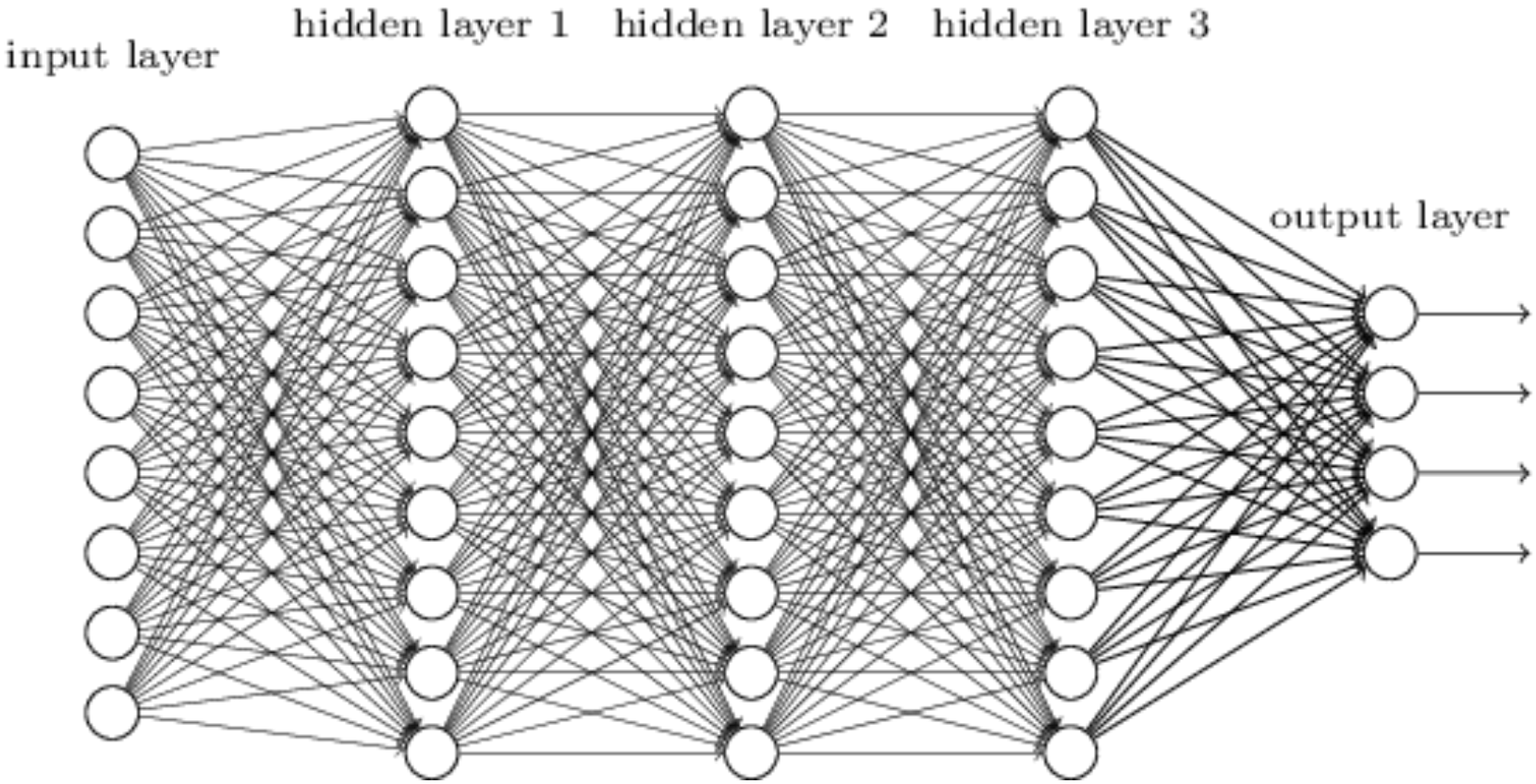
Interpretable Deep Learning



https://christophm.github.io/interpretable-ml-book/

STANDARD MACHINE LEARNING

DECISION

A few classes +
Hinge Loss +
Sequential Minimal Optimization +
HOG +
Deformable Part Models

Loss Function
Batch Optimization

Trainable classifier

Feature Library
Custom Features

Hand-engineered features

Raw Data

https://christophm.github.io/interpretable-ml-book/

DEEP LEARNING

Backpropagation + Tricks

DECISION

Loss Function
Network Architecture
Stochastic Gradient Descend

Trainable classifier
+
Trainable features

Raw Data

Training data: a set of $(x^{(m)}, y^{(m)})$ pairs.
Learn a function $f_w : x \to y$ to predict on new inputs x.

1. Choose a model function family f_w.

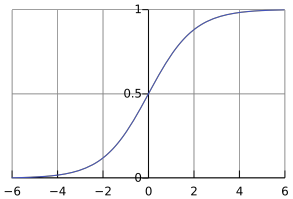
2. Optimize parameters w.

