

# Interpretability in Machine Learning

Kris Sankaran

Nepal Winter School in AI

December 26, 2018

# Outline

Introduction

Distillations

Perturbations

Influence Functions

Concept Activation Vectors

## Learning Objectives

- ▶ Realize that deep learning models aren't necessarily black boxes
- ▶ Distinguish between types of interpretability studied in literature
- ▶ Understand foundational distillation and perturbation-based methods

## What is interpretability?

- ▶ “Ability to explain or present in understandable terms to a human.”
- ▶ I would add: ability to predict (by hand) the model’s behavior under different interventions
  - Do I get the same prediction if I slightly change neuron 132 in layer 3?
  - What if I deploy my self-driving cars in snowy Montreal, after training in sunny Palo Alto?
  - What if change the subject’s race in data point  $x_i$ ?

## General Approaches

- ▶ Resort to a more easily interpretable model
  - Distillation: Compress complex model into simpler, more interpretable one
  - Design: Create new model classes competitive with Deep Learning, but easier to interpret
- ▶ Quantify effect of perturbations
  - Saliency maps: Pixel-level feature importance
  - Influence Functions: Importance of individual training examples
  - Concept Activation: Sensitivity to user-defined concepts

## Linear Models

- ▶ In some ways, a gold standard
  - Sign and size of  $\beta_j$  is meaningful
  - Effects of changes in  $x_j$  easy to predict (even when extrapolating!)
- ▶ But even here, can be issues
  - True effect can be null, but still see nonzero  $\beta_j$
  - Correlated input lead to unstable coefficients
  - Outliers lead to counterintuitive behavior

## Decision Trees

- ▶ In some ways, a gold standard
  - Can trace “decision-making” process
  - Effects of changes in  $x_j$  easy to predict (even when extrapolating!)
- ▶ But even here, can be issues
  - Paths can get complicated for even moderately deep trees
  - Correlated input lead to unstable splitting patterns

# Outline

Introduction

Distillations

Perturbations

Influence Functions

Concept Activation Vectors

## Distillation

- ▶ Intuition: Large models do well because they have better search strategies, but learned decision boundaries can be approximated by simpler model classes
- ▶ Strategy: Train a complicated teacher model, and “distill” it into a simpler student model
- ▶ (Besides interpretability, often useful for model compression)

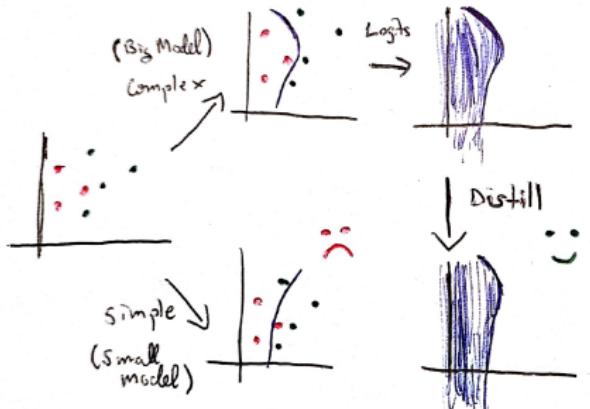


Figure: Distilling a complex model to a simple model class often works better than attempting to directly learn within the simple model class.

## Distillation by Trees

- ▶ Proposal: Approximate teacher logits by class of soft decision trees
  - Branching probabilities at node  $i$ :  $\sigma(x^T w_i + b_i)$
  - Internal nodes still learn filters  $w_i$
  - But now can inspect paths for each decision

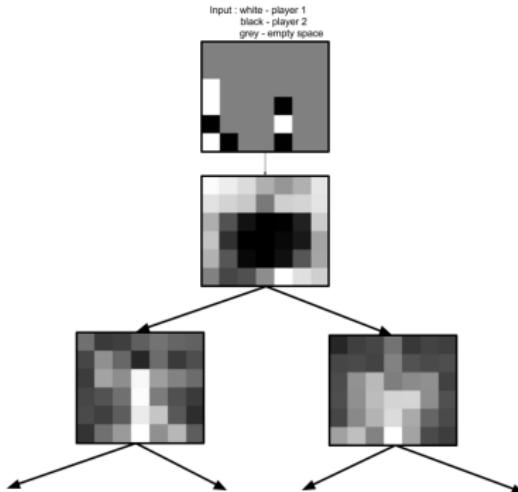


Figure: It's possible to inspect learned features at decision nodes, as in this Connect 4 example.

