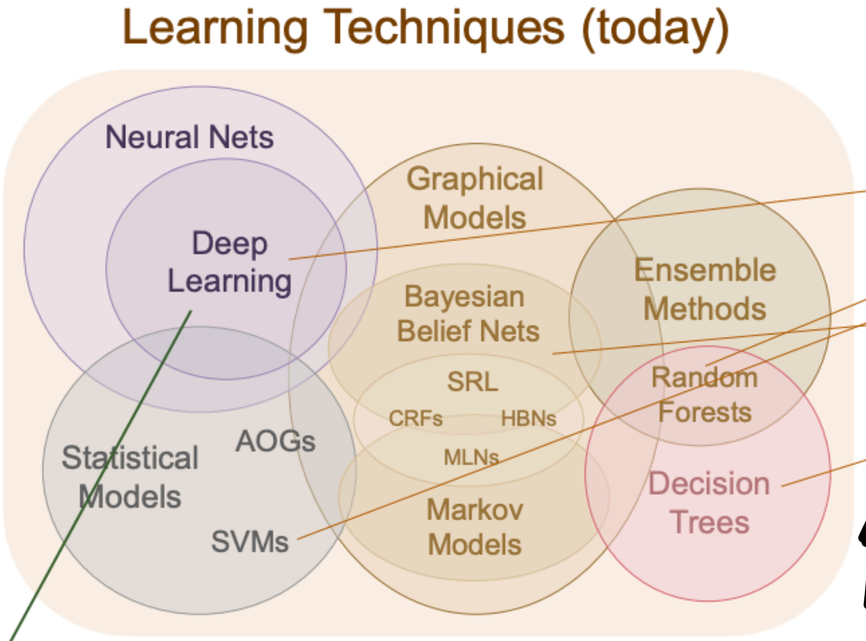
Future Research Directions

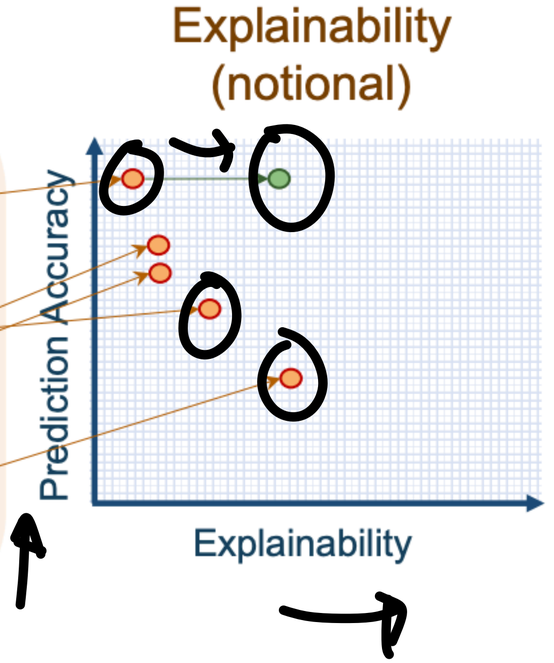


Future Research Directions

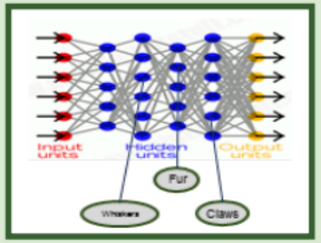
New Approach

Create a suite of machine learning techniques that produce more explainable models, while maintaining a high level of learning performance





XAI

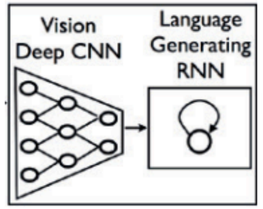


Deep Explanation

Modified deep learning techniques to learn explainable features

Future Research Directions

Generating Image Captions

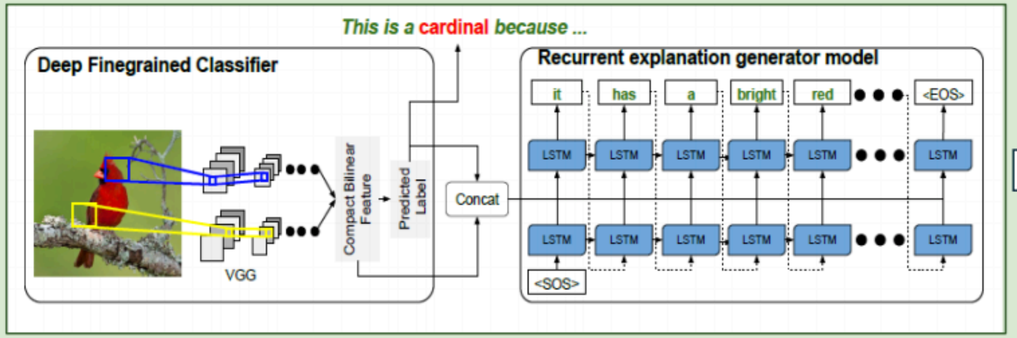


A group of people shopping at an outdoor market

There are many vegetables at the fruit stand

- A CNN is trained to recognize objects in images
- A language generating RNN is trained to translate features of the CNN into words and captions.