Parameters

Weights Bias

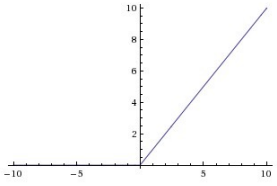
$$f(x) = \sigma(w^T \cdot x + b)$$

Dot Product



Sigmoid Function

$$\sigma(x) = \frac{1}{(1+e^{-x})}$$

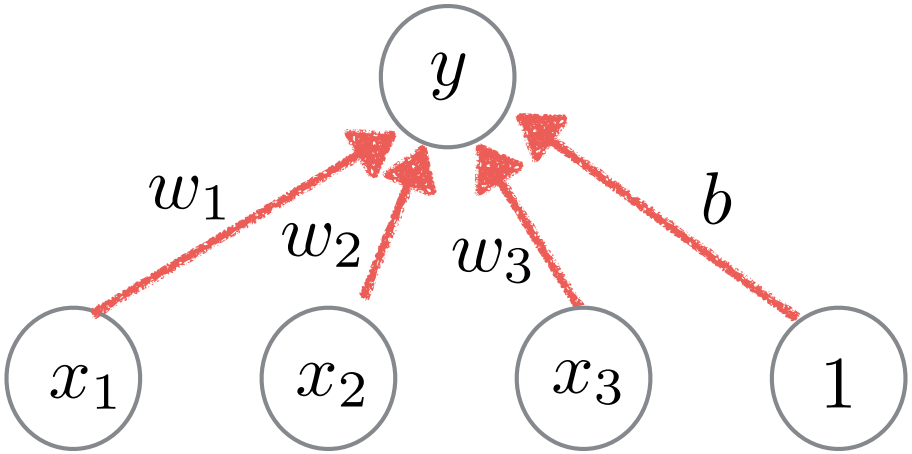


ReLU Function

$$\sigma(x) = max(0, x)$$

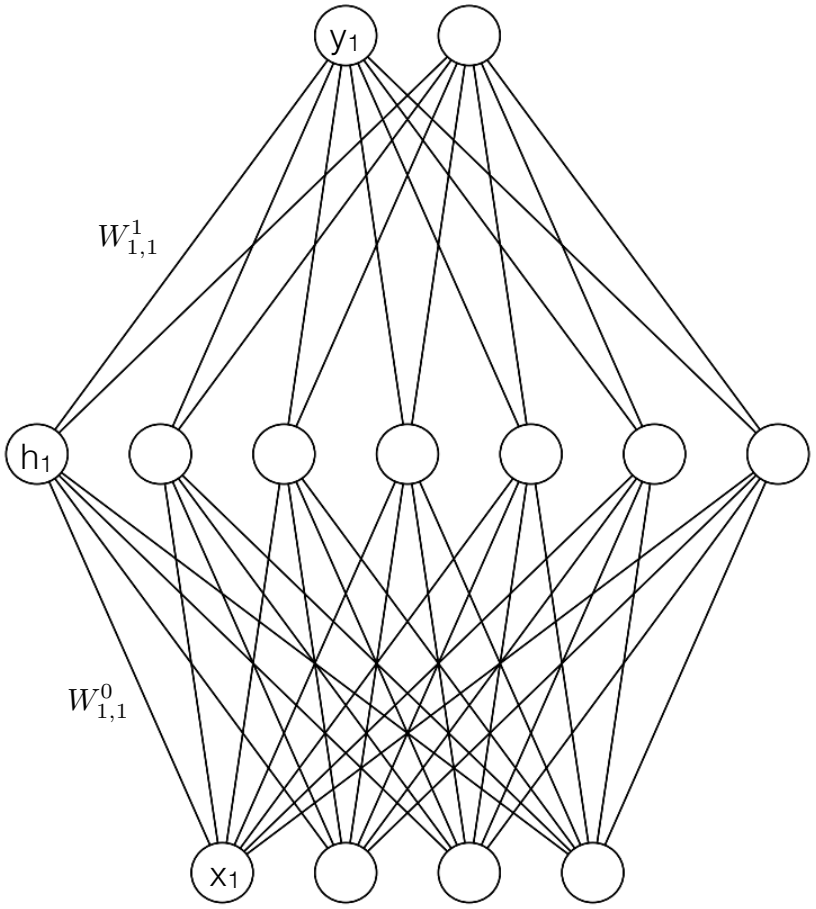
1-layer neural net model

$$f(x) = \sigma(w^T \cdot x + b)$$

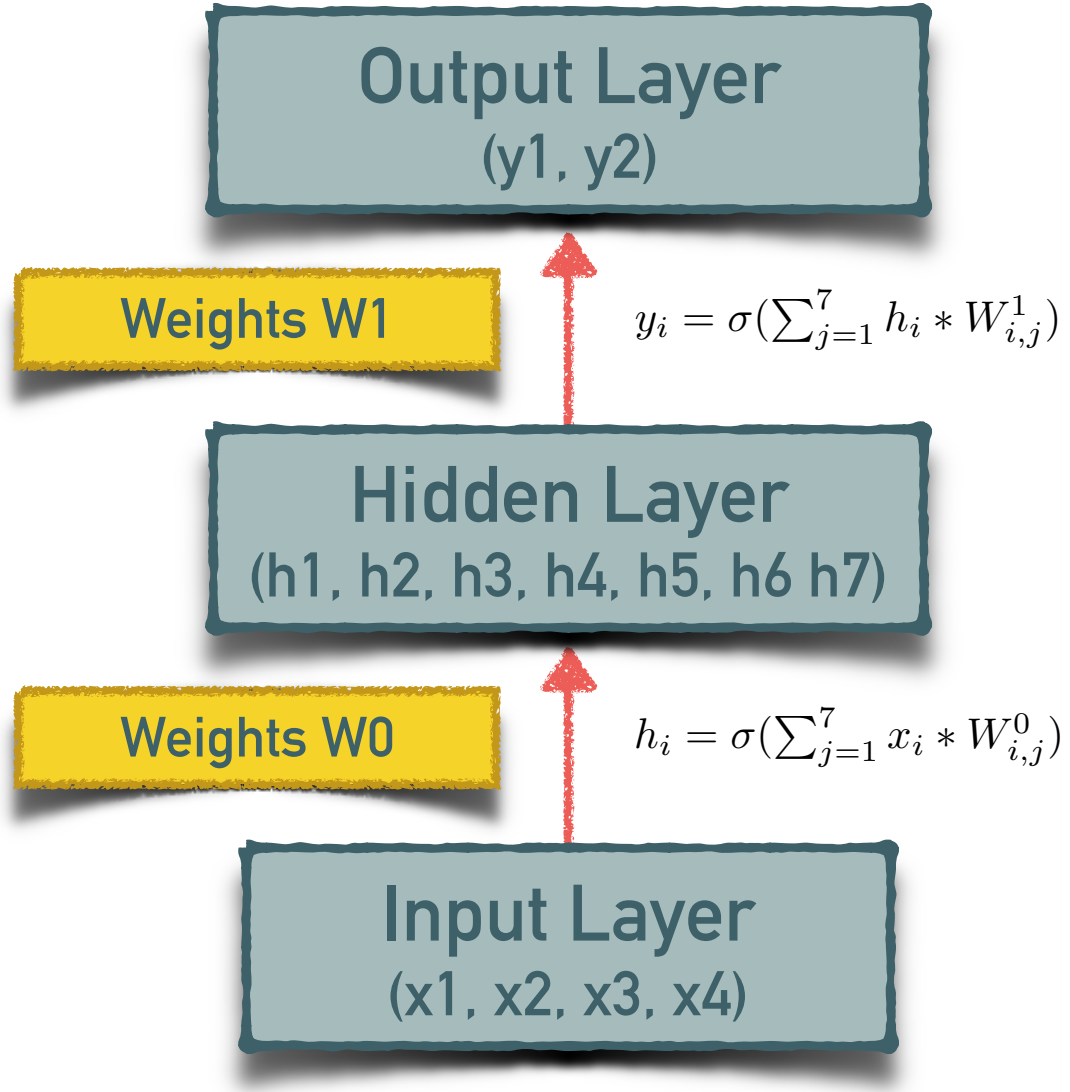


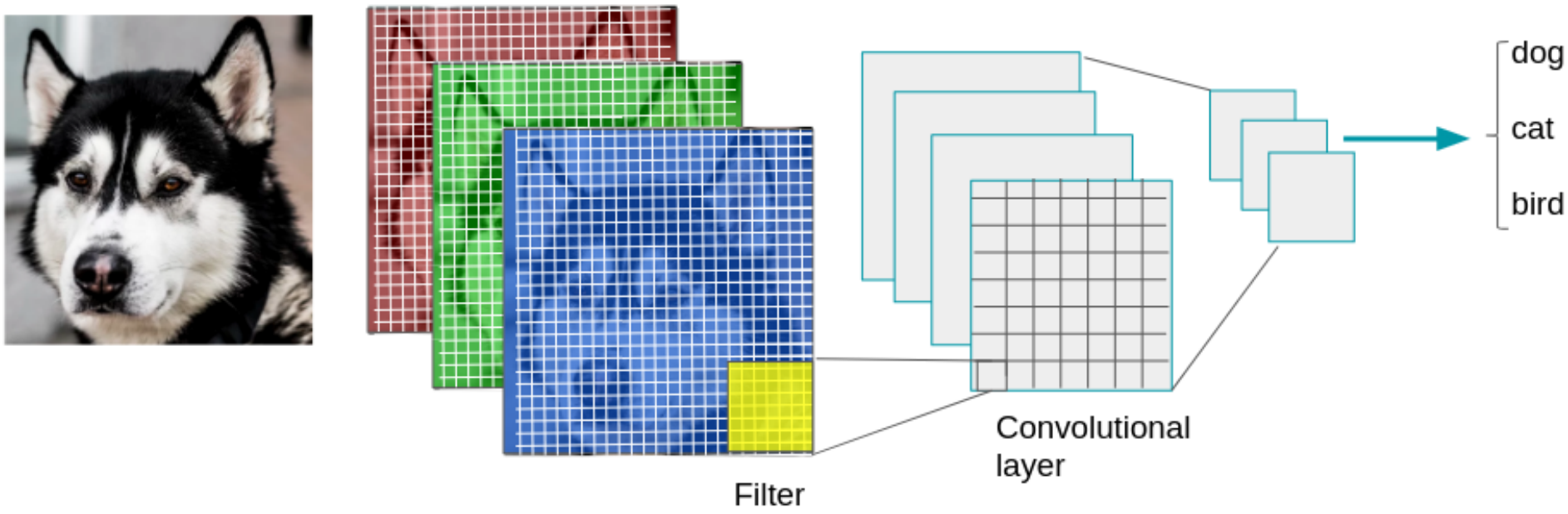
Graphical Representation

2-layer neural net model

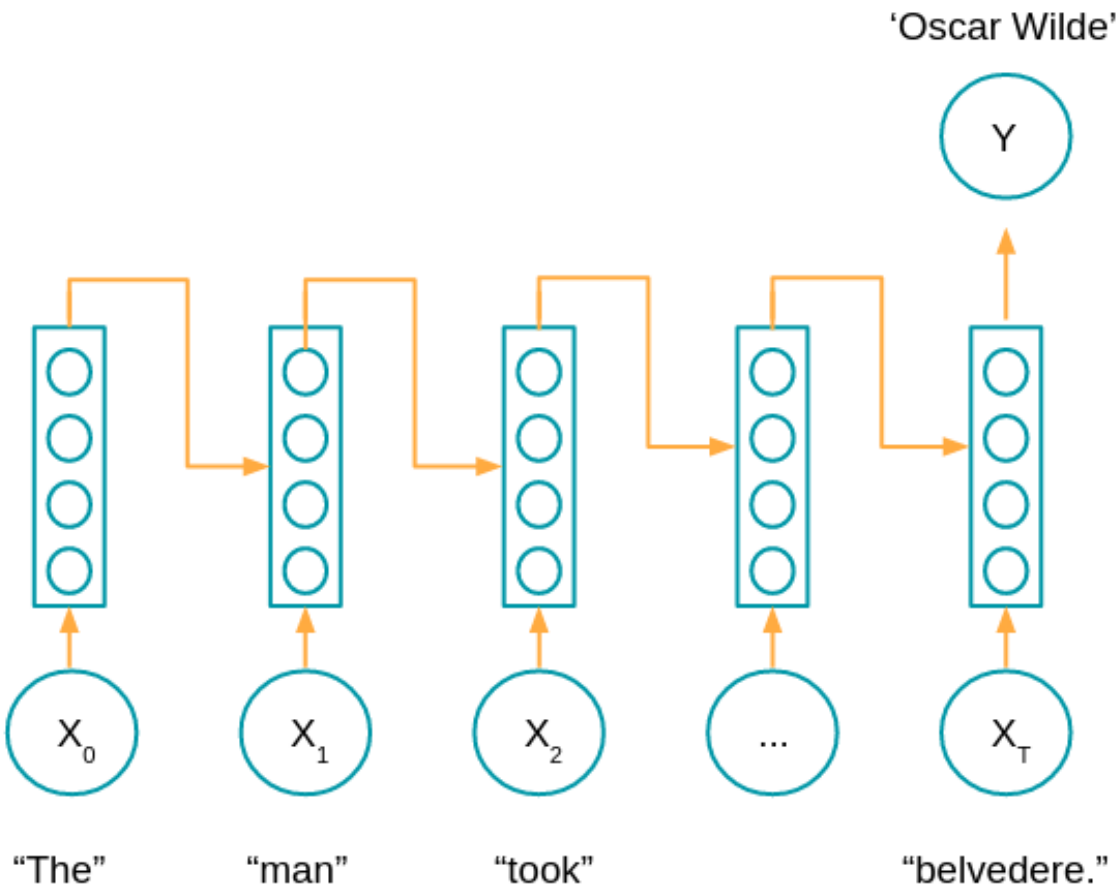


Computing the activation of one layer from the previous one can be written as a matrix-vector multiplication!