## Additive Models

- ▶ Approximate regression / decision surface by additive model with interactions
  - $F(x) = b_0 + \sum h_j(x_j) + \sum_{j \neq k} h_{jk}(x_j, x_k) + \dots$
- ▶ Can directly inspect changes in predictions when changing values of specific features
- ▶ Good for tabular data
- ▶ Bad when features need to be learned from raw inputs

# Additive Models

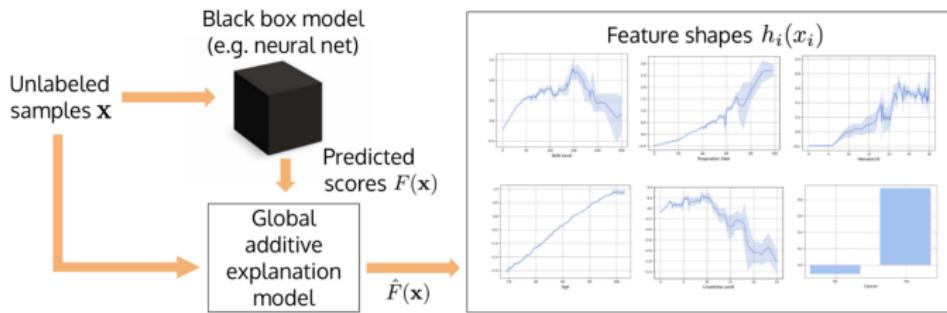


Figure: Example first-order additive features learned by distillation.

# Outline

Introduction

Distillations

Perturbations

Influence Functions

Concept Activation Vectors

## Perturbations

- ▶ Intentional perturbation can be illuminating
- ▶ What happens when you change...
  - Raw pixel values?
  - Presence of training example?
  - Layer activation values?

# Saliency Maps

- ▶ For a given image, how would class predictions change if you manipulated a pixel?
- ▶ Simplest version:  $\nabla_x f_\theta(x)$  change in class given infinitesimal changes in inputs

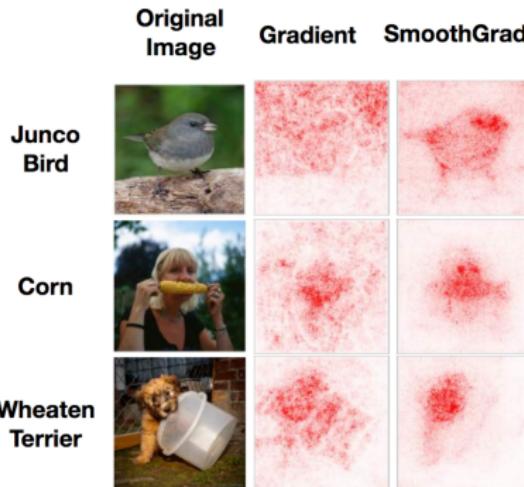


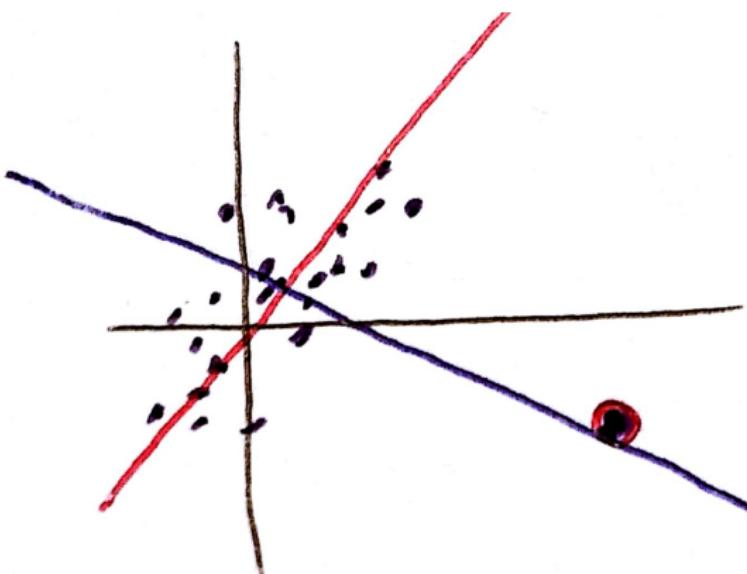
Figure: Example saliency maps. Smooth-grad just averages the gradient map over many noisy versions of input image.