Generating Visual Explanations

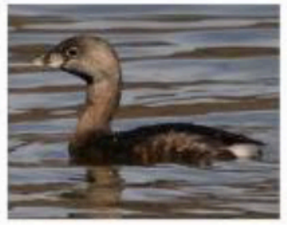


Researchers at UC Berkeley have recently extended this idea to generate explanations of bird classifications. The system learns to:

- Classify bird species with 85% accuracy
- Associate *image descriptions* (discriminative features of the image) with *class definitions* (image-independent discriminative features of the class)

Example Explanations

This is a Kentucky warbler because this is a yellow bird with a black cheek patch and a black crown.



This is a pied billed grebe because this is a brown bird with a long neck and a large beak.

Limitations

- Limited (indirect at best) explanation of internal logic
- Limited utility for understanding classification errors

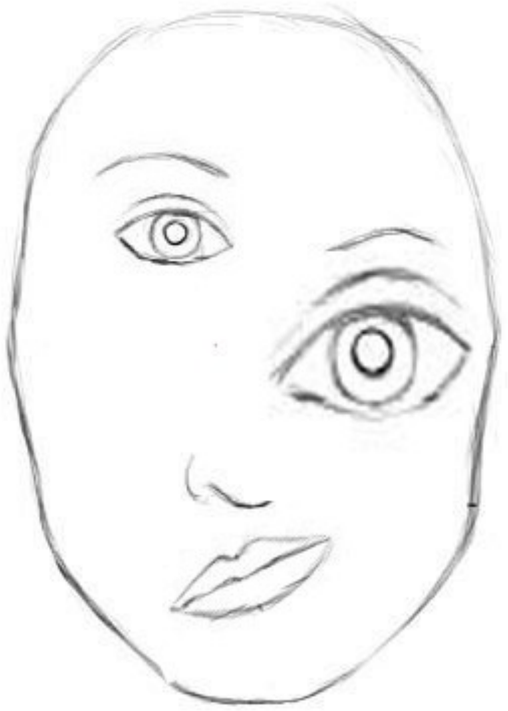
Hendricks, L.A, Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., and Darrell, T. (2016). Generating Visual Explanations, arXiv:1603.08507v1 [cs.CV] 28 Mar 2016

Topics thus far …

- **Visual and Time Series Modeling:** Semantic Models, Recurrent neural models and LSTM models, Encoder-decoder models, Attention models.
- **Representation Learning, Causality And Explainability:** t-SNE visualization, Hierarchical Representation, , gradient and perturbation analysis, Topics in Explainable learning, Structural causal models.
- **Unsupervised Learning:** Restricted Boltzmann Machines, Variational Autoencoders, Generative Adversarial Networks.
- **New Architectures:** Capsule networks, , Transformer Networks.
- **Applications:** Applications in in NLP, Speech, Image/Video domains in all modules.
-

Problem with current deep learning networks

- Convolutional neural networks with filtering and max pooling layers
  - ✓ Generate indicators of object parts
  - ✓ check the presence of object parts and confirm the object identity.
  - ✓ Often fails to understand the spatial relationship between object parts.



Capsule networks

- Traditional networks
  - Weigh the inputs and generate a scalar output
- Key idea in capsule networks (neurons to capsules)
  - ✓ Move inidividual neuron outputs from scalar to vector
  - ✓ Encode the probability of presence of an attribute along the magnitude of the vector output
  - ✓ Encode the pose (translation + rotation) in the angle of the vector output

Neurons versus Capsules

Capsule vs. Traditional Neuron			
Input from low-level capsule/neuron		$\text{vector}(\mathbf{u}_i)$	$\text{scalar}(x_i)$
Operation	Affine Transform	$\hat{\mathbf{u}}_{j\|i} = \mathbf{W}_{ij}\mathbf{u}_i$	–
	Weighting	$\mathbf{s}_j = \sum_i c_{ij}\hat{\mathbf{u}}_{j\|i}$	$a_j = \sum_i w_i x_i + b$
	Sum		
	Nonlinear Activation	$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}$	$h_j = f(a_j)$
Output		$\text{vector}(\mathbf{v}_j)$	$\text{scalar}(h_j)$