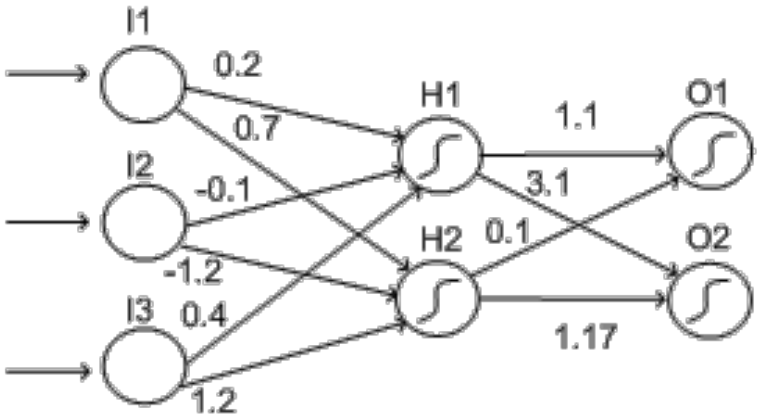


https://www.datakeen.co/3-deep-learning-architectures-explained-in-human-language/



https://www.datakeen.co/3-deep-learning-architectures-explained-in-human-language/

Interpretable Deep Learning



- What does the weights of each connection mean in terms of interpreting the result?
- Which is the set of weights that play the most important role in the final prediction?
- Does knowing the magnitude of the weights tells me anything about the importance of the input variables?

https://www.datakeen.co/3-deep-learning-architectures-explained-in-human-language/

Interpretable Deep Learning

Methods for explaining neural networks generally falls within two broad categories: **feature visualization** and **feature attribution methods**.

The firsts aim at visualizing what is going on inside the network and answering questions such as (1) which weights are being activated given some inputs? or (2) what regions of an image is being detected by a particular convolutional layer? .

https://www.datakeen.co/3-deep-learning-architectures-explained-in-human-language/

Interpretable Deep Learning

What Does the Network See?

Semantic dictionaries give us a fine-grained look at an activation: what does each single neuron detect? Building off this representation, we can also consider an activation vector as a whole. Instead of visualizing individual neurons, we can instead visualize the *combination* of neurons that fire at a given spatial location. (Concretely, we optimize the image to maximize the dot product of its activations with the original activation vector.)

=

+

+

+

+ ...

Activation Vector

Channels

https://distill.pub/2018/building-blocks/

Interpretable Deep Learning

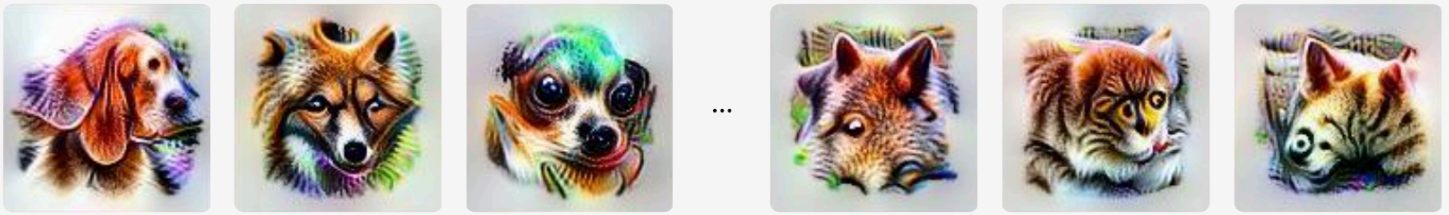
For instance, by combining feature visualization (*what is a neuron looking for?*) with attribution (*how does it affect the output?*), we can explore how the network decides between labels like **beagle** and **tiger cat**.

Several floppy ear detectors seem to be important when distinguishing dogs, whereas pointy ears are used to classify "tiger cat".

CHANNELS THAT MOST SUPPORT ... BEAGLE ⟷ TIGER CAT

feature visualization of channel

hover for attribution maps →

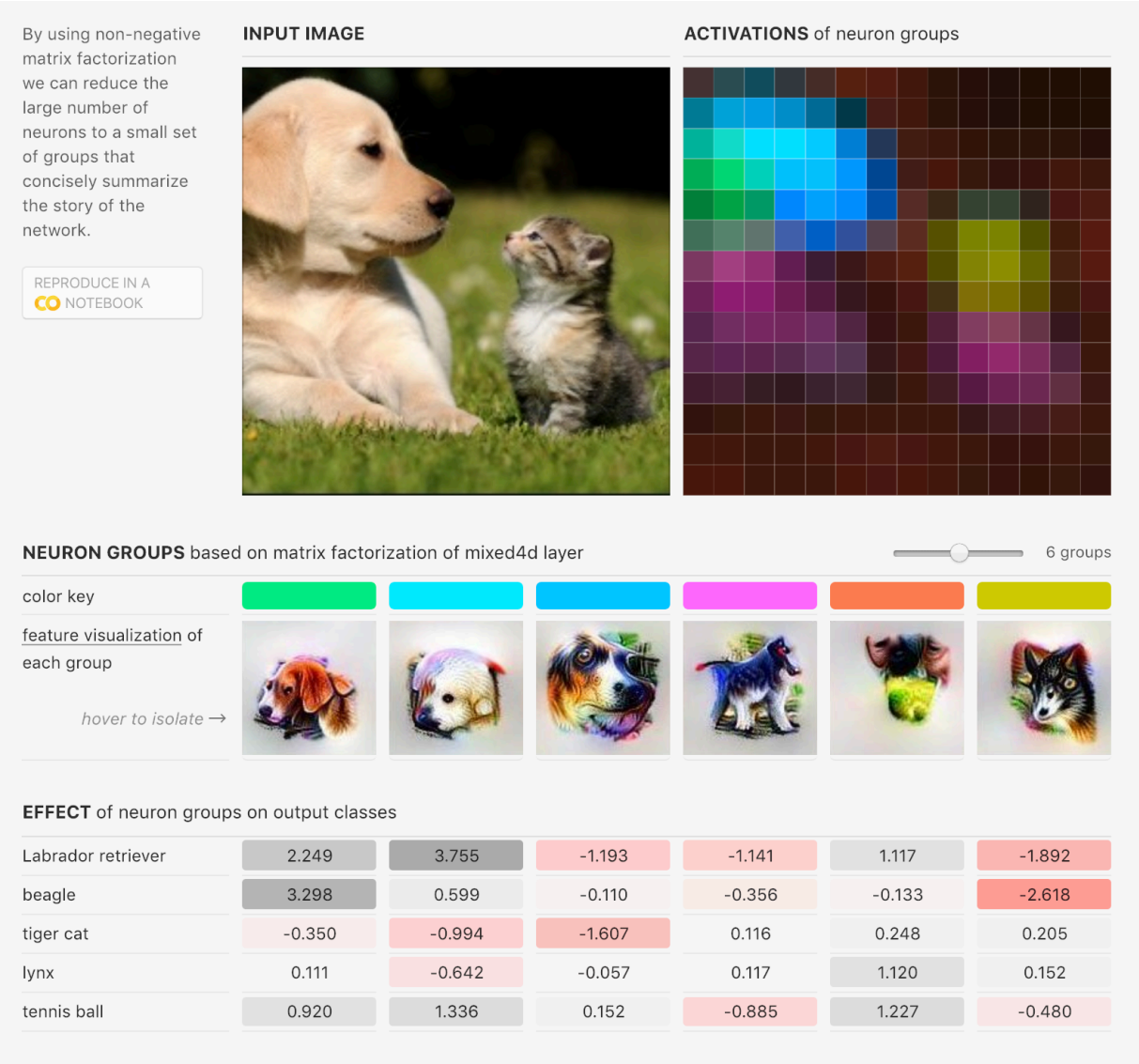


net evidence	3.49	2.25	1.18	1.07	1.45	1.58
for "beagle"	3.08	-0.21	0.45	-1.03	-0.50	-0.29
for "tiger cat"	-0.40	-2.47	-0.73	0.04	0.95	1.29

REPRODUCE IN A CO NOTEBOOK

https://distill.pub/2018/building-blocks/

Interpretable Deep Learning

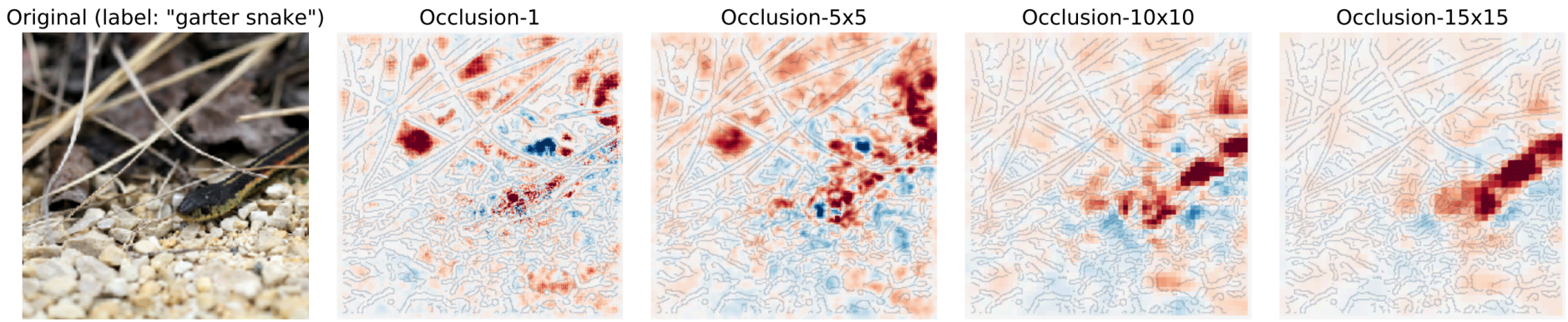


https://distill.pub/2018/building-blocks/

Interpretable Deep Learning

Feature attribution methods.

Perturbation-based methods directly compute the attribution of an input feature by removing, masking or altering them, and running a forward pass on the new input, measuring the difference with the original output.