## Cautionary advice...

- ▶ While simple to implement, use cases aren't entirely clear
- ▶ Still have to inspect one example at a time
- ▶ Many proposals don't pass *sanity checks*

## Influence Functions

- ▶ Classical notion from statistics: Influential datapoints
- ▶ Measures sensitivity of fit to removal of points



**Figure:** The difference between fits when you include vs. remove a point is called that point's influence.

## Contributions

- ▶ Change in loss at (potentially new point)  $z$  when  $\epsilon$ -upweight training pair  $z_i = (x_i, y_i)$

$$I(z, z_i) = -\nabla_{\theta} \ell\left(z, \hat{\theta}\right)^T H_{\hat{\theta}}^{-1} \nabla_{\theta} \ell\left(z_i, \hat{\theta}\right)$$

- ▶ Alignment between gradients of loss at new and old points, after adjusting for local curvature
  - More alignment  $\rightarrow$  larger decrease in loss
- ▶ Applied of SGD-able approximations to Hessian  $H_{\hat{\theta}}^{-1}$

# Applications

- ▶ Identifying most influential points for specific test cases
- ▶ Identifying mislabeled points in training data
- ▶ Not restricted to deep learning!

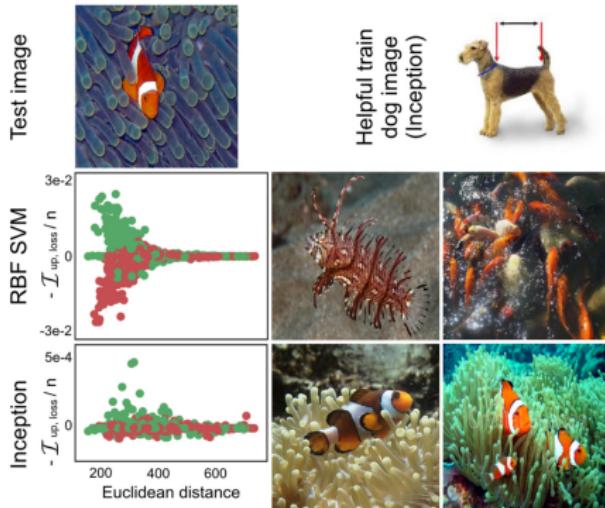


Figure: Example influential observations from koh2017understanding

## Concept Activation Vectors

- ▶ What if you had a specific “concept” that you wanted to test model sensitivity to?
  - Importance of anemone to predicting clownfish?
- ▶ Need workable definition of a concept...

## Formulation

- ▶ Idea: Have the user provide concepts!
- ▶ Study how concept affects feature activations, one layer at a time

## User concepts

- ▶ User creates set of “concept” samples (or chooses them from a database, using image tags, say)
- ▶ Create activations  $f_l$  at layer  $l$  for both concept and random nonconcept examples
- ▶ Define concept activation  $v_C^l$  based on learned linear decision boundary

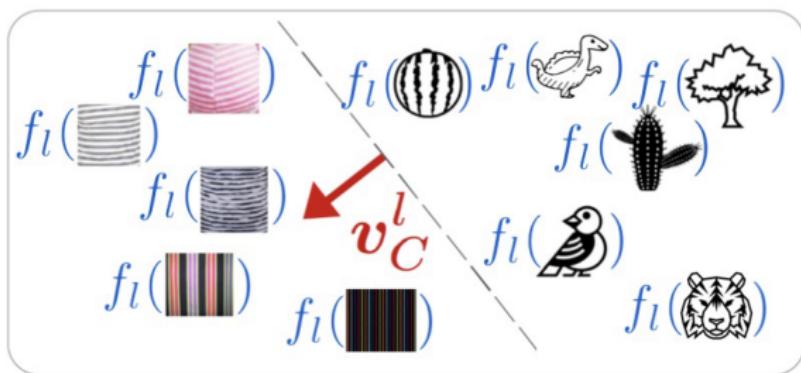


Figure: The CAV vector is defined as the direction that separates samples representative of a concept (like stripes) and unrelated random samples.

## Effect of Concepts

- ▶ See how logit  $h_{lk}$  changes when nudging layer  $l$  activations  $f_l(x)$  for example  $x$  towards concept  $C$

$$S_{k,C,l}(x) = \lim_{\epsilon \downarrow 0} \frac{h_{lk}(f_l(x) + \epsilon v_C^l) - h_{lk}(f_l(x))}{\epsilon}$$

- ▶ Averaging this over examples  $x$  from class of interest

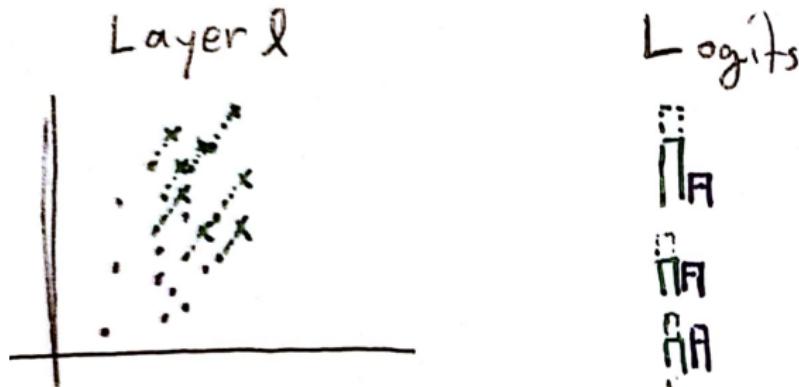
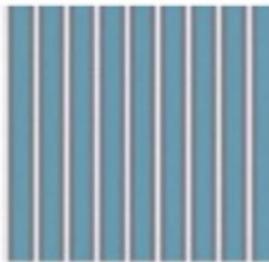


Figure: The CAV score for a class is the amount the logit for that class changes in response to perturbing its examples in the CAV direction.