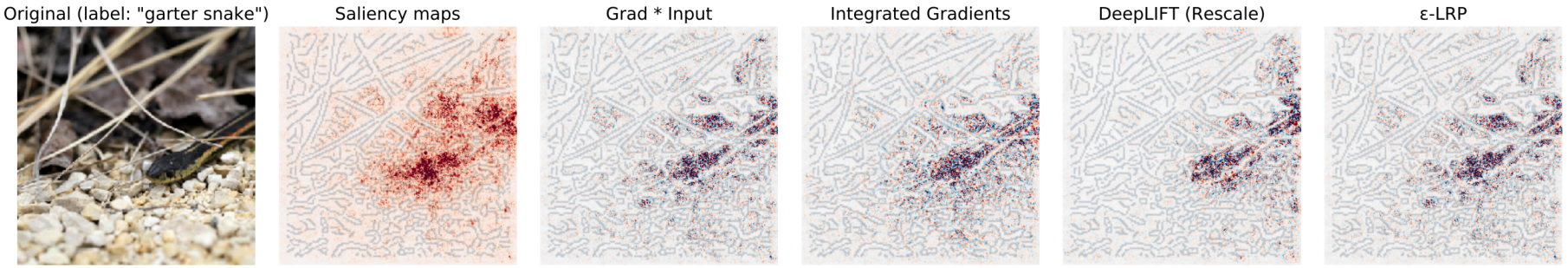


https://pdfs.semanticscholar.org/7a56/72796aeca8605b2e370d8a756a7a311fd171.pdf

Interpretable Deep Learning

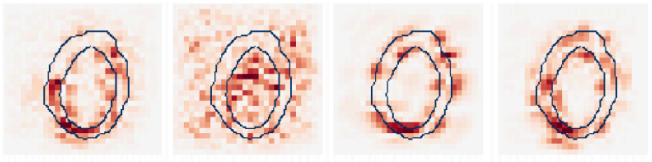
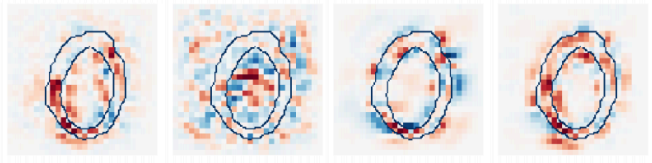
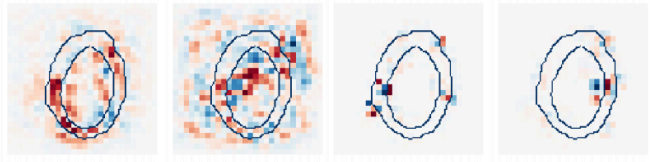
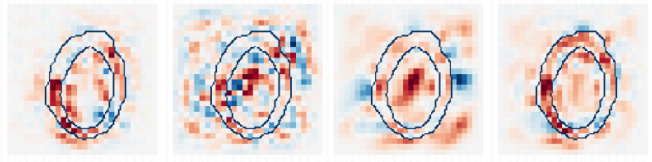
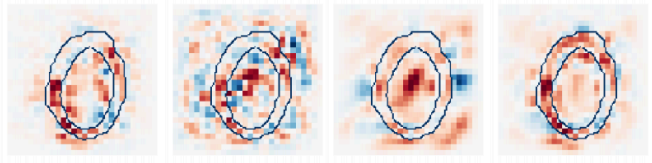
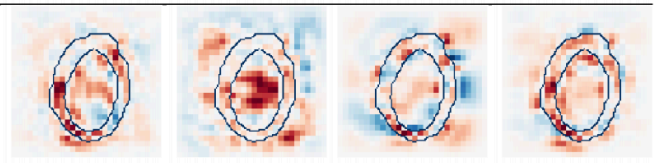
Feature attribution methods.

Gradient based methods constructs attributions by considering the partial derivative of the target output with respect to the input features (pixels).



https://pdfs.semanticscholar.org/7a56/72796aeca8605b2e370d8a756a7a311fd171.pdf

Interpretable Deep Learning

Method	Attribution $R_i^c(x)$	Example of attributions on MNIST
		ReLU Tanh Sigmoid Softplus
Saliency Maps	$\left\lvert \frac{\partial S_c(x)}{\partial x_i} \right\rvert$	
Gradient * Input	$x_i \cdot \frac{\partial S_c(x)}{\partial x_i}$	
ϵ-LRP	$x_i \cdot \frac{\partial^g S_c(x)}{\partial x_i}, \quad g = \frac{f(z)}{z}$	
DeepLIFT	$(x_i - \bar{x_i}) \cdot \frac{\partial^g S_c(x)}{\partial x_i}, \quad g = \frac{f(z) - f(\bar{z})}{z - \bar{z}}$	
Integrated Gradient	$(x_i - \bar{x_i}) \cdot \int_{\alpha=0}^{1} \frac{\partial S_c(\tilde{x})}{\partial(\tilde{x_i})}\Big\rvert_{\tilde{x}=\bar{x}+\alpha(x-\bar{x})} d\alpha$	
Occlusion-1	$x_i \cdot \int_{\alpha=0}^{1} \frac{\partial S_c(\tilde{x})}{\partial(\tilde{x_i})}\Big\rvert_{\tilde{x}=x_{[x_i=\alpha \cdot x_i]}} d\alpha$	

https://pdfs.semanticscholar.org/7a56/72796aeca8605b2e370d8a756a7a311fd171.pdf

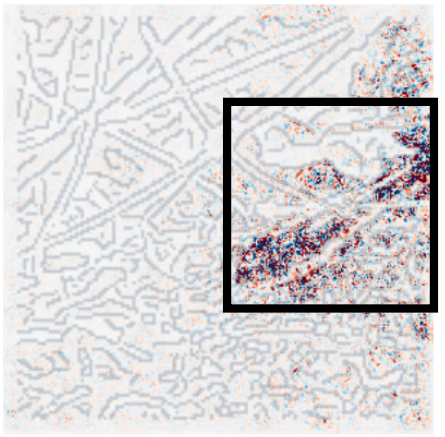
Interpretable Deep Learning

Original (label: "garter snake")

Explanation:

These pixels/region are the evidence of prediction.

DeepLIFT (Rescale)



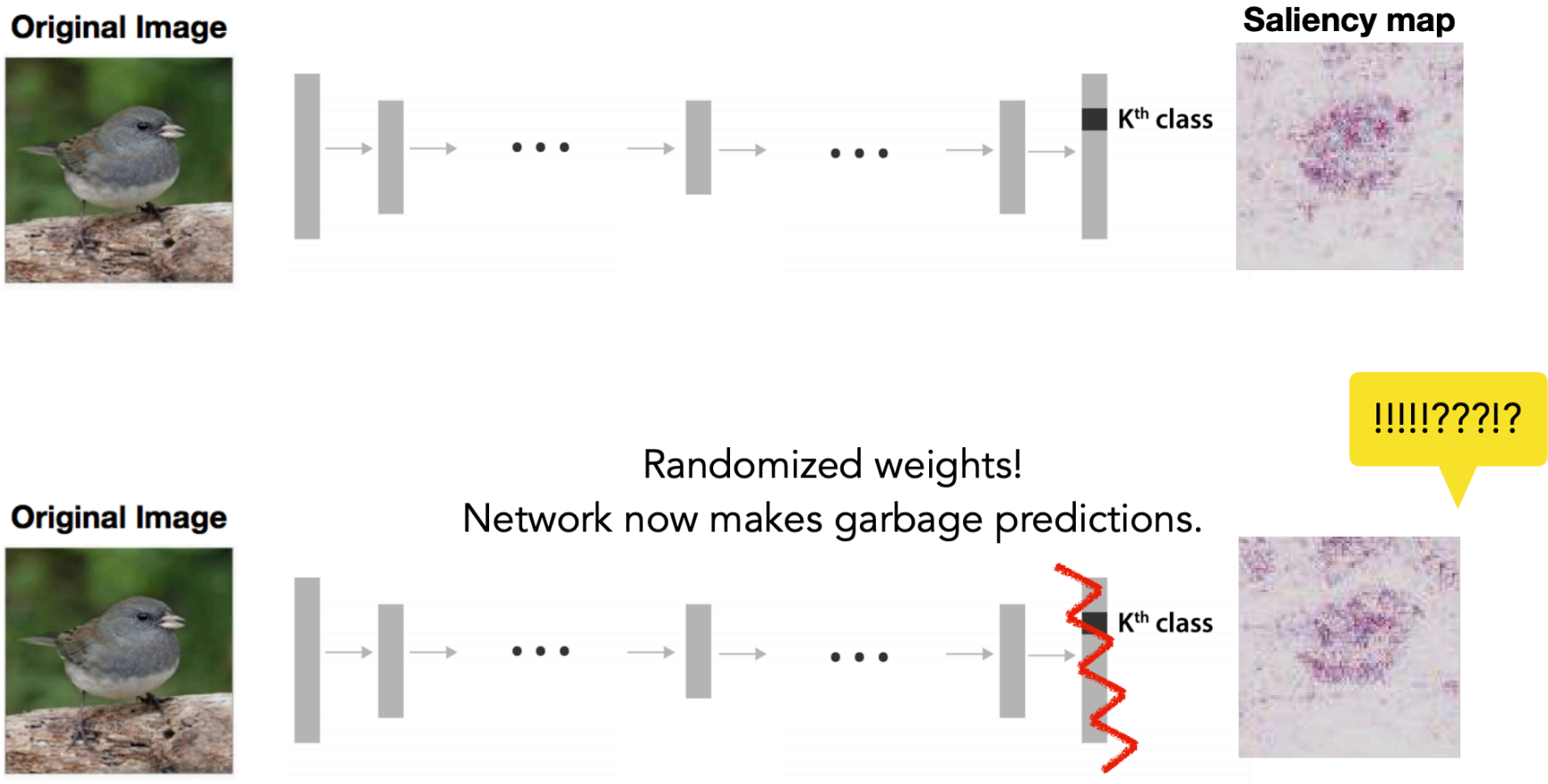
Interpretable Deep Learning

Feature attribution methods.

If the assumtion is correct, when prediction changes, the explanation should change.

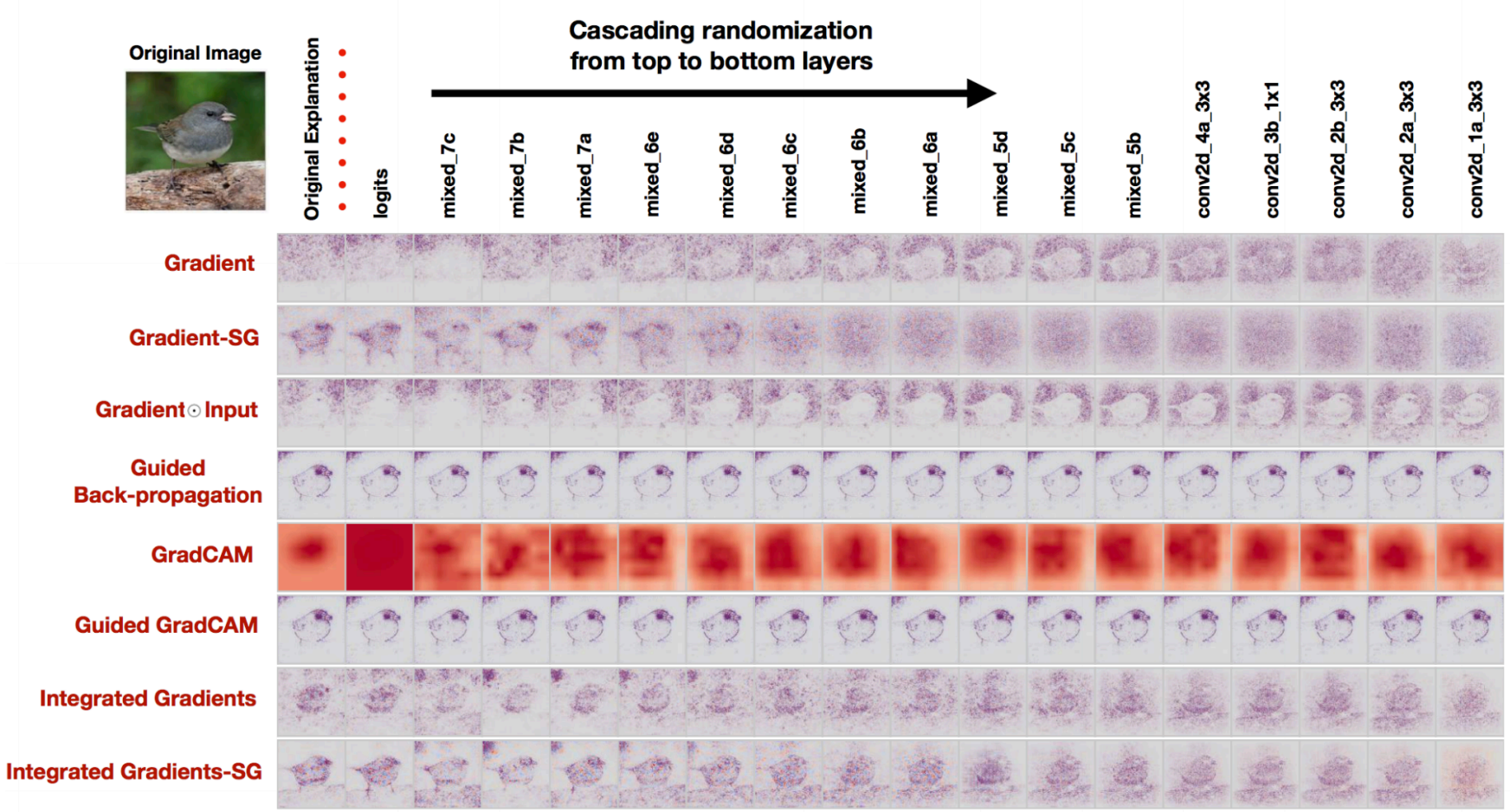
Extreme case: If prediction is random, the explanation should really change.

Interpretable Deep Learning



https://nips.cc/media/Slides/nips/2018/220e(05-09-45)-05-10-30-12640-Sanity_Checks_f.pdf

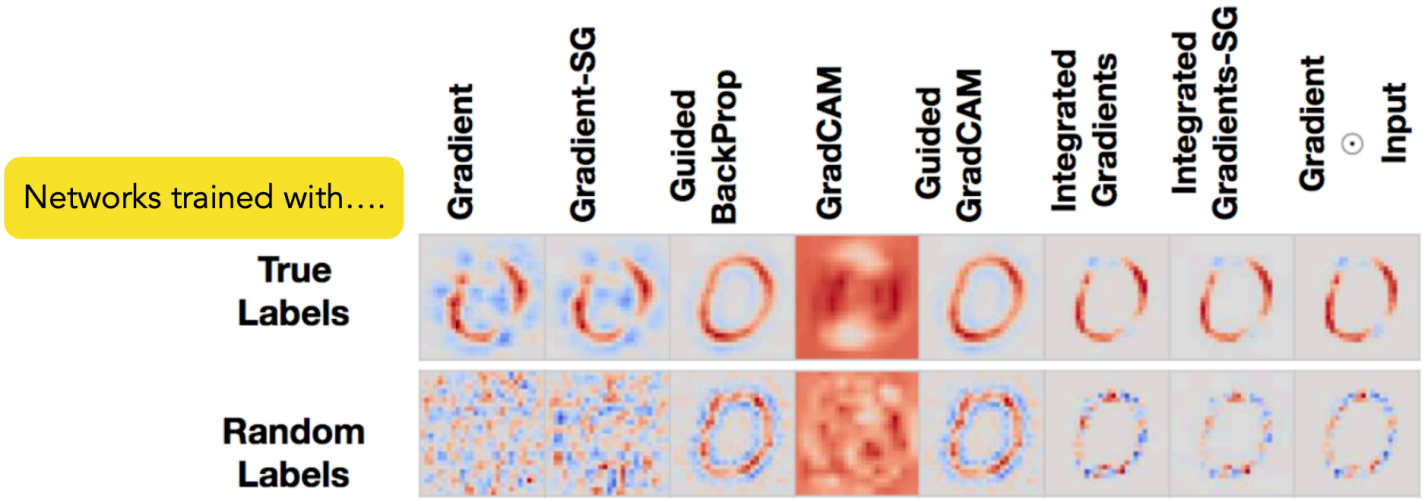
Interpretable Deep Learning



https://nips.cc/media/Slides/nips/2018/220e(05-09-45)-05-10-30-12640-Sanity_Checks_f.pdf

Interpretable Deep Learning

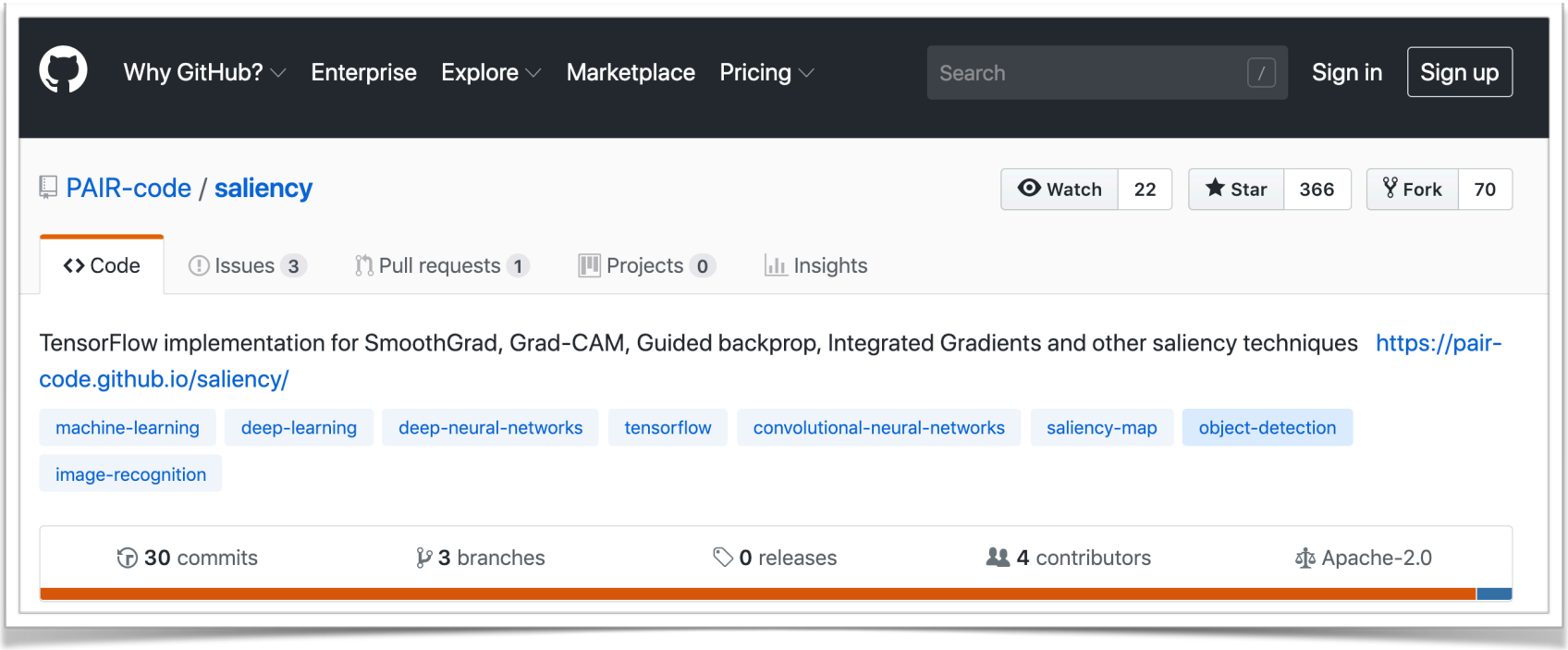
Sanity check2:
Networks trained with true and random labels,
Do explanations deliver different messages?
No!