## Applications

- ▶ Find examples within a class most and least similar to specified concept

***CEO concept:*** most similar striped images

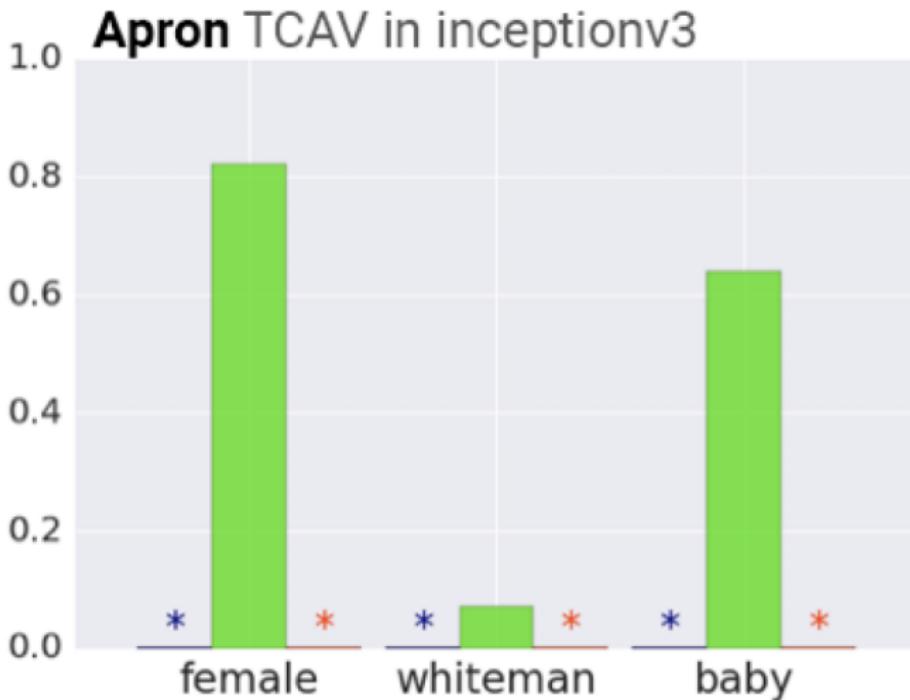


***CEO concept:*** least similar striped images



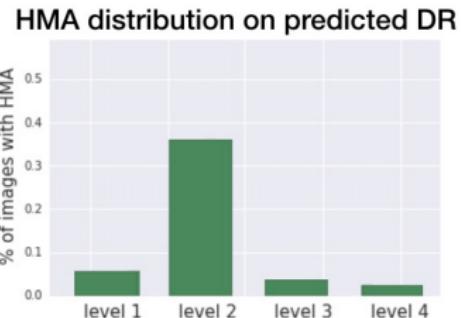
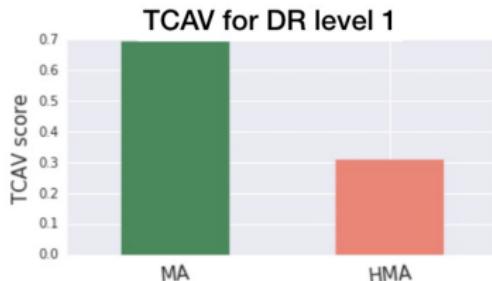
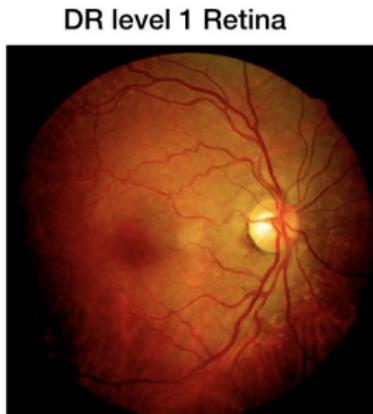
## Applications

- ▶ Quantify potential sources of bias



# Applications

- ▶ Measuring consistency with codified medical practice



## Conclusion

- ▶ There is no consensus on what makes an algorithm interpretable – when reading a paper, try to clarify the working definition
- ▶ As complicated as it can be to interpret models... still usually simpler than interpreting people