https://nips.cc/media/Slides/nips/2018/220e(05-09-45)-05-10-30-12640-Sanity_Checks_f.pdf

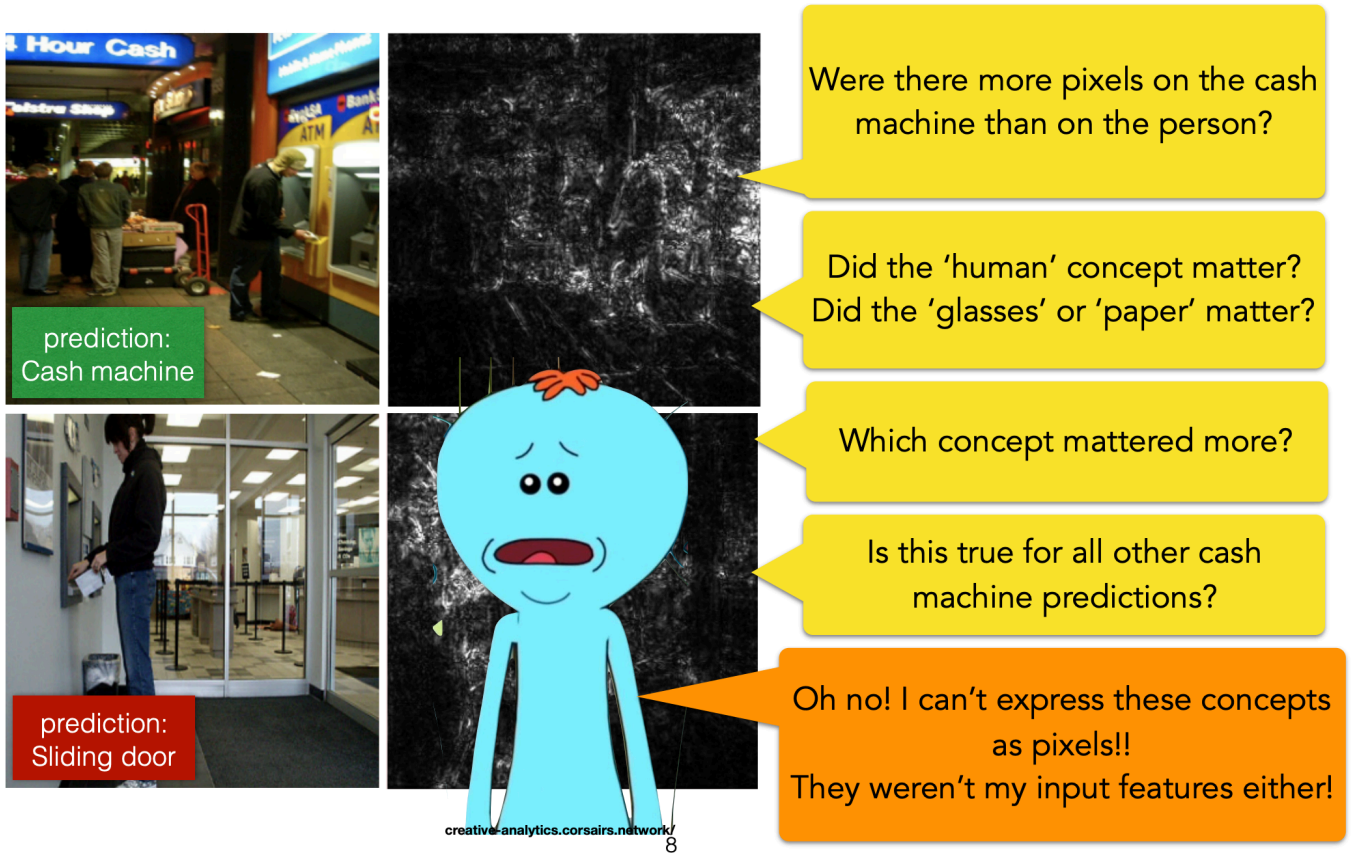
Interpretable Deep Learning

https://github.com/PAIR-code/saliency/blob/master/Examples.ipynb



Interpretable Deep Learning

In general, we cannot express explanations (for humans) as pixels.
We need to explain decisions in terms of **concepts**, not pixels.



https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning

Instead of pixels, we would like to use a **quantitative explanation expressing how much a concept** (gender, visual class, etc.) was important for a decision, **even if the concept was not part of the training**!

This has been recently explored in a method called **Concept Activation Vectors**.

Kim, Been, et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." International Conference on Machine Learning. 2018.

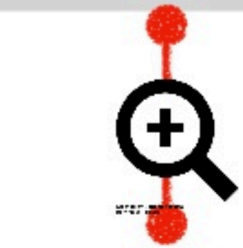
Interpretable Deep Learning

A trained machine learning model (e.g., neural network)

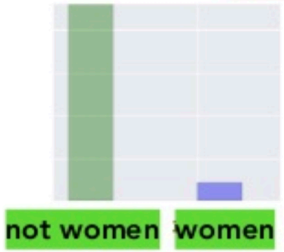
$p(z)$

Doctor-ness

Was gender concept important to this doctor image classifier?



TCAV score for **Doctor**



TCAV provides **quantitative importance** of a concept **if and only if** your network learned about it.

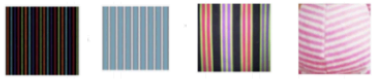
https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning

A trained machine learning model (e.g., neural network)

$p(z)$

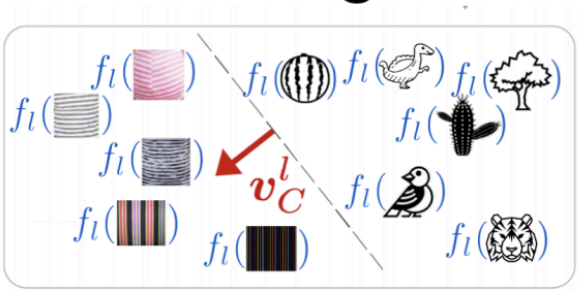
zebra-ness



Was striped concept important to this zebra image classifier?

TCAV

1. Learning CAVs

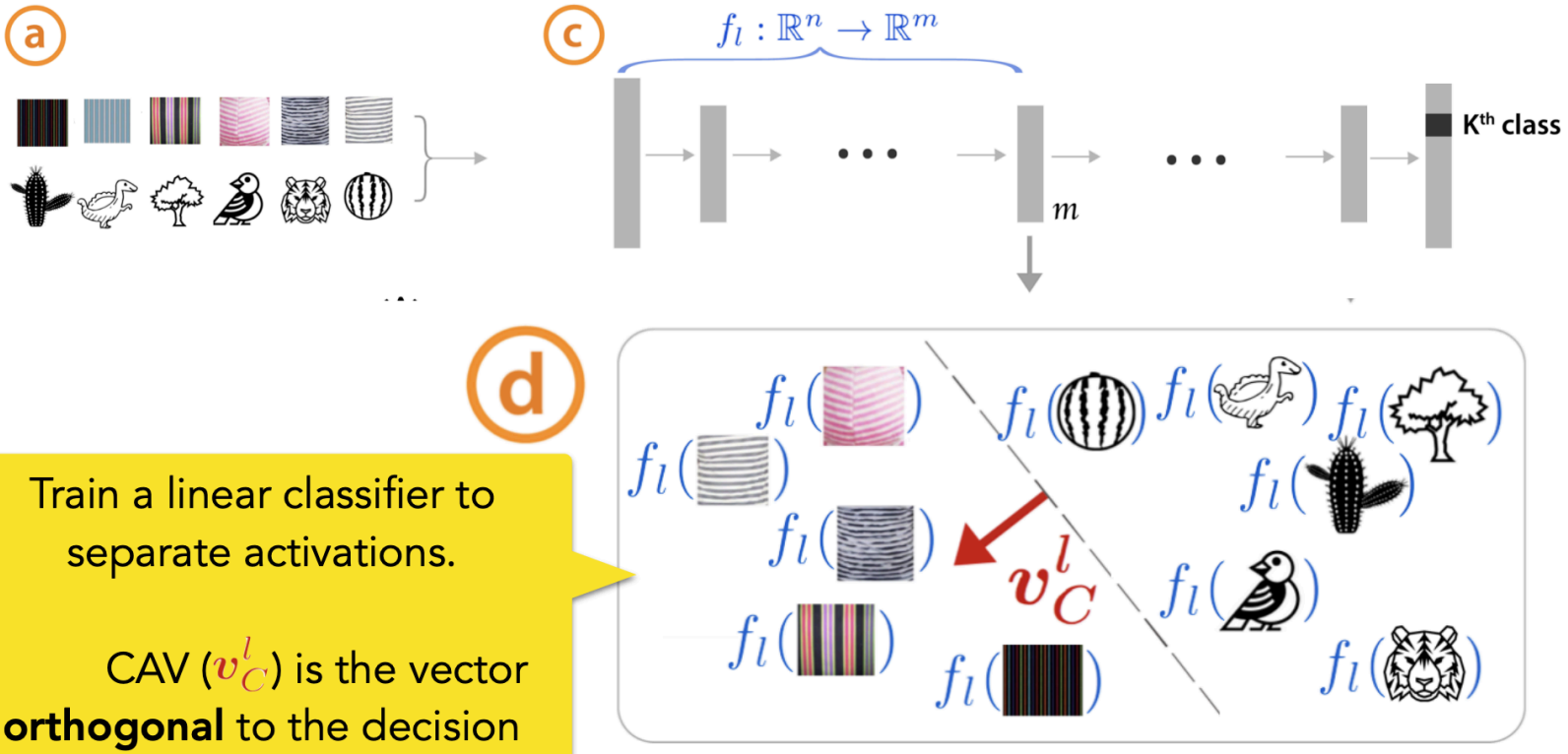


1. How to define concepts?

https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning

Inputs:



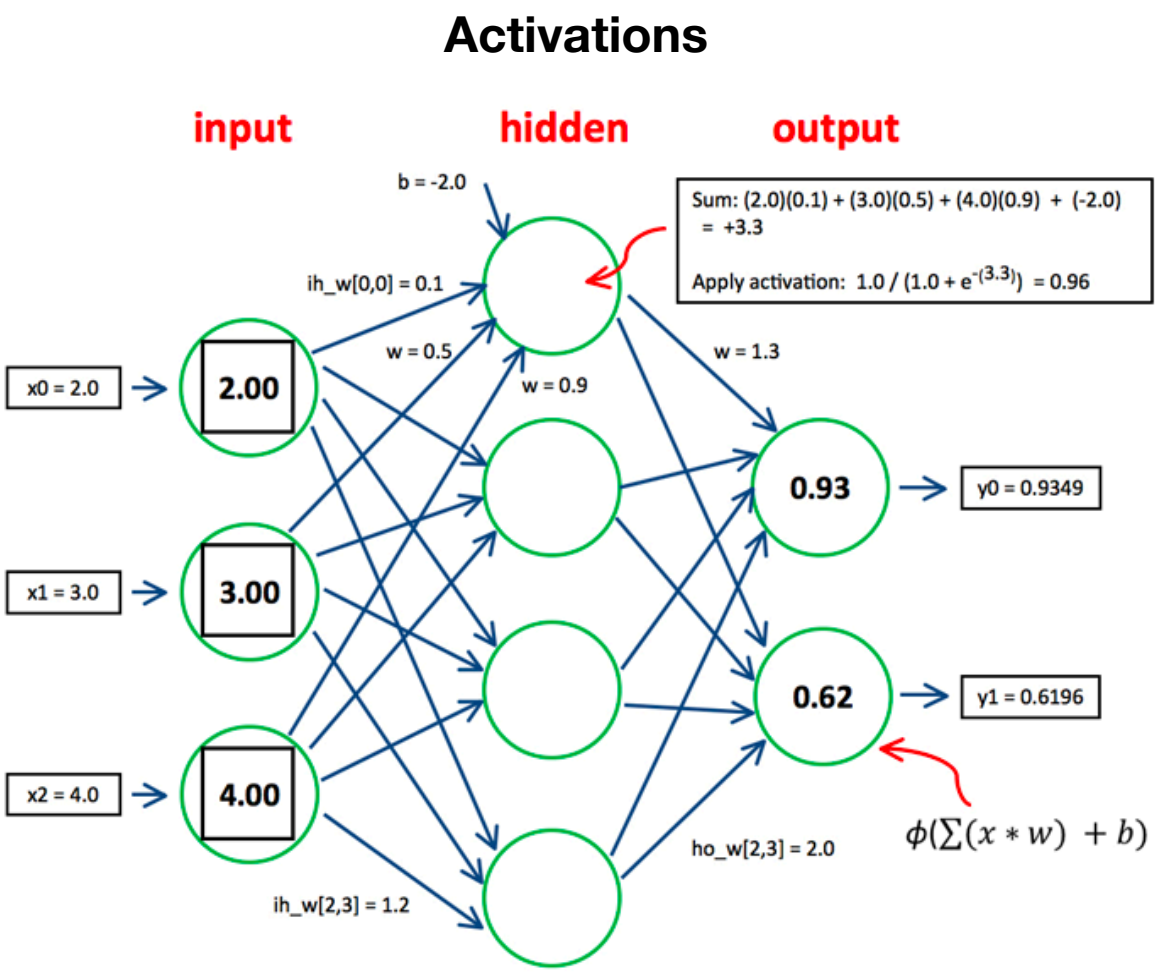
Train a linear classifier to separate activations.

CAV (v_C^l) is the vector **orthogonal** to the decision boundary.

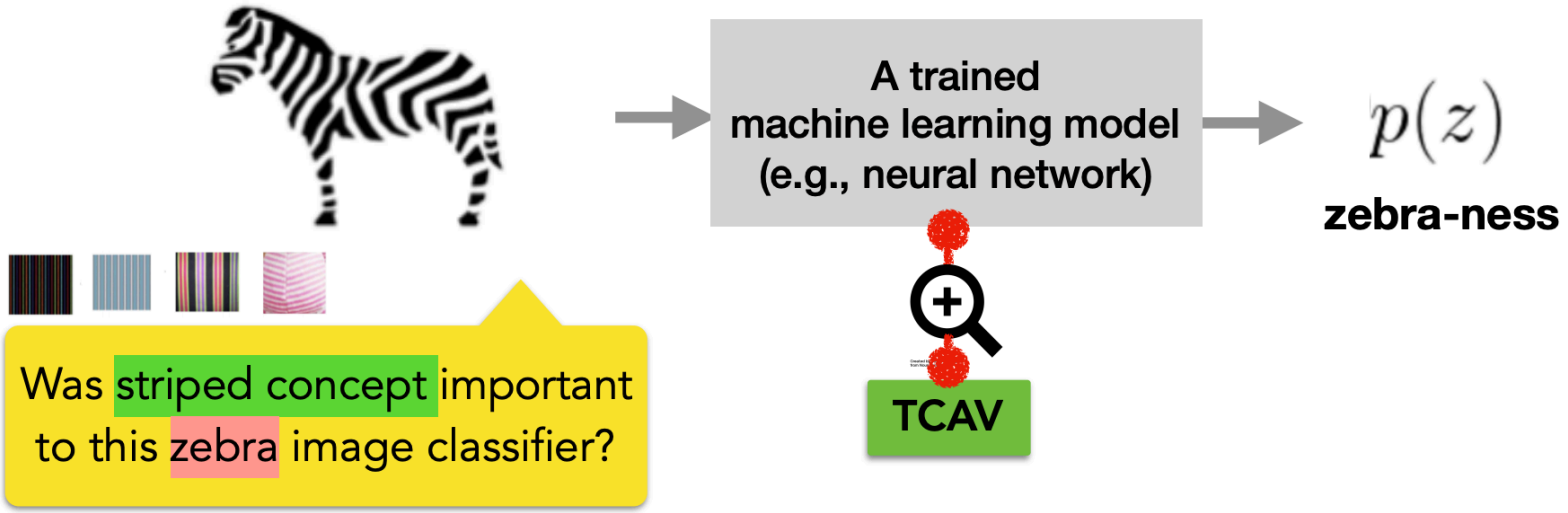
[Smilkov '17, Bolukbasi '16 , Schmidt '15]

https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning

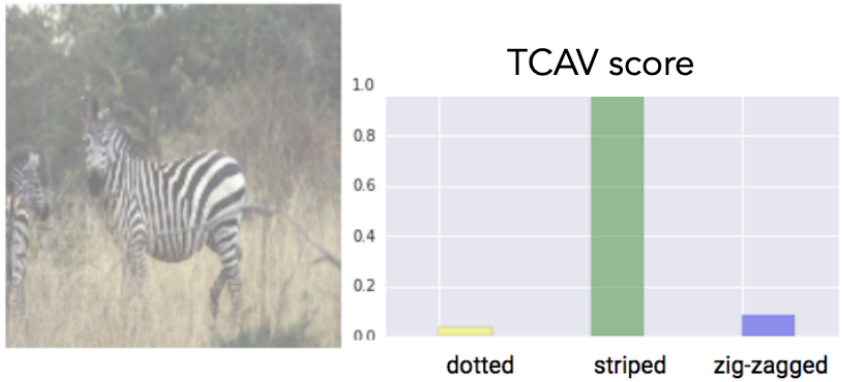


Interpretable Deep Learning



https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning



$$S_{C,k,l}(\text{  })$$
$$S_{C,k,l}(\text{  })$$
$$S_{C,k,l}(\text{  })$$
$$S_{C,k,l}(\text{  })$$

zebra-ness → $\dfrac{\partial p(z)}{\partial \boldsymbol{v}_C^l} = S_{C,k,l}(\boldsymbol{x})$

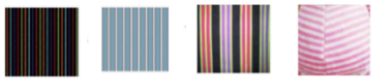
striped CAV →

$$\text{TCAVQ}_{C,k,l} = \frac{|\{\boldsymbol{x} \in X_k : S_{C,k,l}(\boldsymbol{x}) > 0\}|}{|X_k|}$$

Directional derivative with CAV

https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning

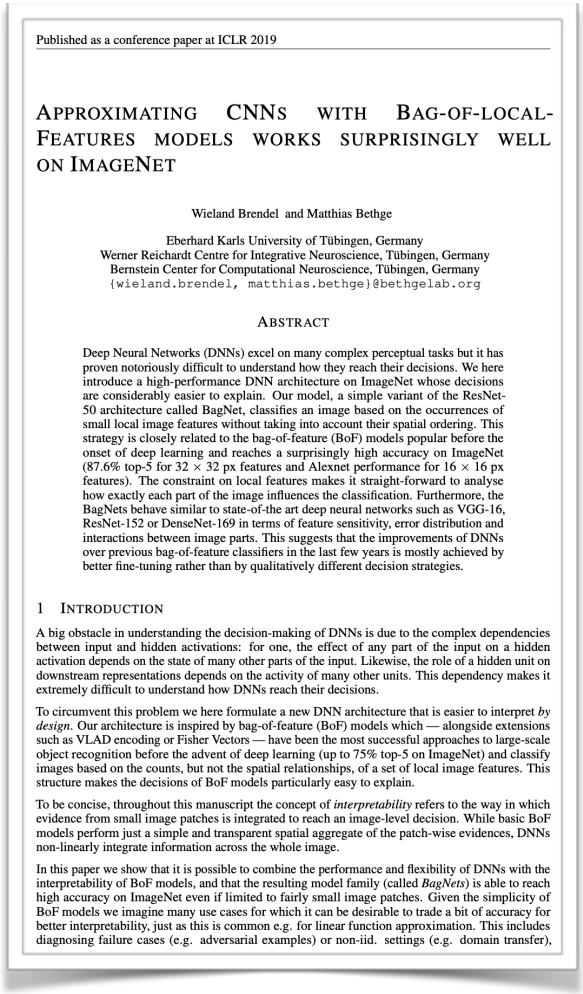


stripes concept (score: 0.9)
was important to **zebra** class
for this trained network. 

https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning

This is still a hot topic…



Published as a conference paper at ICLR 2019

APPROXIMATING CNNS WITH BAG-OF-LOCAL-FEATURES MODELS WORKS SURPRISINGLY WELL ON IMAGENET

Wieland Brendel and Matthias Bethge

Eberhard Karls University of Tübingen, Germany
Werner Reichardt Centre for Integrative Neuroscience, Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
{wieland.brendel, matthias.bethge}@bethgelab.org

ABSTRACT

Deep Neural Networks (DNNs) excel on many complex perceptual tasks but it has proven notoriously difficult to understand how they reach their decisions. We here introduce a high-performance DNN architecture on ImageNet whose decisions are considerably easier to explain. Our model, a simple variant of the ResNet-50 architecture called BagNet, classifies an image based on the occurrences of small local image features without taking into account their spatial ordering. This strategy is closely related to the bag-of-feature (BoF) models popular before the onset of deep learning and reaches a surprisingly high accuracy on ImageNet (87.6% top-5 for 32 × 32 px features and Alexnet performance for 16 × 16 px features). The constraint on local features makes it straight-forward to analyse how exactly each part of the image influences the classification. Furthermore, the BagNets behave similar to state-of-the art deep neural networks such as VGG-16, ResNet-152 or DenseNet-169 in terms of feature sensitivity, error distribution and interactions between image parts. This suggests that the improvements of DNNs over previous bag-of-feature classifiers in the last few years is mostly achieved by better fine-tuning rather than by qualitatively different decision strategies.

1 INTRODUCTION

A big obstacle in understanding the decision-making of DNNs is due to the complex dependencies between input and hidden activations: for one, the effect of any part of the input on a hidden activation depends on the state of many other parts of the input. Likewise, the role of a hidden unit on downstream representations depends on the activity of many other units. This dependency makes it extremely difficult to understand how DNNs reach their decisions.

To circumvent this problem we here formulate a new DNN architecture that is easier to interpret *by design*. Our architecture is inspired by bag-of-feature (BoF) models which — alongside extensions such as VLAD encoding or Fisher Vectors — have been the most successful approaches to large-scale object recognition before the advent of deep learning (up to 75% top-5 on ImageNet) and classify images based on the counts, but not the spatial relationships, of a set of local image features. This structure makes the decisions of BoF models particularly easy to explain.

To be concise, throughout this manuscript the concept of *interpretability* refers to the way in which evidence from small image patches is integrated to reach an image-level decision. While basic BoF models perform just a simple and transparent spatial aggregate of the patch-wise evidences, DNNs non-linearly integrate information across the whole image.

In this paper we show that it is possible to combine the performance and flexibility of DNNs with the interpretability of BoF models, and that the resulting model family (called *BagNets*) is able to reach high accuracy on ImageNet even if limited to fairly small image patches. Given the simplicity of BoF models we imagine many use cases for which it can be desirable to trade a bit of accuracy for better interpretability, just as this is common e.g. for linear function approximation. This includes diagnosing failure cases (e.g. adversarial examples) or non-iid. settings (e.g. domain transfer),

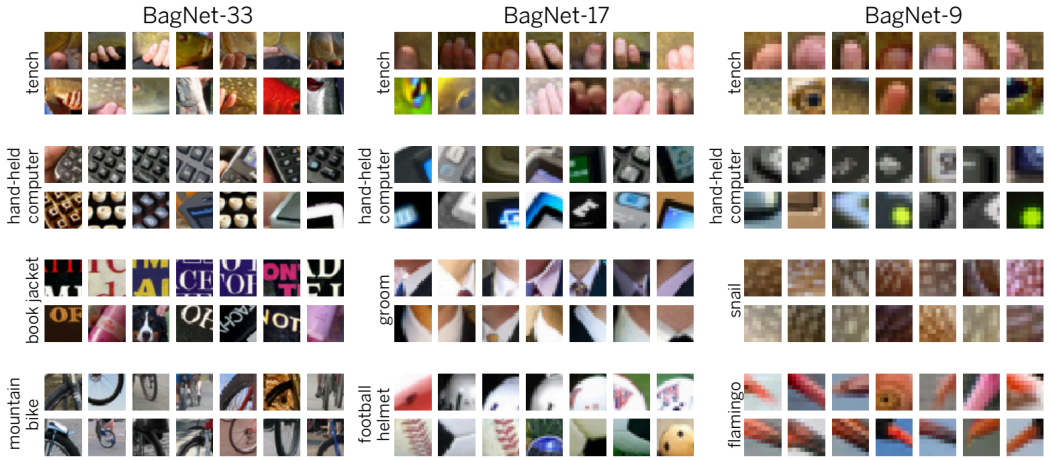


Figure 3: Most informative image patches for BagNets. For each class (row) and each model (column) we plot two subrows: in the top subrow we show patches that caused the highest logit outputs for the given class across all validation images with that label. Patches in the bottom subrow are selected in the same way but from all validation images with a *different* label (highlighting errors).

Interpretable Deep Learning

This is still a hot topic…

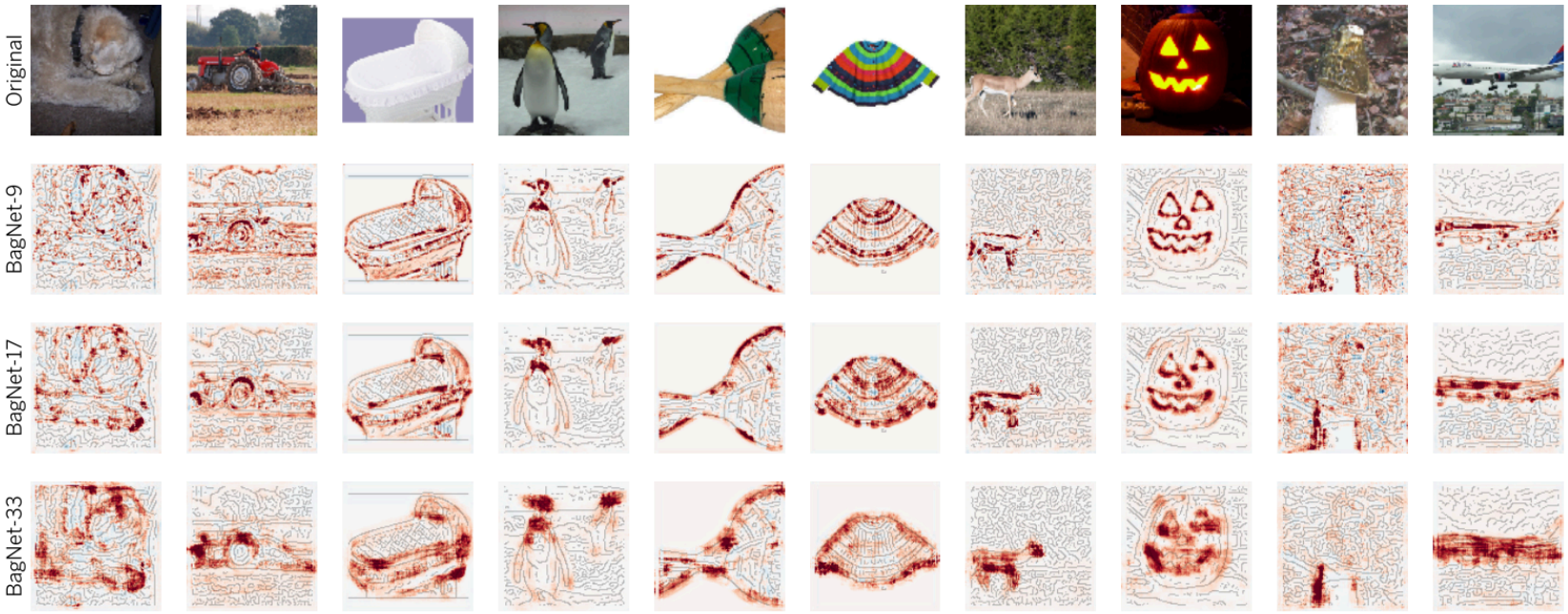


Figure 2: Heatmaps showing the class evidence extracted from of each part of the image. The spatial sum over the evidence is the total class evidence.